Proceedings of the Thirteenth International Workshop on Treebanks and Linguistic Theories (TLT13)

December 12-13, 2014
Tübingen, Germany

Editors

Verena Henrich
Erhard Hinrichs
Daniël de Kok
Petya Osenova
Adam Przepiórkowski

Sponsors

SFB 833

ISBN: 978-3-9809183-9-8

Department of Linguistics (SfS)
University of Tübingen
Wilhelmstr. 19
72074 Tübingen, Germany

`http://www.sfs.uni-tuebingen.de/`

Program Chairs

Verena Henrich, University of Tübingen, Germany
Erhard Hinrichs, University of Tübingen, Germany
Petya Osenova, Sofia University, Bulgaria
Adam Przepiórkowski, Polish Academy of Sciences, Poland

Program Committee

Eckhard Bick, University of Southern Denmark, Denmark
Ann Bies, Linguistic Data Consortium, Philadelphia, USA
Johan Bos, University of Groningen, The Netherlands
Cristina Bosco, University of Turin, Italy
António Branco, University of Lisbon, Portugal
Miriam Butt, Konstanz University, Germany
David Chiang, University of Southern California, Los Angeles, USA
Tomaž Erjavec, Jožef Stefan Institute, Slovenia
Markus Dickinson, Indiana University, Indiana, USA
Stefanie Dipper, Ruhr University Bochum, Germany
Dan Flickinger, Stanford University, California, USA
Anette Frank, Heidelberg University, Germany
Eva Hajičová, Charles University, Czech Republic
Valia Kordoni, Humboldt University Berlin, Germany
Anke Lüdeling, Humboldt University Berlin, Germany
Joakim Nivre, Uppsala University, Sweden
Gertjan van Noord, University of Groningen, The Netherlands
Stefan Oepen, Oslo University, Norway
Kemal Oflazer, Carnegie Mellon University, Qatar
Karel Oliva, Academy of Sciences of the Czech Republic, Czech Republic
Marco Passarotti, Catholic University of the Sacred Heart, Italy
Victoria Rosén, Bergen University, Norway
Kiril Simov, Bulgarian Academy of Sciences, Bulgaria
Caroline Sporleder, Saarland University, Germany
Michael White, Ohio State University, Ohio, USA
Nianwen Xue, Brandeis University, Massachusetts, USA
Heike Zinsmeister, University of Hamburg, Germany

Local Organizing Committee

Erhard Hinrichs
Petra Burkard
Verena Henrich
Daniël de Kok
Dörte de Kok
Bettina Meier

Preface

The *Thirteenth International Workshop on Treebanks and Linguistic Theories* (TLT13) is held at the University of Tübingen, Germany, on December 12-13, 2014. We are happy to see that, ten years after the previous workshop in Tübingen, TLT is still in its prime.

This year's TLT saw 38 paper submissions of which 28 were accepted, highlighting both the popularity of the workshop and the quality of its submissions. Even though the papers in the present proceedings cover a wide array of topics, we see some clear trends:

- Dependency treebanks have become a mainstay in the field. Joakim Nivre observed in his submission to the TLT1 proceedings, *What kinds of trees grow in Swedish soil?*, that dependency annotation schemes such as that of the Prague Dependency Treebank of Czech were quickly gaining importance. Ten years later, a good share of TLT13 papers revolve around dependency structure. Given this development, we are very honored to have two major contributors to dependency parsing – Gertjan van Noord and Sandra Kübler – as this year's keynote speakers.

- Treebanks of historical text is a topic that has seen increasing popularity in the last few years. For instance, in this year there are papers treating Old French, Early New High German, and Old Occitan text. Also, lexical-semantic annotation of treebanks is certainly becoming a 'trending topic' with submissions covering German and Bulgarian.

- Finally, we see some topics that are relatively new to TLT. For instance, TLT13 has two submissions about the syntactic analysis of learner language. Another newer topic for TLT is the annotation of new types of text, such as verses or encyclopedia text.

We think that this year's TLT has a well-rounded program with papers on treebank development, the use of treebanks for linguistic research, and treebank-driven approaches to natural language processing. Of course, this would not have been possible without the program committee, who worked hard to review the many submissions and provided authors with valuable feedback. We would also like to thank CLARIN-D and the Collaborative Research Center SFB 833 for sponsoring TLT13. We are also thankful to the University of Tübingen for hosting the workshop – even though ten years have passed since the last TLT in Tübingen, we traveled back in time from the *Neue Aula* 'New Aula' in 2004 to the *Alte Aula* 'Old Aula' in 2014. Last but not least, we would like to wish all participants a fruitful workshop.

Verena Henrich, Erhard Hinrichs, Daniël de Kok, Petya Osenova, and Adam Przepiórkowski

Contents

I Long papers **1**

Verbal constructional profiles: possibilities and limitations **2**
Aleksandrs Berdičevskis and Hanne Martine Eckhoff

Towards a Universal Stanford Dependencies parallel treebank **14**
Cristina Bosco and Manuela Sanguinetti

Finding Parse Errors in the Midst of Parse Errors **26**
Markus Dickinson and Amber Smith

Querying topological fields in the TIGER scheme with TIGERSearch **37**
Stefanie Dipper

Parsing Poorly Standardized Language: Dependency on Old French **51**
Gaël Guibon, Isabelle Tellier, Matthieu Constant, Sophie Prévost and Kim Gerdes

Consistency of Manual Sense Annotation and Integration into the TüBa-D/Z Treebank **62**
Verena Henrich and Erhard Hinrichs

Deriving Multi-Headed Projective Dependency Parses from Link Grammar Parses **75**
Juneki Hong and Jason Eisner

Different approaches to the PP-attachment problem in Polish **88**
Katarzyna Krasnowska

POS-Tagging Historical Corpora: The Case of Early New High German **103**
Pavel Logacev, Katrin Goldschmidt and Ulrike Demske

Synergistic development of grammatical resources: a valence dictionary, an LFG grammar, and an LFG structure bank for Polish **113**
Agnieszka Patejuk and Adam Przepiórkowski

The Sense Annotation of BulTreeBank **127**
Alexander Popov, Stanislava Kancheva, Svetlomira Manova, Ivaylo Radev, Kiril Simov and Petya Osenova

The Effect of Annotation Scheme Decisions on Parsing Learner Data **137**
Marwa Ragheb and Markus Dickinson

Metrical annotation for a verse treebank **149**
T. M. Rainsford and Olga Scrivner

Cross-lingual Dependency Transfer with Harmonized Indian Language Treebanks **160**
Loganathan Ramasamy and Zdeněk Žabokrtský

POS Tagset Refinement for Linguistic Analysis and the Impact on Statistical Parsing **172**
Ines Rehbein and Hagen Hirschmann

Semi-Automatic Deep Syntactic Annotations of the French Treebank **184**
Corentin Ribeyre, Marie Candito and Djamé Seddah

Formalizing MultiWords as Catenae in a Treebank and in a Lexicon **198**
Kiril Simov and Petya Osenova

Estimating the Utility of Simplified Discriminants in Grammar-Based Treebanking **208**
Arne Skjærholt and Stephan Oepen

A grammar-licensed treebank of Czech **218**
Hana Skoumalová, Alexandr Rosen, Vladimír Petkevič, Tomáš Jelínek, Přemysl Vítovec and Jiří Znamenáček

Evaluating Parse Error Detection across Varied Conditions **230**
Amber Smith and Markus Dickinson

II Short papers **242**

Quantitative Comparison of Different Bi-Lexical Dependency Schemes for English **243**
Norveig Anderssen Eskelund and Stephan Oepen

The definition of tokens in relation to words and annotation tasks **250**
Fabian Barteld, Renata Szczepaniak and Heike Zinsmeister

From <tiger2/> to ISOTiger - Community Driven Developments for Syntax Annotation in SynAF **258**
Sonja Bosch, Kerstin Eckart, Gertrud Faaß, Ulrich Heid, Kiyong Lee, Antonio Pareja-Lora, Laurette Pretorius, Laurent Romary, Andreas Witt, Amir Zeldes and Florian Zipser

Challenges in Enhancing the Index Thomisticus Treebank with Semantic and Pragmatic Annotation **265**
Berta Gonzalez Saavedra and Marco Passarotti

TüBa-D/W: a large dependency treebank for German **271**
Daniël de Kok

What can linguists learn from some simple statistics on annotated treebanks **279**
Jiří Mírovský and Eva Hajičová

Estonian Dependency Treebank and its annotation scheme **285**
Kadri Muischnek, Kaili Müürisep, Tiina Puolakainen, Eleri Aedmaa, Riin Kirt and Dage Särg

Developing a Corpus of Syntactically-Annotated Learner Language for English **292**
Marwa Ragheb and Markus Dickinson

Part I

Long papers

Verbal constructional profiles: reliability, distinction power and practical applications

Aleksandrs Berdičevskis and Hanne Eckhoff
UiT The Arctic University of Norway
E-mail: aleksandrs.berdicevskis@uit.no, hanne.m.eckhoff@uit.no

Abstract

In this paper we explore the notion of constructional profiles (the frequency distribution of a given linguistic item across syntactic environments) from two angles, methodological and applied. We concentrate on verbal constructional profiles, using Russian argument frame data in two different dependency formats. We first test the profiles' stability and distinction power across sample sizes, and then use the profiles in two tasks concerning Russian aspect: to identify the aspectual partner of a given verb and to guess whether a given verb is perfective or imperfective.

1 Introduction

A linguistic profile is a frequency distribution of occurrences of a linguistic item across a given parameter. Because it contains useful quantitative information about an item's usage, such profiles can help us discover fundamental properties of the item [3]. Here we focus on verbal constructional profiles, where the item is always a verb, and the parameter is its syntactic environment. This methodology has been used for various purposes with some success [2, 4, 7], but little is known about the basic properties of the profiles.

We start by addressing two general methodological questions in section 4. First, is there such thing as a reliable constructional profile, i.e. is there a stable distribution that we might want to approximate? If yes, what corpus size is required to capture it? Second, what distinction power do the profiles possess at different corpus sizes? To test that, we used the SynTagRus treebank of modern Russian [1, 11], both in its native dependency format and converted into the PROIEL format (see section 3). As a secondary goal, we compare the two dependency schemes' ability to yield useful argument structure data.

We then zoom in on a more language-specific question and estimate the possibility of using verbal constructional profiles as an objective criterion in the study of Russian aspect, with a view to use it in diachronic studies, in particular, to apply it to Old Russian where the aspectual properties of verbs are often highly unclear. In section 5 we test the method's applicability to modern Russian aspect, taking

into account our methodological results. We test two different hypotheses: first, that constructional profiles can be used to identify the most likely aspectual partner of a verb; second, that constructional profiles can be used to tell whether a verb is perfective or imperfective. We see that both hypotheses hold to some extent, but only the second one yields results that are good enough to be of practical use for our purposes.[1]

2 Constructional profiles

The constructional profile of a given verb is a set of pairs {argument frame; frequency}, where the argument frame is a set of dependents that are considered arguments (as opposed to adjuncts).[2] By frequency we mean the relative frequency of occurrence of the frame with a given verb. Two small example profiles in the two formats (see more about the formats in section 3) are seen in table 1.

	PROIEL			SynTagRus		
verb	frame	abs. freq.	**rel. freq.**	frame	abs. freq.	**rel. freq.**
vysypat'	V+obj	1	**0.20**	V+1-kompl	1	**0.20**
	V+sub+obl	2	**0.40**	V+predik+sravnit	1	**0.20**
	V+sub+obj+obl	1	**0.20**	V+1-kompl+2-kompl+predik	1	**0.20**
	V+obj+obl	1	**0.20**	V+2-kompl+predik	1	**0.20**
				V+1-kompl+3-kompl	1	**0.20**
vylivat'	**V+obj+obl**	1	**0.33**	**V+1-kompl+3-kompl**	1	**0.33**
	V+sub+obj	2	**0.67**	V+1-kompl+predik	2	**0.67**

Table 1: Simple constructional profiles for the verbs *vysypat'* 'pour out (solids)' and *vylivat'* 'pour out (liquids)'

One of the most important things one can do with profiles is to compare them. The similarity of two profiles can be measured as their intersection rate (adapted from [7]), which is calculated as follows. For every frame that occurs in both compared profiles, we look at its relative frequencies in both profiles, take the smallest of these two values and add up all such values. If we compare the profiles of *vysypat'* 'pour out (solids)' and *vylivat'* 'pour out (liquids)', we will get an intersection rate of 0.2 regardless of which annotation scheme we use (see 1). Only one frame (V+obj+obl in PROIEL, V+1-kompl+3-kompl in SynTagRus) occurs in both profiles, and the smallest of its relative frequencies is 0.2.

We tried three different ways of building profiles. Table 1 illustrates the first profile type (simple), where only syntactic relation labels are included. The second

[1] All raw data for this article, results and their statistical significance, and code used to perform experiments can be found at the TROLLing Dataverse, hdl:10037.1/10142.

[2] Verb dependents deemed to be arguments here: direct and indirect/oblique objects (including sources and goals with motion verbs), subjects, passive agents, complement clauses and various types of predicative complements.

type (partly enriched) includes basic morphological information about the verb: whether it is passive and whether it is a participle used in attributive function. The third profile type (fully enriched) additionally uses simplified morphological and lexical information about the arguments: argument infinitives are labeled as such (inf), prepositional arguments are labeled "PP", arguments headed by subjunctions are labeled with the subjunction lemma, and nominal arguments are labeled with their case. Thus, a simple V+obl frame may for instance turn out to be a Vref-lpas_attrib+obl_PP in the fully enriched profile type. Naturally, the two enriched profile types, and especially the fully enriched type, are more granular than the simple type.

3 The two formats

To address the questions outlined in section 1, we turned to SynTagRus, a large treebank of Russian (860,720 words).[3] Since our goal is ultimately to study Russian aspect diachronically, we wanted to have the data in the same format that our Old Russian data are in,[4] effectively creating a treebank spanning over a thousand years. We therefore automatically converted the whole SynTagRus into the PROIEL format, an enriched dependency grammar scheme which is used for an expanding family of treebanks of ancient languages originating in the PROIEL parallel treebank.[5] For the experiments, we used both versions of SynTagRus: the original and the converted one.[6]

Both PROIEL and SynTagRus are dependency schemes, essentially describing the functional relationships between heads and dependents. However, both give considerably more granular argument structure representations than most dependency schemes, e.g. the Prague Dependency Treebank [10], but in very different ways. The two schemes are therefore particularly interesting to compare. The SynTagRus format, based on the Meaning–Text model [9], is the more traditional of the two schemes, in that it only allows primary dependencies and only to a limited extent allows empty nodes. The argument representation is highly granular, but without including much information on the syntactic category of the argument itself. Instead, it heavily relies on lexico-semantic properties of words. For instance, while the 1-kompl ('first complement') relation is typically used for direct objects, with the verb 'to live' it is used for PPs and adverbs denoting locations ('to live in Norway', 'to live here' etc.). The relative rank of an argument in a valency frame, not its form, decides what relation label it gets.

[3] Created by the Laboratory of Computational Linguistics at the Institute for Information Transmission Problems, and hosted by the Russian National Corpus (http://ruscorpora.ru/).

[4] The Tromsø Old Russian and OCS Treebank (TOROT) at https://nestor.uit.no.

[5] Found at foni.uio.no:3000 and built by members of the project *Pragmatic Resources in Old Indo-European Languages*.

[6] A conversion from the PROIEL format to the SynTagRus format is likely to be much less successful, due to the focus on lexical semantics in the latter scheme and its higher granularity. Such a conversion would have to rely heavily on lexical lists.

The PROIEL scheme is inspired by and convertible to LFG F-structures [5, 6]. Structure sharing is indicated by way of secondary dependencies (e.g. in control and raising structures, but also to indicate shared arguments and predicate identity). The format also systematically uses empty verb and conjunction nodes to account for ellipsis, gapping and asyndetic coordination. Both these features enhance argument structure representation, since less structural information is lost than in the SynTagRus scheme. Argument representation is less granular than in SynTagRus: transitive objects (obj) are distinguished from oblique objects (obl), and complement clauses (comp) and arguments with external subjects (xobj, e.g. control infinitives) have separate labels. Both schemes use a wide definition of the term "argument" – for instance, both schemes take sources and goals with motion verbs and locations with positional verbs as arguments – but the PROIEL scheme is somewhat more restrictive, which will necessarily cause differences in the constructional profiles.

The main issues encountered in the conversion process had to do with coordination and null verb insertion, since the necessary information was not always recoverable in the SynTagRus data. We were able to insert secondary dependencies to external subjects very successfully using morphological cues, but were unable to insert secondary dependencies to shared arguments. Apart from that, argument structure dependencies and labels were generally converted very successfully using lexical, morphological and part-of-speech cues. A spot check of 50 random sentences (759 words, including empty tokens) shows that the conversion was 98% accurate if only the errors relevant to argument structure were counted, whereas the overall accuracy was 90%. The two most frequent error types include wrong relation labels and wrong structure (e.g. incorrect head node), next come wrong part of speech, the least frequent type are wrong morphological features (which usually follow from a part-of-speech misclassification).

4 Profile stability and distinction power

As mentioned above, one crucial question is how reliable the constructional profiles are, or, to put it in more practical terms, what sample size is required for a profile to become stable. Profile sample size is understood here as the number of verb tokens that were used to build a profile. The profiles in table 1, for instance, have extremely small sample sizes (5 and 3), and thus are hardly reliable.

In order to estimate the relationship between profile stability and sample size, we carried out the following experiment. For a given sample size n (from 10 to 500 with step 10), all verbs that had a frequency no lower than $2n$ were found in the corpus. For every verb, two non-intersecting but otherwise random samples of the size n were drawn, a profile was built using each of these samples, and the intersection rate between the two profiles was calculated. Average values for each sample size (see more below) are presented in figure 1.[7] The higher the intersection

[7] The fact that large sample sizes yield small sets of verbs (v) to be tested ($v = 991$ for $n = 10$,

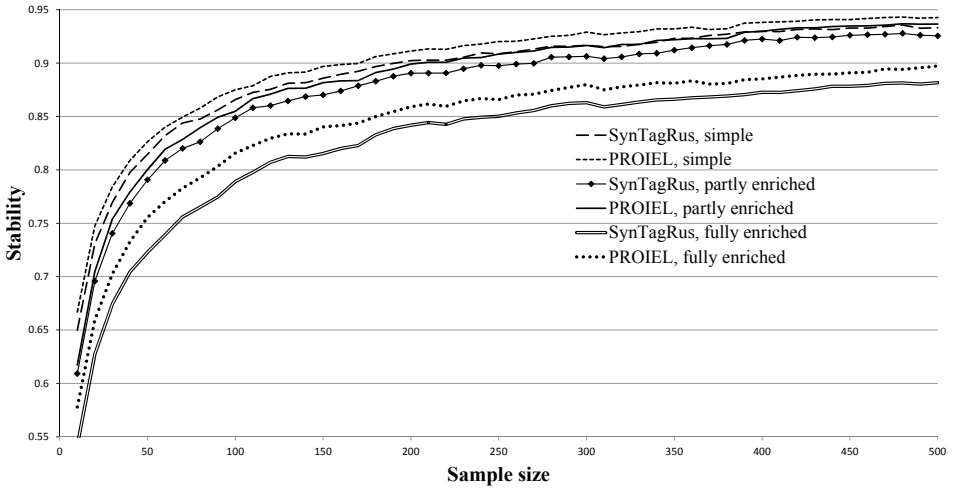


Figure 1: Profile stability across sample sizes.

rate, the closer the two profiles built using independent samples are (ideally, we want them to be identical), and the higher the profile stability is.

As can be seen from figure 1, the simple (purely syntactic) profiles exhibit higher stability when compared to the partly enriched ones of the same sample size, while these, in turn, are more stable then fully enriched ones. This is unsurprising, since as profiles include more information (in this case about morphology), they become more granular, and require larger samples to become stable. For the same reason, the PROIEL format gives slightly higher stability than the SynTagRus format for the profiles of the same type. In general, the differences between simple, partly enriched and fully enriched profiles are more salient than those between the SynTagRus and PROIEL schemes.

Stability per se, however, tells us very little about how useful the profiles can be, how much information they contain. To estimate that, we measure another parameter, the distinction power, or how unique each profile is, how well it can identify its verb. To do that, we performed a similar procedure, going through different sample sizes, drawing two non-intersecting, but otherwise random samples for each verb and using them to build two profiles. One of these profiles was included into a control set, one into a test set. We then tried to use the profiles to match verbs. For every profile from the test set, the intersection rate was calculated between it and every profile from the control set, and the control profile which exhibits the largest intersection rate was considered a match. Afterwards, we checked whether the matched profiles actually belonged to the same verb (success) or not

$v = 5$ for $n = 500$) is accounted for by bootstrapping: the procedure of drawing two random samples for each verb was repeated m times, where m is a minimal possible number that satisfies the condition $v * m \geq 1000$ (for instance, $m = 2$ for $n = 10$, $m = 200$ for $n = 500$) The total number of datapoints is thus never less than 1000.

(failure). To increase robustness, the procedure was repeated 30 times, drawing different samples each time. The average results for selected sample sizes are provided in table 2. Note that sample size here refers to the sample size of profiles, i.e. the number of examples used to build every profile. Since we are interested in how the profile sample size influences its distinction power, or its uniqueness, we need to compare the success rates across sample sizes. For that purpose we fix the number of verbs being matched at 27 (the number of verbs available at the largest analyzed sample size), otherwise the success rate would depend not only on the profiles' properties, but also on the number of verbs in the test set. The baseline is thus always 0.037 (=1/27, the random level).

sample	SynTagRus			PROIEL		
size	simple	partly enriched	fully enriched	simple	partly enriched	fully enriched
5	0.258	0.301	0.423	0.252	0.312	0.386
55	0.759	0.843	0.937	0.798	0.829	0.922
105	0.828	0.854	0.959	0.853	0.886	0.940
155	0.950	0.964	0.992	0.949	0.966	0.990
205	0.949	0.985	0.996	0.983	0.996	0.999

Table 2: Average success rate at the matching task across profile sample sizes.

For both schemes and all profile types, the differences between the achieved success rate and the baseline (the performance of a random guesser) are highly significant at every sample size, and effect sizes are immense. Note that even at a sample size as small as 5, the profiles are of some use, at 55 results are very good, and after 105 they start approaching the ceiling. Importantly, the more enriched the profiles are, the higher distinction power they possess. Unlike with the stability parameter, here increased granularity does the profile good service, allowing it to "recognize" the verb better.

We see that PROIEL does slightly better than SynTagRus with simple and partly enriched profiles, despite the fact that its profiles are less granular. This is probably due to the fact that the PROIEL relations are better correlated with the morphosyntactic properties of the argument, and this policy serves the given purpose better than SynTagRus lexically-based annotation. With fully enriched profiles, when information about argument morphology is available, SynTagRus slightly outperforms PROIEL.

Both types of differences are more salient for small sample sizes, with large samples, the performance is always almost at the ceiling level.

5 Russian aspect

Modern Russian has a system where aspect is expressed lexically through opposing pairs (*pisat'*, *na-pisat'* resp.'write.imperfective' and 'write.perfective'). Traditionally, context substitution tests are used as criteria to establish such partnerships,

and also to tell whether a verb is perfective or imperfective. However, these tests rely on native speaker intuitions, and cannot be applied to historical data. Also, the native speakers are not always sure about partnerships even in the modern system [7, 102–106]. We therefore try to apply constructional profiles to these two tasks. To find out if we can use them as an objective criterion in the study of Russian aspect, we need to know what the profiles measure. We suggest two possibilities.

The first possibility is that constructional profiles may serve as a measure of synonymy, as attempted for nouns in [4]. We expect Russian aspectual partners to be near-perfect synonyms, ideally distinguished only by aspectual properties. If we can use the constructional profiles as a measure of close synonymy, we should be able to identify a verb's aspectual partner by the two verbs' intersection rate, as [7] suggests.

The second possibility is that the constructional profiles of aspectual partners may serve as a measure of dissimilarity between aspectual partners rather than of similarity. First, it may be the case that verbal constructional profile data will not group the best synonyms together, but rather cluster verbs into syntactically relevant verb classes, perhaps similar to Levin classes [8]. Clustering experiments using argument frame data similar to ours (see e.g. [12] and [13]) suggest that this may be so. Second, it may be that aspectual partners differ systematically with respect to argument realisation. We know that perfective and imperfective verbs have at least somewhat different preferences, mostly due to the perfective verbs' affinity for specific arguments and the imperfective verbs' affinity for non-specific or generic arguments. If this is the case, then constructional profiles may serve as a basis for guessing the aspect of a given verb.

5.1 Aspectual partners

In order to test whether constructional profiles can serve as an objective criterion for establishing aspectual partners, we select all verbs that, according to the SynTagRus annotation, have a partner. From this subset of paired verbs we select those where both partners have a frequency higher than a given cutoff. From these verbs, two are considered to be partners iff, first, one is imperfective and one is perfective ("homoaspectual marriages" are forbidden), second, the perfective partner's profile has the highest intersection rate with the imperfective partner's profile among all perfective verbs, third, the imperfective partner's profile has the highest intersection rate with the perfective partner's profile among all imperfective verbs. In other words, "polygamy" is forbidden, too: one verb can have either one partner or no partners (if the guesser fails to find a partner that fulfills the criteria above). We use the fully enriched profiles since they are the most informative, and run the experiment for both annotation schemes. The results for different cutoffs are provided in table 3. Note that the first column shows the frequency cutoff, not the actual profile sample size. Sample sizes are different for every verb, since the guesser tries to use all available information, i.e. all the examples with the given verb.

While the results for high sample sizes (and smaller verb sets to choose a part-

frequency cutoff	SynTagRus	PROIEL	number of pairs
10	0.21	0.17	992
60	0.67	0.55	164
110	0.80	0.71	90
160	0.78	0.81	54
210	0.80	0.87	30
260	1.00	1.00	20
310	1.00	1.00	14

Table 3: Success rates (share of pairs correctly identified) of aspect partner matching

ner from) look quite impressive, it is important to remember that the guesser has been given a lot of useful information: first, it knows that every verb in the test set should have a partner; second, it knows which verbs are perfective and which are imperfective. If this information is removed, the performance decreases significantly. While the results are theoretically interesting (constructional profiles do allow us to find partners under certain conditions), we do not currently see any possibility to put them to practical use. If we, for instance, are faced with Old Russian data, where samples are not going to be large (but verbs are numerous), and it is not known with certainty which aspect a given verb belongs to, and whether it has a partner, the guesser, at least in its current simple form, will not be of much help.

At this task, the SynTagRus native annotation scheme outperforms PROIEL for smaller cutoffs, but loses out at higher ones.

5.2 Identifying aspect

In the second task, we try to identify the aspect of a given verb. We perform the experiment on our modern Russian data, but with a view to apply similar methods to Old Russian data. While for modern Russian it is almost always instantly obvious to a native speaker whether a verb is perfective or imperfective, that is not true for Old Russian, and intuition can easily lead to errors.

First, we create two average aspectual profiles. The imperfective profile contains frames from profiles of all imperfective verbs and arithmetic means of their relative frequencies in every profile. The perfective profile is created by the same method applied to perfective verbs. Verbs with frequencies lower than 10 are not included in the average profiles.

We then perform the comparison task: for every verb with a frequency higher than a given cutoff a constructional profile is created, and the profile is compared with both the average perfective and the average imperfective profile. Importantly, the profile of the verb being tested gets excluded from the respective average profile (since we pretend that we do not know the aspect of this verb, we should not be able to decide whether its profile should be included into perfective or imperfective average profile; in other words, the guesser does not know anything about the verb).

The guesser then chooses the aspect depending on which average profile has the higher intersection rate with the verb profile.

The results can be seen in table 4. We report only the best results, which can be achieved using the partly enriched profiles in the PROIEL format. Fully enriched PROIEL profiles give almost the same results, but do marginally worse at some sample sizes. The same is true for fully enriched SynTagRus profiles, while the discrepancies are bigger. Partly enriched SynTagRus profiles do noticeably worse. It is not quite clear what the reason for these effects is, but the most plausible hypothesis would be that the SynTagRus scheme and the full morphological information make the profiles too granular for the given task (which, as we have shown, decreases their reliability), while the partly enriched PROIEL-format profiles turn out to be "just right".

frequency cutoff	success rate	number of verbs
10	0.71	1710
60	0.75	375
110	0.76	197
160	0.78	120
210	0.74	80
260	0.71	65
310	0.72	50
360	0.76	33
410	0.78	27
460	0.78	18

Table 4: Success rate of the aspect identification task

The differences between our guesser and the baseline (random choice of aspect) are significant at the 0.05 level (two-sided proportions test with Yates' continuity correction) for cutoffs smaller than 210.[8] Higher cutoffs, as can be seen from the table, result in smaller samples, and the significance testing lacks power. For our purposes, however, the smaller sample sizes are the most interesting, and the performance is never lower than 70% here. Given that a more sophisticated guesser can potentially be devised, these results give some hope about constructional profiles serving as an independent diagnostic for determining the aspect of a given verb.

6 Conclusions

In this paper we explored the notion of verbal constructional profiles from two main angles using Russian data on two dependency grammar formats.

[8]It can be argued that the baseline should be represented not by a random guesser, but by one which always chooses the answer "imperfective", since imperfective verbs are more numerous than perfective at any cutoff. However, an adequate measure of performance would then be not accuracy, but F-score (with perfectives being a positive class). While F-scores for our guesser will be lower than plain accuracy, for the "always-imperfective" guesser they will always be 0.

Our first angle was chiefly methodological. In section 4 we established that the existence of a "true" distribution can be claimed. Learning curves like Figure 1 can be used to determine what sample size is required to get a stable profile and claim a reasonable approximation to the "true" distribution. We refrain here from providing any specific thresholds, since the numbers, as has been shown, depend on the profile type and annotation scheme and, probably, also on the language in question. We can, however, note that for small samples (10–20 examples) stability is low. This is important to know, since in some cases researchers are forced to rely on small samples. One example would be studies on infrequent verbs (and, as is well-known, the majority of words are infrequent), another one – studies based on relatively small treebanks (e.g. of exotic or ancient languages). As expected, less granular profiles become stable at smaller sample sizes than more granular profiles, so at this task simple PROIEL profiles performed best.

When we looked at the distinction power of the constructional profiles, however, granularity proved to be an advantage, and the most granular profiles (SynTagRus fully enriched profiles) performed best. Even very small samples allow us to identify some verbs, and decent-size samples do very well. Enriched profiles, being more granular, require larger sample sizes, but for the same reason have higher distinction power than simple profiles. Interestingly, unless fully enriched profiles are used, the less granular PROIEL scheme outperforms the more granular SynTagRus scheme at the matching test nonetheless, probably due to the former scheme's sensitivity to the morphosyntactic category of the argument. With fully enriched profiles, however, the PROIEL scheme loses its advantage, and the more granular SynTagRus shows slightly better results.

In general, we saw that both schemes provided good argument frame data, but that the PROIEL scheme's small set of syntactic relation labels were more informative than the larger set of SynTagRus valency labels when using frames unenriched with argument morphology.

In section 5 we turned to practical applications. We found that verbal constructional profiles may be used both to guess aspectual partners and to establish the aspect of a given verb. Granularity was also an important parameter in these tasks: aspect partner matching works best with fully enriched SynTagRus profiles, whereas aspect identification worked best with partly enriched PROIEL profiles. While the aspect partner matching task was mostly interesting from a theoretical point of view (the method is not applicable for low- and medium-frequency verbs), the aspect identification task gave more hopeful results. Even at very low frequency cutoffs, we had good results with determining aspect. This is a measure that may potentially serve as an independent criterion to decide the aspectuality of verbs in diachronic studies.

7 Acknowledgments

We are indebted to the Laboratory of Computational Linguistics at the Institute for Information Transmission Problems (Moscow) for the use of the offline version of SynTagRus and advice on its usage, and to the members of the CLEAR group at UiT The Arctic University of Norway and three anonymous TLT13 reviewers for their comments and suggestions.

References

[1] Apresian, Ju.D., Boguslavskij, I.M., Iomdin B.L. et al. (2005) Sintaksičeski i semantičeski annotirovannyj korpus russkogo jazyka: sovremennoe sostojanie i perspektivy. In: *Nacional'nyj korpus russkogo jazyka: 2003–2005*. Moscow: Indrik.

[2] Eckhoff, Hanne Martine, Janda, Laura and Nesset, Tore (2014) Old Church Slavonic BYTI Part Two: Constructional Profiling Analysis. *Slavic and East European Journal* 58:3.

[3] Gries, Stefan and Otani, Naoko (2010) Behavioural properties: a corpus-based perspective on synonymy and antonymy. *ICAME Journal* 34.

[4] Janda, Laura and Solovyev, Valery (2009) What constructional profiles reveal about synonymy: A case study of Russian words for SADNESS and HAPPINESS. *Cognitive Linguistics* 20:2.

[5] Haug, Dag (2010) *PROIEL guidelines for annotation*. http://folk.uio.no/daghaug/syntactic_guidelines.pdf

[6] Haug, Dag, Jøhndal, Marius, Eckhoff, Hanne, Welo, Eirik, Hertzenberg, Mari and Müth, Angelika (2009) Computational and Linguistic Issues in Designing a Syntactically Annotated Parallel Corpus of Indo-European Languages. *Traitement Automatique des Langues* 50:2.

[7] Kuznetsova, Julia (2013) *Linguistic profiles: Correlations between form and meaning*. PhD thesis. Tromsø: University of Tromsø.

[8] Levin, Beth (1993) *English verb classes and alternations: a preliminary investigation*. Chicago: University of Chicago Press.

[9] Mel'čuk, Igor' Aleksandrovič (1995) *Russkij jazyk v modeli "Smysl–Tekst"*. Vienna: Škola "Jazyki russkoj kul'tury".

[10] Hajičová, Eva and Sgall, Petr (eds.) (1999) *Annotations at analytical level*. http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/a-layer/html/index.html

[11] Russian National Corpus *Sintaksičeski razmečennyj korpus russkogo jazyka: instrukcija pol'zovatelja*. http://ruscorpora.ru/instruction-syntax.html

[12] Schulte im Walde, Sabine and Brew, Chris (2002) Inducing German Semantic Verb Classes from Purely Syntactic Subcategorisation Information. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia.

[13] Schulte im Walde, Sabine (2004) Automatic Induction of Semantic Classes for German Verbs. In: Langer, Stefan and Schnorbusch, Daniel (eds): *Semantik im Lexikon*. Tübingen: Gunter Narr Verlag.

Towards a Universal Stanford Dependencies parallel treebank

Cristina Bosco and Manuela Sanguinetti

Department of Computer Science
University of Turin
{bosco;msanguin}@di.unito.it

Abstract

The paper describes the process conceived to convert a multilingual parallel treebank, namely ParTUT, into an annotated resource that conforms to the representation format and specifications of the Universal Stanford Dependencies (USD).

The main goal of this work is to create, taking an already existing resource as the starting point, a fully parallel treebank that is featured by a widely-known and used representation format, i.e. that of the Stanford Dependencies, and in particular its cross-linguistic variant, namely the Universal Stanford Dependencies, in order to provide a useful resource for a number of NLP tasks, including those that have typically benefitted from such representation format, such as Information Extraction and statistical parsing, but also translation-related tasks (by virtue of the parallel annotation).

1 Introduction

The increasing need to use multilingual resources for NLP systems goes hand in hand with the opportunity to make such resources available and accessible. Nevertheless, the specific morphological and syntactic features of the language at issue, or the end use the resources have been designed for, can motivate the exploitation of different representation formats, thus limiting their accessibility and portability. As for treebanks in particular, a few steps towards the harmonization of datasets and formats that could be easily shared by the community have recently led to the spread of the Stanford Typed Dependencies (SD) [5] as a *de facto* standard, and to its cross-linguistic variant, namely the Universal Stanford Dependencies (USD) [4].

A further step, beyond the development and application of the SD and USD formats in multilingual resources, is the development of *parallel* and possibly aligned data in SD and USD, which could be crucial for translation-related tasks.

This motivates our present work, which consists in the development of a parallel treebank in USD format that could benefit from the annotation scheme and

therefore could be further exploited for translation purposes.
Therefore, the paper describes a project of conversion into USD of an existing parallel treebank for Italian, English and French. In particular, we discuss the challenges encountered in the application of a SD format in a cross-linguistic perspective.
The paper is organized as follows: after a section where the main SD-related formats are reported, we briefly describe the resource to be converted, while Section 4 and 5 are devoted to the description of the approach adopted and to some of the main challenges encountered during conversion.

2 Related Work on Stanford Dependencies

While the SD format has been primarily conceived for English, as also shown by the recently-developed English Web Treebank [15], in the last few years, it has been partly re-designed to be successfully used for other languages, such as Hebrew [16] and Finnish [6], even converting resources that were previously available in other formats, instead of developing new datasets from scratch: this is the case, for example, of the Italian Stanford Dependencies Treebank [1] and the set of six treebanks (English, French, German, Spanish, Swedish and Korean) annotated in basic SD [10].
The success and applicability of the SD standard to practical tasks is also attested by its use in experiments like in NLP contests, such as the shared tasks on Dependency Parsing for Information Extraction (DPIE) and Cross-Language Dependency Parsing (CLaP) at EVALITA 2014[1], the evaluation campaign for Italian NLP tools. However, observing the SD scheme in the perspective of the application to different text types, two main limitations have been detected and reported in [9], i.e. that the scheme does not offer coverage of a large variety of complex syntactic constructions and of constructions typical of informal language.

An attempt to re-build the SD taxonomy in a cross-linguistic perspective is finally proposed with the so-called Universal SD [4], which has also been applied to experiment the "stanfordization" of thirty dependency treebanks in the HamleDT project [12].

As regards the conversion project presented here, our main reference is namely the USD format, and in a similar fashion to most of the conversion projects mentioned above, we do not aim to create a brand new resource using USD as a native representation format, but we rather develop a USD-based parallel resource starting from an already existing one. This is motivated by the need to use a universal annotation standard that could preserve a higher degree of parallelism between the language pairs involved, in order to exploit the parallel annotation for translation purposes.

[1] http://www.evalita.it/

3 ParTUT and its content

The parallel resource to be converted is ParTUT[2], a dependency-based treebank of Italian, English and French, developed as the multilingual extension (thus using the same principles and annotation format) of the Turin University Treebank (TUT)[3].

The whole treebank currently comprises an overall amount of 148,000 tokens, with approximately 2,200 sentences in the Italian and English sections, and 1,050 sentences for French (being the resource under constant development, the French part of the newest texts recently added to the collection is yet to be analyzed and included).

Although the first release of the resource mostly included parallel texts from the legal domain, its later extensions opened to a larger variety of genres, in order to provide a broader selection of texts for a more efficient training of parsers. As a matter of fact, it has been shown that the performance of parsing systems can be improved by training and testing mixed genres rather than formal and throughly revised texts [15]. Bearing this principle in mind, the resource has been recently enriched with fairly different text genres that include web articles from Project Syndicate[4] and Wikipedia articles retrieved in English and French and then translated into Italian by two teams of graduate students in Translation.

ParTUT consists of human-checked sentences analyzed using the Turin University Linguistic Environment (TULE) [8], a rule-based parser first developed for Italian and then adapted for English and French.

One of the main uses of the treebank is the creation of a system for the automatic alignment of parallel sentences taking into account the syntactic information provided by annotation [14].

4 Converting ParTUT: a rule-based approach

Differences among formats mainly deal with the type of relations, their granularity, and the different notions of specific phenomena assumed in the representation, such as coordination, subordinate clauses, verb groups; these were the aspects we mainly focused on while designing the conversion process.

We conceived our approach to conversion in a similar fashion to Rosa et al. for HamleDT [12], attempting to adhere as much as possible to the original USD core scheme and label set. However, contrarily to HamleDT, where most of the adaptation concerned the label set only, the conversion process from TUT also entailed a partial or complete reshaping of subtrees. We thus identified three classes of relations, according to the way they had to be treated:

- R-renamed: most of the relations had to be simply renamed, as they represent the same grammatical function and link the same elements in both formats;

[2] http://www.di.unito.it/~tutreeb/partut.html

[3] http://www.di.unito.it/~tutreeb/, around 3,500 Italian sentences and 102,000 tokens.

[4] http://www.project-syndicate.org/

e.g. VERB-RMOD+RELCL[5], that is the relation linking the head node of a relative clause to the rest of the sentence, is renamed *relcl*, and PDET-RMOD, which links predeterminers with determiners, is renamed *predet*.

- R-swapped: besides the renaming, the conversion of these relations involves the direct swapping between governor and dependent. This class includes, in particular, the relation linking determiner with noun and preposition with its argument; TUT in fact follows the tenets of the Word Grammar [7], where determiners and prepositions are complementizers of content words, while in SD content words are usually considered as heads of the structure where they occur. Also the relation linking the modal with the main verb, and the one linking the predicative complement to the copula (see Figure 1) are in this class.

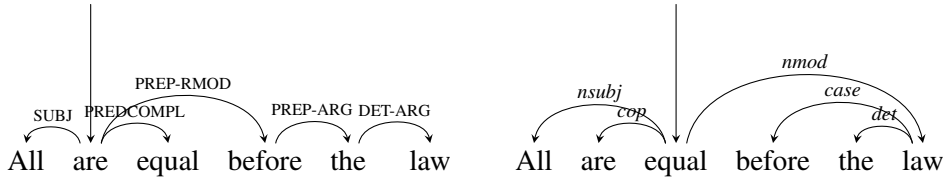


Figure 1: An example of copulative verb in TUT format (on the left) and its conversion into USD (on the right).

- R-reshaped: relations whose conversion consists in a full reshaping of the structure where they occur. This class includes, for example, elliptical structures, where the elided material is usually preserved in the TUT representation by means of co-indexed traces, while USD scheme replicates the representation pattern from Lexical Functional Grammar [2], adopting the *remnant* relation (see Figure 2[6]).

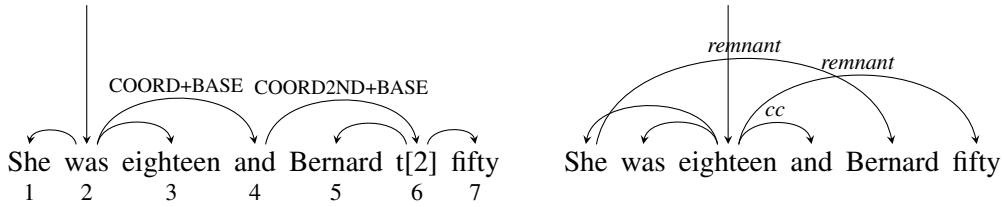


Figure 2: Elliptical structure in TUT format (on the left) and its conversion into USD (on the right).

[5]According to the usual notation of TUT and SD, the relations names are in capital letters and in italics respectively.

[6]For the sake of readability, we just retained the more relevant labels.

Comparing TUT and SD format, different granularities of relations often occur (see also 5). The former has been designed in principle for a detailed representation of morphologically rich languages and also exploits null elements in order to generate a full description of the predicate argument structure. The latter aims to provide a simple and intuitive way of annotating dependencies [9]. Different granularities of relations are treated during conversion in two ways: when SD is less specific than TUT, which is the more frequent situation, two or more TUT relations are labeled with a same SD relation, while when TUT is less specific than SD, we constrain the conversion of the single TUT relation with conditions that allow the selection of two or more different relations in SD. For instance, the relation linking the determiner with the noun, i.e. DET-ARG in TUT, is converted in *det* when the determiner is an article and *poss* when it is a possessive adjective.

Furthermore, the implemented conversion includes preprocessing steps for the adaptation of the notation and the conversion of the Part-of-Speech tagset. The adaptation step basically deals with the shift from the word-centered notation of the TUT format[7] to the USD notation, which is (in compliance with basic SD) centered upon relations.
As regards the PoS tagset, according to the parameters set in the conversion, it can be currently maintained in the native TUT, although we are also developing the conversion into the tagset proposed in the USD guidelines[8], which is an extended version of the Google universal tagset [11].
Finally, the resource will be also released in the revised CoNLL-U format[9].

Following the relation taxonomy used by the authors in the provided documentation (see previous footnotes), Figure 3 summarizes how the main TUT relations are mapped onto the USD scheme. The '(...)' sign that surrounds some of the relations in the TUT column represents the possible morpho-syntactic specifications of the relations.

4.1 Mapping functions

Each mapping function is structured according to the following pattern:

`TUTrelation,SDrelation(Xfeaturetype:condition)`

where `TUTrelation` is namely the relation to be converted and `SDrelation` is its counterpart, the `X` stands for the element (`G` = governor, `D` = dependent) and the `featuretype` for the feature (`mor` and `syn` for morphological and syntactical features respectively) to which the condition is applied, while `condition` represents a condition that can be required for the relation mapping.
If the slot contains one (or more) condition, it should then be verified on the appropriate element (`G` or `D`) and/or feature (`mor` or `syn`):

[7] Each TUT annotation line includes a word W, its PoS tag and the morphological features, the link to the governing node and the relation that labels that link.

[8] `http://universaldependencies.github.io/docs/ud-pos-index.html`

[9] `http://universaldependencies.github.io/docs/format.html`

TUT	USD
Core dependents of clausal predicates	
VERB-SUBJ	*nsubj, csubj*
VERB-OBJ	*dobj, ccomp*
VERB-INDOBJ	*iobj*
VERB-OBJ/VERB-SUBJ	*nsubjpass, csubjpass*
VERB-PREDCOMPL+OBJ	*xcomp*
Non-core dependents of clausal predicates	
ADVB-RMOD	*advmod*
ADVB-RMOD-NEG	*neg*
VERB+FIN-RMOD	*advcl*
NOUN-RMOD	*nmod*
NOUN-RMOD-TIME	*tmod**
Special clausal dependents	
NOUN-APPOSITION-VOCATIVE	*vocative*
VERB-EXPLETIVE/VERB-SUBJ	*expl*
AUX+TENSE, AUX+PROGRESSIVE, VERB+MODAL-INDCOMPL	*aux*
AUX+PASSIVE	*auxpass*
VERB-PREDCOMPL+SUBJ	*cop*
CONJ-ARG	*mark*
END, SEPARATOR	*punct*
Noun dependents	
ADJC(...)-RMOD	*amod*
DET+DEF-ARG	*det, poss**
DET+INDEF-ARG	*det*
DET+QUANTIF-ARG	*neg, nummod, det*
PDET-RMOD	*predet**
NOUN-RMOD, NOUN-OBJ, NOUN-SUBJ	*nmod*
NUM-RMOD	*nummod*
VERB-RMOD+RELCL	*relcl**
(...)APPOSITION(...)	*appos*
Case markers	
PREP-RMOD	*case*
Compounding and unanalyzed	
CONTIN	*goeswith*
CONTIN+LOCUT	*mwe, compound, name*
CONTIN+DENOM, NOUN-APPOSITION-DENOM	*name*
PARTICLE	*prt**
Coordination	
COORD(...)	*cc*
COORD2ND()	*conj, cc*
COORDANTEC(...)	*preconj**
Loose joining relations	
NUM-RMOD-LISTPOS	*list*
VERB-EXTRAOBJ, VERB-EXTRASUBJ	*dislocated*
Other	
TOP(...)	*root*
DEPENDENT, VISITOR	*dep*

Figure 3: Mapping scheme of (the more relevant) TUT relations onto USD. USD relations marked with a * represent all those English-specific labels for which a direct counterpart in the TUT format already exists.

- conditions with `featuretype = mor` require the presence of a particular morphological feature in the Part of Speech of the X element of the TUT relation
  Ex.: `DET+DEF-ARG, det (Gmor: ART DEF)` is the mapping function that falls in the R-swapped category described above, and that specifies that the TUT relation for definite determiners `DET+DEF-ARG` should be renamed as `det` if the governor of the TUT relation satisfies the morphological condition (`mor`) of being a definite article `ART DEF`.

- conditions with `featuretype = syn` require the presence or absence (indicated by + or −) of a particular syntactic structure associated with the X element of the TUT relation;
  Ex.: `VERB-INDOBJ, nmod (Gsyn + DEP& PREP)` is the mapping function that falls in the R-renamed category, and that specifies that the TUT relation for indirect objects `VERB-INDOBJ` must be renamed as `nmod` when the given node has a dependent (`DEP`) which is a preposition (`PREP`).

If a rule has more than one condition, this will be of the form:

```
TUTrelation,SDrelation(Xmor:condition;Xsyn+/-:condition).
```

5 Discussion

This section attempts to illustrate some of the relevant aspects that involved this conversion project.

As mentioned above, in this first stage of development of our work, we decided to fully adhere to the USD annotation principles and label set, in order to comply with the standard, and, at the same time, to assess its actual applicability to the parallel texts in ParTUT. However, a number of issues raised from the conversion, as we tested it on the texts of the collection. These aspects mostly had to do with the specific features of the two formats, and with the theoretical as well as technical reasons that lie behind their conception, rather than the differences between the language pairs involved. The observations emerged both from the design of the conversion model and the actual results obtained with the automatic process can be basically reduced to three main issues, that we will attempt to examine in this section; they regard: the differences in annotation granularity, the uniformity in the parallel annotation, and the actual coverage, using the current label set and representation scheme, of the linguistic phenomena encountered in the collection.

Different granularities: pros and cons of both formats As far as the difference in annotation and label granularity is concerned, with its set of 11 *morphoSyntactic* and 27 *functionalSyntactic* features can be combined together, the TUT format has a far richer scheme compared to the one adopted in USD (where the taxonomy includes an overall amount of 43 relations). This intuitively let one assume

that conversion from finer- to coarse-grained labels would be more straightforward. However, the rationale behind these differences lies namely on the different principles and linguistic layers that are meant to be described, such as the predicate-argument structure in the TUT representation format. Let us consider, for example, the case of predicative modifiers. TUT introduces a distinction between arguments and adjuncts that, because of its subtlety, has been intentionally removed in SD and USD. This results in the absence of a proper counterpart for the TUT relation used to express the predicative modifier (RMODPRED), which has then been considered as a simple modifier (either nominal, adjectival or clausal) according to the USD scheme.
On the other hand, while in USD compounding is distinguished from phrasal modification by using a specific label (that is namely *compound*), such distinction is not drawn in TUT, where they are generically defined as modifiers (NOUN-RMOD or NUM-RMOD, depending on whether they include nouns or numerals). As a result, it is quite impossible to keep this distinction during the automatic conversion, and a manual post-editing step in necessarily required in those cases.

Uniformity in parallel annotation One of the advantages of a "universal" representation scheme, as claimed by [4], is "*the parallelism between different constructions across and within languages*" (p.3). In fact, besides the release of a *parallel* treebank into a widely-recognized format that enables its portability, the second main goal that motivated this conversion is the exploitation of the USD representation for translation purposes and in the ongoing development of a syntactic alignment system that could overcome, benefitting from the dependency-based annotation, the problems raised by the so-called *translation shifts* [3, 13]. We report therefore some qualitative observations about the capacity of USD for preserving a higher degree of parallelism under a structural perspective and better serving the purposes of an automatic alignment system with respect to TUT. We considered a small sample consisting of 20 sentences from each language section and annotated in USD, taking into account two main criteria: *i*) all the sentences must have a 1:1 correspondence, this means that each subset of the sample is a triple containing the same sentence in the English, French and Italian version respectively; and *ii*) they must contain at least one example of translation shifts (see [13]). In line with the *lexicalist* approach adopted for the USD design, whereby grammatical relations should hold between content words, in our observations on the sample data, we considered the latter only and defined the set of dependency edges that link them in a sentence as the *core* structure of that sentence. We thus compared each annotated triple, and attempted to determine the frequency of the following cases: *i*) same core structure, i.e. content words are linked with the same edges and relations in each sentence of the triple, as the example in Figure 4 (which also highlights an important difference with respect to TUT as regards the head-selection criteria); *ii*) similar core structure, i.e. content words are linked with the same edges in each sentence of the triple, but with (either partially or completely) different relations

(see Figure 5); *iii*) different structures.

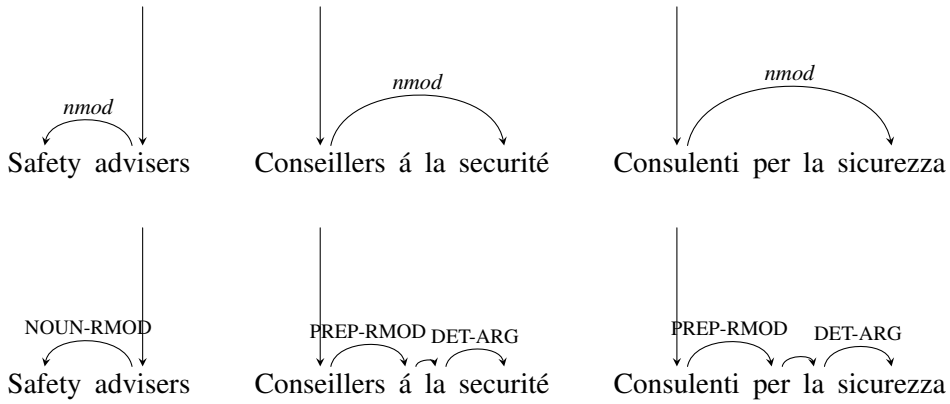


Figure 4: An example of same core structures in USD (upper part), despite the presence of a translation shift from nominal (English version) to prepositional modification (French and Italian counterparts). Because of the differences in the original conception of the representation format, TUT annotation (lower part) offers a divergent version of the example proposed that does not preserve this uniformity .

For the cases *i* and *ii*, we also verified whether such conditions occur in only two out of three sentences of the triple. What emerged from this comparison is that most of the sentence triples had the same core structure (around 60%), and less than 17% had different structures. This seems to confirm the advantage of the USD scheme, with respect to TUT (as also shown in Figure 4).

Coverage of label set and representation conventions As regards this aspect, the coarser granularity of relations offered by the USD scheme makes it possible their applicability to a wide range of linguistic phenomena. This allows us, for the time being, to exploit the resource without the need to enrich the taxonomy with new labels, and to stick to the original intention, which is to remain faithful to the proposed standard. It should also be added that we used for Italian and French as well a large part of those relation labels that in the scheme proposed in the official documentation are referred to as English-specific. This sub-set of relations is easily identified in Table 3 as marked with an asterisk ('*'). Furthermore, except for *prt* (the label expressing particles in phrasal verbs), that is used in ParTUT for English only, all the remaining relations listed in the table can be applied to all the three languages of the treebank.

Beyond the observations reported above, however, we detected that the USD scheme offers a limited coverage for what concerns many, albeit rare, phenomena (which is an aspect that has also been observed in [9]). In particular we did not find in USD the means to deal with comparative structures, which often occur in a rich repertory of variants especially in Italian and French. Also the representation

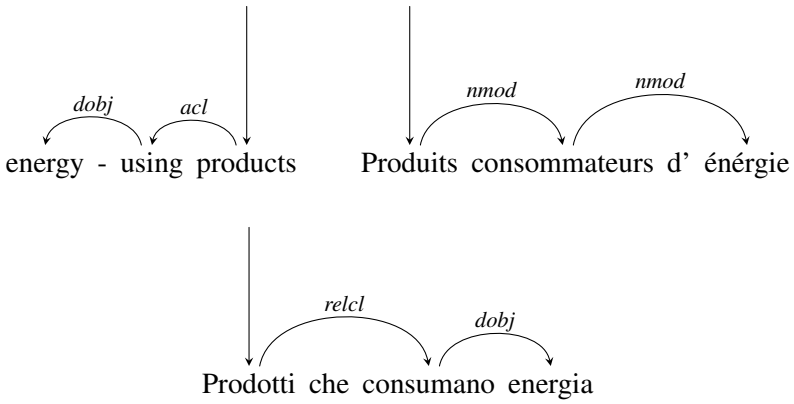


Figure 5: Example of similar core structures in the presence of a shift in the morpho-syntactic category of the root modifier (that is verb in the English and Italian example and noun in the French version).

of various types of relative clauses have been kept underspecified in the current version of ParTUT in USD format.

This motivates our plans for the future advancement of this work, where we intend to design and apply more adequate representation of a larger variety of linguistic phenomena, following the hints of [9] and benefitting from the richness of the source format, in particular the explicit annotation of null elements.

6 Conclusion and future work

In this paper we presented an ongoing work on the conversion of a parallel treebank into the Universal Stanford Dependencies. The main goal of this conversion project is the release of a treebank that is not only annotated according to the principles of a scheme that currently represents a *de facto* standard in treebank design, but also contains parallel texts that, benefiting from a uniform annotation across the different languages, can then be exploited for practical NLP tasks, such as multilingual parsing and Machine Translation.

The next steps are the full release of the treebank and its use as a test of the alignment approach described in [14].

References

[1] Cristina Bosco, Simonetta Montemagni, and Maria Simi. Converting Italian treebanks: Towards an Italian Stanford Dependency Treebank. In *Proceedings of the 7th ACL Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

[2] Joan Bresnan. *Lexical-Functional Syntax*. Blackwell, Oxford, 2001.

[3] John C. Catford. *A Linguistic Theory of Translation: An Essay on Applied Linguistics*. Oxford University Press, 1965.

[4] Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of 9th International Conference on Language Resources and Evaluation (LREC 2014)*, 2014.

[5] Marie-Catherine de Marneffe and Christopher D. Manning. The Stanford Typed Dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 2008.

[6] Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, 2013.

[7] Richard Hudson. *Word grammar*. Basil Blackwell, Oxford and New York, 1984.

[8] Leonardo Lesmo. The rule-based parser of the NLP group of the University of Torino. *Intelligenza artificiale*, 2(IV), 2007.

[9] Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R. Bowman, Timothy Dozat, and Christopher D. Manning. More constructions, more genres: Extending Stanford Dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, 2013.

[10] Ryan Mcdonald, Joakim Nivre, Yvonne Quirmbach-brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu, and Castelló Jungmee Lee. Universal dependency annotation for multilingual parsing. In *Proceedings of 51st annual meeting of the Association for Computational Linguistics (ACL'13)*, 2013.

[11] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12)*, 2012.

[12] Rudolf Rosa, Jan Mašek, David Mareček, Martin Popel, Daniel Zeman, and Zdeněk Žbokrtský. HamleDT 2.0: Thirty dependency treebanks stanfordized. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

[13] Manuela Sanguinetti and Cristina Bosco. Translational divergences and their alignment in a parallel treebank. In *Proceedings of the 11th Workshop on Treebanks and Linguistic Theories (TLT-11)*, pages 169–180, 2012.

[14] Manuela Sanguinetti, Cristina Bosco, and Loredana Cupi. Exploiting *catenae* in a parallel treebank alignment. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

[15] Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel R. Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. A gold standard dependency corpus for English. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, 2014.

[16] Reut Tsarfaty. A unified morpho-syntactic scheme of Stanford Dependencies. In *Proceedings of 51st annual meeting of the Association for Computational Linguistics (ACL'13)*, 2013.

Finding Parse Errors in the Midst of Parse Errors

Markus Dickinson and Amber Smith

Department of Linguistics
Indiana University
E-mail: {md7,smithamj}@indiana.edu

Abstract

We consider a simple method for dependency parse error detection and develop representations which account for parse errors occurring in the context of other parse errors. The core insight is to combine both very detailed representations with more relaxed representations, i.e., ones which allow for errors. We show that for situations with little annotated data to begin with, error detection precision can be greatly increased.

1 Introduction

Parse error detction is useful for improving the quality of large corpora via a semi-automatic correction process [2, 8], for assisting in automatic parse revision [11], and for selecting sentences for active learning [14]. Yet there are few methods for automatic parse error detection, and some are tailored to particular annotation schemes or parsers [e.g., 1, 2]. We consider a simple method for dependency parse error detection and develop representations which account for parse errors occurring in the context of other parse errors [9, 10]. Although our work focuses on one particular method [8], the method is general-purpose and language and parser-independent, and the insights should be applicable to general parse improvement.

Our contributions start with identifying the core reasons for how anomalies are identified in an error detection method [8], in section 2. This analysis highlights how new insights can be integrated and shows why the method is well-suited for situations with little annotated data to learn from. We then outline in general terms how surrounding errors occur in parses, in the beginning of section 3. Such surrounding errors are problematic because methods use full parse information to detect anomalies, and an error in one part affects finding an error in another part. While a human annotator may focus on a sentence-by-sentence correction and thus identify surrounding errors on the basis of one identified error, surrounding errors nonetheless can lead to a decrease in error detection precision and/or a misjudgment in how severe an error is. Furthermore, for tasks such as parse correction, there may be no human involved in the correction process.

Building on the types of surrounding errors, we propose in the rest of section 3 new information to include in the model, which increases the precision of the method for a workable portion of the corpus, as shown in the evaluation in section 4. The core insight is to combine both very detailed representations with more relaxed representations, i.e., ones which allow for errors.

2 Parse error detection

2.1 The DAPS method

The method in [8] detects anomalous parse structures (DAPS), using *n*-gram sequences of dependency structures. The training corpus is reduced to a set of rules that consist of a head and its dependents, and then the rules from the parsed testing corpus are scored based on their similarity to rules from training, for heads of the same category. This is a rather simple method (discussed below), but it serves the linguistic function of identifying anomalies in word valency [17].

For example, consider the partial tree in figure 1 from the Wall Street Journal (WSJ) portion of the Penn Treebank [12], converted to dependencies with the Stanford CoreNLP tool [4]. From this tree, rules are extracted as in (1), where all dependents of the `NNS` head (`-H`) are realized.

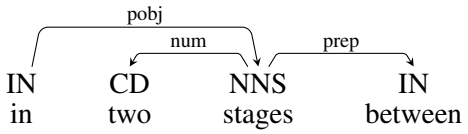


Figure 1: A sketch of a basic dependency tree

(1) pobj → num:CD NNS-H **prep:IN**

Such a parsed rule is then broken down into its component *n*-grams and compared to rules from training, using the formula for scoring an item (e_i) in (2). *N*-gram counts ($C(ngrm)$) come from the same corpus used to train the parser. An instantiation for this rule is in (3), where we obtain a score for the `prep:IN` in (1). Tokens are ranked by score, with lower scores more likely to be errors.

(2) $$s_{base}(e_i) = \sum_{ngrm:e_i \in ngrm \wedge n \geq 2} C(ngrm)$$

(3) s_{base}(prep:IN) = C(NNS **prep:IN**) + C(**prep:IN** `END`)
+ C(num:CD NNS **prep:IN**) + C(NNS **prep:IN** `END`)
+ C(`START` num:CD NNS **prep:IN**)
+ C(num:CD NNS **prep:IN** `END`)
+ C(`START` num:CD NNS **prep:IN** `END`)

We use two scoring variants: 1) the high-gram method (*high*) uses all *n*-grams of length 3 or greater ($n \geq 3$), as bigrams do not encode much context [8]; and 2) a weighted version of an all-gram method (*wall*), which multiplies bigram counts by a dampening weight of 0.01: thus, if a rule lacks a bigram, it is more likely that the rule is of poor quality than if it simply lacked a trigram [see also 11].[1]

2.2 Analysis of the method

The method is simple: count up *n*-grams of varying lengths. While there are reasons to consider using more sophisticated methods in the future (e.g., tree kernels [15]), this framework has its advantages. First, it is relatively quick to run; it is easy to replicate; and the output is interpretable. Furthermore, it seems to be effective across a variety of conditions, including for manual annotation. This fits with the general notion that effective error detection methods, acting as sanity checks on complicated processes, are often relatively simple [6]. Secondly, it allows for arbitrarily long and complex pieces of information. Thirdly, it handles small amounts of data well, even with large numbers of unique *n*-grams.

The method counts up features (*n*-grams) during training and then points out in the parse those positions whose features have very low counts, often zero. In other words, the positions identifed are those whose features are sparse. If more features are added, the chance of accidentally obtaining a low score is decreased, potentially increasing precision—assuming that the features are truly informative. The benefit for scenarios with small amounts of annotated data is that, while there may be few ways to gather new data, there are new ways to splice up the data one already has. We investigate some of these new ways in this paper.

3 Accounting for contextual errors

One problem with the method as it stands stems from errors occurring together in the same rule: an item's (in)validity may be clouded by an erroneous sister. In the constructed (4), for instance, the validity of `DT:Det` attaching to the `Noun` head is occluded by the presence of the erroneous `MOD:Adv`. Instead of `DT:Det MOD:Adj Noun-H`, we get `DT:Det MOD:Adv MOD:Adj`, a trigram absent from training. In this particular case, then, we may unfairly flag `DT:Det` as an error.

(4) **DT:Det** MOD:Adv MOD:Adj Noun-H

Our goal is to account for erroneous contextual items (i.e., sister dependents) in scoring a rule. Types of contextual errors include:

1. (a) **Mislabeling:** A dependency has the wrong dependency label (`A Y:X C` should be `A Z:X C`).

[1] The code here implements *high* by default and requires only minor modification for *wall*: `http://cl.indiana.edu/~md7/papers/dickinson-smith11.html`

(b) **Mistagging**: A dependency has the wrong POS label (`A Y:X C` should be `A Y:W C`).

2. **Substitution:** A dependency has the wrong attachment, but something else could go in this position with this head (`A B C` should be `A D C`).
3. **Comission:** A dependency has the wrong attachment, but nothing else goes in this position with this head (`A B C` should be `A C`).
4. **Omission:** There is a missing attachment (`A C` should be `A B C`).

We approach the errors by adding *n*-grams that reflect each type of change, essentially compiling out Levenshtein edits. Given the analysis of the method above, the effect of adding more *n*-grams should serve to increase *precision*, not recall, of errors found: since each dependency relation now has more chances to obtain a higher score, we expect fewer which are still low-scoring, but those cases should be more accurate. Note that: a) each solution is restricted to a single type of edit, some of which are partial substitutions; and b) modifications are for all *n*-grams, not just the rule as a whole [5]. The set-up of adding new *n*-grams also makes clear that the issue is fundamentally one of representation.

Focusing on representation, we hope, makes the insights applicable to other domains, in the same way that representations for corpus annotation error detection have led to, e.g., particular features for native language identification [3] and methods for uncovering annotation definitions [7]. Additionally, although we have only tried individual solutions, future work can explore combinations of solutions; initial results indicate that not all combinations are favorable.

Solution #1a: Removing a dependency label For potentially incorrect dependency labels, our solution is to include *n*-grams which replace a dependency:POS pair with only the POS. Recalling the base formula for calculating an anomaly score in (2) and using a set of modified *n*-grams N', the formula is given in (5) and an instantiation in (6). Here, the POS `CD` replaces `num:CD`.

(5) $$s(e_i) = s_{base} + \sum_{ngrm' \in N' \wedge ngrm' : e_i \in ngrm' \wedge n \geq 3} C(ngrm')$$

(6) s(prep:IN) = s_{base}(prep:IN)
+ C(CD NNS-H **prep:IN**)
+ C(`START` CD NNS-H **prep:IN**)
+ C(CD NNS-H **prep:IN** `END`)
+ C(`START` CD NNS-H **prep:IN** `END`)

The set of modifications N' is subject to constraints. First, only one item in a rule is replaced at a time. Thus, we obtain `START CD NNS-H prep:IN END` and `START num:CD NNS-H IN END`, but not `START CD NNS-H IN END`. This keeps the number of new *n*-grams managable and reflects the assumption that it is less likely

for there to be multiple contextual errors. The second constraint is not to allow heads to be modified—more of an issue when using a dummy label (solution #2) or removing the item entirely (solution #3). We restrict this because the comparison of rules is based on having similar heads; e.g., only rules with NNS heads are compared to the rule in (1). The third constraint only applies when an item is removed (solution #3): items on the edge of an *n*-gram are not allowed to be removed. In this case, we wind up with an already-known bigram, e.g., num:CD NNS-H prep:IN becomes NNS-H prep:IN. Relatedly, for all solutions we do not modify any bigrams (e.g., CD NNS-H), given the questionable nature of bigrams.

Solution #1b: Removing a POS label Having a wrong POS label (A Y:X C should be A Y:W C) seems at first glance to be mainly a concern for when automatic POS tagging is employed, but, as example (4) illustrates, sometimes the root of an inconsistency can be traced to an improper POS tag (e.g., the problem for (4) is the presence of the Adv(erb) POS tag). We thus remove the POS tag and keep the dependency label (e.g., in (6) replace every CD with num).

Solution #2: Skipping items To handle an incorrectly-attached sister item when some other item could attach in this position (e.g., A M:X C should be A N:Y C), we replace items in a rule with a dummy token (SKIP) (e.g., START SKIP NNS-H prep:IN). This solution is fairly broad, as the dummy SKIP stands in for any substituted item (including relabelings, as in solution #1a), as well as the original item.

Solution #3: Removing items When a dependency has the wrong attachment, but nothing else goes in this position (e.g., A B C should be A C), the simple modification is to remove a potentially erroneous context item (e.g., B). However, training and testing must be handled differently, as removing an item in training would be an incorrect sequence (e.g., A C in this case). Thus, we obtain training rules in the basic way (s_{base} in (3)), and then in testing add rules which remove an item.

Solution #4: Inserting items Items which should have been attached to a head (e.g., A C should be A B C) lead to positing the insertion of an item between two other items in a rule. While we might want to posit every single intervening token, with every possible label, as the inserted item (B), this is cumbersome. For this exploration, we simply ask whether *something* could be inserted between the two items. Training consists of generating extra *n*-grams with dummy SKIPs appropriately inserted, exactly as in solution #2. For testing, we then insert SKIP items between items in a rule if there is an item between them in the linear string (A SKIP C). If neighboring items in a rule are linearly adjacent, no SKIP is inserted.

Context changes vs. revision checking The methods of changing items are different than the revision checking algorithm in [8], in that they account for contextual errors, whereas revision checking rescores the item in focus. To examine the

score of A in A B C, for instance, the current solutions simulate a change to B or C to see how that affects the score of A, whereas revision checking notes the effect of revising A. Context item changing is rather robust, in that it subsumes rules the system has not seen before. For example, generalizing A B C to A SKIP C covers A D C, too, which may have never been seen before. Thus, the scoring may also be of help for new rules within new kinds of data.

4 Evaluation

4.1 Experimental conditions

We use the WSJ corpus [12] for all experiments, converted to the Stanford dependencies [4], allowing us to vary the training data size and keep constant the testing data. In particular, we train both the parsers and the error detection methods on a small portion of the corpus (section 02) and a larger portion (sections 02–15), and then test on section 00. To increase the variety in parser quality, we use two parsers with differing methods, MaltParser [16] and MSTParser [13]. This set-up is in line with recommendations in [18] for selecting appropriate experimental conditions to robustly test error detection methods.

Error detection precision is given in table 1, i.e., the percentage of parser errors accurately identified for corpus positions under a threshold. Since each method identifies a differing number of positions at differing thresholds, we report precision for *segment sizes*: for the lowest-scoring 5% of tokens, for example, what is precision across the methods? Since the number of positions is fixed for a given segment, an increase in precision means there is a corresponding increase in recall for that segment [18]. We choose low segment sizes (1% and 5% of the testing corpus) because activities like manually correcting annotation for a huge amount of text rely on high precision for a relatively small percentage of the data [18].

4.2 Results

Turning to the trends in table 1, we note a number of things. 1) For small training (02), it almost always helps to have some new representation, i.e., multiple representations indeed work better for situations with little annotated data. This matches our analysis of the method (section 2.2), where multiple perspectives on the data provide more opportunities to pinpoint a spot as erroneous.

2) For large training (02-15), adding the new representations generally helps for checking a small amount of data (1% segment), but becomes less beneficial as more corpus positions are examined (5%). Indeed, precision goes *up* for the 1% segment, but significantly *down* for the 5% segment in moving from the small (02) to the large (02-15) training scenario.

In table 2, which reports the score thresholds for each condition, we can see part of the reason for the downward trends: as more information is added to the model—adding bigrams (moving from High to Wall), adding new representations

	1% seg.	Train: 02		Train: 02-15	
	Condition	Malt	MST	Malt	MST
High	Base	77.0	73.6	86.7	78.7
	No-dep	79.4	76.5	89.8	**83.4**
	No-POS	83.5	78.4	91.8	82.8
	Skip	**90.2**	**84.6**	**95.2**	74.4
	Remove	84.0	80.0	92.9	83.1
	Insert	78.7	75.2	88.0	78.5
Wall	Base	91.2	83.7	93.1	81.8
	No-dep	91.5	85.0	93.9	83.8
	No-POS	92.5	85.0	**94.6**	**84.0**
	Skip	**94.1**	**88.5**	94.3	75.5
	Remove	91.5	88.1	94.3	81.6
	Insert	92.5	84.4	92.8	65.7

	5% seg.	Train: 02		Train: 02-15	
	Condition	Malt	MST	Malt	MST
High	Base	77.0	73.6	**66.8**	**62.5**
	No-dep	79.4	76.5	65.7	61.3
	No-POS	**81.6**	**77.4**	65.4	62.1
	Skip	77.0	71.4	59.8	51.9
	Remove	79.8	74.0	66.0	57.0
	Insert	78.1	75.2	66.3	59.8
Wall	Base	80.4	76.0	**66.6**	60.7
	No-dep	80.4	77.1	65.9	60.5
	No-POS	**81.3**	**77.9**	65.2	**61.4**
	Skip	78.4	71.3	59.8	51.6
	Remove	81.0	75.2	65.4	56.9
	Insert	77.3	67.2	59.8	47.8

Table 1: Precision (%) for 1% & 5% segments (1342 & 6710 positions). Baseline LAS: Malt.02: 81.1%, Malt.02-15: 86.4%, MST.02: 80.5%, MST.02-15: 87.6%.

(comparing Base to other models), or, most crucially, adding more training data and thus more training rules (moving from 02 to 02-15)—the score threshold required to hit 1% or 5% of the data rises, in some cases dramatically. The reason for this is straightforward: by including information from a greater number of (training) rules, the chance of having a low score for a parsed rule is less likely. A score of 158.92, for the Malt.02-15.Wall.Skip case, for example, means that many of the rules flagged here as errors have a great deal of supporting evidence (roughly speaking, 158 instances from the training data support it), and so we should not be surprised that many of the cases are not errors, i.e., precision is low.

The upshot is that multiple representations work best for increasing precision when low-scoring positions account for a large percentage of the data to be corrected, and this tends to happen when: a) the amount of training data is small, and/or b) the amount of data to be corrected is a small percentage of the corpus. (For very large corpora, a small percentage of the corpus is still a large absolute number of positions.) Following the mathematical reasoning above, when more representations are added, a position is unlikely to get a low score by chance, and so low scores mean more, i.e., are more likely to truly indicate errors. The 1% segments confirm this: precision is high for every condition where the score threshold is around zero, whereas the higher thresholds for the Skip (14.11) and Insert (8.05) models lead to lower precision (74.4% and 65.7%, respectively).

Turning to the generally improved precision for the 02-15 training experiments with the 1% segments (vs. 02 training), we believe this improvement stems from the fact that the set of training rules is more accurate. The method is still examining zero-scoring rules, but, as compared to the 02 training experiments, more training data gives a better representation of what an acceptable rule is, and so the zero scores seem to be more meaningful.

	1% seg.	Train: 02		Train: 02-15			5% seg.	Train: 02		Train: 02-15	
	Condition	Malt	MST	Malt	MST		Condition	Malt	MST	Malt	MST
High	Base	0	0	0	0	High	Base	4	4	16	14
	No-dep	0	0	0	0		No-dep	4	4	42	42
	No-POS	0	0	0	0		No-POS	4	4	63	51
	Skip	0	0	2	13		Skip	7	10	156	176
	Remove	0	0	0	1		Remove	4	4	41	45
	Insert	0	0	0	0		Insert	4	4	27	28
Wall	Base	0.00	0.00	0.03	0.37	Wall	Base	0.19	0.45	20.81	19.97
	No-dep	0.00	0.00	0.04	1.03		No-dep	0.52	1.15	46.32	48.35
	No-POS	0.00	0.00	0.13	1.33		No-POS	2.01	1.83	67.20	59.42
	Skip	0.00	0.00	3.14	14.11		Skip	7.04	11.04	158.92	182.29
	Remove	0.00	0.00	0.20	2.31		Remove	1.29	2.13	45.23	49.51
	Insert	0.00	0.00	1.05	8.05		Insert	3.05	6.05	80.59	95.44

Table 2: Score thresholds for 1% & 5% segments (1342 & 6710 positions).

3) The more beneficial models seem to be the ones which maintain the attachments in a rule but make them more abstract (No-dep, No-POS, Skip), and not the models which attempt to actually modify the attachments of the rules (Remove, Insert). This is not necessarily tied in to the score thresholds, either: notice how the No-POS model performs better than the Remove model for MST.02-15.Wall with a 5% segment (table 1: 61.4% vs. 56.9%), but actually has a higher score threshold (table 2: 59.42 vs. 49.51). The difference here is in how closely tied the representation is to the data: for No-POS, there is a slight bit of abstraction for an element which is already present in the parse, whereas the Remove model completely changes the elements in the rule. The simplicity of the DAPS method likely means that it is risky to deviate too far from the original parses without utilizing further information.

4) Comparing the High and Wall methods confirms a finding in [18], namely: Wall methods tend to work better than High when less training data is involved, but then the two methods are more equivalent for greater amounts of training data. This underscores the main thrust of our analysis here: when there is less training data, it helps to provide additional representations, whether in the form of bigrams or in abstracted n-gram representations.

This comparison also raises the issue of how best to use the abstract representations: for bigrams, we used a weight of 0.01 and found good results. While precision is not better for higher thresholds, it is at least not generally worse than the High method. It seems, then, that it might be fruitful to explore weighting the abstract representation; indeed, it may help for all n-grams to explore a more proper weighting system, e.g., experimentally determining the optimal weights.

5 Conclusion

Building from the simple DAPS method for dependency parse error detection, we developed representations which account for parse errors occurring in the context of other parse errors. The core insight is to combine both very detailed representations with more relaxed representations, i.e., ones which allow for errors. We have shown that for situations with little annotated data to begin with, error detection precision can be greatly increased. How one evaluates crucially affects the conclusions to be drawn, underscoring the point in [18] that many different corpus settings need to be employed.

Within the DAPS method, there are several avenues to explore: even richer representations, e.g., incorporating lexical information, grandparent information, etc.; different methods of comparing and combining information (e.g., tree kernels); schemes for weighting information; and so forth. Additionally, by accounting for incorrect context, we hope to not only develop better error detection models, but also to get one step closer to developing a parse corrector, which will need to account for multiple, interrelated errors.

Acknowledgements

We would like to thank the participants of the IU computational linguistics colloquium, as well as the three anonymous reviewers, for useful feedback.

References

[1] Rahul Agarwal, Bharat Ambati, and Dipti Misra Sharma. A hybrid approach to error detection in a treebank and its impact on manual validation time. *Linguistic Issues in Language Technology (LiLT)*, 7(20):1–12, 2012.

[2] Bharat Ram Ambati, Mridul Gupta, Samar Husain, and Dipti Misra Sharma. A high recall error identification tool for hindi treebank validation. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010.

[3] Serhiy Bykh and Detmar Meurers. Native language identification using recurring *n*-grams – investigating abstraction and domain dependence. In *Proceedings of COLING 2012*, pages 425–440, Mumbai, India, December 2012.

[4] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC 2006*, 2006.

[5] Markus Dickinson. Ad hoc treebank structures. In *Proceedings of ACL-08*, Columbus, OH, 2008.

[6] Markus Dickinson. Detection of annotation errors in corpora. *Language and Linguistics Compass*, forthcoming.

[7] Markus Dickinson and Charles Jochim. A simple method for tagset comparison. In *Proceedings of the 6th Language Resources and Evaluation Conference (LREC 2008)*, Marrakech, Morocco, 2008.

[8] Markus Dickinson and Amber Smith. Detecting dependency parse errors with minimal resources. In *Proceedings of IWPT-11*, pages 241–252, Dublin, October 2011.

[9] Tadayoshi Hara, Yusuke Miyao, and Jun'ichi Tsujii. Descriptive and empirical approaches to capturing underlying dependencies among parsing errors. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1171, Singapore, August 2009. Association for Computational Linguistics.

[10] Tadayoshi Hara, Yusuke Miyao, and Jun'ichi Tsujii. Effective analysis of causes and inter-dependencies of parsing errors. In *Proceedings of IWPT-09*, pages 180–191, Paris, October 2009.

[11] Mohammad Khan, Markus Dickinson, and Sandra Kübler. Does size matter? text and grammar revision for parsing social media data. In *Proceedings of the Workshop on Language Analysis in Social Media*, Atlanta, GA USA, 2013.

[12] M. Marcus, Beatrice Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[13] Ryan McDonald, Kevin Lerman, and Fernando Pereira. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of CoNLL-X*, pages 216–220, New York City, June 2006.

[14] Seyed Abolghasem Mirroshandel and Alexis Nasr. Active learning for dependency parsing using partially annotated sentences. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 140–149, Dublin, Ireland, October 2011. Association for Computational Linguistics.

[15] Alessandra Moschitti. Making tree kernels practical for natural language learning. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 113–120, Trento, Italy, 2006.

[16] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[17] Adam Przepiórkowski. What to acquire from corpora in automatic valence acquisition. In Violetta Koseska-Toszewa and Roman Roszko, editors, *Semantyka a konfrontacja jezykowa, tom 3*, pages 25–41. Slawistyczny Ośrodek Wydawniczy PAN, Warsaw, 2006.

[18] Amber Smith and Markus Dickinson. Evaluating parse error detection across varied conditions. In *Proceedings of the 13th International Workshop on Treebanks and Linguistic Theories (TLT13)*, Tübingen, Germany, 2014.

Querying topological fields in the TIGER scheme with TIGERSearch

Stefanie Dipper

Institute of Linguistics
Ruhr-University Bochum
E-mail: dipper@linguistics.rub.de

Abstract

The TIGER corpus is a German treebank with a hybrid dependency-constituency annotation. In this paper, I address the question how well topological fields (e.g. Vorfeld, Verb second) can be searched in this treebank, using the search tool TIGERSearch. For most queries, a version without crossing branches is used. It turns out that queries can be formulated that result in quite good F-scores for the Vorfeld and left and right brackets. Mittelfeld and Nachfeld are hard to query. This is due partly to properties of the language, partly to design decisions in the TIGER scheme, and partly to restrictions imposed by the search tool.

1 Introduction

This paper deals with the TIGER scheme [1], one of the two main annotation schemes for German syntax. The other main scheme is the TüBa-D/Z scheme [18]. The two schemes implement quite different design principles. The TIGER scheme is famous for its extensive use of *crossing branches* for encoding non-local dependencies. The TüBa-D/Z scheme is special in that it puts a layer with *topological fields* on top of the constituency structure.[1]

Topological fields are widely acknowledged as a useful concept by modern syntactic theories of German. Hence, linguists using German treebanks often would like to refer to these notions in formulating a query expression. The TIGER corpus [5] was created to serve both as training data for automatic applications and as a source for linguistic investigations. The question, addressed in this paper, is then whether linguist users are able to query topological fields not only in the TüBa-D/Z treebank, where they are explicitly encoded, but also in the TIGER treebank. The TIGER corpus comes with its own search tool, TIGERSearch [10], which is also used in this paper for searching the corpus.

[1] For a comparison of the two schemes, see [6].

VF	LK	MF			RK	NF		
Hans	*hat*	*heute*	*Maria*		*getroffen*	*die*	*einkaufen*	*war*
H.	has	today	M.		met	who	shopping	was
Hans	*traf*	*heute*	*Maria*			*die*	*einkaufen*	*war*
H.	met	today	M.			who	shopping	was
	dass	*Hans*	*heute*	*Maria*	*getroffen hat*	*die*	*einkaufen*	*war*
	that	H.	today	M.	met has	who	shopping	was

Figure 1: Topological field analysis of different sentences (‘(that) Hans met Maria today, who was shopping’)

The paper first gives a short introduction to German syntax (Sec. 2). Sec. 3 introduces the TIGER annotation scheme, and Sec. 4 presents the evaluation, followed by the conclusion (Sec. 5). The appendix contains sample templates.

2 German syntax: topological fields

German has a relatively free constituent order. Following a long tradition, German sentences are usually analyzed and split into different *topological fields* [9]. The element that functions as the separator between these fields is the verb or verbal parts (in most cases). The verb can be located in two different positions, either the second (“verb second”) or the final position (“verb final”) of the clause. Fig. 1 shows three sentences with their field analyses. “LK” and “RK” (“Linke/Rechte Klammer”, ‘left/right bracket’) indicate the two verbal positions (LK can also be occupied by subordinating conjunctions). The brackets divide the sentences into “VF” (“Vorfeld”, ‘prefield’), containing exactly one constituent, “MF” (“Mittelfeld”, ‘middle field’) with multiple constituents, and “NF” (“Nachfeld”, ‘postfield’), which often contains clausal constituents (which can be assigned a separate layer with topological fields). The brackets and the fields can also stay empty.

If the sentence contains only a simple verb form, one of the brackets remains empty, possibly resulting in ambiguous structures, see Fig. 2: (ia/b) and (iia/b) contain identical strings each, which can be analyzed by different brackets and fields, though. To (manually) disambiguate such structures, the simple verb form is replaced by some complex verb form, e.g. a particle verb or a combination of an auxiliary or modal plus verb. In (i’) the simple verb *ging* ‘went’ has been replaced by the particle verb *ging weg/wegging* ‘went away’; in (ii’) the simple form of the preterite *traf* ‘met’ has been replaced by perfect tense *hat getroffen* ‘has met’. The test paraphrases in (i’) reveal that (i) can be a verb-second (a) or verb-final (b) clause. The two options in (ii’) are stylistic variants, and it is sometimes hard to tell which is “the right” one. The TüBa-D/Z scheme defines a default rule for such cases [18, p. 93]: *Unless there is strong evidence for a position in MF, the relative clause is located in NF.* In the TIGER scheme, which does not annotate topological fields, the variants result in the same analysis.

The different topological slots — fields and brackets — are highly relevant

	VF	LK	MF	RK	NF
(i) a	***wer***	***ging***			
(i) b	***wer***			***ging***	
	who	went		went	
(i') a	*wer*	*ging*			*weg*
	who	went			away
(i') b	*wer*			*wegging*	
	who			away-went	
(ii) a	*Hans*	*traf*	***Leute, die*** ...		
(ii) b	*Hans*	*traf*	***Leute,***		***die*** ...
	H.	met	people who		who
(ii') a	*Hans*	*hat*	*Leute, die* ...	*getroffen*	
(ii') b	*Hans*	*hat*	*Leute*	*getroffen,*	*die* ...
	H.	has	people who	met	who

Figure 2: (Fragments of) syntactically-ambiguous (i/ii) and non-ambiguous (i'/ii') sentences ('who went (away)?'; 'Hans met people who ...')

for research in German syntax. E.g. the Vorfeld often serves as a test position for constituency because it usually contains exactly one constituent — there are exceptions, though (see e.g. [11]). The Vorfeld is also interesting from an information-structural point of view because it seems to be the prime position for sentence topics — it often contains constituents with other information-structural functions, though (e.g. [16, 7]). Constituent order ("scrambling") within the Mittelfeld has been investigated extensively (e.g. [2]), as well as the question which constituents can occur *extraposed*, i.e. in the Nachfeld slot (e.g. [17]). Finally, the relative order of verbal elements in the Rechte Klammer has been researched a lot (e.g. [8]).

3 The TIGER annotation scheme

The TIGER scheme implements a hybrid approach to syntactic structure, combining features from constituency and dependency structures. On the one heand, it uses virtual nodes like "NP" and "VP" for constituents. On the other hand, non-local dependents are connected by crossing branches, directly linking the head and its dependent; edges are labeled by grammatical functions such as "SB" (subject) or "MO" (modifier).

The TIGER scheme omits "redundant" nodes, assuming that these nodes can be recovered automatically by combining information from the POS tags and/or the functional labels. This concerns two types of nodes: *unary* nodes, i.e. non-branching nodes like NP nodes that dominate one terminal node only; and NP nodes dominated by PPs.

This design principle — omitting redundant nodes — poses obvious problems for treebank users. If users are interested e.g. in VPs with an NP daughter, they have

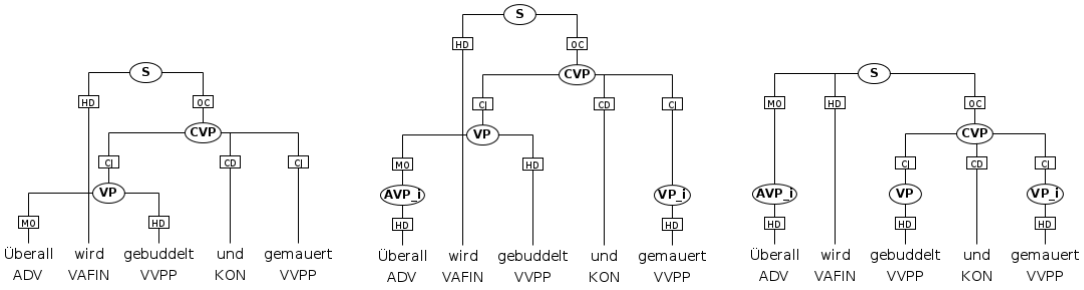


Figure 3: TIGER sentence no. 291 in original version (left), enriched version ("ENR", center), and endriched context-free version ("CF", right) ('Everywhere people are digging and constructing')

to make additional efforts to retrieve all actual NPs. The search tool that comes with the TIGER corpus, TIGERSearch [10], allows the user to define *templates*, which facilitates such queries enormously. (1a) defines a sample template for pronouns. The first conjunct exploits the fact that all pronominal POS tags start with "P" (e.g. "PPER" for personal pronouns, "PPOSAT" for attributive possessive pronouns, etc. [15]). Other tags that start with "P" (pronominal and interrogative adverbs and particles) are excluded by the second conjunct. (1b) shows how to use the template in a query to constrain the otherwise unspecified node variable "#a" to pronouns. The query searches for VPs that directly dominate some pronoun.

(1) a. `PRON(#x) <- #x: [pos=/P.*/ & pos!=/PROAV|PWAV|PTK.*/];`

b. `[cat="VP"] > #a:[] & PRON(#a)`

In a similary way, a template for NPs in general could be defined. An alternative way is to apply a script that expands TIGER's minimalistic structures and inserts such redundant nodes, thus creating an enriched, user-friendly version of the treebank, as has been suggested e.g. by [14]. Fig. 3 illustrates both formats. The figure shows a TIGER structure in the original version (left) and in the enriched version (center), with two inserted nodes: AVP_i and VP_i.[2]

Non-local dependencies are encoded by crossing branches in the TIGER scheme. Such structures are difficult to process automatically, so scripts have been created to re-attach these branches in a way to avoid crossings. I call the resulting structures "context-free" because they could have been created by a context-free grammar. The rightmost structure shown in Fig. 3 is such a context-free structure.[3] It attaches the AVP_i node higher up, eliminating the crossing branch. The evaluation

[2]The enriched version of the corpus has been created by the tool *TIGER Tree Enricher* [13]. The marker "_i" for inserted nodes is optional, and is used here to highlight inserted nodes. All sentence numbers in this paper refer to the TIGER corpus, release 2.2; URL: `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.html`.

[3]The context-free version of the corpus has been created by the program *treetools* by Wolfgang Maier, URL: `https://github.com/wmaier/treetools`.

Field	Template description
VF	– In main clauses: leftmost constituent of a sentence, preceding a finite verb; coordinating conjunctions may precede – In subord. clauses: leftmost constituent of a sentence, dominating a relative or interrogative word
LK	– In main clauses: the finite verb following VF – In subord. clauses: the subordinating conjunction
MF	The part between LK and RK (i.e. both brackets must be filled); the template marks the beginning (MFB) and end (MFE) of MF
RK	– Single-element RKs (RKS): the finite verb in a subordinate clause, or a verb particle, infinitive or participle – Multi-element RKs: a cluster of several verbs; the template marks the beginning (RKB) and end (RKE) of complex RKs
NF	The part following a verb particle, infinitive, participle or verb cluster; the template only marks the beginning (NFB) of NF

Table 1: Description of the topological templates

will show that querying the topological fields is rather difficult if not impossible if crossing branches may occur.[4]

4 Querying the scheme: an evaluation

The TIGER corpus has been used successfully to search for elements in specific topological fields. For instance, [12] investigates (a subset of) extraposed clauses, i.e. clauses in the Nachfeld. However, [12] only considers clauses that are dependent from a noun (object or relative clauses), which facilitates querying enormously. As we see below, querying for Nachfeld constituents *in general* is actually very hard.

For the evalution, a student of linguistics annotated the first 600 sentences of the TIGER corpus with topological fields.[5] Sentences 1–100 were used in the development of the query templates, sentences 101–600 were reserved for the evaluation.

Of course, the results of the evaluation heavily depend on the quality of the templates. Table 1 summarizes the most relevant properties of the individual templates. In the definitions of the templates, I tried to avoid exploiting linguistic knowledge about the topological fields, such as "sentential constituents often are located in the NF" (because this is a statement that one might want to verify). However, I did use information such as "relative clauses are verb-final clauses".

I evaluate the scheme by applying the templates to the TIGER corpus in an

[4]This observation can be transferred to treebanks annotated with pure dependency structures.

[5]The annotation tool was WebAnno [19]. The fields were annotated mainly according to the TüBa-D/Z scheme. However, the student located interrogative and relative pronouns in VF, and subordinating conjunctions in LK (rather than C).

enriched version with redundant nodes and in context-free format ("CF"). Querying the version with crossing branches ("ENR") results in highly complex (and inefficient) queries, so I defined only the template for VF in ENR.[6]

The appendix displays the template definitions for the Vorfeld position and the precedence relation in the CF version.[7,8] For efficiency reasons, the VF template is split in two parts: VF in main (VFmain) and subordinate (VFsub) clauses. VFmain also covers the verb-second (V2) position in the LK slot, since VF and V2 depend on each other. A query using the VF template is shown in (2).

(2) `#vf:[] & #v2:[] & VFmain_cf(#vf,#v2)`

The templates are designed to result in high precision rather than high recall. For instance, only VF instances are covered where the sentence either directly starts (i) with the VF or (ii) with a coordinating conjunction that directly precedes the VF. Other sentence-initial elements or elements following the VF are not allowed to maintain the constraint that there is exactly one constituent preceding the finite verb. This constraint, e.g., excludes VF in sentences with preposed material (3a) or with parentheticals intervening between VF and the finite verb (3b).[9]

(3) a. [$_{AVP}$ Gewiß] — [$_{NP}$ <u>die wirtschaftliche Liberalisierung und Öffnung des Landes</u>$_{VF}$] schreiten voran . (s62)

b. [$_{CAP}$ <u>Früher oder später</u>$_{VF}$] , [$_{S}$ da sind sich alle einig] , muß Perot Farbe bekennen und Konzepte vorlegen . (s47)

4.1 Qualitative results

Qualitative results from the development process show that there are certain types of constructions that cannot be handled properly by the templates. The problems can be traced back to (i) difficult constructions, (ii) systematic ambiguities of the language, (iii) constraints of the search tool, and (iv) the design of the annotation scheme, in particular (v) crossing branches.

[6] Using the enriched versions facilitates querying since we do not have to care about omitted NP nodes etc. The vast majority of the conversion steps of the enrich-script are trivial so they do not affect the evaluation, cf. [13]. Creating the context-free version involves more complex operations, see Fn. 4. Still, the conversion does not seem to introduce problematic structures.

[7] All template definitions used in this paper can be found at `http://www.linguistics.ruhr-uni-bochum.de/~dipper/tiger-templates.html`.

[8] TIGERSearch uses a purely left corner-based definition of precedence, which is not sufficient in most cases (a node #n1 is said to precede another node #n2 if the left corner of #n1 precedes the left corner of #n2 [10, p. 80]; according to this definition, a node consisting of two or more words does not precede its following sibling). In addition, the precedence template allows for intervening quotes (via the template "prec_quotes"; and similarly with "prec_comma"). The VF template further refers to a template "hasLeftChild", which defines left-corner dominance. This template extends the corresponding TIGER relation to one that holds between terminal or non-terminal nodes.

[9] The parenthetical sentence in (3b) contains a VF, which is correctly found by the VF template. Here and in the following examples, the underlined, labeled part indicates the "target" slot, as annotated in the gold data, and the part in boldface indicates the string matched by the template (if any).

(i) Difficult constructions In general, parentheticals, non-constituent coordination and elliptical constructions are difficult to handle by templates, so a large number of these are not covered. (4) shows instances of coordinated elliptical sentences: In (4a), the VF is missing in the second conjunct so that the verb in second position (LK) cannot be recognized. In (4b), the second conjunct consists of the VF only, and the predicate is missing.

(4) a. Er **tritt**$_{LK}$ in die GM-Verwaltung ein und wird$_{LK}$ Großaktionär des Autokonzerns . (s25)

b. " **Geschäftemachen**$_{VF}$ ist seine Welt und nicht die Politik$_{VF}$. (s44)

(ii) Systematic linguistic ambiguities First, sentences with empty Mittelfeld and simple finite verbs are systematically ambiguous, as shown in Fig. 2.[10] The finite verb in such sentences would be (possibly incorrectly) matched by the VF template. A pertinent example from our development corpus is the verb of the relative clause *die meinen* 'who think' in (5).

(5) Allerdings gibt es dem Magazin zufolge in kleinen und mittleren Firmen viele Unternehmer , die meinen$_{RK}$, Perot sei einer von ihnen , und die den Texaner unterstützen . (s18)

A similar ambiguity arises whenever the right bracket is not filled. In such cases, it is hard to tell (automatically) where to draw the boundary between MF and NF, as in (6a). One option would be to use the syntactic category (S, VP, NP, PP, etc.) as an indicator of the position: usually, S and (most) VPs are located in NF, NPs in MF, and PPs can be in MF or NF. However, one aim of annotating (and querying) corpora is exactly to verify such common wisdom.

The MF and NF templates both require that the right bracket be filled, to minimized incorrect matches that result from an unclear position of the right bracket. This excludes a lot of instances (false negatives), such as (6a). At the same time, the (very) simple heuristics applied in the template also yields false positives (6b) (the beginning of the (incorrect) NF matches are marked in boldface).

(6) a. " Ich glaube kaum , daß mit seinem , naja , etwas undiplomatischen Stil im Weißen Haus dem Land ein Gefallen getan wäre $_{NF}$. (s24)

b. So will der politische Außenseiter beispielsweise das Steuersystem vereinfachen , **das** Bildungssystem verbessern , **das** gigantische Haushaltsdefizit abbauen , **Einfuhren** aus Japan drosseln **und** die geplante Freihandelszone der USA mit Mexiko verhindern . (s37)

[10] Such cases do occur: in the TüBa-D/Z treebank, there are 720 (0.84%) MF-less instances of the form VF-LK(-NF), and 125 (0.15%) of the form VF-RK (in the TüBa-D/Z scheme, the VF constituent is placed under a C node in the second type of constructions, cf. Fn. 5).

(iii) Constraints of the search tool The query language of TIGERSearch supports searches for linguistic relations such as precedence and dominance relations. It is not a programming language, though. Hence, certain query constraints cannot be formulated (or would require complex constraints). This includes cases where mother and daughter constituents match the query but only the highest, maximal one is correct. This happened, e.g., with the first version of the Nachfeld (NF) template that searched for a (i.e. *some*) constituent following the right bracket (RK), see (7): the S node occupies the Nachfeld but both the S and NP nodes matched the NF query. (The current NF template only matches the first word of the NF.)

(7) " Es ist wirklich schwer $[_{RK}$ zu sagen] , $[_{S}$ $[_{NP}$ **welche Positionen**] **er einnimmt**$_{NF}$] **,** da er sich noch nicht konkret geäußert hat " , beklagen Volkswirte . (s36)

Another example are cases where a topological field does not correspond to a single TIGER constituents. Variables in TIGERSearch queries always correspond to single constituents. Hence, for complex fields like the Mittelfeld (MF), which can consist of multiple constituents, two variables have to be used, one marking the beginning of the MF (MFB), one marking the end (MFE). Similarly, complex verb clusters in the right bracket (RK) and multiple Nachfeld constituents cannot be matched by a single variable.

(iv) Design of the the annotation scheme The crossing edges of the TIGER scheme are hard to query in general (see below). Certain sentences contain edges that encode dependencies rather than constituents, without resulting in crossing branches, though. This concern different types of left dislocation with resumptive elements, as in (8). In such cases, constraints on the number of constituents (e.g. in VF) cannot be applied sensibly.

(8) $[_{PP}$ $[_{S}$ **Daß Perot ein Unternehmen erfolgreich leiten kann**] , **davon**$_{VF}$] sind selbst seine Kritiker überzeugt . (s6)

The last example shows that the queries would have to provide exceptions for individual cases. Such an approach is not desirable in general because it uses queries to *encode* a lot of information rather than to simply *extract* information from the treebank.

(v) Crossing branches Turning now to the enriched (ENR) scheme with crossing branches, it is obvious that extra efforts have to be made to correctly treat discontinuous constituents. Fig. 4 shows an example sentence (left) that would not be matched by the VFmain template in the appendix because the right corner of the NP node does not precede the finite verb. In contrast, the VFsub template incorrectly matches the phrase *was [. . .] eigentlich machen* of the other example sentence in Fig. 4 (right) because the right corner of the VP node is adjacent to the finite verb.

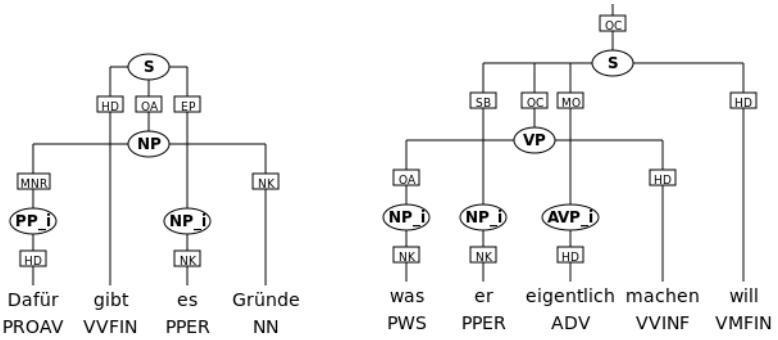


Figure 4: TIGER sentences no. 86 (left) and 48 (fragment; right) in ENR version with crossing branches ('There are reasons thereof' (left) and '... what he actually wants to do' (right))

What we need here is a way to address the daugther nodes that are continuous. TIGERSearch provides an operator "discontinuous()" which can be used here. Relevant parts of the template are shown in the appendix.[11] This way, (many of the) missed discontinuous cases (i.e. false negatives like in Fig. 4, left) can be matched. What is rather unclear, however, is how false positives (Fig. 4, right) can be excluded.

4.2 Quantitative results

I evaluted the templates by comparing the query results of the sentences 101–600 with the manually-annotated fields. Fields and brackets of the queries must span strings of words that are identical with the gold fields for counting as a match, i.e. overlapping spans were considered errors.

Table 2 shows the results from the evaluation set (TIGER sentences s101–s600). Due to performance issues and out-of-memory errors, not all templates could be run successfully by TIGERSearch, and had to be modified accordingly.[12] For multi-constituent fields, the gold annotations were transformed into boundary markers.

The table shows that the F-scores for VF, LK and RK are near or above 90% and quite good. In contrast, the MF templates have a bad recall, and the (simple) NF template yields the lowest F-score.[13]

[11] [4] investigates Vorfeld positions in the CGN corpus of spoken Dutch, which is annotated according to the TIGER scheme. [4] deals with discontinuous VF constituents roughly by assuming that either the entire constituent or at least its head must precede the finite verb to qualify as the VF constituent [4, p. 76]. It seems to me that this definition would fail to correctly determine the VF in the first example in Fig. 4 because the head occurs sentence-final.

[12] E.g. the original MF template referred to both the VF/LK template and the RK template, which made it computationally too expensive.

[13] [3] present a topological parser for German. The parser was trained on a version of the Negra corpus (which is annotated similar to the TIGER corpus) that has been automatically enriched with

Field		F1	Prec	Rec	#Gold	#System
VF		87.26	88.43	86.11	648	631
LK		93.01	97.11	89.23	678	623
MF:	MFB	58.27	96.74	41.69	854	368
	MFE	56.78	87.96	41.92	854	407
RK:	RKS	89.06	87.41	90.77	390	405
	RKB	83.42	78.67	88.77	187	211
	RKE	88.22	82.63	94.62	186	213
NF:	NFB	45.70	56.71	38.27	243	164

Table 2: Results of TIGERSearch template-based queries for topological fields: F-Score, Precision, Recall (all in %), number of instances in the gold and system data (i.e. query results)

5 Conclusion

To sum up the findings of this paper: The dependency-oriented TIGER annotation scheme (in its original form) does not really seem suitable for syntactic investigations at the level of topological fields. In particular, crossing branches that result from long-distance dependencies are difficult to handle, and especially excluding false positives is difficult.

Hence, converting the treebank to a context-free format is a good idea in general and facilitates further (automatic and manual) processing to a great extent. However, searching for topological fields in this format still requires complex templates and a considerable amount of processing time. What we actually need is a version of the TIGER corpus enriched with topological-field annotations. For some of the fields (VF, LK, RK), automatically adding topological fields seems feasible (especially if a powerful programming language is used). Other fields (MF, NF) would require manual work.

Approaches like the one taken by the TüBa-D/Z scheme seem favorable, by explicitly annotating topological fields from the beginning. So why not just stick to the TüBa-D/Z corpus? I think there are two main reasons why it is favorable to be able to use both treebank, TüBa-D/Z and TIGER. First, both are only medium-sized (TüBa-D/Z, release 9: around 85,000 sentences; TIGER: around 50,000 sentences). Second, while both cosist of texts from newspapers from the 1990s, the style differs to some extent: TIGER contains texts from *Frankfurter Rundschau*, TüBa-D/Z from *taz*, which is a rather progressive newspaper. So in an ideal world, users would probably like to exploit both treebanks.

topological field annotations. They report 93.0% precision and 93.7% recall for the enrichment script. Unfortunately the script is no longer available.

Acknowledgements

I would like to thank the anonymous reviewers for helpful comments. Many thanks to Ronja Laarmann-Quante for creating the gold topological annotation of 600 TIGER sentences, and to Adam Roussel for the evaluation script.

References

[1] Stefanie Albert et al. TIGER Annotationsschema, 2003. Technical Report, Universität des Saarlandes, Universität Stuttgart, Universität Potsdam, `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/TIGERCorpus/annotation/tiger_scheme-syntax.pdf`.

[2] Markus Bader and Jana Häussler. Word order in German. a corpus study. exploring the left periphery. *Lingua*, 120(3):717–762, 2010.

[3] Markus Becker and Anette Frank. A stochastic topological parser for German. In *Proceedings of COLING-2002*, Taipei, Taiwan, 2002.

[4] Gerlof Bouma. *Starting a sentence in Dutch. A corpus study of subject- and object-fronting*. PhD thesis, University of Groningen, 2008.

[5] Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation, Special Issue*, 2(4):597–620, 2004.

[6] Stefanie Dipper and Sandra Kübler. German Treebanks: TIGER and TüBa-D/Z. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*. Springer, Berlin, forthcoming.

[7] Werner Frey. A medial topic position for German. *Linguistische Berichte*, 198:53–190, 2004.

[8] Erhard Hinrichs and Tsuneko Nakazawa. Linearizing AUXs in German verbal complexes. In John Nerbonne, Carl Pollard, and Klaus Netter, editors, *German in Head-Driven Phrase Structure Grammar*, pages 11–38. CSLI, Stanford, 1994.

[9] Tilman Höhle. Der Begriff "Mittelfeld", Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329–340, Göttingen, Germany, 1986.

[10] Wolfgang Lezius. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora*. PhD thesis, Universität Stuttgart, 2002. Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung (AIMS), volume 8, number 4.

[11] Stefan Müller. Zur Analyse der scheinbar mehrfachen Vorfeldbesetzung. *Linguistische Berichte*, pages 29–62, 2005.

[12] Stefan Müller. Qualitative Korpusanalyse für die Grammatiktheorie: Introspektion vs. Korpora. In Gisela Zifonun and Werner Kallmeyer, editors, *Sprachkorpora — Datenmengen und Erkenntnisfortschritt*, IDS-Jahrbuch 2006. de Gruyter, Berlin, New York, 2007.

[13] Adam Roussel. Documentation of the tool TIGER Tree Enricher. `http://www.linguistics.ruhr-uni-bochum.de/resources/software/tte`, 2014.

[14] Yvonne Samuelsson and Martin Volk. Automatic node insertion for treebank deepening. In *Proceedings of the Third Workshop on Treebanks and Linguistic Theories (TLT)*, pages 127–136, Tübingen, 2004.

[15] Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset), 1999. Technical report, Universitäten Stuttgart und Tübingen, `http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-1999.pdf`.

[16] Augustin Speyer. Die Bedeutung der Centering Theory für Fragen der Vorfeldbesetzung im Deutschen. *Zeitschrift für Sprachwissenschaft*, 26:83–115, 2007.

[17] Jan Strunk and Neal Snider. Subclausal locality constraints on relative clause extraposition. In Heike Walker, Gert Webelhuth, and Manfred Sailer, editors, *Rightward Movement from a Cross-linguistic Perspective*, pages 99–143, Amsterdam, 2013. John Benjamins.

[18] Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Germany, 2012.

[19] Seid Muhie Yimam, Iryna Gurevych, Richard Eckart de Castilho, and Chris Biemann. WebAnno: A flexible,web-based and visually supported system for distributed annotations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (System Demonstrations) (ACL 2013)*, pages 1–6, 2013.

Appendix: TIGERSearch templates

- Templates for the Vorfeld position in main and subordinate clauses in context-free format (VFmain_cf and VFsub_cf)
- Extension of the VFmain_cf template for discontinuous constituents (VF-main_enr)
- Definition of the precedence relation

```
// VF (and V2) in main clauses
VFmain_cf(#vf,#v2) <-
  #s: [cat="S"]
& #v2: [pos=/V.FIN/]    // #v2: Verb in second position
& #s > #vf              // #vf: Vorfeld constituent
& #s >HD #v2

// VF is first constituent
& ( // 1. VF is very first element in the sentence
    hasLeftChild(#s,#vf)   // #vf is left-most child
  | // 2. Or some coordinating conjunction precedes VF
    #s >@l #conj
  & [] >JU #conj
  & prec(#conj,#vf)
  )

// VF precedes VFIN
& ( // 1. VF directly precedes V2
    prec(#vf,#v2)
  | // 2. A comma may intervene after clausal or appositive VF
    ( #vf: [cat=("S"|"VP")]           // either VF itself precedes comma
    & prec_comma(#vf,#v2)
    | #vf >* #clause_app              // or some embedded constituent
    & ( #clause_app: [cat=("S"|"VP")]
      | [] >APP #clause_app
      )
    & prec_comma(#clause_app,#v2)
    )
   );

// VF in subordinate clauses
VFsub_cf(#vf) <-
  #s: [cat="S"]
& #s > #vf               // #vf: Vorfeld constituent
& // VF is very first element in the sentence
  hasLeftChild(#s,#vf)   // #vf is left-most child
& #vf >* [pos=/.*(REL|W).*/];  // relative or interrogative elements

// Discontinuous VF
VFmain_enr(#vf,#v2) <-
// VF contains discontinuous element -> take daughter node
  #s: [cat="S"]
& #v2: [pos=/V.FIN/]
```

```
& #s > #vfin
& #s >* #vf_disc // #vf_disc: disontinuous mother of VF constituent
& discontinuous(#vf_disc)
& #vf_disc > #vf
& ...

// Precedence relation
prec(#x,#y) <-
( // 1. #x is a terminal node
  #x: [word=/.*/]
& #x . #y
| // 2. #x is non-terminal
  #x: [cat=/.*/]
& #x >@r #xchildR
& #xchildR . #y
| // 3. quotes may intervene (everywhere)
  prec_quote(#x,#y)
);
```

Parsing Poorly Standardized Language Dependency on Old French

Gael Guibon (1), Isabelle Tellier (1,2)
Matthieu Constant (3), Sophie Prevost (1) and Kim Gerdes (2,4)

(1) Lattice CNRS
(2) université Paris 3 - Sorbonne Nouvelle
(3) université Paris-Est, LIGM
(4) LPP CNRS
E-mails: gael.guibon@gmail.com, isabelle.tellier@univ-paris3.fr,
Matthieu.Constant@u-pem.fr, sophie.prevost@ens.fr, kim@gerdes.fr

Abstract

This paper presents results of dependency parsing of Old French, a language which is poorly standardized at the lexical level, and which displays a relatively free word order. The work is carried out on five distinct sample texts extracted from the dependency treebank *Syntactic Reference Corpus of Medieval French* (SRCMF). Following Achim Stein's previous work, we have trained the *Mate* parser on each sub-corpus and cross-validated the results. We show that the parsing efficiency is diminished by the greater lexical variation of Old French compared to parse results on modern French. In order to improve the result of the POS tagging step in the parsing process, we applied a pre-treatment to the data, comparing two distinct strategies: one using a slightly post-treated version of the TreeTagger trained on Old French by Stein, and a CRF trained on the texts, enriched with external resources. The CRF version outperforms every other approach.

1 Introduction

Today's research on historic language data is still profoundly different from usage based analyses of modern languages. Historic language data are generally sparse and intrinsically inhomogeneous. Common statistical corpus analysis methods are thus poorly suited and less successful as even simple frequency counts on raw corpora fail to provide reliable results. Moreover, one central goal of diachronic linguistics is the analysis of structural change over time, which is a gradual process calling for quantitative methods. However, very few resources of historic language are available in digital formats, and even fewer are provided with any type of annotation that could allow the application of standard corpus linguistic methods. There

is a variety of reasons for this situation, ranging from epistemological difficulties to the lack of economic interest. In this paper, we address the technical problems of producing, extending, or consolidating these resources with the help of statistical parsers.

Treebank development is often made easier and more precise by the use of machine learning techniques in a bootstrapping approach. Today's successful machine learning techniques rely on the underlying assumption that word forms are spelled the same way with only few exceptional and thus unknown forms whose analysis can be guessed correctly from the context. In this paper, we explore how difficult dependency parsing on non-standardized text actually is, also compared to equivalent tasks on more homogeneous texts of modern languages.

The treebank we use for these measures is the manually annotated *SRCMF* treebank (Syntactic Reference Corpus of Medieval French)[1] [10]. We explore at which point in the standard incremental parsing setup (lemmatization, POS-tagging, dependency parsing) the inhomogeneous character of the data interferes most strongly. In particular, two strategies are tested for POS-tagging to overcome this difficulty: one based on a slightly post-treated version of the TreeTagger trained on a large corpus of Old French, the other applying Conditional Random Fields (CRF) learning for various distinct texts separately. We show that CRFs allow to greatly improve previous results.

In the following, we first introduce the SRCMF treebank (section 2), and the portions of it we have used. We also provide some indicators to quantify its variability relatively to contemporary French. We then briefly present a related work (section 3). We finally explain the experiments conducted to minimize the impact of the lack of standardization on the final parsing quality (section 4).

2 Presentation of the Corpus

2.1 General presentation

Our research is based on the *Syntactic Reference Corpus of Medieval French* (SRCMF) [10], a heterogeneous treebank of Old French which was developed in a joint research project funded by the *Deutsche Forschungsgemeinschaft*[2] and the *Agence Nationale de la Recherche*[3] (ANR) from March 2009 to February 2012. The origin of this project was a collection of important medieval French texts whose electronic versions are stemming from the *Base de Francais Médiéval*[4] (BFM) [4] and the *Nouveau Corpus d'Amsterdam*[5] (NCA) [5]. It has been built to serve as a reference treebank of Old French.

[1] http://srcmf.org/

[2] http://www.dfg.de/

[3] http://www.agence-nationale-recherche.fr/

[4] http://bfm.ens-lyon.fr/

[5] http://www.uni-stuttgart.de/lingrom/stein/corpus/

Text	Date	Nb words	Nb sent.	Type
Chanson de ***Roland***	1100	29 338	3843	verse
Yvain by Chretien de Troyes	1177-1181	42 103	3735	verse
La ***Conqueste*** *de Constantinople* by Robert de Clari	>1205	33 994	2282	prose
Queste del Saint ***Graal***	1220	40 000	3049	prose
Aucassin *et Nicolete*	late 12c.-early 13c.	9387	985	verse & prose

Table 1: Texts from the SRCMF used in our experiments

Although the original texts contained few punctuations or other indications of segmentation, they have been segmented into clauses made around a finite verb. These clauses will be referred to as "sentences" in the following, even if a subordinate clause is not exactly a sentence. The original electronic versions of the texts already came with a POS tagging (50 POS tags), whereas the fine-grained dependency annotation (31 syntactic functions) was added manually, using Mazziotta's tool *NotaBene* [8].

In SRCMF, the POS tags were verified and each clause was syntactically analyzed by means of a dependency tree. Only *Yvain* includes a manually verified lemmatization.

From the SRCMF we choose five texts, shown in Table 1, of different periods, genres, and dialects. The first four of these texts are similar in size and date from the early 12th to the 13th century, written either in prose or verse. By way of comparison, the fifth selected text has a different size and is composed of a mix of verse and prose. The texts differ in the regional dialect they have been written in: Norman for *Roland*, Champenois for *Yvain*, Picard for *La Conqueste* and *Aucassin*, while *Graal* is unmarked for regional dialect. In fact, our experiments also include other texts, but the results for these five texts are representative of the whole results.

2.2 Heterogeneity of the corpus

The main reason for the heterogeneous character of the data is not so much the time span in which the different texts have been produced (120 years), but rather the lack of spelling norms, which only gradually developed historically under the influence of printing and the emergence of grammar books. In the middle ages, each author could develop their own written preferences influenced by their dialect. However, medieval texts display important variations which correspond not only to the dialects, since some spelling variants between texts belonging to the same dialect can also be observed. Still more surprisingly, even a single text by the same author may display spelling variations in some words. Table 2 provides some examples of words appearing in various forms in the same text (*Yvain*), whereas only a single form persists in contemporary French.

In order to measure SRCMF's word form variability, we can compare it with

Contemporary word	Form variations
Bretagne	bretaingne | bretaigne
vilain(e)(s)	vilains | vileins | vilainne
ainsi	ensi | einsi | ainsi

Table 2: Examples of word form variability

contemporary French. Unfortunately, we only have verified lemmas for one corpus (*Yvain*, from Chretien de Troyes), so we can only study the word form variability of a small part of SRCMF and we cannot quantify the differences due to the various kinds of dialects, authors, or even centuries. We used the French Treebank (FTB) [6] [1] as a sample of contemporary French. For both corpora, we computed the number of distinct forms corresponding to a single lemma, and averaged this number for each distinct POS tag. Table 3 shows the values obtained for the main morpho-syntactic categories. This indicator allows us to quantify the variability of spelling, at least for *Yvain*.

POS \ Corpus	French Treebank	SRCMF's Yvain
proper noun	1	1.25
common noun	1.31	1.31
infinitive verb	1	1.10
finite verb	2.48	3.15
determinant	1.06	2.21
adjective	1.63	1.68
adverb	1.01	1.40
Average number of forms per lemmas	1.57	2.25

Table 3: average number of forms for a lemma, for the FTB and *Yvain*

As expected, the values for *Yvain* are always higher than those for the FTB. Some categories of words which are nowadays considered as invariable (proper nouns, infinitive verbs) can correspond to various forms in *Yvain*. For example, the name *Yvain* itself can appear under four different forms: *Yvains*, *Yveins*, *Yvain*, and *Yvein*. This name being the main character of the text, it shows how poorly standardized Old French can be.

3 Previous works

Training statistical parsers is becoming a common step in linguistic ressource development in general and treebank construction in particular, often mixed with manual and rule-based approaches. But we believe that it is an interesting endeavor in itself, widely under-used, as a tool of linguistic analysis because it can

[6] http://www.llf.cnrs.fr/Gens/Abeille/French-Treebank-fr.php

provide information about the consistency of the syntactic annotation or about the variability of different sub-corpora. Cross-training a parser on a sub-corpus and applying the resulting parser on another corpus gives interesting insights not only in the difficulties of parsing these heterogeneous texts, but also in the historic and genre differences between these texts, as well as the dialectal spelling differences.

Achim Stein first conducted such syntactic parsing experiments on the SRCMF corpus [9]. The results are (partially) reported in Table 4. For this work, he trained the TreeTagger[7] on the Old French *Nouveau Corpus d'Amsterdam* in order to obtain POS and lemmas, and used Bernd Bohnet's dependency parser *Mate parser* [2] for the syntactic analyses. Although only about 70% of the graphemic forms received lemmas, many of which are ambiguous, this unverified lemmatization was used as the initial data for the parsers. The results of cross-training a parser are reported below.

Train \ Test		Auc.	Rol.	Graal	Yvain	Conq.
Aucassin	UAS		63.84	70.23	63.57	74.00
	LAS		44.56	57.16	48.04	61.88
Roland	UAS	67.73		71.03	64.48	67.80
	LAS	52.93		57.67	49.71	55.07
Graal	UAS	75.92	66.87		72.79	76.20
	LAS	63.06	46.67		58.24	64.49
Yvain	UAS	74.71	68.00	80.80		72.27
	LAS	61.96	48.45	70.06		58.68
Conq.	UAS	70.27	61.93	70.53	61.58	
	LAS	56.32	42.00	57.98	45.44	

Table 4: Stein's scorings

Except for this work, the SRCMF has mainly been used for linguistic purposes. We do not refer to these other studies here, as our work is clearly a continuation of Stein's experiments, which serve as our baseline.

4 Our experiments

Our purpose is to improve Stein's results. We expect to obtain a better performance of the Mate parsers by improving the initial POS labeling and lemmatization phases. Thus, we first explain the strategy used to obtain a good POS tagger, then we detail the results obtained for the parsing phase.

4.1 POS Tagging and Lemmatization

In order to achieve a comparable experimental setup with reliable performance measures, we produced sample extracts similar in size as the ones used in the pre-

[7] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

vious experiments: approximately 16000 words per sample. *Aucassin* contains only 9387 words but it was kept since it was the only text containing both verse and prose, which allowed us to see if this implied different results. Just as Stein, we used the *Mate* dependency parser, but we used it only for the dependency annotation. For the preliminary POS tagging, we tried two different strategies:

- Tagging the data with the *TreeTagger* trained by Stein. We also applied basic surface rules that remove useless tagger outputs and improve the lemmatization. When a lemma is not recognized, the word is associated with a specific "<nolem>" string.
- Training a specific POS tagger with Conditional Random Fields (CRF) [6], implemented by *Wapiti* [8] [7] for each training text separately. CRFs allow to take into account various types of contextual and external information such as the (slightly post-treated) *TreeTagger* lemmatization results and a lexicon that we extracted from the BFM corpus. This lexicon associates to each word present in the BFM corpus the set of its possible POS tags. The feature templates defined in this CRF using these external resources take the following forms:
  - check whether the current word (resp. the previous word, resp. the next word) is associated with the <nolem> lemma value by the TreeTagger
  - check the value of the lemma predicted by TreeTagger for the current word (resp. the previous word, resp. the next word)
  - check the value of two consecutive lemmas predicted by TreeTagger (for the previous and the current words, for the current and the next words)
  - for each distinct POS tag, check whether it can be associated with the current word in the BFM lexicon
  - concatenate all the distinct POS tags associated with the current word in the BFM lexicon

  Other features were used, such as checking the near contextual words, the final letters of the words (up to 4 letters), the lowercase value of the words, whether or not word forms begin by an uppercase, whether word forms begin neither by a letter nor by a number, whether the word's final letter is a special character (e.g an apostrophe for elision).

The main advantage of CRFs is to take into account more external resources and more contextual information, which appears to be crucial for a POS labeling of good quality for our historic language data. While *TreeTagger* uses a model trained on the *Nouveau Corpus d'Amsterdam*, with CRFs (which require fewer training data) we treat each text separately. Tables 5 and 6 display the accuracies

[8] `http://wapiti.limsi.fr/`, we used the 1.4.0 version

of the various POS taggers obtained on our data. CRFs usually obtain far better results than the TreeTagger.

Train \ Test	Auc.	Rol.	Graal	Yvain	Conq.
Aucassin		80.00	85.76	80.03	87.86
Roland	80.48		82.66	78.20	84.13
Graal	85.38	80.58		82.70	86.84
Yvain	83.13	80.22	89.05		82.11
Conq.	80.48	74.51	79.98	71.04	

Table 5: Accuracies of cross-trained POS taggers learned by CRF

Test	Accuracy
Aucassin	70.94
Roland	71.59
Graal	84.28
Yvain	66.76
Conq.	65.65

Table 6: Accuracies of the POS produced by the TreeTagger

4.2 Parsing

As previously mentioned, like Stein, we used the *Mate parser* (anna-3.61 version) for the syntactic analysis of our texts. For each experiment, we tried two distinct POS labeling strategies: either the (slightly post-treated) *TreeTagger* trained by Stein, or a specific POS tagger learned on the training text by a CRF. In each case, the lemmatization was provided by the TreeTagger, improved by surface rules. The "cross learning" results we obtained are shown in Table 7 and Table 8.

Train \ Test		Auc.	Rol.	Graal	Yvain	Conq.
Aucassin	UAS		66.12	73.57	68.67	76.02
	LAS		49.34	60.96	51.10	64.84
Roland	UAS	71.30		72.00	68.08	69.20
	LAS	58.36		62.61	54.16	54.80
Graal	UAS	75.34	67.40		72.84	77.21
	LAS	66.38	51.27		61.11	66.01
Yvain	UAS	74.67	69.46	81.05		73.83
	LAS	64.06	50.16	70.51		61.32
Conq.	UAS	72.07	65.20	71.08	62.37	
	LAS	59.65	45.33	60.18	48.04	

Table 7: Syntactic analysis results with the POS produced by the TreeTagger

Train \ Test		Auc.	Rol.	Graal	Yvain	Conq.
Aucassin	UAS		76.27	79.00	72.70	79.20
	LAS		58.98	65.02	57.63	68.50
Roland	UAS	72.26		73.02	70.64	73.86
	LAS	56.84		58.45	55.19	61.27
Graal	UAS	78.48	77.82		75.16	80.88
	LAS	65.52	59.79		61.15	69.08
Yvain	UAS	77.07	79.20	82.42		76.74
	LAS	64.72	61.58	70.41		63.81
Conq.	UAS	75.02	72.85	76.03	66.07	
	LAS	60.59	54.71	61.87	50.14	

Table 8: Syntactic analysis results with the POS produced by the CRFs

As can be seen in these results, the syntactic analyses based on CRF-induced POS tags outperform every other approach, improving Stein's results by nearly 10% in average.

We can see that there is a huge gap between UAS and LAS in our results, as it was the case in Stein's experiments. We suspect that this gap, which is not common in dependency parsing, is due to the size of the dependency label set (around 30 in our case, as compared with around 10 in standard treebanks) and/or the higher rate of variability in Old French, i.e. the fact that there exist several different forms for a same lemma (cf. Table 3).

4.3 Influence of the Lemmas on the Parsing

To evaluate the influence of the lemmas on the parser, we have conducted other experiments using *Yvain*, for which corrected lemmas are available. We have divided our gold version of *Yvain* with verified lemmas into two different sub-corpora of about 16 000 words each, one dedicated for training the parser, the other for testing it. We did the exact same division on *Yvain* with *TreeTagger* predicted lemmas.

	TT predicted lemmas	Verified lemmas
lemma acc.	58.84	100
UAS	88.99	89.36
LAS	79.55	80.53

Table 9: Lemmas'influence on the parsing (with gold POS)

The experiment whose results are displayed in table 9 only shows the lemmas' influence on dependency parsing, not on POS tagging. Here, as opposed to our previous works, the training and testing corpora are extracted from the same text (the only one provided with verified lemmas) and the parser could take advantage of gold POS tags, which explains why the parse scores are much higher than in

previous experiments, even with a small training set. The 1% improvement in LAS of this experiment confirms that lemmas have an influence on the parser quality, even without considering their influence on correct POS tags (which indirectly appear in the final parse results).

We can also compare our results to a simple baseline of dependency parsing using *Mate* for various portions of the French Treebank (FTB) with gold POS (some of them similar in size to our training set from *Yvain*). We obtain, without any further treatments, the scores in Table 10.

In fact, these results are hard to compare, for the following reasons:

- in the variant of the FTB we used, multi word units are not pre-treated: they have to be recognized during the parsing phase, which is a harder task than just parsing. In fact, the current state of the art for the dependency parsing on contemporary French with pre-recognized multi word units can reach 90.3% in UAS and 87.6% in LAS [3].
- the average length of a "sentence" in *Yvain* is about 10 words, while it is about 30 for the FTB, which implies that the task of parsing the FTB is much more difficult, as the syntax of the sentences it contains is more complex

It is nevertheless possible to draw several conclusions from these experiments. First, as already known, the training corpus size is very important to obtain high scores in parsing. But we could not obtain a large size corpus with perfect lemmas for Old French. Secondly, we can see that, for the available quantity of training data, the results obtained by the trained parser are already not bad. This means that, at the syntactic level, Old French is "regular" enough for training a parser.

	FTB train $\approx$ 450000 w.	FTB train $\approx$ 16000 w. (average of 5 experiments)
UAS	88.80	81.19
LAS	85.58	76.04

Table 10: Baseline on the French Treebank using *Mate*

5 Conclusion

In this paper, we have explored the difficulties and possible improvements of statistical parsing of a poorly standardized language. The results of our experiments show that a main issue in the process is the lack of a correct lemmatization, which percolates through the whole parsing process and is partly responsible for the final parsing quality. We managed to get around the poor lemmatization by trying to directly influence the quality of POS tagging and by doing so we obtained far better results than what has been achieved previsously on Old French. Adding external resources seems to be one of the keys to increase the final score. With these experiments, we managed to obtain a correct parsing quality, which could provide a

reasonable base for manual correction, but the results remain well below the level of the scores obtained for contemporary more standardized languages. Note, however, that our training sets of dependency trees were relatively small compared to other treebanks.

In order to obtain a better comparison we decided to test on corpora of approximatively the same size. These experiments confirmed again that a poorly standardized corpus results in a huge drop on LAS scoring. Moreover, the fact that the results show similar scores in UAS can be analyzed as a symptom of the higher variability in Old french.

In the present state of our experiments, it remains difficult to draw some solid linguistic conclusions. The relatively good scores when training *Aucassin* on *Roland* is somewhat unexpected, since *Aucassin* is far later than *Roland* and partially written in prose (whereas *Roland* is written in verse). Both also differ in genres. *Conqueste* is known to be somewhat untypical with regard to some syntactic features, as well as rather marked from a lexical and morphological point of view, which could explain the fact that we obtain worse scores with it than with the other texts. *Graal* and *Yvain*, though differing in their form, are not very distant in time, and moreover they share some common literary themes: this could explain the relatively good scores.

These brief linguistic conclusions certainly deserve further investigation. The asymmetries of our cross-trained tables should be further analysed. It is also still not clear whether the efficiency of a parser trained on one text and applied to another one is correlated with the historic proximity of the writing period, with the texts' genres, or more basically simply with the texts' length (remember that *Aucassin* is smaller than the other texts). If it appears to be linguistically relevant, the results of cross-training a syntactic parser could be used as a distance measure between genres and origin time of texts.

Note also that the variability explored here is mainly of a lexical nature. Only a serious study of the syntactic variations (e.g. word order of Old French is freer than in contemporary French) and its influence on the machine learning process could improve the scope of the results.

References

[1] Anne Abeillé, Lionel Clément, and François Toussenel. Building a treebank for french. In *Treebanks*, pages 165–187. Springer, 2003.

[2] Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *The 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China, 2010.

[3] Marie Candito, Benoît Crabbé, Pascal Denis, et al. Statistical french dependency parsing: treebank conversion and first results. In *Proceedings of*

the Seventh International Conference on Language Resources and Evaluation (LREC 2010), pages 1840–1847, 2010.

[4] Céline Guillot, Alexei Lavrentiev, and Christiane Marchello-Nizia. Document 2. les corpus de français médiéval : état des lieux et perspectives. *Revue française de linguistique appliquée*, XII:125–128, 2007.

[5] Pierre Kunstmann and Achim Stein. Le nouveau corpus d'amsterdam. In *"Actes de l'atelier de Lauterbad"*, pages 9–27, 2007.

[6] John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pages 282–289, Seattle, Washington, 2001.

[7] Thomas Lavergne, Olivier Cappé, and François Yvon. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July 2010.

[8] Nicolas Mazziotta. Logiciel notabene pour l'annotation linguistique. annotations et conceptualisations multiples. In *Recherches qualitatives. Hors-série "Les actes"*, volume 9, 2010.

[9] Achim Stein. Parsing heterogeneous corpora with a rich dependency grammar. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).

[10] Achim Stein and Sophie Prévost. Syntactic annotation of medieval texts: the syntactic reference corpus of medieval french (srcmf). In Tübingen: Narr, editor, *New Methods in Historical Corpus Linguistics*. 2013. Corpus Linguistics and International Perspectives on Language, CLIP Vol. 3, P. Bennett, M. Durrell, S. Scheible and R. Whitt (eds).

Consistency of Manual Sense Annotation and Integration into the TüBa-D/Z Treebank

Verena Henrich and Erhard Hinrichs

Department of Linguistics
University of Tübingen
E-mail: {vhenrich,eh}@sfs.uni-tuebingen.de

Abstract

Since sense-annotated corpora serve as gold standards for the development, training, and evaluation of word sense disambiguation (WSD) systems, their availability is a necessary prerequisite for WSD. The purpose of the present paper is to describe the manual annotation of a selected set of lemmas in the TüBa-D/Z treebank [9, 21] with senses from the German wordnet GermaNet [7, 8]. With the sense annotation for a selected set of 109 words (30 nouns and 79 verbs) occurring together 17 910 times in the TüBa-D/Z, the treebank currently represents the largest manually sense-annotated corpus available for GermaNet.

This paper describes the annotation process, presents statistics, analyzes inter-annotator agreement (and disagreement), and documents the technical integration of the sense annotations into the most recent release 9.1 of the TüBa-D/Z. The publication of this paper is accompanied by making the described sense annotations available to the research community.

1 Introduction and Motivation

The purpose of this paper is to describe the manual sense-annotation of a selected set of lemmas in the TüBa-D/Z treebank [9, 21] with special focus on inter-annotator agreement and technical integration into the TüBa-D/Z.[1] The sense inventory used for tagging word senses is taken from GermaNet [7, 8]. The underlying textual resource, the TüBa-D/Z treebank, is a German newspaper corpus already semi-automatically enriched with high-quality annotations at various levels of language including parts of speech, morphology, syntactic constituency, etc. The use of treebank data is motivated by at least two considerations: (i) the grammatical information contained in a treebank makes it possible to utilize deep syntactic and

[1] The present paper substantially extends the research described earlier in [9]: it presents updated statistics on the sense annotation, reports frequency, polysemy, and inter-annotator agreement for each lemma, and documents the technical integration into the treebank.

semantic information for word sense disambiguation [6, 3]; (ii) since the TüBa-D/Z is based on newspaper text, this ensures a broad coverage of topical materials such as politics, economy, society, environmental issues, sports, arts and entertainment.

This paper starts with a brief overview of the resources TüBa-D/Z and GermaNet (Section 2) and with statistics on the sense annotations, including frequency, polysemy, and inter-annotator agreement for each lemma (Section 3). Sections 4 and 5 describe the annotation process and analyze inter-annotator agreement (and disagreement) for the annotated lemmas. The technical integration of the sense annotations into the most recent release of the TüBa-D/Z is documented in Section 6. Finally, Section 7 concludes with a comparison to related work and an outlook to future work.

The publication of this paper is accompanied by making the described sense annotations available to the research community. Therefore, the sense annotations have been integrated into release 9.1 of the TüBa-D/Z (as of December 2014).

2 Resources

The Underlying Textual Resource for the sense-tagged corpus presented here is the syntactically annotated Tübingen Treebank of Written German (TüBa-D/Z) [21], the largest manually annotated treebank for German. It includes the following annotation layers: part-of-speech, inflectional morphology, lemmatization, syntactic constituency, grammatical functions, named entity classification, anaphora and coreference relations. The textual material for the treebank is taken from the daily newspaper "die tageszeitung" (taz). The current release 9.1 of the TüBa-D/Z (as of December 2014) contains 1 569 916 tokens occurring in 85 358 sentences that are taken from 3 444 newspaper articles.

The Sense Inventory for the sense-annotation is taken from GermaNet [7, 8], a lexical semantic network that is modeled after the Princeton WordNet for English [5]. It represents semantic concepts as *synsets*, i.e., as sets of (near-)synonymous words (referred to as *lexical units*), that are interlinked by semantic relations. GermaNet covers the three word categories of adjectives, nouns, and verbs, each of which is hierarchically structured in terms of the hypernymy relation of synsets. GermaNet's version 9.0 (release of April 2014) contains 121 810 lexical units, which are grouped into 93 246 synsets. Using a wordnet as the gold standard for the sense inventory is fully in line with standard practice for English where the Princeton WordNet is typically taken.

3 Sense-Annotated Lemmas

The sense annotation in the TüBa-D/Z is geared toward the lexical sample task in word sense disambiguation (as in many existing corpora [2, 11, 13, 15, 17, 19]), rather than toward the all-words task. The decision for sense annotation of a lexical sample is motivated by the requirements of machine learning. Sense-annotated

data are useful for training (semi-)automatic machine learning models only if there are sufficiently many instances for each item to be classified. Due to limitations of how much text can reasonably be annotated manually in an all-words, sense-annotated corpus, the resulting numbers of instances for each token are only of sufficient frequency for machine-learning applications if manual sense annotation concentrates on a lexical sample.

A total of 109 lemmas (30 nouns and 79 verbs) were selected for manual sense annotation. These lemmas have two or more senses in GermaNet and occur at least 21 times in the TüBa-D/Z – to ensure a reasonable lower bound of data instances for machine learning purposes. The average frequency for an annotated lemma is 164. Altogether, 17 910 occurrences of the 109 lemmas are sense annotated in the TüBa-D/Z 9.0.[2]

The 30 annotated nouns occur 8 803 times in the treebank – at least 24 times and at most 1 699 times. On average, there are 293 annotated occurrences per noun lemma. The average polysemy (number of senses in GermaNet) is 4.1 for the annotated nouns, ranging from 2-7 senses. Table 3 lists the nouns in decreasing order of their number of occurrences in the treebank (column *Freq.*). Column *GN* contains the noun's number of senses in GermaNet, while column *#s* lists the numbers of senses for which at least one annotation exists in the treebank. A superscript + indicates the existence of at least one token for which no GermaNet sense is annotated. These cases occur, for example, for idiomatic expressions or figurative meanings where it is not obvious from the context which sense to chose. Inter-annotator agreement (columns *IAA* and κ) of the manual sense annotation will be discussed in Section 5 below.

Nouns	**Freq.**	**GN**	**#s**	**IAA**	κ
Frau	1699	3	3	98.7	96.0
Mann	1114	3	3	98.8	94.7
Land	1112	7	7^+	97.9	95.5
Partei	811	3	3	97.3	62.6
Haus	789	5	5	85.8	60.7
Grund	460	5	5	99.8	96.9
Stunde	426	4	4	98.1	92.8
Stimme	289	4	4	98.0	96.0
Mal	284	2	1	100.0	NaN
Kopf	269	6	4^+	97.8	84.0
Band	159	6	5	98.7	96.9
Tor	137	4	4	100.0	100.0
Fuß	129	3	3	99.1	79.7
Höhe	126	4	4	65.8	11.3
Freundin	122	3	2	97.1	95.9

Nouns	**Freq.**	**GN**	**#s**	**IAA**	κ
Anschlag	99	5	3^+	95.9	61.4
Spur	94	5	5^+	84.7	73.1
Bein	91	3	1^+	97.4	-1.3
Runde	83	6	6	92.0	88.1
Karte	76	4	4	99.6	100.0
Sender	76	5	4	83.8	60.8
Stuhl	60	3	3	98.0	100.0
Ausschuß	50	2	1	100.0	NaN
Bestimmung	48	6	4	90.8	79.2
Gewinn	48	3	3	95.7	89.2
Überraschung	42	3	3	96.6	93.0
Teilnahme	37	3	1	98.9	NaN
Kette	25	4	4	73.9	62.5
Abfall	24	4	2	100.0	100.0
Abgabe	24	5	3	100.0	100.0

Table 1: 30 sense-annotated nouns.

[2] All statistics reported in the present paper are taken from the most recent versions of the TüBa-D/Z (release 9.1) and GermaNet (release 9.0) – with the only exception of inter-annotator agreement figures, which were calculated on the basis of TüBa-D/Z 8.0 and GermaNet 8.0, as the most current versions available at the time the annotation process started.

For verbs, 9 107 occurrences are annotated with the senses of 79 verb lemmas. The average occurrence per verb lemma is 115 with the least frequent verb occurring 21 times, the most frequent one 801 times. The average polysemy is 2.8, with the most polysemous verb showing 14 senses in GermaNet. Table 3 lists the 79 selected lemmas for verbs – again ordered by their frequency in the TüBa-D/Z (column *Freq.*) and with their number of senses in GermaNet (column *GN*), the number of occurring senses (column *#s*), as well as inter-annotator agreement (columns *IAA* and κ).

Verbs	**Freq.**	**GN**	**#s**	**IAA**	**κ**
heißen	801	4	4	94.0	90.9
gelten	502	5	5	97.7	96.2
setzen	404	14	9^{+}	79.5	75.4
erhalten	399	4	4^{+}	89.2	76.4
sitzen	345	7	6^{+}	92.4	85.6
fragen	344	2	2	99.7	99.0
aussehen	231	2	2	92.1	75.8
reden	227	3	3^{+}	80.0	45.8
sterben	220	2	2	98.4	79.2
ankündigen	211	2	2	100.0	100.0
unterstützen	188	2	2	95.7	38.1
bedeuten	187	3	3	98.1	79.2
verkaufen	186	5	4^{+}	92.5	75.8
verurteilen	180	2	2^{+}	96.0	86.6
leisten	176	3	3	90.3	83.1
bauen	167	3	3^{+}	95.4	83.2
verschwinden	159	2	2	73.8	47.4
gründen	148	4	4	99.2	93.0
reichen	146	4	4	95.6	91.8
geschehen	145	2	2^{+}	98.4	49.5
herrschen	128	2	2	99.0	90.4
präsentieren	127	2	2	98.9	98.0
informieren	125	2	2	98.2	93.0
freuen	121	2	2	90.9	64.0
verdienen	115	2	2	100.0	100.0
demonstrieren	111	3	3	87.5	77.5
holen	110	5	5^{+}	74.2	65.4
aufrufen	105	2	2	99.0	66.3
verfolgen	105	6	6	91.8	89.0
weitergehen	101	2	2	98.9	88.3
besitzen	99	2	2	92.7	84.2
versichern	99	3	3	96.5	80.9
vorliegen	96	2	2	95.0	84.9
enthalten	94	2	2	100.0	100.0
liefern	93	4	4^{+}	88.0	82.5
erweisen	89	3	3	97.2	84.6
existieren	88	2	2	98.6	0.0
drängen	84	3	3	88.2	81.7
behandeln	82	4	4	85.6	78.0
begrüßen	79	2	2	98.5	96.2

Verbs	**Freq.**	**GN**	**#s**	**IAA**	**κ**
beschränken	78	2	2	96.7	93.2
betragen	73	2	1	100.0	NaN
beraten	70	3	3	90.8	81.4
merken	62	2	2	98.1	89.9
entziehen	61	3	2	96.2	92.6
widmen	60	2	2	100.0	100.0
empfehlen	58	3	3	100.0	100.0
gestalten	57	2	2	96.3	78.0
bekennen	54	2	2	97.9	95.7
wundern	54	2	2	97.8	94.5
auffallen	51	2	2	97.8	95.1
engagieren	51	2	2	100.0	100.0
raten	46	2	2	100.0	100.0
rücken	45	2	2	100.0	100.0
bedenken	43	3	2	97.6	94.0
versammeln	42	2	2	97.0	93.9
vollziehen	42	2	2	96.9	93.4
erweitern	41	2	2	97.1	84.1
zugehen	41	6	4^{+}	94.3	87.6
gestehen	40	2	2	90.0	80.0
berufen	39	2	2	100.0	100.0
klappen	39	3	2	100.0	100.0
kündigen	39	2	2	71.4	42.9
trauen	39	4	4	92.2	91.1
stehlen	36	2	1^{+}	100.0	100.0
verstoßen	36	2	2	100.0	100.0
zurückgeben	36	2	2	100.0	100.0
ärgern	36	2	2	100.0	100.0
befassen	35	2	2^{+}	96.9	86.0
einschränken	35	2	2	100.0	100.0
identifizieren	34	3	3	92.6	88.3
beschweren	33	3	2	100.0	100.0
vorschreiben	31	2	1	96.2	0.0
nützen	29	2	2	100.0	NaN
kleben	28	2	2	87.5	74.3
verdoppeln	26	2	2	100.0	100.0
fressen	25	3	2	72.7	54.3
wiedergeben	24	3	2	76.2	40.0
verlesen	21	2	1	100.0	NaN

Table 2: 79 sense-annotated verbs.

4 Annotation Process

In order to assure good quality of the manual sense annotation and to calculate inter-annotator agreement, sense annotation is independently performed by two annotators (native German computational linguists) for all word lemmas and occurrences. The annotators have the possibility to indicate problematic word occurrences with comments to be discussed separately.

The manual annotation is performed lemma-by-lemma (as in many related annotation projects, for example, [6], [11], [17], and [20]), i.e., an annotator first takes a look at all senses of a word in GermaNet and then – having in mind all possible senses – annotates each occurrence of that word in the TüBa-D/Z with the corresponding sense from GermaNet.

For each occurrence of a word in the treebank, the annotators are supposed to select exactly one GermaNet sense from the list of possible word senses, if possible. Since it is not always possible to select exactly one sense, i.e. it is either unclear or undecidable which of two senses is illustrated or none of the senses is plausible, the annotation guidelines allow the assignment of multiple senses or no sense for a word occurrence. The need to annotate more than one sense does not arise very often. This confirms both the results of [19] who annotated only 79 out of 2 421 occurrences with multiple senses, as well as the findings of [22, page 6] that "the average number of senses [...] is not very high, which shows that annotators have a tendency to avoid multiple answers."

An experienced lexicographer, who is a native speaker of German and who has been the main responsible expert for the lexicographic extension of GermaNet for several years, supervises the two annotators. In an adjudication step, the experienced lexicographer goes through all occurrences, where the two annotators either do not agree or at least one of them had a comment, and resolves disagreements. This procedure of conducting independent annotations with an adjudication step afterwards is along the lines with most other sense-annotation projects, including for example [6], [12], and [17].

Where annotators found during the annotation process that a sense is missing from GermaNet, GermaNet is updated to include that sense. If the TüBa-D/Z contains occurrences of senses for the selected lemmas that are currently not covered by GermaNet, the two annotators indicate for these occurrences that a sense is missing in GermaNet. For example, the noun *Mann* had the following two senses in GermaNet 7.0: (i) 'man' in the general sense of an adult male person, and (ii) 'husband' in the more specific sense of a married man. In sentence (1), the noun *Mann* is used as a 'unit for counting manpower'.

(1) *Er will die Personalstärke der Bundeswehr auf 270.000 Mann reduzieren.*[3]
('He wants to reduce the manning level of the German Armed Forces to 270,000 men.')

[3] Sentence 10 692 from TüBa-D/Z 9.

The lexicographic expert decides whether to add a missing sense to GermaNet. In the case of *Mann*, the mentioned *counting unit* sense has been included. The subsequent update of the sense inventory during the sense annotation process brings about mutual benefits both for the sense inventory which is being extended as well as for the sense-annotated corpus which profits from a feasible sense inventory. Such an update is common practice for all those annotation projects where the sense-annotated corpus is being created by the same research group which maintains the sense inventory (e.g., [11], [14], [15], and [17]).

5 Inter-Annotator Agreement

An inter-annotator agreement (IAA) score is calculated to assess the reliability of the manual sense annotations. The calculated percentage of IAA accounts for partial agreement using the Dice coefficient [22]. The overall percentage of agreement, which is obtained by averaging the Dice coefficient for all annotated occurrences of the word category in question, is 96.4% for nouns and 93.7% for verbs. This corresponds to Cohen's kappa κ [4] values of 85.4% and 82.4% for nouns and verbs, respectively.

The agreement for each of the 30 nouns is documented in columns *IAA* and κ in Table 3. With the two exceptions of *Höhe* (65.8%) and *Kette* (73.9%), the calculated IAA values for all other nouns are at least above 80%, mostly even above 90%. The explanation for the low agreement of the noun *Höhe* is due to the semantic distinction of the two word senses in GermaNet which are very fine-grained and turned out to be difficult to distinguish during the manual annotation process. The reason for a low performance for *Kette* stems from a subsequent restructuring of the sense inventory during the annotation process. A revision of senses in GermaNet has been performed after one annotator had already tagged 5 occurrences (which constitute already about 20%) with a sense of *Kette* that has been deleted. The tagging by the second annotator is conducted on the already revised set of senses and thus the deleted word sense is never chosen.

The kappa coefficients (column κ in Table 3) show a much higher deviation compared to the percentages of IAA. Here, the two by far worst results are obtained for *Höhe* (11.3) and *Bein* (-1.3), while all other kappa values lie above 60. The low κ for *Höhe* was to be expected as a result of an already low percentage of IAA. The explanation for a negative value for κ is that there is even less agreement between the annotators than an agreement by chance would be. The reason for the negative kappa value for *Bein* is due to the skewed distribution of annotated senses, i.e., the same predominant sense is assigned to all except one occurrence. The agreement by chance is thus nearly 1 and a deviation in the manual annotation influences the calculated coefficient enormously. For lemmas where both annotators always pick the same sense for all occurrences, Cohen's kappa is not informative. It is technically not possible to calculate the coefficient for these lemmas, because the agreement by chance is 1, which would result in a division

by zero. This is the reason why there are no κ values for the three nouns *Mal*, *Ausschuss*, and *Teilnahme*.

A detailed inspection of the IAA for single words did not show a correlation between the IAA and the polysemy of a noun. The Pearson correlation coefficient [18] between the IAA and the number of senses a noun has is -0.03, with a p-value of 0.88, i.e., without statistical significance. For example, the most problematic nouns mentioned above show different numbers of senses, i.e., 2 for *Höhe*, 3 for *Bein*, and 4 for *Kette*. Further, the reasons for the annotator disagreement are diverse and obviously not connected to the polysemy of a word, i.e., unclear sense distinction, skewed distribution of annotated senses, and subsequent update of the sense inventory, respectively. The other way around, the IAA values for the most polysemous nouns, i.e., 97.9% for *Land* (7 senses), 97.8% for *Kopf*, 92.0% for *Runde*, and 90.8% for *Bestimmung* (6 senses each) are comparable to the average of 96.4% for all annotated nouns. For nouns, this finding that there is no obvious correlation between the IAA and a word's polysemy corroborates the results reported by [6] on sense-annotating the Penn Treebank with senses from WordNet.

For each of the 79 verbs, the percentage of inter-annotator agreement (column *IAA*) and Cohen's kappa (column κ) are listed in Table 3. In general, the inter-annotator agreement for verbs is slightly lower than for nouns. However, similar to nouns, the calculated IAA values for most verbs are at least above 80%, mostly even above 90%. The few exceptions with a higher disagreement are *verschwinden* with 73.8%, *holen* with 74.2%, *kündigen* with 71.4%, *fressen* with 72.7%, and *wiedergeben* with 76.2%. For most of them (i.e., for *verschwinden*, *holen*, *kündigen*, and *wiedergeben*), the difficulty is mainly caused by a fine-grained distinction of senses which make an unambiguous annotation difficult. This detrimental effect for very fine-grained word senses was already observed by [16, page 97]. They report an improvement in the inter-annotator agreement from 71.3 to 82% for the same SensEval-2 lexical sample task when more coarse-grained verb senses are used instead of the fine-grained distinctions taken from WordNet 1.7. In the case of *holen*, an additional complexity arises due to the addition of two new word senses during the annotation process. For the verb *fressen*, most disagreements occur for transferred usages of the verb.

For the same reasons that cause a lower percentage of IAA, it was expected for those verbs to yield lower κ scores, which turned out to be true (i.e., *verschwinden* (47.4), *holen* (65.4), *kündigen* (42.9), *fressen* (54.3), and *wiedergeben* (40.0)). The explanation for kappa coefficients of 0.0 for the two verbs *existieren* and *vorschreiben* is a skewed distribution of annotated senses. Both for *existieren* and for *vorschreiben*, all except one occurrence (by one of the two annotators) are assigned the same word sense. This results in an agreement by chance of nearly 1 which in turn results in a very low kappa coefficient.[4] For the verbs *betragen*,

[4]Since the chance agreements for these cases are nearly 1, Cohen's kappa is basically not informative for those cases, but since it is not exactly 1, it is technically possible to calculate the coefficient.

nützen, and *verlesen* no κ values are given because both annotators always picked one sense for all occurrences and thus the coefficient is not informative for those lemmas.

For verbs, the observation for the TüBa-D/Z sense-annotation differs from [6]'s finding that there is no obvious correlation between the IAA and a word's polysemy when sense-annotating the Penn Treebank. For the sense-annotation in the TüBa-D/Z, the Pearson correlation coefficient between the inter-annotator agreement and the polysemy of a verb is -0.39. The coefficient's absolute value is not remarkably high[5] to claim a strong correlation, but there is at least a higher correlation than for nouns, and, with a p-value smaller than 0.001, the correlation for verbs is statistically significant.

Overall, the reported percentage of IAA is very high. The values are comparable to the agreement statistics reported in [19] for their work in creating a German sense-annotated corpus. The observed agreement values are much higher than those observed for English. [22], for example, observes a pairwise Dice coefficient of 73% for nouns and 63% for verbs. [16] report an inter-annotator agreement of 71.3% for the English verb lexical sample task for SensEval-2. The reason for much higher IAA values for German than for English is the different number of distinct senses: an average of 4.1 for German nouns and 2.8 for German verbs as opposed to an average of 7.6 for English nouns and 12.6 for English nouns in the case of [22, Table 3].

6 Technical Integration into the Treebank

The TüBa-D/Z is released in different data formats, including NeGra export, export XML, Penn Treebank, TIGER-XML, and CoNLL. However, only three of the data formats in which the TüBa-D/Z is released are suitable to be extended with sense annotation. For each data format, the sense annotation refers to the sense in GermaNet 9.0, encoded by the ID of the corresponding lexical unit. In those cases, where no GermaNet sense can be annotated, the corresponding ID is set to `-1`. This occurs, for example, for idiomatic expressions or figurative meanings where it is not obvious from the context which sense to choose. For those annotations for which more than one GermaNet sense is selected, multiple lexical unit IDs are encoded. Such multiple IDs signal that either two senses are jointly represented by a specific word token or that it is undecidable which of two senses is represented by a specific word token.

The following subsections describe for each data format the appropriate extension needed for sense annotation. The newly added WSD information is underlined in the corresponding examples.

[5] Since 'not remarkably high' describes the absolute value of the correlation coefficient, it means that the coefficient is 'not remarkably distinct from zero'.

NeGra export format

The line-oriented NeGra export format [1] represents each word token and each syntactic phrase node in a separate line. The following example shows an example line that encodes a word token:

```
sitzt  sitzen  VVFIN  3sis  HD  503  %% LU=112247
```

The line encodes the word token itself (`sitzt`), followed by its lemma (`sitzen`), by its part-of-speech tag (`VVFIN`), its morphological tag (`3sis`), its phrasal edge label (`HD`), and its phrasal parent's node ID (`503`). In analogy to the annotation of referential relations in the TüBa-D/Z, sense annotations encoding the IDs of GermaNet lexical units are appended as comments to the corresponding lines representing the sense-annotated words in question (`%% LU=112247`).

Export XML format

This format represents all treebank information in a hierarchical XML structure. It encodes word tokens as separate XML elements. The following export XML example shows the same example as given above for the NeGra export format:

```
<word xml:id="s9_4" form="sitzt" pos="VVFIN" morph="3sis"
lemma="sitzen" func="HD" parent="s9_503" wsd-lexunits="112247"/>
```

The XML element `word` represents the word token in question with the XML attribute `form` for the token itself, the attribute `pos` for its part-of-speech tag, attribute `lemma` for its lemma, attribute `func` for its phrasal edge label, and attribute `parent` for the node ID of its phrasal parent. An `wsd-lexunits` XML attribute is added to the `word` element, which encodes the ID of the corresponding GermaNet lexical unit.

CoNLL 2012

The CoNLL format (in version 2012)[6] also encodes each token in a separate line. It provides a separate column for word senses. The following example again represents the same token as before:

```
T990507.2  9  4  sitzt  VVFIN  (LK:-(VXFIN:HD*))  sitzen  112247
```

The line first encodes a document ID (`T990507.2`), then the sentence number (`9`), followed by the word number within the sentence (`4`), followed by the word itself (`sitzt`), followed by its part-of-speech tag (`VVFIN`), followed by the corresponding parse bit (`(LK:-(VXFIN:HD*))`), by the lemma (`sitzen`), and the word sense ID (`112247`).[7]

[6] http://conll.cemantix.org/2012/data.html

[7] Please note that the above example only displays a subset of all columns included in the canonical CoNNL 2012 format. Some columns, which encode information such as named entities, coreference, etc. and which are irrelevant for the example at hand, are omitted for reasons of space. Note

7 Related and Future Work

By integrating the sense annotations described in this paper into the most recent release of the TüBa-D/Z (release 9.1 as of December 2014), the sense annotations have been made available to the research community.[8] In terms of quantity, the present study significantly goes beyond previous efforts on creating a German sense-annotated corpus in that 17 910 occurrences from the TüBa-D/Z treebank are annotated with the GermaNet senses of 30 nouns and 79 verbs.

Related studies which created sense-annotated corpora for German include [2, 10, 19, 23]. [2] annotated the GermaNet senses of 40 word lemmas (6 adjectives, 18 nouns, and 16 verbs) in 1 154 occurrences in the deWaC corpus. [19] annotated 2 421 occurrences of the EuroWordNet-GermaNet senses of 25 nouns in a medical corpus obtained from scientific abstracts from the Springer Link website. The same medical corpus was annotated by [23] with 24 ambiguous UMLS types – each of which occurs at least 11 times in the corpus (seven occur more than 100 times). [10] constructed the sense-annotated corpus WebCAGe semi-automatically by annotating more than 10 000 word occurrences in web-harvested texts with GermaNet senses of more than 2 000 word lemmas.

In future work, the sense annotations described in this paper will be used as a gold standard to evaluate German word sense disambiguation systems. The implementation of automatic disambiguation systems will employ the treebank's grammatical information and investigate the impact of deep syntactic and semantic information for word sense disambiguation.

Acknowledgements

The research reported in this paper was jointly funded by the SFB 833 grant of the DFG and by the CLARIN-D grant of the BMBF. We are very grateful to Valentin Deyringer and Reinhild Barkey for their help with the annotations. Special thanks go to Yannick Versley for his support with the sense-annotation tool, to Scott Martens and Christian M. Meyer for their input on how to calculate inter-annotator agreement, and to Marie Hinrichs and Jianqiang Ma for their help with the technical integration into the TüBa-D/Z.

References

[1] Brants, Thorsten (1997) *The NeGra Export Format for Annotated Corpora (Version 3)*. Technical report, Computerlinguistik, Universität des Saarlandes, Germany.

also that the CoNLL 2012 format for the TüBa-D/Z data uses the column 'lemma' to specify the citation form of a word. This differs from the original CoNLL 2012 format where the column 'lemma' refers to predicates that have semantic role information.

[8] http://www.sfs.uni-tuebingen.de/en/ascl/resources/corpora/sense-annotated-tueba-dz.html

[2] Broscheit, Samuel, Frank, Anette, Jehle, Dominic, Ponzetto, Simone Paolo, Rehl, Danny, Summa, Anja, Suttner, Klaus and Vola, Saskia (2010) Rapid bootstrapping of Word Sense Disambiguation resources for German. In *Proceedings of the 10. Konferenz zur Verarbeitung Natürlicher Sprache*, pp. 19–27, Saarbrücken, Germany.

[3] Chen, Jinying and Palmer, Martha (2009) Improving English Verb Sense Disambiguation Performance with Linguistically Motivated Features and Clear Sense Distinction Boundaries. In *Language Resources and Evaluation*, volume 43, pp. 181–208, Springer Netherlands.

[4] Cohen, Jacob (1960) A Coefficient of Agreement for Nominal Scales. In *Educational and Psychological Measurement*, 20(1), pp. 37–46.

[5] Fellbaum, Christiane (ed.) (1998) *WordNet – An Electronic Lexical Database*. The MIT Press.

[6] Fellbaum, Christiane, Palmer, Martha, Dang, Hoa Trang, Delfs, Lauren and Wolf, Susanne (2001) Manual and Automatic Semantic Annotation with WordNet. In *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources Applications Customizations*, pp. 3–10.

[7] Hamp, Birgit and Feldweg, Helmut (1997) GermaNet – a Lexical-Semantic Net for German. In *Proceedings of ACL Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, Madrid, Spain.

[8] Henrich, Verena and Hinrichs, Erhard (2010) GernEdiT – The GermaNet Editing Tool. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, pp. 2228–2235, Valletta, Malta.

[9] Henrich, Verena and Hinrichs, Erhard (2013) Extending the TüBa-D/Z Treebank with GermaNet Sense Annotation. In *Gurevych, Iryna, Biemann, Chris and Zesch, Torsten (eds.) Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, ISBN 978-3-642-40721-5, pp. 89–96. Springer Berlin Heidelberg.

[10] Henrich, Verena, Hinrichs, Erhard and Vodolazova, Tatiana (2012): WebCAGe — A Web-Harvested Corpus Annotated with GermaNet Senses. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France, pp. 387–396.

[11] Kilgarriff, Adam (1998) Gold standard datasets for evaluating word sense disambiguation programs. In *Computer Speech & Language*, volume 12, number 4, pp. 453–472.

[12] Kilgarriff, Adam and Rosenzweig, Joseph (2000) Framework and Results for English SENSEVAL. In *Computers and the Humanities*, 34(1-2) pp. 15–48.

[13] Mihalcea, Rada, Chklovski, Timothy and Kilgarriff, Adam (2004) The Senseval-3 English lexical sample task. In *Mihalcea, Rada and Edmonds, Phil (eds.) Senseval-3: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 25–28, Association for Computational Linguistics, Barcelona, Spain.

[14] Miller, George A., Leacock, Claudia, Tengi, Randee and Bunker, Ross T. (1993) A Semantic Concordance. In *Proceedings of the 3 DARPA Workshop on Human Language Technology*.

[15] Palmer, Martha, Fellbaum, Christiane, Cotton, Scott, Delfs, Lauren and Dang, Hoa Trang (2001) English Tasks: All-Words and Verb Lexical Sample. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pp. 21–24, Association for Computational Linguistics, Toulouse, France.

[16] Palmer, Martha, Ng, Hwee Tou and Dang, Hoa Trang (2006) Evaluation of WSD Systems. In *Agirre, Eneko and Edmonds, Philip (eds.) Word Sense Disambiguation: Algorithms and Applications*, chapter 4, volume 33, Text, Speech and Language Technology series, pp. 75–106, ISBN 978-1-4020-4808-1, Springer Netherlands.

[17] Passonneau, Rebecca, Baker, Collin, Fellbaum, Christiane and Ide, Nancy (2012) The MASC Word Sense Sentence Corpus. In *Proceedings of the Eighth Language Resources and Evaluation Conference*, Istanbul, Turkey.

[18] Pearson, Karl (1896) Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity, and Panmixia. In *Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, volume 187, pp. 253–318, doi:10.1098/rsta.1896.0007.

[19] Raileanu, Diana, Buitelaar, Paul, Vintar, Spela and Bay, Jörg (2002) Evaluation Corpora for Sense Disambiguation in the Medical Domain. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas, Canary Islands, Spain.

[20] Saito, Jahn-Takeshi, Wagner, Joachim, Katz, Graham, Reuter, Philip , Burke, Michael and Reinhard, Sabine (2002) Evaluation of GermanNet: Problems Using GermaNet for Automatic Word Sense Disambiguation. *Proceedings of the Workshop on "Wordnet Structures and Standardization, and how these affect Wordnet Applications and Evaluation" at LREC 2002*, pp. 14–19, Las Palmas, Grand Canaria.

[21] Telljohann, Heike, Hinrichs, Erhard W., Kübler, Sandra, Zinsmeister, Heike and Beck, Kathrin (2012) *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Technical report, Department of General and Computational Linguistics, University of Tübingen, Germany.

[22] Véronis, Jean (1998) A study of polysemy judgments and inter-annotator agreement. In *Proceedings of SENSEVAL-1*, Herstmonceux Castle, England.

[23] Widdows, Dominic, Peters, Stanley, Cederberg, Scott, Chan, Chiu-Ki, Steffen, Diana and Buitelaar, Paul (2003) Unsupervised monolingual and bilingual word-sense disambiguation of medical documents using umls. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*, Volume 13, BioMed '03, pages 9–16, Stroudsburg, PA, USA, Association for Computational Linguistics.

Deriving Multi-Headed Planar Dependency Parses from Link Grammar Parses*

Juneki Hong

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
juneki@cs.cmu.edu

Jason Eisner

Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218, USA
jason@cs.jhu.edu

Abstract

Under multi-headed dependency grammar, a parse is a connected DAG rather than a tree. Such formalisms can be more syntactically and semantically expressive. However, it is hard to train, test, or improve multi-headed parsers because few multi-headed corpora exist, particularly for the projective or planar case. To help fill this gap, we observe that link grammar already produces *undirected* planar graphs, and so we wanted to determine whether these could be converted into directionalized dependency parses. We use Integer Linear Programming to assign consistent directions to the labeled links in a corpus of several thousand parses produced by the Link Grammar Parser, which has broad-coverage hand-written grammars of English as well as Russian and other languages. We find that such directions can indeed be consistently assigned in a way that yields valid multi-headed dependency parses. The resulting parses in English appear reasonably linguistically plausible, though differing in style from CoNLL-style parses of the same sentences; we discuss the differences.

1 Motivation

Link Grammar [29] is a syntactic formalism in which a parse of a sentence is an undirected, edge-labeled, planar graph. The labeled edges of the graph represent syntactic relationships among the words. The vertices of the graph are simply the words $1, 2, \ldots, n$ of the sentence, along with a distinguished "root" vertex 0.

The small Link Grammar community has invested effort in creating link grammars for several languages. In this short paper, we consider whether their *undirected* parses can be converted automatically to *directed* ones. We have three motivations:

*This material is based upon work supported by the National Science Foundation under Grant No. 1423276. The work was mainly conducted while the first author was at Johns Hopkins University.

1. We were curious about the relation between link grammar annotation and dependency grammar annotation. We suspected that the link grammar parses could be interpreted as multi-headed dependency grammar parses. Although the link grammar authors did not bother to specify directions for the different edge types, we suspected that they essentially had such directions in mind.

2. Our problem provides a good case study for how to automatically enrich a corpus. Hand-constructed grammars or corpora sometimes provide a lighter level of annotation than desired. In our setting, the edges lack directions; in other settings, the syntactic categories may be coarser than desired, or some relations may be omitted. One may want to automatically enrich the annotation in such cases, whether by doing some kind of learning [17, et seq.], or by exploiting implicit constraints. In this paper, we use Integer Linear Programming to exploit implicit constraints of consistency and acyclicity.

3. The resulting parses may be useful data for experimenting with new parsing *algorithms*. There has been a good deal of recent research on projective dependency parsing, variously using global optimization or sequential classification (see [16, 3, 5] for surveys). Some of these algorithms could be extended to the *multi-headed* case, which is of linguistic and computational interest for reasons discussed below. However, for training and testing such algorithms, one would need a plausible sample of multi-headed projective dependency parses of real sentences. Our method cheaply manufactures such a sample, to compensate for the current lack of gold-standard data of this form.

Our automatic method for reconstructing the latent directions also had an unexpected benefit. It revealed an inconsistency in the hand-written English link grammar, regarding the handling of embedded sentences with missing (PRO) subjects.

2 Multi-Headed Dependency Parsing

Dependency parsing maps a sentence to a directed graph whose vertices are the words $1, 2, \ldots, n$ of the sentence along with a distinguished "root" vertex 0. A labeled directed edge $u \xrightarrow{L} v$ or $v \xleftarrow{L} u$ indicates that the "child" v is some kind of argument or modifier of its "parent" u. The edge label L indicates the specific syntactic or semantic relationship between the two words.

In the special case $u = 0$, the edge designates v as playing some special top-level role in the sentence, e.g., as the main verb. We disallow $v = 0$.

As discussed by [13, 9], one might impose various requirements on the parse graph:

- SINGLE-HEAD: Each word has ≤ 1 parent.
- ACYCLIC: There are no directed cycles.

- CONNECTED: Each pair of words has a undirected path between them.
- REACHABLE: Each word can be reached from 0 by a directed path (which implies CONNECTED). Note that 0 may have multiple children.
- PLANAR: edges may not "cross." That is, if there are edges between i, j and between u, v, where $i < u < j$, then PLANAR requires $i \leq v \leq j$.

It is common to impose all of these requirements at once, leading to a *projective dependency parser* that produces projective trees rooted at 0. However, parsing algorithms can be devised that relax any or all of the requirements [9].

In this paper, we are interested in relaxing the SINGLE-HEAD requirement while preserving all the others. This is the setting of *multi-headed projective dependency parsing*. Just as in the single-headed case, the other requirements ensure that all edges are projective. (A projective edge is one where the parent is an ancestor of all words between the parent and the child [19].)

Relaxing SINGLE-HEAD means that the parse can have more than n edges, allowing it to express more relationships between words. In English, for example, here are some constructions that seem to call for a multi-headed analysis:

control In *"Jill likes to skip,"* the word *Jill* is the subject of two verbs. In *"Jill persuaded Jack to skip," Jack* is the object of one verb and the subject of another. Without recognizing this, our parser would miss the syntactic invariants that *skip* always has a subject and *persuaded* always has an object. It would also be unable to exploit the selectional preferences of both verbs to help disambiguate the parse. This is why we prefer to make the parser aware of multi-headedness, rather than using a single-headed parser and then extracting the additional semantic roles from its output.

relativization In *"The boy that Jill skipped with fell down,"* the word *boy* is the object of *with* as well as the subject of *fell*. Without recognizing this, we would miss the syntactic invariant that *with* always has an object.

conjunction In *"Jack and Jill went up the hill," Jack* and *Jill* serve as the two arguments to *and*, but they are also semantically subjects of *went*. Without recognizing this, we would have no (local) reason for expecting the arguments of *and* to be nouns.

In linguistics, it is common to analyze some of these structures using trees with "empty categories." The subject of *skip* is taken to be a silent morpheme *PRO*: *"Jill$_i$ likes PRO$_i$ to skip."* However, this is no longer a tree if one considers the implicit undirected edge between *Jill* and *PRO* (denoted by their shared index i). Our simpler representation contracts this coreference edge, eliminating *PRO* and creating a *Jill* $\leftarrow$ *skip* link.

An anonymous reviewer objects to research on projective parsing algorithms, since the PLANAR restriction is linguistically questionable even for single-headed parsing, and even more so for multi-headed parsing. However, efficient algorithms

often exist in the projective case, and these projective algorithms—which are typically first developed on a projective corpus of the sort that we will construct—can be useful even when the true parses are not quite projective. Natural-language parses have a low rate of non-projective edges [19] and the non-projective parses tend to be "almost projective" [25]. Thus, one can apply a fast projective parser as an approximate method, or as one ingredient in a more complex model [15], or as the first step in a coarse-to-fine or stacking architecture [27, 16] in order to obtain preliminary edge scores that are then supplied to a non-projective parser. Another approach is to transform non-projective parses into a projective annotation style so that projective parsers can be used [22, 20].

3 Link Grammars

Graph representations of syntactic and semantic structure have been widely considered of late [7, 6, 10, 2, 23]. A few past NLP papers have explored multi-headed dependency parsing [4, 18, 28, 9]. They constructed their multi-headed dependency corpora by automatically converting from other formats such as HPSG. Currently there seem to be no corpora that were directly annotated in this form, other than the Danish Dependency Treebank [12].

The above work considered non-projective parses. It seems at first that no one has worked out annotation conventions for *projective* multi-headed dependency parsing. However, this is only half-true. Link Grammar [29] is a grammar-based formalism for projective dependency parsing with *undirected* links. It produces undirected connected planar graphs. Annotation conventions are implicit in the detailed lexicon for the Link Grammar Parser [30]. The 122 link types in the English lexicon are documented at `http://www.abisource.com/projects/link-grammar/dict/summarize-links.html`, which specifies for every word a constraint on the *sequence* of labeled leftward and rightward edges attached to it. As remarked by [8], this is analogous to dependency grammar's use of head automata to constrain a word's sequence of left and right children. For example, in *"The boy that Jill skipped with fell down,"* the word *with* uses a lexical entry that requires it to link to a governing verb to the left, an extracted object farther to the left, and nothing to the right. Each entry has a hand-assigned cost in {0,1,2}, and the parser finds the parse of minimum total cost [30, 31].

Given a link grammar parse, it would be straightforward to convert it to an acyclic dependency parse by orienting all edges rightward. However, the result may violate the REACHABLE constraint. Instead we could orient all edges by depth-first search from the root node, which yields a DAG satisfying all our constraints. However, this might result in inconsistent annotation conventions, with some `S`-labeled links pointing from subject to verb and others from verb to subject.

In the English link grammar, an `S` edge encodes a "subject-verb" relation *whose left word serves as the subject*. We would expect verbs to point to their subject arguments in dependency grammar, and so we surmise that all `S` links should be

interpreted as pointing leftward (from verb to subject: *"Jack* $\overset{S}{\leftarrow}$ *is falling"*).

In general, we supposed that the link grammar lexicon designers actually had a *consistent* direction in mind for each *edge label*. This does not imply that English subjects must always appear to the left of the verb! The link grammar designers took care to use a distinct `SI` label in cases of subject-verb inversion, and we surmise that `SI` links are intended to point rightward (again from verb to subject: *"Is* $\overset{SI}{\rightarrow}$ *Jack falling?"*). Similarly, different edge labels are used for English "object-verb" relations according to whether it is the left or the right word that serves as the object. These labels are presumably intended to encode different edge directions.

Our goal in this paper is to recover these implicit directions by global optimization. We seek a fixed mapping from labels to directions such that link grammar parses become directed dependency parses that satisfy all of our constraints.

Our first thought was to seek a direction mapping such that no parsed word sequence allowed by the link grammar lexicon could possibly violate our constraints after directions were imposed. This is a well-defined constraint programming problem. For example, to prevent cyclicity, we would require (roughly speaking) that no word type in the lexicon could follow a sequence of directed rightward links through other word types and then a leftward link back to itself.

However, we feared that there would not be a feasible solution—because of errors in the lexicon or linguistically unnatural word sequences not anticipated by the grammar designers. In this case it would be unclear how to relax our constraints.

Thus, rather than considering all theoretically possible word sequences, we chose to use a sample of *naturally occurring* sentences parsed by the link grammar, and to seek a direction mapping so that *these* parses would not violate our constraints after directions were imposed. If no such mapping exists, then we are willing to orient a few edge tokens in the wrong direction to ensure that the parses are still well-formed—but we minimize the number of such violations. In this way, the empirical distribution of sentences guides our assignment of directions. We are releasing the resulting multi-headed directed corpus via our personal websites.

4 Data Sets

We used the English sentences from the CoNLL 2007 Shared Task [21]—a subset of the Penn Treebank for which *single*-headed reference parses are available. We also used a prefix of the Russian News Commentary data from the ACL 2013 Shared Task of Machine Translation,[1] which is unparsed.

We generated link parses using the AbiWord/CMU link grammar parser version 5.0.8 [24]. The parser's coverage is less than ideal: we obtained connected parses for only 10,960 (of 18,577) English sentences and only 4,913 (of 18,577) Russian sentences, discarding the other sentences.[2] These two languages have the

[1] `http://www.statmt.org/wmt13/training-monolingual-nc-v8.tgz`

[2] When the link parser fails, it outputs a disconnected graph representing its best effort parse within a time limit. We removed these sentences for fear that the parses would be unreliable.

most mature lexicons at present, although lexicons for 7 other languages are available.

On English, the link grammar parses have 8% more edges overall, indicating that their directed versions will have a few multi-headed constructions per sentence. They do differ in style from the single-headed CoNLL parses of the same English sentences. Only 52% of the links match CoNLL arcs, and only 57% of the CoNLL arcs match links.

5 Integer Linear Programming Model

For each undirected labeled edge ij in the link corpus, where i, j denote tokens in the same sentence with $i < j$, we introduce nonnegative integer variables x_{ij} and x_{ji} with a constraint $x_{ij} + x_{ji} = 1$. We interpret $x_{ij} = 1$ or $x_{ji} = 1$ to mean that the link has direction $i \rightarrow j$ or $i \leftarrow j$, respectively.[3]

For each non-0 token v, we ensure that it has at least one parent by constraining[4]

$$\sum_u x_{uv} \geq 1 \tag{1}$$

where u ranges only over tokens such that the relevant variable exists. To prevent cycles,[5] for each token v we introduce a depth variable d_v in the range $[0, n_v]$ (not constrained to be integer), where n_v is the length of the sentence containing v. We require a child's depth to be at least 1 greater than each of its parents' depths—constraints that can be satisfied iff the sentence has no directed cycles:

$$(\forall u)\ d_v + (1 + n_v) \cdot (1 - x_{uv}) \geq 1 + d_u \tag{2}$$

The second summand ensures that (2) is trivially satisfied (hence has no effect) when u is *not* the parent of v.

Finally, we encourage all links with the same label to have the same direction. For each label L, we introduce binary variables r_L and ℓ_L, which say whether a link of type L is "allowed" to point right or left, respectively. For each undirected edge ij of label L, with $i < j$, we write

$$x_{ij} \leq r_L + s_{ij} \qquad\qquad x_{ji} \leq \ell_L + s_{ij} \tag{3}$$

where $s_{ij} \geq 0$ is a slack variable that allows an edge token to point in a disallowed direction if needed to ensure (1)–(2).

[3] In practice we halve the number of variables by replacing x_{ji} with $1 - x_{ij}$ for $j > i$, but that obscures the exposition.

[4] To denote two linked tokens, we use variables i, j when i is to the left of j, or variables u, v when u is the parent of v.

[5] This also ensures REACHABLE, given (1).

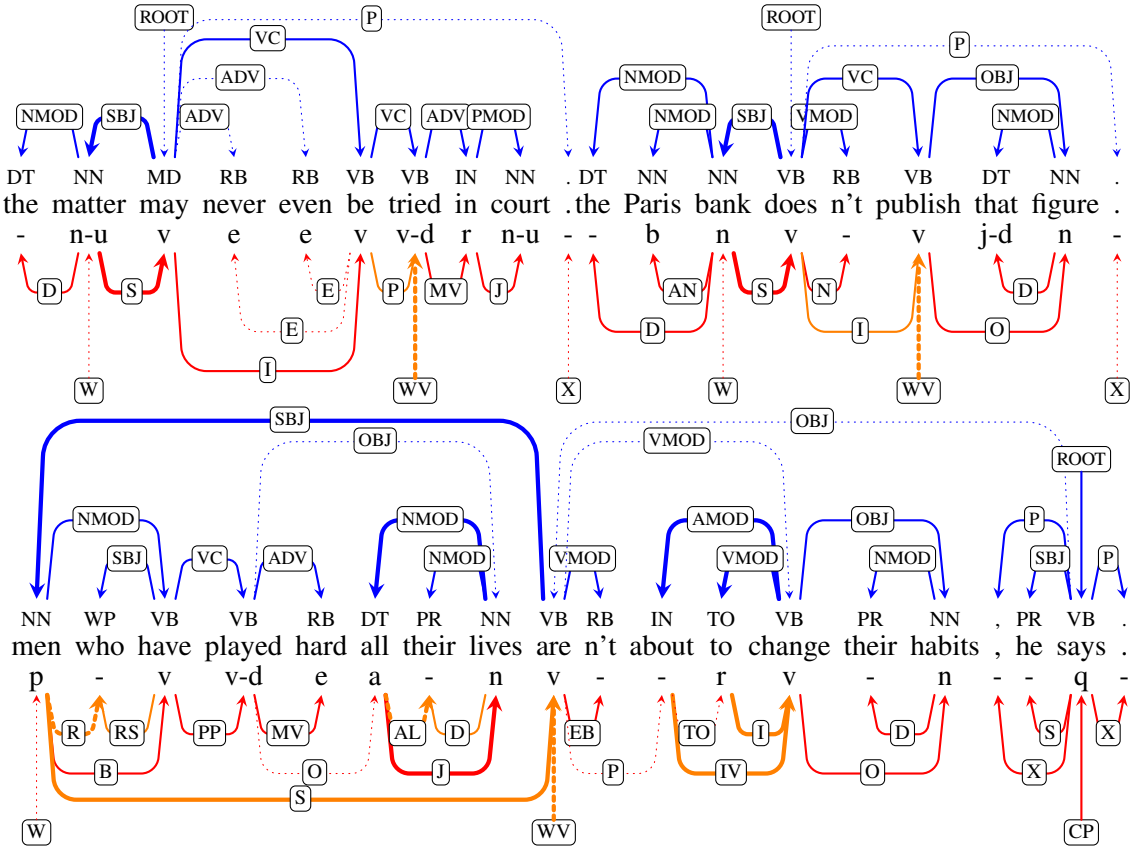


Figure 1: The blue upper edges show CoNLL gold dependency parses; the red lower edges show our oriented version of the link grammar parses. Edges are shown as dotted lines if they appear only in one parse. Edges are highlighted in orange if the child has multiple parents. Edges that appear in both parses are solid lines, drawn thicker if the directions do not match. Vertical edges have parent 0. For 100 example parses, see Appendix B of the supplementary material.

Our objective tries to minimize the number of allowed directions (by link type—cf. [26]) and the total slack (by link token):

$$\min \left(\sum_L r_L + \ell_L \right) \cdot \frac{N_L}{4} + \sum_{ij} s_{ij} \tag{4}$$

where N_L is the number of link tokens with label L. Objective (4) is willing to tolerate up to 1/4 of those link tokens' using a disallowed direction before it prefers to allow *both* directions. One could experiment with adjusting the constant 1/4; we selected that value simply because it seemed like a reasonable threshold *a priori*.

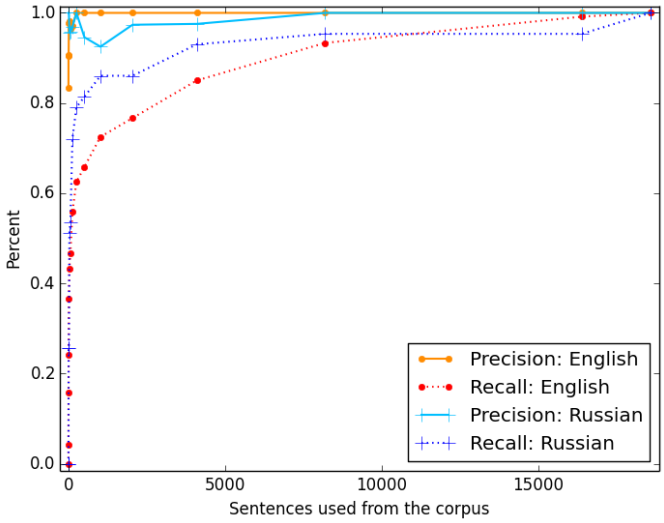


Figure 2: Rapid convergence to the direction mapping obtained on the largest dataset. The direction mappings obtained on small datasets have high *precision* relative to the one obtained on the largest dataset. Their *recall* grows as more link types are seen and directionalized.

6 Experiments and Results

We solved our ILP problem using the SCIP Optimization Suite [1], encoding it using the ZIMPL language [11]. Our largest run took 1.5 hours. On English, only 7 of 113 link *types* allowed both directions, and only $\sum_{ij} s_{ij} = 4043$ of 195000 link *tokens* required a disallowed direction via slack. 72.09% of the English sentences but (alas) only 0.04% of the Russian ones had at least one multi-headed word. See Table 1 and Appendix A for detailed results.

6.1 Stability of Results

We worried that the direction mapping might be unstable and sensitive to the input corpus. Happily, Figure 2 shows otherwise (for both English and Russian). Using even a small prefix of the data got very high-precision results, in the sense that nearly all r_L or ℓ_L variables that were 1 in this lightly trained mapping were also 1 in our largest run. The only disadvantage to using small data is low recall relative to the large run—many of the labels L are not observed yet and so we do not yet allow either direction ($r_L = \ell_L = 0$).

We used only coarse link tags as our labels, keeping only the capital letters of a tag (and merging all `ID` tags). This is because other characters in a tag indicate fine-grained features such as plurality that generally do not affect link direction. However, when we tried using fine tags as our labels instead, we found that all refinements of the same coarse tag would almost always spontaneously agree on their preferred direction. This indicates that there is indeed a "natural" direction

Label	Rightward	Multiheaded	CoNLL Match	CoNLL Dir Match	CoNLL Label
A	0% (0/8501)	0% (0/8501)	84% (7148/8501)	98% (7002/7148)	NMOD 98% (7000/7148)
AA	0% (0/4)	0% (0/4)	-	-	-
AF	84% (16/19)	37% (7/19)	32% (6/19)	0% (0/6)	VMOD 83% (5/6)
AJ	50% (131/262)	0% (0/262)	86% (225/262)	99% (223/225)	COORD 97% (218/225)
AL	100% (71/71)	99% (70/71)	-	-	-
AM	0% (0/45)	0% (0/45)	51% (23/45)	65% (15/23)	AMOD 65% (15/23)
AN	0% (0/9401)	0% (0/9401)	83% (7825/9401)	98% (7639/7825)	NMOD 96% (7523/7825)
AZ	100% (2/2)	0% (0/2)	100% (2/2)	100% (2/2)	ADV 100% (2/2)
B	100% (1514/1515)	61% (919/1515)	53% (806/1515)	84% (678/806)	NMOD 75% (603/806)
BI	100% (34/34)	0% (0/34)	38% (13/34)	100% (13/13)	VMOD 77% (10/13)
BW	100% (1/1)	100% (1/1)	100% (1/1)	0% (0/1)	OBJ 100% (1/1)
C	100% (3272/3272)	0% (0/3272)	3% (85/3272)	53% (45/85)	NMOD 27% (23/85)
CC	100% (176/176)	4% (7/176)	9% (16/176)	0% (0/16)	PRN 56% (9/16)
CO	0% (0/2478)	1% (32/2478)	5% (114/2478)	68% (78/114)	NMOD 39% (44/114)
CP	100% (283/283)	13% (36/283)	88% (249/283)	100% (249/249)	ROOT 100% (249/249)
CQ	100% (7/7)	0% (0/7)	100% (7/7)	0% (0/7)	VMOD 57% (4/7)
CV	100% (3237/3237)	100% (3237/3237)	56% (1827/3237)	28% (512/1827)	VMOD 52% (956/1827)
CX	100% (6/6)	0% (0/6)	83% (5/6)	20% (1/5)	VMOD 60% (3/5)
D	0% (56/19535)	0% (71/19535)	85% (16656/19535)	100% (16608/16656)	NMOD 100% (16629/16656)
DD	0% (0/629)	0% (3/629)	26% (165/629)	99% (164/165)	NMOD 99% (163/165)
DG	0% (0/1051)	0% (0/1051)	90% (950/1051)	100% (950/950)	NMOD 100% (948/950)
DP	0% (0/13)	0% (0/13)	23% (3/13)	100% (3/3)	SBJ 100% (3/3)
DT	0% (0/509)	0% (0/509)	100% (508/509)	99% (505/508)	NMOD 99% (505/508)
E	0% (0/1897)	0% (2/1897)	67% (1279/1897)	99% (1263/1279)	ADV 84% (1079/1279)
EA	1% (6/473)	2% (11/473)	83% (394/473)	96% (377/394)	AMOD 95% (376/394)

Table 1: Our solution, i.e., our reconstruction of the "intended" direction for each link type in the English Link Grammar. We also indicate the extent to which each of these link types (1) has a single dominant direction, (2) participates in multi-headed constructions, and (3) corresponds to CoNLL links of a predictable direction and type. For space reasons, we show only the start of this table—the full table can be found in Appendix A of the supplementary material.

for the coarse tag and that we can find it.

6.2 Linguistic Analysis

The resulting English corpus uses a syntactic annotation scheme that is somewhat different from the CoNLL annotations. Differences are tabulated in Appendix A of the supplementary material, while the actual parses are contrasted in Appendix B. Fragments of these appendices are shown in Table 1 and Figure 1.

The link grammar results in multi-headed treatments of infinitivals, compound determiners, relative clauses, and embedding. The other annotation differences are generally reasonable, e.g., letting *'s* be the head of a possessive, and different handling of punctuation and lists. One could easily modify the ILP to explicitly encourage agreement with the CoNLL link directions (for word pairs that are linked

in CoNLL). Of course, a few of the differences are due to parser attachment errors.

The main vexation is the handling of subject-verb links. Under the English link grammar, the verb (or 0) that governs a clause will link to both the clause's subject and its last (main) verb. This would permit our desired treatment of *"Jill persuaded him to skip"*, in which *"him"* has two parents. But the ILP solution generally treats subjects as *parents* of verbs (thus we get *him* → *to*). The reason for this is an inconsistency in the link grammar itself.[6] Fixing the link grammar would presumably correct the link direction. As noted in section 1, it is arguably a positive result that our method was able to detect a problem in the grammar engineering.

7 Conclusions

We have presented an automatic ILP-based method to "orient" link grammar parses in multiple languages, turning them into rooted connected DAGs. This improves their linguistic interpretability and provides new corpora for experimenting with multi-headed dependency parsers.

ILP may *in general* be a valuable technology for enriching existing annotated corpora. For example, the Penn Treebank project [14] deliberately omitted types of annotations that plausibly could be added automatically. ILP can help by leveraging unsupervised corpus-wide information [26], enforcing annotations that are simultaneously well-formed per sentence and consistent across sentences.

References

[1] Tobias Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.

[2] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

[6] Specifically, the link grammar is inconsistent about how it handles clauses with missing subjects (*"Jill wanted to skip"*). In this case, the governing verb links to the clause's first (tensed) verb in lieu of its subject. The problem is that it no longer also links to the clause's last (main) verb as it would if the subject were present. Hence our method concludes that a VP is always headed by its first verb (*to* → *skip*). That is a respectable convention on its own, but recall that unfortunately, the governing verb does *not* link to this first verb when the subject is present. As a result, the only remaining possible parent for the first verb is the subject (*him* → *to* → *skip*). This leads to the odd solution where subjects are treated as parents of verbs in general.

Note that subjects are not *always* parents, due to another inconsistency in the link grammar. In the construction *"It was impossible, they said"*, the subject (*"they"*) is inconsistently not linked to 0 but only to the verb (*"said"*) and so must be the verb's child, no other parent being available.

[3] Bernd Bohnet. Comparing advanced graph-based and transition-based dependency parsers. In *Proceedings of the First International Conference on Dependency Linguistics*, pages 282–289, 2011.

[4] Matthias Buch-Kromann. *Discontinuous Grammar. A Model of Human Parsing and Language*. Dr.ling.merc. dissertation, Copenhagen Business School, 2006.

[5] Wenliang Chen, Zhenghua Li, and Min Zhang. Dependency parsing: Past, present, and future. In *Proceedings of COLING 2014: Tutorial Abstracts*, pages 14–16, Dublin, August 2014. Tutorial slides available online.

[6] Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R. Bowman, Timothy Dozat, and Christopher D. Manning. More constructions, more genres: Extending Stanford dependencies. In *Proceedings of the Second International Conference on Dependency Linguistics*, pages 187–196, 2013.

[7] Marie-Catherine de Marneffe and Christopher D. Manning. The Stanford typed dependencies representation. In *Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, 2008.

[8] Jason Eisner. Bilexical grammars and their cubic-time parsing algorithms. In Harry Bunt and Anton Nijholt, editors, *Advances in Probabilistic and Other Parsing Technologies*, pages 29–62. Kluwer Academic Publishers, October 2000.

[9] Carlos Gómez-Rodríguez and Joakim Nivre. Divisible transition systems and multiplanar dependency parsing. *Computational Linguistics*, 39(4):799–845, 2013.

[10] Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. Who did what to whom? A contrastive study of syntacto-semantic dependencies. In *Proceedings of the 6th Linguistic Annotation Workshop*, pages 2–11, Jeju, Korea, 2012.

[11] Thorsten Koch. *Rapid Mathematical Programming*. PhD thesis, Technische Universität Berlin, 2004. ZIB-Report 04-58.

[12] Matthias T. Kromann. The Danish Dependency Treebank and the underlying linguistic theory. In Joakim Nivre and Erhard Hinrichs, editors, *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*, 2003.

[13] Marco Kuhlmann and Joakim Nivre. Mildly non-projective dependency structures. In *Proceedings of COLING-ACL*, pages 507–514, Sydney, 2006.

[14] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[15] Andre Martins, Miguel Almeida, and Noah A. Smith. Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of ACL*, pages 617–622, Sofia, Bulgaria, August 2013.

[16] André F. T. Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. Stacking dependency parsers. In *Proceedings of EMNLP*, pages 157–166, Honolulu, October 2008.

[17] Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. Probabilistic CFG with latent annotations. In *Proc. of ACL*, pages 75–82, Ann Arbor, Michigan, June 2005.

[18] Ryan McDonald and Fernando Pereira. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*, pages 81–88, 2006.

[19] Joakim Nivre. Beyond MaltParser: Advances in transition-based dependency parsing. Available at `http://stp.lingfil.uu.se/~nivre/docs/BeyondMaltParser.pdf`. Slides from invited talks.

[20] Joakim Nivre. Non-projective dependency parsing in expected linear time. In *Proceedings of ACL-IJCNLP*, pages 351–359, Singapore, August 2009.

[21] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, 2007.

[22] Joakim Nivre and Jens Nilsson. Pseudo-projective dependency parsing. In *Proceedings of ACL*, pages 99–106, Ann Arbor, June 2005.

[23] Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 63–72, Dublin, August 2014.

[24] OpenCog Project. Link grammar parser. Available from `http://www.abisource.com/projects/link-grammar/`, April 2014.

[25] Emily Pitler, Sampath Kannan, and Mitchell Marcus. Finding optimal 1-endpoint-crossing trees. *Transactions of the Association for Computational Linguistics*, 1:13–24, 2013.

[26] Sujith Ravi and Kevin Knight. Minimized models for unsupervised part-of-speech tagging. In *Proceedings of ACL-IJCNLP*, pages 504–512. Association for Computational Linguistics, 2009.

[27] Alexander Rush and Slav Petrov. Vine pruning for efficient multi-pass dependency parsing. In *Proceedings of NAACL*, pages 498–507, Montréal, Canada, June 2012.

[28] Kenji Sagae and Jun'ichi Tsujii. Shift-reduce dependency DAG parsing. In *Proceedings of COLING*, pages 753–760, Manchester, UK, August 2008.

[29] Daniel Sleator and Davy Temperley. Parsing English with a link grammar. Computer Science Technical Report CMU-CS-91-196, Carnegie Mellon University, October 1991.

[30] Davy Temperley. An introduction to the link grammar parser. Available at `http://www.link.cs.cmu.edu/link/dict/introduction.html`, March 1999.

[31] Linus Vepstas. Re: Warning: Combinatorial explosion. Message to the `link-grammar` Google Group. Available at `https://groups.google.com/forum/#!msg/link-grammar/eeJw1Ofgc9U/diqPYSwuFfoJ`, February 2014.

Different approaches to the PP-attachment problem in Polish

Katarzyna Krasnowska

Institute of Computer Science
Polish Academy of Sciences
E-mail: k.krasnowska@phd.ipipan.waw.pl

Abstract

A number of approaches, using different available resources, were applied to the PP-attachment problem in Polish. Some methods were reimplementations of supervised and partially supervised models for English described in literature, others were our modifications and extensions, mostly using a wordnet for Polish. The best accuracy achieved on the final testing dataset was 75.7%, which is not much below the accuracy of an expert's decisions obtained in a pilot study.

1 Introduction

The PP-attachment problem consists in identifying correct attachment sites for prepositional phrases occurring in natural language utterances. A high-accuracy method for solving this problem can be useful in parsing and parse disambiguation for the purposes of creating treebanks as well as in any NLP application which requires full syntactic analysis of text. The typical formulation of the problem's single instance is a quadruple $(v, n, p, n2)$, with verb v and noun n being two possible attachment sites for a phrase headed by preposition p[1] with a dependent noun $n2$. This work describes experiments on applying different approaches, using different available resources, to the PP-attachment problem in Polish.

2 Related work

A considerable amount of work has been devoted to the problem of PP-attachment, especially in English. Extensive research in what could be called a "partially supervised" framework was started by Hindle and Rooth [10] and followed by, among

[1] Polish has some prepositions which have the same surface form, but select for different grammatical cases and have different meanings. Therefore, throughout this text, unless explicitly stated otherwise, by *preposition* we will mean its surface form together with the case.

others, Resnik and Hearst [21], Resnik [20], Ratnaparkhi [17, 18] and Pantel and Lin [14]. Another line of research was devoted to supervised methods, including work by Brill and Resnik [5], Ratnaparkhi et al. [19], Collins and Brooks [7], Stetina and Nagao [23], Zavrel et al. [28] and McLauchlan [13]. Volk [24], Kawahara and Kurohashi [11], Bharathi et al. [4], Roh et al. [22] and Coppola et al. [8] proposed approaches combining supervised and partially supervised methods or data. Foth and Menzel [9], Agirre et al. [2], Roh et al. [22] and Anguiano and Candito [3] presented work on incorporating PP-attachment resolution into parsing or parse correction. The problem of phrase attachment for Polish has already been addressed by Acedański et al. [1]. It is however difficult to directly compare their results with ours since their task is different and not restricted to PP-attachment.

3 Data sources

3.1 Fully annotated data: dependency treebanks

Currently, the largest manually constructed treebank for Polish is Krzaki[2] (*Bushes*), a collection of 20 000 unlabelled dependency structures for sentences picked randomly from the manually tagged subcorpus of National Corpus of Polish[3] (NKJP, Przepiórkowski et al. [16]). 5734 lemmatised $(v,n,p,n2,a)$ quintuples, where a stands for either V (verb) or N (noun) attachment, were extracted from Krzaki. Each quintuple represents a dependency substructure where a verb v has an NP dependent (headed by n) or a PP dependent (with the head preposition's dependent n), and a PP headed by p (with a dependent noun $n2$) is governed by either v or n (so that a is known), but both are syntactically possible as the PP's attachment sites. Figure 1 shows schematically the types of dependency substructures from which the tuples were extracted.

The data was split into 3 groups: 50% training, 25% development and 25% set aside for final testing. Two split methods were used. In the first one, each tuple was assigned to a group separately, so tuples coming from one sentence could end up in different groups. Since the supervised methods explored in this work rely on lexical similarities between training data and new instances to be classified, this could cause an "information leak".[4] In the second split method, all quintuples generated from one sentence were required to end up in one group. The data sets obtained using these two strategies will be henceforth referred to as BY-TUPLE and BY-SENTENCE. The numbers of tuples in each dataset are given in Table 1.

A large, automatically obtained dependency treebank for Polish was created by Wróblewska and Przepiórkowski [26]. It consists of about 3 million trees for

[2] http://zil.ipipan.waw.pl/Krzaki

[3] http://nkjp.pl

[4] For example, if a sentence contains a sequence V NP $[p_1\ n_1]_{\mathrm{PP}_1}$ $[p_2\ n_2]_{\mathrm{PP}_2}$ with the NP (headed by n) and the two PP's being dependents of V, it generates, among others, two very similar quintuples (v,n,p_2,n_2,V) and (v,n_1,p_2,n_2,V).

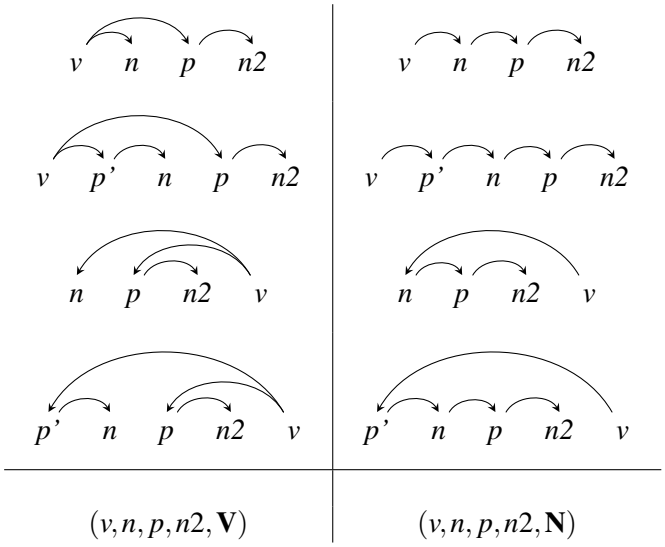


Figure 1: Dependency substructures and corresponding extracted quintuples.

	KRZAKI-TRAIN	KRZAKI-DEV	KRZAKI-TEST
BY-TUPLE	2867 (50%)	1434 (25%)	1433 (25%)
BY-SENTENCE	2810 (49%)	1504 (26%)	1420 (25%)

Table 1: Size of KRZAKI datasets (BY-TUPLE and BY-SENTENCE data splits).

sentences from a parallel, Polish-English corpus. The structure for each Polish sentence was projected from a dependency parse for its English counterpart. It is our belief that automatically obtained linguistic resources should be treated as less reliable than manually annotated ones, but the large size of the projected dependency treebank seems to have the potential to compensate for its possible overall correctness shortcomings. The same extraction procedure was applied to the treebank, yielding a set of 382 580 quintuples that will be referred to as PROJECTED.

3.2 Partially annotated data: NKJP

Some information useful for resolving the PP attachment problem can also be found in data that is not syntactically annotated. To explore this possibility, lemmatised triples of the form (*v*, *p*, *n2*) and (*n*, *p*, *n2*) were obtained from NKJP (restricted to book and press texts) using two heuristic queries. The *V* attachment query found snippets consisting of a VP followed by a PP.[5] The *N* attachment query found fragments consisting of an NP followed by a PP, with the additional requirement that the two are preceded by a punctuation mark and an optional conjunction

[5]In Polish, such VP is not guaranteed to be the actual attachment site of the PP, but probably is.

or complement.[6] One variant of the queries is given in Appendix A. Approximately 18 000 000 verb and 3 000 000 noun attachment examples were found. This data can be seen as training material in a partially supervised framework: examples of probable verb or noun attachments of PPs are available, but there are no examples of correct choice between two particular attachments, like in the previous dataset. Let us define here some counts obtained from this data that will be used in the paper ($*$ means any word):

- $c(x,p)$: triples $(x,p,*)$, i.e., triples where p is governed by the word x,
- $c(x)$: triples $(x,*,*)$, i.e., triples where some preposition is governed by x,
- $c(V,p)$: triples $(v,p,*)$ where v is a verb,
- $c(N,p)$: triples $(n,p,*)$ where n is a noun,
- $c(V)$: triples $(v,*,*)$ where v is a verb (all examples of verb attachment),
- $c(N)$: triples $(n,*,*)$ where n is a noun (all examples of noun attachment).

A similar data extraction procsedure can be found, e.g., in Kawahara and Kurohashi [11], who used a large raw corpus crawled from the web.

4 Experiments

4.1 Baselines

As a baseline, three simplistic models were tested:

Always verb. Always assign a verb attachment.

Partially supervised. Using the NKJP data, estimate the probability of a verb or noun being a governor of a given preposition p by $\frac{c(V,p)}{c(V)}$ and $\frac{c(N,p)}{c(N)}$ respectively; choose verb assignment iff the former is $\geq$ than the latter.

Supervised. This is the "Most likely for each preposition" baseline proposed by Collins and Brooks [7]: choose the most frequent attachment seen for p in training data (KRZAKI-TRAIN from the respective split).

Accuracies achieved by the baselines are presented in Table 2. The results show that naïve models perform poorly: the overall accuracy is not much above 50%, although visibly better for the supervised baseline. The partially supervised baseline performs slightly worse than the "always verb" heuristic, although both are useless as PP-attachment models for Polish. For some comparison with baselines for the same problem in English, Hindle and Rooth [10] report an accuracy of 67% achieved by always choosing noun attachment on a manually constructed dataset. Collins and Brooks [7] provide two baseline results for another dataset. They report two strategies: "Always noun attachment"[7] and "Most likely for each preposition" (as described above) to achieve accuracies of 59.0% and 72.2% respectively. The results for Polish are considerably lower, which is quite possibly

[6] The punctuation mark was introduced to heuristically reduce the snippets where some preceding phrase is the actual attachment site.

[7] Note that both cited English datasets are biased differently than our: towards noun attachment.

due to the problem being more difficult, but one has to bear in mind that this is a comparison across different languages and datasets (the impact of datasets variability on comparability of results for PP-attachment is discussed by Volk [25]).

	BY-TUPLE	BY-SENTENCE
Always verb	55.4%	57.0%
Partially supervised	54.9%	56.3%
Supervised	61.7%	63.4%

Table 2: Baseline accuracy on KRZAKI-DEV.

4.2 Partially supervised approach

Given a quadruple $(v,n,p,n2)$, the information from triples extracted from NKJP was used to determine the PP's attachment site. Following the estimation proposed by Hindle and Rooth [10] and also used by Ratnaparkhi [18], the probabilities of verb and noun attachment were calculated respectively as $P(p|v) = \frac{c(v,p)+\frac{c(V,p)}{c(V)}}{c(v)+1}$ and $P(p|n) = \frac{c(n,p)+\frac{c(N,p)}{c(N)}}{c(n)+1}$. Verb attachment was chosen if $\log P(p|v) - \log P(p|n)$ was above or equal certain threshold (experimentally set to -0.1). A minimal required number of occurrences of $(v,p,*)$ and $(n,p,*)$ triples was experimentally set to 50. If an attachment site n or v occurred with p less than 50 times, the corresponding triples were discarded from training data.

In a second experiment, each preposition p was replaced with a pair (p,c) where c is the semantic category of $n2$ from PLWORDNET[8] (Piasecki et al. [15]). Semantic categories, such as *place*, *animal* or *event*, are a coarse-grained classification of PLWORDNET's lexical units (i.e., words with particular meanings assigned). A uniform distribution of meanings was assumed in case of ambiguous words: each triple $(x,p,n2)$ was treated as k triples with a $\frac{1}{k}$ weight (where k is the number of possible meanings of $n2$), one for each meaning's category. When a quadruple $(v,n,p,n2)$ was to be classified, attachment was chosen for each of $n2$'s possible meanings separately, and the more frequent decision was taken as the final attachment.

Accuracies of both methods are presented in Table 3. The first experiment shows a substantial improvement over the supervised baseline presented before: the accuracies on both DEV datasets were raised by over 10 percentage points, reducing the error by 30.3% and 27.6% respectively. Incorporating semantic categories into the model brings about further, although less spectacular, improvement: 3.6 and 2.0 percentage points respectively (13.5% and 7.5% error reduction). These figures show that lexical information (even obtained in a heuristic way) is crucial for solving the PP attachment. For comparison, Hindle and Rooth [10] report an

[8] http://plwordnet.pwr.wroc.pl/wordnet

accuracy of about 80% for English achieved by a very similar method (without semantic categories) on their data.

	BY-TUPLE	BY-SENTENCE
Preposition	73.3%	73.5%
Preposition+category	76.9%	75.5%

Table 3: Partially supervised method's accuracy on KRZAKI-DEV.

4.3 Supervised approach: backed-off model

In an experiment with a supervised approach, the backed-off model (Collins and Brooks [7]) was used. This method of PP attachment disambiguation relies on examples from the training data which are similar to the instance to be classified. Given a quadruple $(v,n,p,n2)$ for which the PP attachment should be chosen, the procedure is as follows:

- Check whether any quintuples of the form $(v,n,p,n2,a)$ appeared in the training data. If so, the attachment is chosen to be the more frequent one among those training quintuples (V in case of a tie).
- If no matching data was found, search for quintuples matching on only 3 among v, n, p and $n2$, but require matching p. Therefore, all quintuples of the form $(v',n,p,n2,a)$, $(v,n',p,n2,a)$ and $(v,n,p,n2',a)$ are taken into account. As above, the more frequent attachment is chosen.
- If no matching data was found, back-off to quintuples matching on 2 coordinates including p: $(v',n',p,n2,a)$, $(v',n,p,n2',a)$ and $(v,n',p,n2',a)$.
- If no matching data was found, back-off to quintuples matching only on p.
- If all the above failed, choose V attachment as default.

Together with the decision about attachment, the level at which it was taken is returned (4 if all four v, n, p and $n2$ could be taken into account at once; 3 if it was backed-off to matching only three; ...; 0 if the default V attachment was chosen). Results obtained in experiments with different testing/training data setups are listed in Table 4.

The model's accuracy generally deteriorates with the back-off level, which is intuitive and in compliance with the results reported by Collins and Brooks [7]. In both cases, the results for levels 4 and 3 are similar and relatively high. Unfortunately, the quadruples classified at those levels constitute 20% and 8% of the respective test data. The coverage does not approach 100% until level 1. In the case of the model trained on the PROJECTED dataset, the testing data coverage on levels 4 and 3 is higher (36%/38% for BY-TUPLE/BY-SENTENCE). Moreover, coverage exceeds 95% already at level 2. Nevertheless, the performance is worse, which leads to a lower total accuracy. These results show that, despite being much smaller, the manually annotated data provide a more reliable source of training

train	test	4	3	2	1	0	total
KRZAKI-TRAIN BY-TUPLE	KRZAKI-DEV BY-TUPLE	94.7%	95.4%	72.8%	59.7%	50.0%	**73.2%**
instances classified		19	260	717	434	4	
coverage		1.3%	19.5%	69.5%	99.7%	100%	
KRZAKI-TRAIN BY-SENTENCE	KRZAKI-DEV BY-SENTENCE	86.7%	83.5%	72.2%	66.3%	76.2%	**71.3%**
instances classified		15	103	893	472	21	
coverage		1.0%	7.8%	67.2%	98.6%	100%	
PROJECTED	KRZAKI-DEV BY-TUPLE	82.5%	66.4%	65.3%	63.2%	100.0%	**66.3%**
instances classified		57	458	880	38	1	
coverage		4.0%	35.9%	97.3%	99.9%	100%	
PROJECTED	KRZAKI-DEV BY-SENTENCE	72.4%	70.9%	66.4%	61.5%	100.0%	**68.0%**
instances classified		58	506	887	52	1	
coverage		3.9%	37.5%	96.5%	99.9%	100%	

Table 4: Accuracy of the backed-off model tested on KRZAKI-DEV datasets. Besides total results, accuracies achieved at each back-off level are given. The numbers of classified quadruples as well as coverage at each level are also given. A level's coverage is the percentage of quadruples classified at this or higher levels.

examples. It also suggests that the backed-off model can be rather sensitive to more noise in training data. What is more, the model yields worse results than the partially supervised method. For comparison, Collins and Brooks [7] report an accuracy of 84.1% for English.

In the case of models trained on KRZAKI-TRAIN datasets, a quite large discrepancy is visible between results for BY-TUPLE and BY-SENTENCE splits, both in coverage and performance on different levels. In the case of BY-TUPLE datasets, the accuracies are generally higher. The increase of coverage on level 3 is also more visible. This is possibly due to the BY-TUPLE split making it easier for the backed-off model to achieve better results (as discussed above), which was a motivation for testing different data split strategies in the first place. Assuming that the adopted split strategy influences the results, it should be expected that with other training data, unrelated to KRZAKI and independent of the particular split, this tendency would disappear. As a matter of fact, when the respective KRZAKI-TRAIN dataset is replaced with PROJECTED as training data (see the lower part of Table 4), the results for particular levels are no longer so different between BY-TUPLE and BY-SENTENCE datasets. Following those observations, the results of further experiments will be given only for BY-SENTENCE split.

4.4 Lexical generalisations of the backed-off model

Since the backed-off model performs at its best for quadruples for which there are very similar examples in the training data, it seems worthwhile to try increasing its coverage at higher levels without much accuracy loss. One can therefore think of generalising the information contained in training data in order to find matches for more classified instances. One idea is to relax the requirement for matching tuples by taking into account synonymy relations between words.

It seems that the verb lemma is a kind of information which is very specific and should not be lost. This is because verbs tend to have strong selective preferences on prepositional phrases and even ones very close in meaning may differ in that respect, especially when the PP can fill an argument position. However, it is our intuition that replacing a noun with its synonym can help find better evidence in training data. A modification to the backed-off model as described above was therefore introduced. For a quadruple $(v,n,p,n2)$, all the combinations $(v,n',p,n2')$ were generated where n' and $n2'$ share a PLWORDNET synset with any meaning of n and $n2$ respectively. Classification was performed for each "synonymous" tuple separately. The final decision was the most frequent one on the highest possible level (V in case of a tie). The results obtained for this procedure are listed in Table 5. A slight increase in coverage at levels 3 and 4 is visible, but no significant improvement was observed. The overall accuracy is lower than that of the "standard" backed-off model (71.3%).

train	test	4	3	2	1	0	total
KRZAKI-TRAIN BY-SENTENCE	KRZAKI-DEV BY-SENTENCE	87.5%	80.7%	71.3%	65.9%	76.2%	**70.8%**
instances classified		16	119	931	417	21	
coverage		1.1%	9.0%	70.9%	98.6%	100%	

Table 5: Accuracy of the backed-off model augmented with synonyms.

Another experiment was performed using lists of distributional semantic similarity[9] created at Wrocław University of Technology using the SuperMatrix tool (Broda and Piasecki [6]). For each word accounted for, a list of 20 most similar words extracted from a large collection of texts is provided. The words in the lists are not sense-disambiguated. The experiment with similarity lists was analogous to the synset-based one, except that the synonyms retrieved from synsets were replaced with contents of similarity lists. The results are listed in Table 6. The accuracy results are better that the previous ones, an improvement in terms of tuple coverage is also visible.

The synset-based approach described above only took into account synonymy understood as being in the same synset: two words are either considered synonymous or completely unrelated. It seems that such approach has two major limita-

[9] http://nlp.pwr.wroc.pl/en/tools-and-resources/msr-list

train	test	4	3	2	1	0	total
KRZAKI-TRAIN BY-SENTENCE	KRZAKI-DEV BY-SENTENCE	78.9%	80.7%	70.9%	68.1%	76.2%	**72.3%**
instances classified		19	259	995	210	21	
coverage		1.3%	18.5%	84.6%	98.6%	100.0%	

Table 6: Accuracy of the backed-off model augmented with similar words.

tions. First, PLWORDNET's synsets tend to be small: in version 2.2 used in the experiment described in this section, the average number of lexical units per synset is 1.37 (76% of synsets containing one lexical unit) for nouns and 1.49 (73% of synsets containing one lexical unit) for verbs. Therefore, each word has very few (if any) synonyms. Second, a much better use could be made of the wordnet's structure by calculating a wordnet-based word similarity measures and therefore "quantifying" the similarity between words.

In the next experiment, we use the following formula given by Wu and Palmer [27] for measuring similarity between two concepts (synsets) c_1 and c_2:

$$dist(c_1,c_2) = \frac{2 \cdot depth(C)}{d(c_1,C) + d(c_2,C) + 2 \cdot depth(C)}$$

where C is the lowest common hypernym of c_1 and c_2; $depth(c)$ is the depth of concept c, i.e., length of the path connecting it to a root concept; and $d(c,c')$ is the length of the path connecting concepts c and c'. The distance between c_1 and c_2 can be calculated using the above formula as $1 - dist(c_1,c_2)$. Note that the resulting value will always be between 0 and 1. Following the approach adopted in many works concerning wordnet-based similarity/distance measures, the distance between two words is the minimum distance between their possible meanings.

Some transformations (most notably extending the hypernymy relation using some other relations) were performed on PLWORDNET to better suit it to computing Wu-Palmer distance; due to space limitations, we omit the details of those transformations. The original backed-off model was then modified by redefining the notion of matching tuples. Instead of requiring lemmata of words forming two tuples to be identical, the tuples are considered matching if the Wu-Palmer distances between words at respective positions do not exceed a given threshold. Two variants of this method were tested. In the first one, the distance between two words was calculated as in the formula above – by finding the minimum over all their meanings. In the second one, Plukb, a WSD tool created at Wrocław University of Technology (Kędzia et al. [12]), was used to restrict the senses to a single one where possible. The tool is still under development, but the experiment was nevertheless performed to see how the use of WSD would affect the results. Table 7 presents results obtained using different thresholds, with and without WSD. The highest accuracy of 72.5%, was achieved using a 0.05 threshold with WSD. It is very slightly better than the similarity lists approach, and still below the accuracy of the partially supervised method. The failure to outperform it may be due to

the massive amounts of data collected from NKJP outweighing the benefit from syntactically annotated data and supervised learning.

variant	4	3	2	1	0	total
thr=0.05	85.0%	78.6%	72.5%	65.8%	76.2%	
instances classified at each level	20	145	1002	316	21	**71.9%**
coverage	1.3%	11.0%	77.6%	98.6%	100.0%	
thr=0.1	85.7%	76.7%	71.8%	67.7%	76.2%	
instances classified at each level	21	202	1034	226	21	**72.1%**
coverage	1.4%	14.8%	83.6%	98.6%	100.0%	
thr=0.2	78.0%	71.6%	72.4%	66.2%	76.2%	
instances classified at each level	50	563	793	77	21	**72.0%**
coverage	3.3%	40.8%	93.5%	98.6%	100.0%	
thr=0.05, +WSD	84.2%	80.5%	73.0%	67.5%	76.2%	
instances classified at each level	19	133	980	351	21	**72.5%**
coverage	1.3%	10.1%	75.3%	98.6%	100.0%	
thr=0.1, +WSD	84.2%	79.5%	72.0%	66.9%	76.2%	
instances classified at each level	19	151	1038	275	21	**72.0%**
coverage	1.3%	11.3%	80.3%	98.6%	100.0%	
thr=0.2, +WSD	85.2%	74.3%	71.3%	64.9%	76.2%	
instances classified at each level	27	327	995	134	21	**71.7%**
coverage	1.8%	23.5%	89.7%	98.6%	100.0%	

Table 7: Accuracy of different variants of the the backed-off model with wordnet distance on KRZAKI-DEV (BY-SENTENCE split).

McLauchlan [13] also experimented with extending the backed-off method: by smoothing the backed-off estimates using different thesauruses for English, he improved the accuracy from 84.3% to 85.1%.

4.5 Results on final testing data

The experiments described above allowed to establish the variants of the partially supervised and supervised (backed-off) methods performing best of the KRZAKI-DEV dataset from BY-SENTENCE split. Among the two tested partially supervised models, the one with PLWORDNET semantic categories turned out to achieve the higher accuracy of 75.5%. The best-performing backed-off variant was the wordnet-distance based one with WSD, with a result of 72.5% for its best settings. In order to complete the experiments, the two models were tested on KRZAKI-TEST dataset from BY-SENTENCE split. The partially supervised model with semantic categories and the supervised model achieved accuracies of 75.7% and 69.6% respectively (see Table 8 for detailed results of the backed-off model). The results confirm the superiority of the partially supervised model. They also show a certain stability of its performance when tested on KRZAKI-DEV and KRZAKI-TEST. The backed-off model performs surprisingly poorly, which is even more disappointing

given that, unlike the NKJP-trained, partially supervised one, it was trained on data originating from the same resource. It would be much more interesting to compare the models' accuracies across testing datasets extracted from different sources, but we currently have no possibility to obtain reliable data of this kind.

variant	4	3	2	1	0	total
thr=0.05, +WSD	100.0%	85.8%	69.3%	64.7%	38.5%	
instances classified at each level	8	134	911	354	13	**69.6%**
coverage	0.6%	10.0%	74.2%	99.1%	100.0%	

Table 8: Accuracy of the selected backed-off model variant on KRZAKI-TEST (BY-SENTENCE split).

4.6 Comparison with human performance

To provide a kind of upper bound for assessing the tested methods' performance, a linguist manually disambiguated 200 cases taken from Krzaki in a pilot study. The human annotator was given only the information available to the models: two possible governors and the PP truncated to the preposition and noun. Agreement between the attachments chosen by the annotator and extracted from Krzaki was 79% (83% if discarding 10 cases deemed too ambiguous by the linguist). The measures should be treated as a very rough approximation given the small size of data, but they suggest that the described task is difficult. On the positive side, the presented methods are not very far below this human performance. Perhaps a hybrid method, combining multiple models, data and features, could achieve results comparable to a human annotator when given a limited context.

5 Conclusions and further work

Various methods of solving the PP-attachment problem were tested for Polish. Some of them were reimplementations of techniques described in literature, some were our own extensions and ideas. The obtained results are quite good, but still below the "human" upper bound. The amount of work devoted to this problem for English shows that there is much room for further exploration. The models tested in this work mostly rely on the limited context of the $(v,n,p,n2)$ quadruple (only the senses from the WSD tool were obtained using whole corresponding sentences) and all achieve similar results. It seems that at least two general areas are worth investigation. One is using a wider context for PP disambiguation, the other is moving from the "isolated" $(v,n,p,n2)$ case to efficient PP attachment disambiguation in full sentence parsing.

Acknowledgements

The paper is co-founded by the European Union from resources of the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications".

References

[1] Szymon Acedański, Adam Slaski, and Adam Przepiorkowski. Machine Learning of Syntactic Attachment from Morphosyntactic and Semantic Co-occurrence Statistics. In *Proceedings of the ACL 2012 Joint Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*, pages 42–47, Jeju, Republic of Korea, 2012. Association for Computational Linguistics.

[2] Eneko Agirre, Timothy Baldwin, and David Martinez. Improving Parsing and PP Attachment Performance with Sense Information. In *Proceedings of ACL-08: HLT*, pages 317–325. Association for Computational Linguistics, 2008. URL http://aclweb.org/anthology/P08-1037.

[3] Enrique Henestroza Anguiano and Marie Candito. Parse Correction with Specialized Models for Difficult Attachment Types. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1222–1233, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL http://dl.acm.org/citation.cfm?id=2145432.2145561.

[4] Akshar Bharathi, U. Rohini, P. Vishnu, S.M. Bendre, and Rajeev Sangal. A Hybrid Approach to Single and Multiple PP Attachment Using Wordnet. In *Second International Joint Conference on Natural Language Processing: Full Papers*, 2005. URL http://aclweb.org/anthology/I05-1019.

[5] Eric Brill and Philip Resnik. A Rule-based Approach to Prepositional Phrase Attachment Disambiguation. In *Proceedings of the 15th Conference on Computational Linguistics - Volume 2*, COLING '94, pages 1198–1204, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/991250.991346.

[6] Bartosz Broda and Maciej Piasecki. Parallel, Massive Processing in SuperMatrix – a General Tool for Distributional Semantic Analysis of Corpora. *International Journal of Data Mining, Modelling and Management*, 5(1), 2013.

[7] Michael Collins and James Brooks. Prepositional Phrase Attachment through a Backed-Off Model. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 27–38, 1995.

[8] Gregory F. Coppola, Alexandra Birch, Tejaswini Deoskar, and Mark Steedman. Simple Semi-supervised Learning for Prepositional Phrase Attachment. In *Proceedings of the 12th International Conference on Parsing Technologies*, IWPT '11, pages 129–139, Stroudsburg, PA, USA, 2011. Asso-

ciation for Computational Linguistics. ISBN 978-1-932432-04-6. URL http://dl.acm.org/citation.cfm?id=2206329.2206345.

[9] Kilian A. Foth and Wolfgang Menzel. The Benefit of Stochastic PP Attachment to a Rule-based Parser. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 223–230, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1273073.1273102.

[10] Donald Hindle and Mats Rooth. Structural Ambiguity and Lexical Relations. *Computational Linguistics*, 19(1):103–120, March 1993. ISSN 0891-2017. URL http://dl.acm.org/citation.cfm?id=972450.972456.

[11] Daisuke Kawahara and Sadao Kurohashi. PP-attachment Disambiguation Boosted by a Gigantic Volume of Unambiguous Examples. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, IJCNLP'05, pages 188–198, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-29172-5, 978-3-540-29172-5. URL http://dx.doi.org/10.1007/11562214_17.

[12] Paweł Kędzia, Maciej Piasecki, Jan Kocoń, and Agnieszka Indyka-Piasecka. Distributionally Extended Network-based Word Sense Disambiguation in Semantic Clustering of Polish Texts. *{IERI} Procedia*, 10(0):38 – 44, 2014. ISSN 2212-6678. URL http://dx.doi.org/10.1016/j.ieri.2014.09.073. International Conference on Future Information Engineering (FIE 2014).

[13] Mark McLauchlan. *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, chapter Thesauruses for Prepositional Phrase Attachment. 2004. URL http://aclweb.org/anthology/W04-2410.

[14] Patrick Pantel and Dekang Lin. An Unsupervised Approach to Prepositional Phrase Attachment Using Contextually Similar Words. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 101–108, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1075218.1075232.

[15] Maciej Piasecki, Stanisław Szpakowicz, and Bartosz Broda. *A Wordnet from the Ground Up*. Oficyna Wydawnicza Politechniki Wroclawskiej, Wrocław, 2009.

[16] Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors. *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw, 2012.

[17] Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, 1998. AAI9840230.

[18] Adwait Ratnaparkhi. Statistical Models for Unsupervised Prepositional Phrase Attachment. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 2*, COLING '98, pages 1079–1085,

Stroudsburg, PA, USA, 1998. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/980432.980746.

[19] Adwait Ratnaparkhi, Jeff Reynar, and Salim Roukos. A Maximum Entropy Model for Prepositional Phrase Attachment. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 250–255, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. ISBN 1-55860-357-3. URL http://dx.doi.org/10.3115/1075812.1075868.

[20] Philip Resnik. Semantic Classes and Syntactic Ambiguity. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 278–283, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics. ISBN 1-55860-324-7. URL http://dx.doi.org/10.3115/1075671.1075733.

[21] Philip Resnik and Marti A. Hearst. Structural Ambiguity and Conceptual Relations. In *Proceedings of the Workshop on Very Large Corpora*. Ohio State University, 1993.

[22] Yoon-Hyung Roh, Ki-Young Lee, and Young-Gil Kim. Improving PP Attachment Disambiguation in a Rule-based Parser. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, 2011. URL http://aclweb.org/anthology/Y11-1060.

[23] Jiri Stetina and Makoto Nagao. Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 66–80, 1997.

[24] Martin Volk. Combining Unsupervised and Supervised Methods for PP Attachment Disambiguation. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. URL http://dx.doi.org/10.3115/1072228.1072232.

[25] Martin Volk. How Bad is the Problem of PP-attachment?: A comparison of Eglish, German and Swedish. In *Proceedings of the Third ACL-SIGSEM Workshop on Prepositions*, Prepositions '06, pages 81–88, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1621431.1621443.

[26] Alina Wróblewska and Adam Przepiórkowski. Projection-based Annotation of a Polish Dependency Treebank. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pages 2306–2312, Reykjavík, Iceland, 2014. ELRA. ISBN 978-2-9517408-8-4. URL http://www.lrec-conf.org/proceedings/lrec2014/index.html.

[27] Zhibiao Wu and Martha Palmer. Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Asso-

ciation for Computational Linguistics. doi: 10.3115/981732.981751. URL http://dx.doi.org/10.3115/981732.981751.

[28] Jakub Zavrel, Walter Daelemans, and Jorn Veenstra. Resolving PP attachment Ambiguities with Memory-Based Learning. In *Proc. of the Workshop on Computational Language Learning (CoNLL'97), ACL*, pages 136–144, 1997.

A NKJP queries

- verb attachment:

```
[pos="qub" & base="się"]? [pos="adv"]{,2}
[pos="fin|praet|impt|imps|inf|pcon|pact|ger"]
[pos="qub" & base="by"]? [pos="aglt"]?
[pos="qub" & base="się"]? [pos="adv"]{,2}
[pos="prep" & case~$1] [pos="adv"]{,2}
[pos="adj" & case~$1]* [pos="subst" & case~$1]
meta channel="prasa|ksiazka"
```

- noun attachment:

```
[pos="interp"] [pos="conj|comp"]?
[pos="adv"]{,2} [pos="adj" & case~$2]* [pos="subst" & case~$2]
[pos="prep" & case~$1] [pos="adv"]{,2}
[pos="adj" & case~$1]* [pos="subst" & case~$1]
meta channel="prasa|ksiazka"
```

POS-Tagging Historical Corpora: The Case of Early New High German

Pavel Logačev, Katrin Goldschmidt, Ulrike Demske

Department of German Studies
University of Potsdam
E-mail: pavel.logacev@uni-potsdam.de

Abstract

A key problem in automatic annotation of historical corpora is inconsistent spelling. Because the spelling of some word forms can differ between texts, a language model trained on already annotated treebanks may fail to recognize known word forms due to differences in spelling. In the present work, we explore the feasibility of an unsupervised method for spelling-adjustment for the purpose of improved part of speech (POS) tagging. To this end, we present a method for spelling normalization based on weighted edit distances, which exploits within-text spelling variation. We then evaluate the improvement in taging accuracy resulting from between-texts spelling normalization in two tagging experiments on several Early New High German (ENHG) texts.

1 Introduction

A key problem in automatic annotation of historical corpora is inconsistent spelling [1, 7, 8, 4]. The spelling of words varies across texts, and sometimes even within a text. For example, the preposition *über* (*'over'*) can be spelled as *über* or as *vber*. The word *frauen* (*'women'*) may also occur in the form *fraun* or as *frauwn*. A large part of the spelling inconsistency can be attributed to the lack of clear orthographic conventions at the time, as well as to the fact that some texts were incrementally written over a period of time, thus increasing the chance of inconsistent spelling even by a single author. This ubiquitous variation in spelling may complicate (semi-) automatic annotation of historical texts, because some word forms may appear to be unknown to a language model trained on already annotated treebanks, because their spelling differs between texts.

In the present work, we explore the feasibility of an unsupervised method for spelling-adjustment for the purpose of improved part of speech (POS) tagging and semi-automatic syntactic annotation. To this end, we conducted a series of POS tagging experiments on four syntactically annotated Early New High German

(*ENHG* henceforth) texts from the "Referenzkorpus Frühneuhochdeutsch",[1] and compared the accuracy of POS tagging models based on original texts and texts with normalized spelling.

2 Methods for recognition of spelling variation

Several approaches were developed in the recent years to recognize a spelling variant as an instance of a standard spelling. One is to use dictionaries if a language period exhibits normative spelling rules. In the case of ENHG two big digital sources are available: the "Deutsches Rechtswörterburch"[2] and "Deutsches Wörterbuch" by Jacob und Wilhelm Grimm.[3] Pilz [8] searched for 100 word forms from three ENHG texts in both dictionaries and found that over 66% of the tested spelling variants were not present in the dictionaries. Thus, searching for various spellings in dictionaries does not appear to be promising approach by itself.

Another approach is the application of manually specified rules transforming one string into another. This is quite time-consuming and may not be reliable, because the created rules may not generalize to unobserved text. Therefore, some approaches combine manually and automatically generated rules. Human post-editors then decide whether two suggestions are related spelling variants. Such approaches can reach fairly high accuracy rates as Ernst-Gerlach and Fuhr show [4]. Bollmann et al. [1] developed a normalization tool for ENHG spelling variants. They combine manually developed rewrite guidelines, word lists, Levenshtein's edit distance, modern lexicon lookup and human evaluation to optimize rewrite rules and to improve normalization suggestions. Normalized word forms then can be compared with modern New High German (*NHG* henceforth) word forms in order to connect ENHG and NHG lemmatization.

Another approach is to use text corpora instead of dictionaries or rules, particularly if the usage of dictionaries or rules is not possible or is not promising enough. Pilz [8] uses ENHG texts and combines edit distances with machine learning techniques to produce a metric (*FlexMetrik*) estimating whether two word forms are spelling variants of one another.

Jurish [7] tested several approaches for recognizing spelling variants on NHG texts from the *Deutsches Textarchiv*[4] published between 1780 and 1880. In one of the approaches, Jurish uses Hidden Markov Models to identify the most likely lexical candidate with respect to the sentential context. This procedure allows token-wise disambiguation instead of single type-wise identification and reaches a precision of over 99%, because even difficult word forms which belong to different lexical categories can be recognized (such as, for example, the ENHG word form *'im'*, which can function as preposition *'in'*, or as the pronoun *'him'*). As Jurish

[1] http://www.ruhr-uni-bochum.de/wegera/ref/

[2] http://www.rzuser.uni-heidelberg.de/ cd2/drw/

[3] http://dwb.uni-trier.de

[4] http://www.deutschestextarchiv.de

Table 1: Overview of text statistics.

	Text	Year	Tokens	Types	Types/Tokens
1	BdN	1350-1375	21,104	3,463	0.16
2	Denkwürdigkeiten	1451	16,614	2,200	0.13
3	Pfaffe	1471	19,379	3,299	0.17
4	BvK	1465	25,213	3,360	0.13

points out, the performance of Hidden Markov Models on sparse data and on texts with rather heterogeneous spelling, such as ENHG texts, remains to be tested.

3 Corpora

In order to evaluate the effect of spelling normalization across texts on POS-tagging, we used four syntactically annotated Early New High German texts from the *"Referenzkorpus Frühneuhochdeutsch"*. The texts were written between 1350 and 1650, and belong to different language fields, i.e., to different epochs, regions and text genres.[5] The transliterated texts were annotated with the semi-automatic annotation tool @nnotate [3]. We used a part of speech tag set for historical texts that allows detailed lexical categorization considering historical word order [5].[6] Furthermore, the annotated texts contain syntactic and, in some cases, morphological information.

The four texts we used were "Das Buch der Natur" (*'Book of nature'*), "Denkwürdigkeiten" (*'Reminiscences'*), "Des pfaffen geschicht vnd histori vom Kalenberg" (*'The history of the cleric of Kalenberg'*) and "Das Buch aller verbotenen Künste" (*'The book of all forbidden arts'*). All texts were dated from the second half of the 14th until the second half of the 15th century. Table 1 provides an overview over the text statistics.

The first text in table 1 is the Bavarian text "Das Buch der Natur" (*'Book of nature'*, *'BvK'* henceforth). It was written in the second half of the 14th century by Konrad Megenberg in Regensburg and is thus the earliest text in the corpus.

The second text, "Denkwürdigkeiten" (*'Reminiscences'*) is a travel diary written around 1451 by Helene Kottanerin, who was a handmaiden to Queen Elizabeth of Hungary and her daughter Elizabeth of Austria. In the diary, she documented a journey with the pregnant princess Elizabeth. The Upper German text exhibits various spelling variations, including instances of highly frequent words. A possible reason for that is that the text was written incrementally over the course of a long period of time and that the author was not a professional writer.

[5]The corpus development is covered by grant DE 677/7 of the Deutsche Forschungsgemeinschaft (DFG).

[6]For example, our tag set differs from STTS [9] in having additional tags for adjectives to cover post-nominal modification and nominalized usage. Furthermore, our tag set marks discontinuous usage of prepositional adverbs, which is quite frequent in ENHG.

The third text, "Des pfaffen geschicht vnd histori vom Kalenberg" (*'The history of the cleric of Kalenberg'*; *'Pfaffe'* henceforth) is a satirical story in verses written by Philipp Frankfurter during the second half of the 15th century. The first edition of the West Middle German text was published in Augsburg in 1471 and has less apparent spelling variation than *'Denkwürdigkeiten'*.

The fourth text, named "Das Buch aller verbotenen Künste. Des Aberglaubens und der Zauberei" (*'Book of all forbidden arts, superstition and sorcery'*, *'BvK'* henceforth), was written by Johannes Hartlieb in the 1450s and concerns the forbidden magical arts like geomancy, pyromancy or aeromancy. Our edition of the Swabian text was published in Augsburg in 1465.

4 Examples of spelling variation in historical texts

A typical example of spelling alternations between texts involve allographs such as *zz* and *ss*, or *ch* and *ck*. Such alternations are exemplified in (1a) and (1b).

(1) a. Wan ez geschiht oft daz darin velt von **ezzen** oder von **trinchen**
Because it happens often that therein lacks of eat or of drink
'Because it often happens that food and drinking is lacking therein' *(BdN_301)*

b. Dieselb kranckhait mag man mit kainem **essen** oder **trincken** [..] erfüllen
The-same illness might one with no eating or drink [..] fulfill
'One cannot heal that illness with either food or drinking' *(BvK_795)*

Examples of intratextual frequent spelling variants in *'BdN'* (text 1) involve alternations of *c* and *ch* in words such as *nihts, nichts, nichtz* (nothing) or *chlain, clain* (short).

In *'Denkwürdigkeiten'*, while the definite article *die* occurs 338 times, it occurs another 57 times in its alternative version *dy*. Furthermore, the adjective *edel* (*noble*) is also spelled as *edl* (62 and 20 occurrences, respectively). Another spelling variation we found with the subordinating conjunction *that*: It occurs 192 times as *daz* (2a) and only three times as *das* (2b). The spelling *das* is mainly used for pronouns and articles, while *daz* marks the usage as conjunction. The other texts, in contrast, exhibit consistent spellings of *that* either as *daz* (*'BdN'*) or as *das* (*'Pfaffe'*, *'BvK'*).

(2) a. [die herren] waren fro **daz**-*KOUS*[7] sich ir gnad gewilligt het
[the gentlemen] were glad that-*KOUS* herself her Grace acquiesced has
den von polan ze nemen.
the one of Poland to take.
'[the gentlemen] were glad that her Grace had acquiesced to take the one of Poland [in marriage].' *(Denkwürdigkeiten_71)*

b. jch solt mich wol gehaben / **das**-*KOUS* wer aus komen [..]
I should me well being / that-*KOUS* we out come [..]
'I should be confident / that we come out [..]' *(Denkwürdigkeiten_141)*

In *'Pfaffe'*, we found four spelling variants of the personal pronoun *sie* (singular, *she*) and *sie* (plural, *they*): *si* (plural: 1 occurrence), *sy* (singular: 19 times; plural: 25 times), *sie* (singular: 44 times; plural: 60 times), *sye* (singular: 17 times; plural: 56 times). Furthermore, we found nine different spelling variations on one of the dominant themes of the book: *tiüffel* (*devil*), five of which occur 15 times or more.

There is also some variation in the spelling of technical terms. Names of magical arts are typically formed with the morpheme *-mancia*, as in *geomancia* (*geomancy*). However, they also occur in the alternative spelling *mantia*. Furthermore, the lexemes *aeromantia* and *nigramantia* occur in alternative spellings with the prefixes *are-*, *aro-* and *nygra-*, respectively.

5 POS tagging accuracy in historical corpora

In order to evaluate whether spelling normalization can increase POS tagging accuracy, we used the TnT tagger [2] to tag each text using, in turn, each of the three remaining texts or the remaining texts as whole as the underlying language model. We did so in order to establish a base line to compare the tagging accuracy before and after spelling normalization.

Table 2 shows the results of the tagging experiments for each target text averaged over all language models used for tagging that text. It shows the percentage of unknown tokens, the percentage of correctly assigned tags, as well as the tagging accuracies for known as well as unknown types separately.

Table 2: Overview of the results of the POS-tagging experiments using unmodified corpora. Average percentages with standard deviations in parentheses. Text ids correspond to ids in table 1.

target text	training corpus	% correct	% unknown	% correct (known)	% correct (unknown)
1	2, 3, 4, 2&3&4	69 (5)	35 (6)	87 (1)	37 (6)
2	1, 3, 4, 1&3&4	76 (4)	27 (5)	87 (1)	46 (7)
3	1, 2, 4, 1&2&4	70 (5)	31 (6)	85 (2)	36 (6)
4	1, 2, 3, 1&2&3	75 (5)	28 (6)	86 (2)	48 (7)

Across texts, the overall tagging accuracy was 73%, with a significantly higher accuracy for known tokens (86%) and than for unknown tokens (42%). The relatively low accuracy for previously unseen types for text 1 (approximately 36%, as

[7] The part of speech tag 'KOUS' stands for subordinating conjunctions.

opposed to approximately 47% in the other two texts) is possibly caused by differences in syntactic patterns due to the fact that text 1 (*'BdN'*) is significantly older than the remaining texts. The lower accuracy for text 3 (*'Pfaffe'*), on the other hand, is likely caused by the verse structure of that text, which is not present in the remaining texts. Thus, because the unusual syntactic constructions in parts of the text were rarely present in the training data, the TnT tagger's second-order Markov model cannot reliably use an unknown word's POS-context to assign it to a POS category. This difference suggests that the text genre affects the tagging accuracy.

Not unexpectedly, the highest proportions of unknown word forms were found among open class words (adjectives, nouns, and verbs). Among these classes, the average proportion of unknown tokens was 51% (SD=19%). Adverbs had a somewhat lower unknown token rate of 26% (SD=10%). The lowest rates of unknowns were found among closed class words (articles, conjunctions, prepositions, pronouns) with an average of 11% ($SD = 6$). Closed class words were associated with an average accuracy of 76%. The latter is not unexpected, because these lexical categories contain only few words. However, the tagger does not perform well on unknown closed-class words: tagging correctness for unknown conjunctions was under 20%, with an average of 10% and only 2% on the text "Denkwürdigkeiten".

6 Word similarity metric

In order to broaden the coverage of the language model, we normalized word spelling across texts using a metric of distance between word forms. We assumed that two word forms are spelling variants of a single lexeme and should be conflated when they occurred in the same environments (as assessed by a bigram model), and when they were sufficiently similar to each other in terms of spelling. We assessed word similarity by means of a weighted edit distance metric [6], which is an indicator of the number of character changes required to transform one word into the other.

For example, the unweighted edit distance between *cat* and *mat* is 1, because a *'c'* needs to be substituted by an *'m'* or vice versa to transform one word into the other. The edit distance between *rice* and *ice* is also 1, because an *'r'* needs to be deleted or inserted. However, if we decided to penalize insertions and deletions twice as much as substitutions, the weighted edit distance between *rice* and *ice* would be 2. In a similar manner, different substitutions can be penalized differently. For example, the substitution 'b' $\rightarrow$ 'z' appears, *a priori*, less likely to produce an alternative spelling of the same lexeme than the substitution 'b' $\rightarrow$ 'p'. We used a weighted edit distance to capture this fact.

More formally, we assumed that the word forms w_i and w_j were instances of the same lexeme when the inequality in equation 1 was satisfied. In this equation, $P_{bigram}(texts|w_i \leftrightarrow w_j)$ is the bigram probability of the texts *assuming* the conflation of w_i and w_j (i.e., assuming that w_i are instances of the same lexeme in different spelling), while $P_{bigram}(texts|w_i \not\leftrightarrow w_j)$ is the probability of the texts *not assuming*

a conflation of w_i and w_j (i.e, that w_j and w_i are distinct words). Furthermore, $P(w_i \leftrightarrow w_j)$ is the prior probability of w_i corresponding to w_j, as assessed by means of a weighted edit distance.

In other words, we conflated two word forms when the increase in the probability of the two texts according to a bigram model of both texts outweighed the prior probability of the change in word form according to equations 2 and 3.

$$P_{bigram}(texts|w_i \leftrightarrow w_j) \cdot P(w_i \leftrightarrow w_j) > P_{bigram}(texts|w_i \not\leftrightarrow w_j) \tag{1}$$

The prior probability of word w_i corresponding to w_j was computed according to equation 2 as the product of the prior probabilities of all edit operations required to transform w_i to w_j.

$$P(w_i \leftrightarrow w_j) = \prod_{\forall e \in edits(w_i \rightarrow w_j)} P(e) \tag{2}$$

In setting prior probabilities of single edit operations we exploited the within-text spelling variation in the training corpus. First, we identified all pairs of word forms in the training corpus which can be transferred into each other with exactly one edit operation (such as *cat* and *mat*). Next, for every pair of such words w_k and w_m, we computed the probability p_{km} that w_k and w_m each occur with the POS tags assigned to them in the training corpus, assuming that they are spelling variants of the same word, and therefore theoretically occur with the same POS tags equally often. This means that if w_k and w_m mostly occurred with the same POS-tags, p_{km} was large, while it was relatively small if the two words were rarely assigned the same POS tags.

We set the prior probability for every edit operation e to the geometric mean of all p_{km} according to equation 3, where w_k can be transferred into w_m with the edit operation e and where n is the number of such word form pairs. $P(e)$ was set to 1 for all edits operations for which no minimal pairs existed.[8] Set up in this way, the prior probability captures the negative evidence against some edit operations which is present in the corpus, while it does not penalize edit operations for which no minimal pair was observed.

$$P(e) = \sqrt[n]{\prod_{\forall w_k, w_m:\ w_k \xrightarrow{e} w_m} p_{km}} \tag{3}$$

7 Tagging normalized texts

In order to test our metric, we conflated all sufficiently similar words in all pairs of training and test sets, if those words did not differ by more than three characters and if at least one of them occurred in the test set but not in the training set. The latter condition was used because conflating two words which are not in the test set

[8] The prior probabilities do not sum to 1, because they concern independent events.

cannot possibly result in higher POS tagging accuracy. In the next step, we used the TnT tagger [2] to tag the normalized texts.

The differences in accuracy between the results in table 2 and the tagging experiments are presented in table 3. It shows that texts 1 and 2 appear to substantially benefit from spelling normalization, with an average improvement of 2.2 percentage points in accuracy, while the tagging accuracy on texts 3 and 4 seems largely unchanged. This difference appears largely due to the fact that only 0.5% of all tokens in texts 3 and 4 were affected by the conflation (the percentage of unknown tokens decreased by approximately 0.6 percentage points), *vis-à-vis* the approximately 2% affected in texts 1 and 2.

Table 3: Differences between the results of the POS tagging experiments using normalized and original spelling.

test corpus	training corpora	% correct	% unknown	% correct (known)	% correct (unknown)
1	2, 3, 4, 2&3&4	1.4 (1.7)	-2.0 (2.1)	-0.3 (0.3)	1.7 (2.3)
2	1, 3, 4, 1&3&4	1.6 (0.8)	-1.9 (1.1)	0.1 (0.1)	2.9 (1.5)
3	1, 2, 4, 1&2&4	0.2 (0.2)	-0.4 (0.2)	-0.1 (0.1)	0.1 (0.1)
4	1, 2, 3, 1&2&3	-0.2 (0.3)	-0.7 (0.4)	-0.2 (0.4)	-1.2 (1.7)

Across the board, the largest contribution to the reduction in the number of unknown tokens for text 2 was due to the conflation of four pairs of highly frequent word forms, with the major share being due to the conflation of the word forms *das* (198 occurrences in text 2) and *daz* (228 occurrences in text 2). While *das* appeared with various POS tags (e.g., demonstrative pronoun, definite article, and conjunction) and was present in all texts, *daz* appeared almost exclusively as a conjunction in text 2. The situation in text 1 was similar: the conflation of *daz* and *das* appeared to play a significant role, but also that of other closed-class tokens, such as *darumb* and *darvmb* (*therefore*), or *uon* and *von* (*from*). Better coverage of highly frequent function types also explains the increase in accuracy for unknown word forms — because more highly frequent word forms are known, and are therefore recognized more reliably, the tagger can use that context to guess the POS of more unknown tokens in their surroundings.

In summary, our results suggest that spelling normalization may improve the quality of (semi-)automatic annotation, especially if the target text exhibits a high degree of spelling variability, or if there are systematic differences in spelling between texts.

8 Discussion

In the present paper, we investigated the idea that normalizing spelling across historical texts may improve the coverage of language models in automatic annotation. We presented an unsupervised method for identifying alternative spellings of the

same lexeme in historical texts. We tested its performance on four Early New High German texts.

Although the performance was mixed, our results suggest that especially texts with large spelling variation such as our texts 1 and 2 (*'BdN'* and *'Reminiscences'*) may benefit from such normalization. This is especially so, when the spelling of highly frequent functional lexemes varies between texts, because the identification of alternative spelling variants can have a positive effect not only on the correct assignment of POS categories to known, but also to previously unseen, unknown tokens.

A clear limitation of our approach is that it considers edit operations only at the level of a single character and is therefore unable to capture higher-level generalizations, for example, the alternation of character sequences 'ie' by 'y' (such as in *die*, a definite article, and its alternative spelling 'dy'), as the substitution of one character ('i' or 'e') by 'y', and the deletion of the other character. A consequence of that is the inability to represent the fact that the deletion of 'i' or 'e' may produce an entirely different word, and not a spelling variant, in other contexts.

We suspect that more sophisticated methods, which are able to capture alternations between character sequences, such as variations on [1, 7], may lead to significant performance increases over our metric.

References

[1] M. Bollmann, S. Dipper, J. Krasselt, and F. Petran. Manual and semi-automatic normalization of historical spelling. In *Proceedings of KONVENS 2012*, pages 342–350, 2012.

[2] T. Brants. TnT: a statistical part-of-speech tagger. In *Proceedings of the 6th conference on Applied NLP*, pages 224–231, 2000.

[3] T. Brants and O. Plaehn. Interactive corpus annotation. In *Procedings of LREC-2000*, pages 224–231. Association for Computational Linguistics, 2000.

[4] Andrea Ernst-Gerlach and Norbert Fuhr. Semiautomatische Konstruktion von Trainingsdaten fur historische Dokumente. *Lehren–Wissen–Adaptivität (LWA 2010)*, 2010.

[5] S. Dipper et al. HiTS: ein Tagset für historische Sprachstufen des Deutschen. *JLCL*, 28(1):85–137, 2013.

[6] D. Jurafsky and H James. Speech and language processing. 2000.

[7] B. Jurish. More than words. *JLCL*, 25(1):23–39, 2010.

[8] T. Pilz. *Nichtstandardisierte Rechtschreibung*. Logos Verlag Berlin GmbH, 2010.

[9] C. Thielen, A. Schiller, S. Teufel, and C. Stöckert. Guidelines für das Tagging deutscher Textkorpora mit STTS. *Rapport technique, Université de Stuttgart et Université de Tübingen*, 1999.

Synergistic development of grammatical resources: a valence dictionary, an LFG grammar, and an LFG structure bank for Polish

Agnieszka Patejuk and Adam Przepiórkowski

Institute of Computer Science, Polish Academy of Sciences
E-mail: {aep,adamp}@ipipan.waw.pl

Abstract

The aim of this paper is to present the simultaneous development of three interrelated linguistic resources for Polish: a valence dictionary, a formal grammar, and a syntactic corpus. This parallel development creates a strong synergistic effect: the valence dictionary is constantly verified by its use in the formal grammar, and both are verified in the process of constructing the structure bank.

1 Introduction

The aim of this paper[1] is to introduce a new linguistic resource of Polish and discuss the role it plays in the verification of the quality and completeness of two other resources. The new resource is a corpus of sentences annotated with LFG syntactic structures: the usual constituency trees and functional structures bearing information about grammatical functions of various constituents in the tree, about their morphosyntactic features, and about the predicates introduced by the heads of these constituents.

The other two resources – the valence dictionary and the LFG grammar – were originally developed relatively independently. However, once the process of converting the valence dictionary to an LFG lexicon started, many inconsistencies and gaps in the dictionary were discovered; since then, the employment of the dictionary in the grammar has been the source of additional quality and completeness control of the dictionary, with a constant flow of bug reports and data requests.

The third resource, an LFG structure bank (presented here for the first time), is empirically based on an earlier constituency treebank, but the LFG structures are constructed independently of those in that treebank: sentences are parsed with

[1]The work described here was partially financed by the CLARIN-PL project (http://clip.ipipan.waw.pl/CLARIN-PL).

an XLE parser implementing the LFG grammar and then manually disambiguated using the INESS tool. In the process of manual disambiguation, problems in the LFG grammar and in the valence dictionary are discovered and corrected, leading to new versions of both resources.

The main thesis of this paper is that the parallel development of such resources is preferable to the usual – and often unavoidable for practical reasons – procedure of, say, developing a valence dictionary on the basis of a closed treebank, as the flow of information to upstream resources is considerable and leads to massive improvements.

The paper is structured as follows: §2 presents the three resources, §3 discusses the synergy effect during their parallel development and §4 concludes the paper.

2 Resources

2.1 *Walenty*, a valence dictionary of Polish[2]

One of the two initial resources is the valence dictionary *Walenty*, presented in detail elsewhere [22, 19], so we will only illustrate it with a couple of valence schemata.

A simplified example of an entry for the verb ADMINISTROWAĆ 'administrate' is given below:

(1) `administrować: imperf: subj{np(str)} + obj{np(inst)}`

This is an imperfective verb and the schema specifies two arguments: the subject and a passivisable[3] object (*somebody administrates something*). They are both nominal phrases (NPs), and while the object bears the fixed instrumental case, the subject's case is specified as structural, as its morphological realisation depends on the category of the head assigning case (gerunds uniformly assign genitive case to their subject) and – at least on some theories [16] – on the category of the subject, namely whether it is a certain kind of numeral phrase (in which case it is accusative) or not (in which case it is nominative).

A slightly more complex schema is needed for the verb DEDYKOWAĆ 'dedicate', as used in the following example from the National Corpus of Polish:

(2) Gola dedykuję [dla rodziców] i [sympatii Iwonie].
goal.ACC dedicate for parents.GEN and girlfriend.DAT Iwona.DAT
'I dedicate this goal to my parents and my girlfriend Iwona.' (NKJP)

In the above example, the first person subject is *pro*-dropped, the pre-verbal object occurs in the accusative case, but it would occur in the genitive if the verb were

[2]This section provides information about the version of *Walenty* of 20 November 2014. However, entries from plain text export, (1) and (3), were simplified by reducing them to the following fields: lemma, aspect, valence schema. The remaining ones are not directly relevant to the discussion.

[3]The label `obj` is used to mark arguments which can become the subject under passive voice.

negated, so its case is marked as structural in the valence schema, and there is one more argument, whose grammatical function is not marked explicitly in the schema below:

(3) `dedykować: imperf:`
`subj{np(str)} + obj{np(str)} + {np(dat); prepnp(dla,gen)}`

This argument must be specified as being realisable by two kinds of phrases: a dative NP or a prepositional phrase (PP; `prepnp` above) headed by the preposition DLA 'for' and including a genitive NP. The fact that these two kinds of phrases occupy the same argument position follows from the possibility to coordinate them, as in (2) above.

These two valence schemata illustrate only a couple of a number of interesting or unique features of *Walenty*. First of all, as already pointed out above, it explicitly defines an argument position via the coordination test, so one position in one valence schema may be filled by categorially diverse constituents, as in the famous English example *Pat became a republican and quite conservative* [27, p. 142], where the noun phrase *a republican* is coordinated with the adjectival phrase *quite conservative* within an argument position of *became*. It turns out that such coordination of unlike categories is relatively common in Polish.

Second, *Walenty* – while remaining relatively theory-neutral – is informed by contemporary linguistic theories and encodes linguistic facts often ignored in other valence dictionaries, e.g., control and raising, structural case, passivisation, non-chromatic arguments [15], etc.

Third, the dictionary contains a very rich phraseological component [18] which makes it possible to precisely describe lexicalised arguments and idiomatic constructions, e.g., the fact that one may welcome (Pol.: *witać*) somebody "with open arms" (Pol.: *z otwartymi ramionami*) or "with arms wide open" (Pol.: *z szeroko otwartymi ramionami*), but not just "with arms" (Pol.: **z ramionami*) or "with unusually wide open arms" (Pol.: **z niezwykle szeroko otwartymi ramionami*).

Fourth, while the process of adding deep semantic information to *Walenty* has begun only recently, some arguments are already defined semantically, e.g., the manner arguments as occurring with the verbs ZACHOWYWAĆ SIĘ 'behave (in some way)' or TRAKTOWAĆ 'treat (somebody in some way)' – such arguments may be realised via adverbial phrases of a certain kind, but also via appropriate prepositional or sentential phrases.

Finally, the dictionary, continually developed within various projects, is already the biggest and most detailed valence dictionary of Polish: as of 20 November 2014, it contains 54 328 schemata for 11 913 lemmata. Moreover, by the end of 2015 *Walenty* is planned to cover 15 000 lemmata, including at least 3000 non-verbal ones. Snapshots of the dictionary are released on an open source licence roughly half-yearly; see `http://zil.ipipan.waw.pl/Walenty`.

2.2 *POLFIE*, an LFG grammar of Polish

POLFIE [13] is an LFG [1, 6] grammar of Polish implemented in XLE [4]. As described in more detail in [13], rules used in *POLFIE* were written on the basis of two previous formal grammars of Polish: the DCG [30] grammar *GFJP2* used by the parser *Świgra* [31] and the HPSG [14] grammar described in [20]. While the former provided the basis for constituent structure rules, the latter was used as the basis of f-descriptions. The basis provided by these previous grammars was the starting point for extensions which were introduced in areas such as coordination and agreement (see, e.g., various publications by the current authors in proceedings of LFG conferences 2012–2014; `http://cslipublications.stanford.edu/LFG`).

Also the lexicon of *POLFIE* is heavily based on other resources. Morphosyntactic information is drawn from a state-of-the-art morphological analyser of Polish, *Morfeusz* [32, 33], from the *National Corpus of Polish* (*NKJP*; [17]) and from *Składnica*, a treebank of parses produced by the *Świgra* parser [29, 34]. While some (very limited) syntactic information is added manually to selected lexical entries – e.g., those of *wh*-words (such as *kto* 'who' or *dlaczego* 'why'), *n*-words (such as *nikt* 'nobody', *nigdy* 'never' or *żaden* 'none'), etc. – valence information is automatically converted from *Walenty*. For example, the schema for ADMINISTROWAĆ 'administrate' in (1) is converted to an XLE entry whose simplified version is given below:

(4)
```
(^ PRED)='administrować<(^ SUBJ)(^ OBJ)>'
(^ SUBJ PRED:
        {(<- CASE)=c nom | (<- CASE)=c acc (<- ACM)=c rec})
(^ OBJ CASE)=c inst
(^ TNS-ASP ASP)=c imperf
```

The first line of this lexical entry specifies the so-called semantic form of the verb, i.e., that the predicate is `administrować` and that it takes two arguments: SUBJ and OBJ. The last line says that it is an imperfective verb, and the penultimate line – that the case of its object is instrumental. The subject specification, split into two lines for typographic reasons, is more complex: it says that the subject is either in the nominative case or it is a governing numeral (see the slightly cryptic `(<- ACM)=c rec`) in the accusative.[4,5]

The XLE system with *POLFIE* parses around a third of sentences in the 1-million-word manually annotated balanced subcorpus [7] of *NKJP*. This may sound like a poor result, but it is typical of deep parsers not propped with any fall-back pre-processing or post-processing strategies. (Such supporting strategies are currently being developed for *POLFIE*.)

[4]Such numerals are assumed to be defective and have no nominative form, so no ambiguity follows from the fact that the first disjunct is not specified as not being a governing numeral.

[5]This information is encoded via the mechanism of off-path constraints [6, p. 148] for reasons explained in detail in [12].

2.3 An LFG structure bank of Polish sentences

The structure bank of Polish sentences is the youngest of these resources, and it is presented here for the first time. It is based on the aforementioned *Składnica* treebank, but only in a weak sense: the same morphosyntactically annotated Polish sentences – originally drawn from the 1-million-word subcorpus of *NKJP* – are assigned syntactic structures here, but these structures are not based on those in *Składnica*. This way interesting cross-theoretical comparisons should be possible in the future between the DCG representations contained in *Składnica* and the LFG representations in the structure bank described in this section.

The resource currently contains almost 6 500 sentences (over 58 000 segments, in the *NKJP* sense of this term). It has been created semi-automatically. First, the sentences were parsed using the *POLFIE* grammar and the XLE system mentioned above. In effect, often multiple analyses were produced for many sentences, since any grammar of a reasonable size must be ambiguous; in case of *POLFIE*, the average number on parses is 717 and the median is 10. (This means that there are a few sentences with a very large number of parses and many with very few analyses.) After this automatic process, analyses were manually disambiguated by a group of linguists – each sentence independently by two linguists, to ensure the high quality of the resulting structure bank.[6] 4 linguists spent 4 half-time months each (i.e., 2 person-months) on the task, 1 spent 1 half-time month, and all of them spent some 2–3 half-time months on learning LFG and the disambiguation system used for this task. During annotation, the linguists were not allowed to individually communicate or to see each other's comments. On the other hand, they could communicate via a mailing list accessible to all of them and to the developers of the grammar. The process was supervised by the main grammar writer (the first author), who responded to all questions and many comments.

This high speed of annotation could be attained thanks to the use of the INESS infrastructure for building structure banks [23, 25]. Figure 1 (on the next page) presents a screenshot of the system for the sentence *Jak wygląda przepiórka* 'What does a quail look like?', lit. 'How looks quail?', before it is disambiguated. Both the c-structure and the f-structure are shown in a compact format encompassing a number of analyses (here, two) at the same time. For example, in the c-structure in the middle of the screenshot, the choice is at the level of the highest IP node: should it be rewritten to ADVP IP (the analysis marked as [a2]) or to IP XPsem (analysis [a1], with the order of nodes reversed, as the lower IP is shared between these two analyses)? The correct parse may be selected by the annotator by clicking on one of the two rules in the bottom left corner of the screenshot: IP –> XPsem IP or IP –> ADVP IP.

This choice at the level of c-structure is correlated with a choice at the level of f-structure. For example, the f-structure will contain the feature ADJUNCT only if a2 is selected. Otherwise, if a1 is chosen, it will contain the feature OBL-MOD.

[6] As in case of the manual annotation of *NKJP* [21], pairs of annotators were not constant; instead annotators were shuffled so as to avoid co-learning the same mistakes.

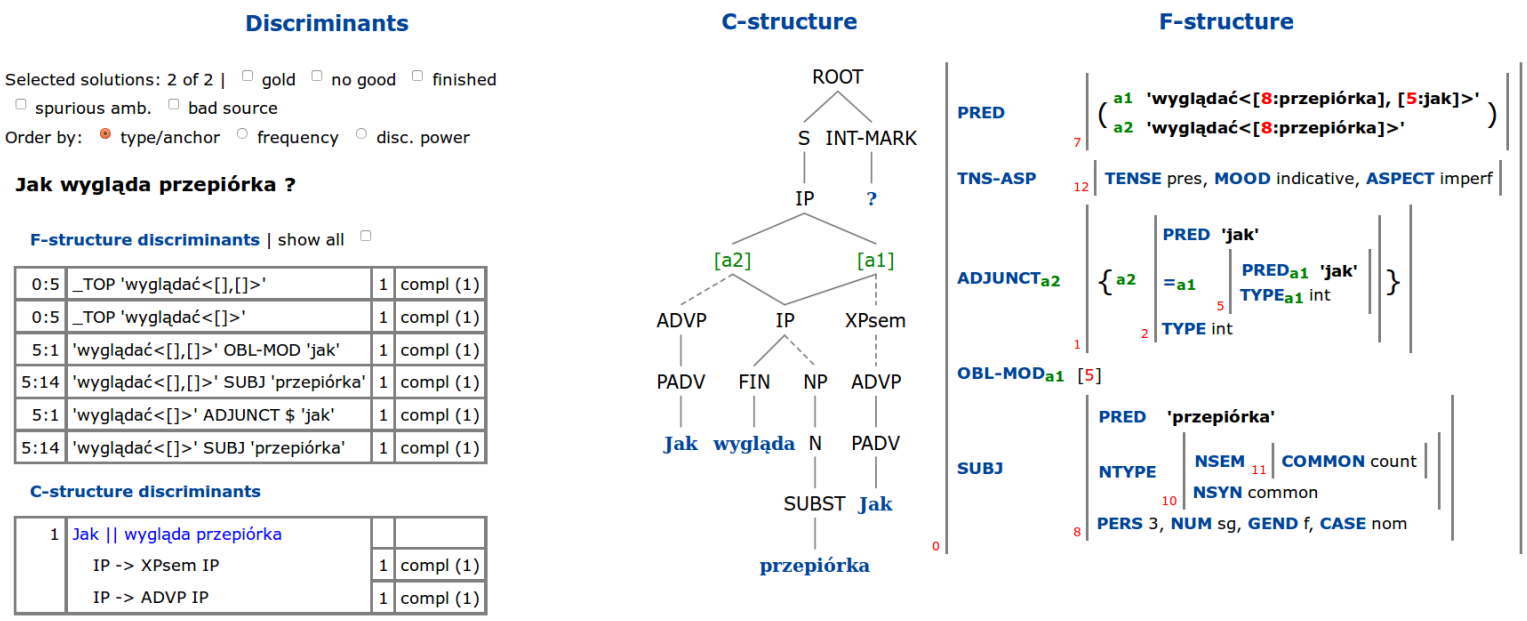


Figure 1: *Jak wygląda przepiórka* before disambiguation

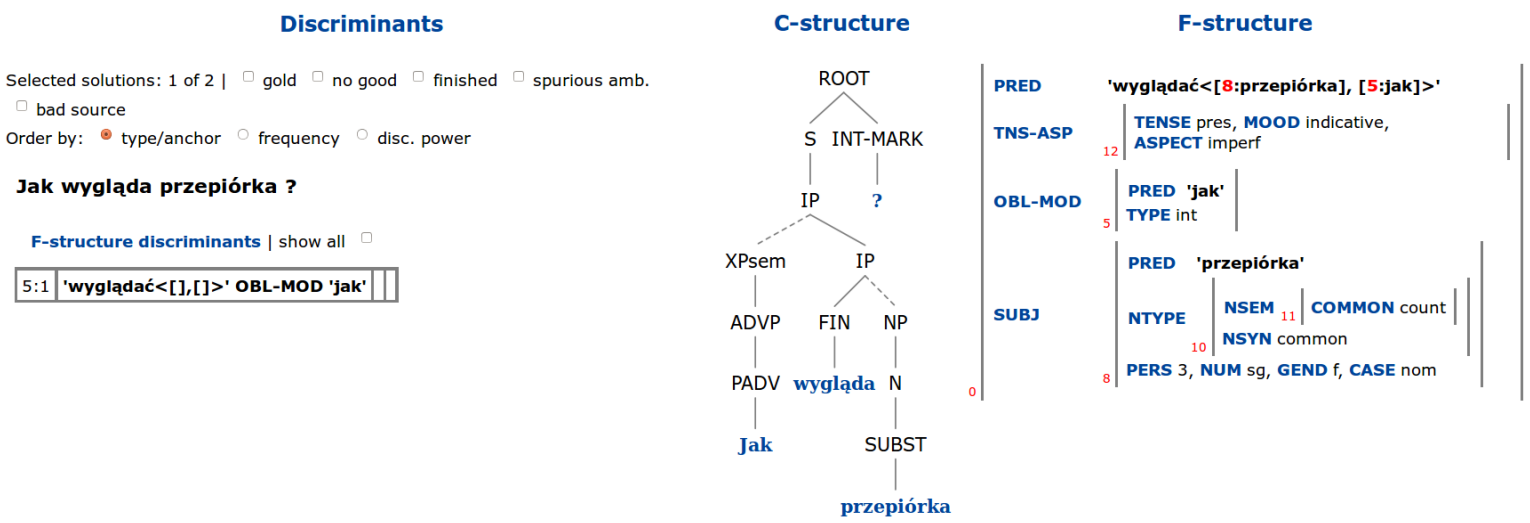


Figure 2: *Jak wygląda przepiórka* after disambiguation

So, instead of relying on c-structure discriminants in the table at the bottom left of Figure 1, annotators may rely on f-structure discriminants in the table above, and select either the third row of the table, mentioning OBL-MOD 'jak', or the fifth row, mentioning ADJUNCT $ 'jak'. In fact, the choice boils down to whether the verb WYGLĄDAĆ 'look like' is a two-argument verb (see the first row in this table) or a one-argument verb (see the second row). As the first of these options seems correct, the annotator may disambiguate this sentence by clicking on the first row or – equivalently – on the third row. The result of choosing the latter discriminant is shown in Figure 2 (on the previous page).

3 Synergy effect with parallel development

As should be clear from the above descriptions of the three Polish resources, the valence dictionary feeds the formal grammar, which is in turn used to build the structure bank. Work on each of these resources also results in the verification and significant improvements of the upstream resources.

First, *Walenty* is automatically converted to LFG constraints to be used in the grammar, and many inconsistencies in the valence dictionary can be identified already at this stage. During this process, morphosyntactic information stored in *Walenty* is compared with information provided by the morphological analyser *Morfeusz*, which makes it possible to discover such problems as wrong aspect of the predicate, wrong case required by the preposition, etc. More importantly, potentially problematic schemata are also discovered, e.g., ones containing no subject when a passivisable object is present or ones with mismatched control relations.

Second, omissions in the valence dictionary are identified when the resulting grammar is used for parsing a corpus of Polish sentences. Analysed sentences are inspected and, if the lack of correct parses results from the incompleteness of *Walenty*, new schemata are added to the dictionary.

Third, sentences parsed with XLE are fed into INESS (including sentences for which XLE returned no good parse: there were over 9 000 sentences). Those sentences which have syntactic analyses (there are over 6 500 such sentences, see §2.3) are disambiguated. The annotators are encouraged to look at f-structure discriminants rather than c-structure discriminants and, especially, at values of PRED, which contain information about the number and type of arguments of particular predicates. This way wrong valence in f-structures is discovered, which may be caused by errors in the *Walenty*-to-LFG conversion procedure, but is more often caused by problems in the valence dictionary itself. Of course, other errors in f-structures are also spotted, relating directly to the grammar. This way, the construction of the structure bank verifies both the formal grammar and the valence dictionary.

Error reports during the construction of the structure bank are facilitated by the rich system of comments offered by INESS. For this task, there are three main types of comments: issue, todo and bad_interp. The last is reserved for reports on wrong morphosyntactic annotation of some words, i.e., it is concerned with the

Składnica treebank and the *NKJP* subcorpus from which the morphosyntactically annotated sentences are taken. This way, the development of the LFG structure bank also influences resources other than the valence dictionary and the grammar. Problems with valence constitute one subtype of todo comments, other subtypes are concerned with the grammar. Finally, comments of type issue signal more subtle problems, e.g., doubts about the proper attachment place of a constituent, doubts about the choice of a grammatical function for an argument, a multi-word expression which should probably have a separate entry in the dictionary, etc. It should be noted that annotators are encouraged to leave comments to suboptimal analyses even when one of the analyses of the sentence is fully correct. Currently, there are almost 3 000 comments in the system.

The whole annotation process is divided into rounds, each involving around 1 000 sentences and lasting 2–3 weeks. After a round of annotation is completed, comments created by annotators are inspected by the grammar writer, who responds to each of them (after they have been anonymised) using the mailing list. The purpose of this review is to give feedback to annotators: explain some analyses, improve their skills by making them aware of certain linguistic issues, encourage them to contribute comments.

Subsequently, relevant comments containing confirmed issues are passed together with responses (and additional comments, if needed) to the developers of relevant resources. Developers of *Walenty* are asked to inspect relevant entries and introduce appropriate changes, if the suggestion is right. Issues related to the conversion are handled by the grammar writer. Finally, comments related to problems in the grammar are collected and passed to the grammar writer to introduce appropriate modifications to improve the treatment of relevant phenomena.

After relevant changes have been introduced in *Walenty* and the grammar, a new lexicon is created, sentences are reparsed and a new version of analyses is fed into INESS so that discriminants can be reapplied from the previous disambiguated version of the structure bank. This takes advantage of an ingenious feature of INESS, based on an idea of [3] and on earlier work on the LinGO Redwoods HPSG treebank [10, 11]: choices made for one version of the grammar remain valid for the next version of the grammar. After discriminants have been reapplied, annotators are asked to return to those sentences which did not have a complete good solution in the previous version, consult their comments and check if the relevant problem is solved in the current version.

The entire procedure described above is repeated until a good solution is obtained for all the sentences. As a result, all three resources, the valence dictionary, the formal grammar and the structure bank, are improved incrementally in parallel, as illustrated in Figure 3.

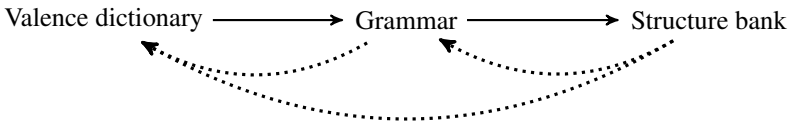


Figure 3: Flow of information to downstream resources (straight solid arrows) and feedback to upstream resources (curved dotted arrows)

4 Conclusion

Evaluation of the quality and completeness of valence dictionaries is difficult. By the concurrent development of a relatively theory-independent dictionary and a comprehensive LFG grammar taking advantage of almost all types of information in this dictionary, the quality of the dictionary is partially verified. By applying the grammar to a relatively balanced corpus of Polish, both the quality and the completeness of the dictionary – as well as the quality of the grammar – are verified.

Obviously, this is not the first attempt at the parallel development of language resources. Grammars have been developed together with treebanks, e.g., the HPSG grammar of English [10, 11] or the LFG grammar of Norwegian [26], as well as – more recently – the DCG grammar of Polish [29, 34]. Similarly, for Czech, valence dictionaries have been extracted from treebanks automatically [28] or developed manually in sync with treebank construction [8]; see also [9] for similar work on German and [24] for a discussion of various improvements of a Norwegian lexicon when constructing a Norwegian parsebank (treebank based on automatic parsing and manual disambiguation). The current setup extends such work by showing the benefits of the simultaneous developments of three different resources, including an independent valence dictionary, not based on the linguistic theory underlying the grammar and the structure bank. While this approach to the parallel development of multiple linguistic resources is often difficult – due to the scarcity of non-linguistic resources (budgetary and human) – we maintain that such a holistic approach should always be strived for.

Our future plans extend this approach even further and involve the addition of semantic information to the valence dictionary, and subsequent verification of this information via the use of semantic representations produced by the grammar based on this extended dictionary in the task of recognising textual entailment ([5]).

References

[1] Joan Bresnan. *Lexical-Functional Syntax*. Blackwell Textbooks in Linguistics. Blackwell, Malden, MA, 2001.

[2] Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios

Piperidis, editors. *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, Reykjavík, Iceland, 2014. ELRA.

[3] David Carter. The TreeBanker: A tool for supervised training of parsed corpora. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 598–603, Providence, RI, 1997.

[4] Dick Crouch, Mary Dalrymple, Ron Kaplan, Tracy King, John Maxwell, and Paula Newman. XLE documentation. `http://www2.parc.com/isl/groups/nltt/xle/doc/xle_toc.html`, 2011.

[5] Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. *Recognizing Textual Entailment: Models and Applications*. Morgan & Claypool, 2013.

[6] Mary Dalrymple. *Lexical Functional Grammar*. Academic Press, 2001.

[7] Łukasz Degórski and Adam Przepiórkowski. Ręcznie znakowany milionowy podkorpus NKJP. In Przepiórkowski et al. [17], pages 51–58.

[8] Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, Veronika Kolářová, and Petr Pajas. PDT-VALLEX: Creating a large-coverage valency lexicon for treebank annotation. In Joakim Nivre and Erhard Hinrichs, editors, *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*, Växjö, Norway, 2003.

[9] Erhard W. Hinrichs and Heike Telljohann. Constructing a valence lexicon for a treebank of German. In Frank van Eynde, Anette Frank, Koenraad De Smedt, and Gertjan van Noord, editors, *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7)*, pages 41–52, Groningen, The Netherlands, 2009.

[10] Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christoper D. Manning. LinGO Redwoods: A rich and dynamic treebank for HPSG. In Erhard Hinrichs and Kiril Simov, editors, *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT2002)*, pages 139–149, Sozopol, 2002.

[11] Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christoper D. Manning. LinGO Redwoods: A rich and dynamic treebank for HPSG. *Research on Language and Computation*, 4(2):575–596, 2004.

[12] Agnieszka Patejuk and Adam Przepiórkowski. A comprehensive analysis of constituent coordination for grammar engineering. In *Proceedings of the 24rd International Conference on Computational Linguistics (COLING 2012)*, 2012.

[13] Agnieszka Patejuk and Adam Przepiórkowski. Towards an LFG parser for Polish: An exercise in parasitic grammar development. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012*, pages 3849–3852, Istanbul, Turkey, 2012. ELRA.

[14] Carl Pollard and Ivan A. Sag. *Head-driven Phrase Structure Grammar*. Chicago University Press / CSLI Publications, Chicago, IL, 1994.

[15] Paul M. Postal. *Skepical Linguistic Essays*, chapter Chromaticity: An overlooked English grammatical category distinction, pages 138–158. Oxford University Press, 2004.

[16] Adam Przepiórkowski. *Case Assignment and the Complement-Adjunct Dichotomy: A Non-Configurational Constraint-Based Approach*. Ph.D. dissertation, Universität Tübingen, Germany, 1999.

[17] Adam Przepiórkowski, Mirosław Bańko, Rafał L. Górski, and Barbara Lewandowska-Tomaszczyk, editors. *Narodowy Korpus Języka Polskiego [Eng.: National Corpus of Polish]*. Wydawnictwo Naukowe PWN, Warsaw, 2012.

[18] Adam Przepiórkowski, Elżbieta Hajnicz, Agnieszka Patejuk, and Marcin Woliński. Extended phraseological information in a valence dictionary for NLP applications. In *Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP 2014)*, pages 83–91, Dublin, Ireland, 2014. Association for Computational Linguistics and Dublin City University.

[19] Adam Przepiórkowski, Elżbieta Hajnicz, Agnieszka Patejuk, Marcin Woliński, Filip Skwarski, and Marek Świdziński. Walenty: Towards a comprehensive valence dictionary of Polish. In Calzolari et al. [2], pages 2785–2792.

[20] Adam Przepiórkowski, Anna Kupść, Małgorzata Marciniak, and Agnieszka Mykowiecka. *Formalny opis języka polskiego: Teoria i implementacja [Eng.: Formal description of Polish: Theory and implementation]*. Akademicka Oficyna Wydawnicza EXIT, Warsaw, 2002.

[21] Adam Przepiórkowski and Grzegorz Murzynowski. Manual annotation of the National Corpus of Polish with Anotatornia. In Stanisław Goźdź-Roszkowski, editor, *Explorations across Languages and Corpora: PALC 2009*, pages 95–103, Frankfurt am Main, 2011. Peter Lang.

[22] Adam Przepiórkowski, Filip Skwarski, Elżbieta Hajnicz, Agnieszka Patejuk, Marek Świdziński, and Marcin Woliński. Modelowanie własności składniowych czasowników w nowym słowniku walencyjnym języka polskiego. *Polonica*, XXXIII:159–178, 2014.

[23] Victoria Rosén, Paul Meurer, and Koenraad De Smedt. Designing and implementing discriminants for LFG grammars. In Miriam Butt and Tracy Holloway King, editors, *The Proceedings of the LFG'07 Conference*, pages 397–417, University of Stanford, California, USA, 2007. CSLI Publications.

[24] Victoria Rosén, Petter Haugereid, Martha Thunes, Gyri S. Losnegaard, and Helge Dyvik. The interplay between lexical and syntactic resources in incremental parsebanking. In Calzolari et al. [2], pages 1617–1624.

[25] Victoria Rosén, Paul Meurer, Gyri Smørdal Losnegaard, Gunn Inger Lyse, Koenraad De Smedt, Martha Thunes, and Helge Dyvik. An integrated web-based treebank annotation system. In Iris Hendrickx, Sandra Kübler, and Kiril Simov, editors, *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (TLT 11)*, pages 157–168, Lisbon, Portugal, 2012.

[26] Victoria Rosén, Paul Meurer, and Koenraad De Smedt. Constructing a parsed corpus with a large LFG grammar. In Miriam Butt and Tracy Holloway King, editors, *The Proceedings of the LFG'05 Conference*, pages 371–387, University of Bergen, Norway, 2005. CSLI Publications.

[27] Ivan A. Sag, Gerald Gazdar, Thomas Wasow, and Steven Weisler. Coordination and how to distinguish categories. *Natural Language and Linguistic Theory*, 3:117–171, 1985.

[28] Anoop Sarkar and Daniel Zeman. Automatic extraction of subcategorization frames for Czech. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, pages 691–697, Saarbrücken, 2000.

[29] Marek Świdziński and Marcin Woliński. Towards a bank of constituent parse trees for Polish. In *Text, Speech and Dialogue: 13th International Conference, TSD 2010, Brno, Czech Republic*, Lecture Notes in Artificial Intelligence, pages 197–204, Berlin, 2010. Springer-Verlag.

[30] D. H. D. Warren and Fernando C. N. Pereira. Definite clause grammars for language analysis — a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13:231–278, 1980.

[31] Marcin Woliński. *Komputerowa weryfikacja gramatyki Świdzińskiego [Eng.: A computational verification of Świdziński's grammar]*. Ph.D. dissertation, Institute of Computer Science, Polish Academy of Sciences, Warsaw, 2004.

[32] Marcin Woliński. Morfeusz — a Practical Tool for the Morphological Analysis of Polish. In Mieczysław Kłopotek, Sławomir T. Wierzchoń, and Krzysztof Trojanowski, editors, *Intelligent information processing and web mining*, pages 503–512. Springer-Verlag, 2006.

[33] Marcin Woliński. Morfeusz reloaded. In Calzolari et al. [2], pages 1106–1111.

[34] Marcin Woliński, Katarzyna Głowińska, and Marek Świdziński. A preliminary version of Składnica—a treebank of Polish. In Zygmunt Vetulani, editor, *Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 299–303, Poznań, Poland, 2011.

The Sense Annotation of BulTreeBank

Aleksander Popov, Stanislava Kancheva, Svetlomira Manova,
Ivaylo Radev, Kiril Simov, Petya Osenova

Linguistic Modeling Department
Institute of Information and Communication Technologies, BAS
{alex.popov|stanislava|svetlomira}@bultreebank.org
{radev|kivs|petya}@bultreebank.org

Abstract

The paper focuses on the sense annotation of BulTreeBank. It discusses three levels of annotation: valency frames, lexical senses and DBPedia URIs. The lexical sense annotation is considered in more detail and in relation to the other two processes. Special attention is paid to the quality validation with respect to two aspects: inter-annotator agreement and cross-resource control.

1 Introduction

Treebanks are typically considered as syntactically annotated corpora. However, the recent tendencies have shown that at some point they actually turn into knowledge-rich resources with semantic and discourse information. The most canonical example for this is the development of Penn Treebank into proposition and discourse treebanks. BulTreeBank was developed as a syntactically annotated corpus for Bulgarian, which currently exists in several formats: in its original format (HPSG-based) and in its conversions (dependency based [1]; Penn treebank based [3]; stanfordized [10]).

In this paper we present the methodology of sense annotation in the BulTreeBank original format. Our efforts have been invested in adding value at several levels, yielding three layers of sense annotation - of lexical senses, valency information and DBPedia instances. The annotation process involved extraction of valency frames from the treebank, addition of senses to verbs, nouns, adjectives and adverbs, as well as combining valency, DBPedia and sense annotation into one merged resource. Thus, the difference with sense annotation in non-treebank corpora is the availability of valency frames in the treebanks, which are used for support of the lexical sense annotation.

The paper is structured as follows: in section 2 the related works are mentioned; in section 3 the methodology of the sense annotation is described; section 4 presents our strategies for annotation quality control; section 5 concludes the paper.

2 Related Work

There are a number of resources which are sense annotated. Most of them rely on WordNets and/or other lexical resources that provide sense differentiation, such as language-specific lexicons. Sense annotated corpora take their origins from seminal corpora, such as SemCor, and are realized as particular variants of them in other languages, such as Dutch, Basque, Bulgarian, etc.[1] Unfortunately, most of them are not freely available in their full capacity and for further third-party research.

At the same time, there are not so many treebanks available that have been annotated with senses. Here the following ones need to be mentioned, among others: for English, the sense annotated developments of Penn Treebank — PropBank ([8]) and NomBank ([5]) — as well as OntoNotes, which combines sense information from several resources; for German, the TÜBa-D/Z sense annotated treebank [4]; for Italian, the syntactic-semantic treebank [6]. In OntoNotes an ontology was used for mapping the WordNet senses. This is the Omega Ontology [9].

Our resource differs from PropBank and Prague Dependency Treebank in that it does not provide detailed semantic role labels. We expect this information to come from the ontological labels in valency frames over the grammatical roles (subject, complement, adjunct). The sense annotated BulTreeBank keeps closer to the OntoNotes strategy of combining syntactic analysis with sense annotations. We also use an ontology - DOLCE - for constraining the senses and controlling their granularity.

The novelty in our sense annotation endeavour lies, as far as we are aware, in the combination of assigned valencies, lexical senses and DBPedia URIs into a syntactic resource.

3 Sense Annotation

For the purposes of the sense annotation, three phases were envisaged: preparation, sense annotation and quality control. The first two phases were discussed in [11]. In the preparatory phase substantial efforts were invested into mapping the definitions of a Bulgarian explanatory dictionary to the intersected senses of Core and Base Concepts in Princeton WordNet. These amount to 5000. The mapping was done automatically with the help of a Bulgarian-English dictionary. Then, these mappings were manually checked and curated according to the following schema: selection of the correct sense among available ones; addition of a sense which is missing in the Bulgarian dictionary; update of a definition. We tried to provide richer definitions in comparison with the Princeton WordNet glosses, since we plan to use this information for enriching the ontology and cross-data relations.

The annotation process was organized in three layers: verb valency frames [7]; senses of verbs, nouns, adjectives and adverbs; DBPedia URIs over named

[1] http://globalwordnet.org/wordnet-annotated-corpora/

entities. First, the valency frames of the verbs were extracted from the treebank with assigned ontological labels over the participants, and then checked manually.

The sense annotation was organized as follows: the lemmatized words per part-of-speech from the treebank got all their senses from the explanatory dictionary of Bulgarian and from our WordNet. When two competing definitions came from both resources - the preference went to the one that was mapped to the WordNet. In the ambiguous cases the correct sense was selected according to the context of usage. For the purpose of evaluation some of the files were checked by two people. More information on this is given in the next section. 92 000 running words have been mapped to senses from the WordNet. Thus, about 43 % of all tokens of the treebank have been covered.

After the manual checks, the sentences with the valency frames were merged with the added word senses. In this step several issues have been addressed: the valency frames have been generalized from the initial syntax surface realizations to argument structure lists, which contain also the potential participants in the event; the various senses have been mapped to the corresponding valency frames; any remaining inconsistencies have been fixed. Thus, the treebank initially provided specific verb frames, realized in the texts, and these were later generalized with respect to participants, as well as senses.

The annotation with DBPedia URIs was performed as a separate activity, but closely related to the valency frames and sense annotation tasks. It covered 10 885 named entities — 2877 organizations, 2938 locations, 4195 people, the rest are from different categories: events, books, others. Unfortunately, the coverage of the Bulgarian DBPedia is rather small. For that reason, the the Bulgarian Wikipedia was used for adding the respective links into the data.

In the process of lexical sense annotation and DBPedia annotation, multiword expressions have been handled as well. During sense annotation all the idiomatic expressions (idioms, light verb constructions, etc.) have been specifically labeled as multiword expressions, in contrast to the previously pure syntactic approach, taken in the annotation of the treebank. Since many of these expressions have a rather narrow potential for combination with other units, the differences show in the ontological constraints. This means that instead of labeling the participants of a predicate as high level concepts, such as Person, Cognitive Fact, Social Event, Machine, etc., the labels remain very specific. Here are two examples, in which the subject has more abstract constraints, while the complement remains specific: 'Litse/grupa ot litsa ostana bez pokriv', "PERSON/GROUP OF PERSONS remains/remain without roof" (PERSON/GROUP OF PERSONS becomes/become homeless); 'Materialni sredstva/deystvie/deynost otida na vyatara', "MEANS/ACT/ACTIVITY goes to the wind" (MEANS/ACT/ACTIVITY is wasted).

During the DBPedia annotation process, the URIs were pointed to the full names of the entities, while the text box kept the specific occurrences of the names in the text. Several kinds of challenging situations were encountered: the text provides a metaphoric name for the entity, while the DBPedia link uses its real name

(for example, the politician Ahmed Dogan is referred to as Sokola (the Falcon) in many texts, but in DBPedia the link is constructed from his actual name); there is insufficient context for the selection of the correct URI; there is no matching URI in the Bulgarian DBPedia; there is no direct URI mapping to the name, available only under another URI.

In the following table some statistics is given on the number of tokens, lemmas and definitions annotated per part-of-speech. Also the ratios Token-to-Lemma and Definition-per-Lemma are presented.

	tok	lemma	def	Tok/Lemma	Def/Lemma
Adj	17741	2077	4344	08,5416	2,0915
Adv	7571	533	726	14,2045	1,3621
Noun	49658	3477	8046	14,2819	2,3141
Verb	17977	2058	5163	08,7352	2,5087
Total	92947	8145	18279		
Tok/Lemma	11,4115				
Def/Lemma	02,2442				

It can be seen that the largest POS group of annotated tokens is that of the Noun. Verbs and Adjectives are comparable to each other. Unsurprisingly, the adverbs constitute the smallest group. The same POS distribution holds for lemmas as well. As for the definitions, the POS hierarchy for their number per lemma in decreasing order is: **Verbs > Nouns > Adjectives > Adverbs**. Interestingly, the ratio Token-to-Lemma combines together Nouns and Adverbs, on the one hand, which both show lesser variety, and Adjectives and Verbs, on the other, which both exhibit greater variety. The definitions per lemma remain stable: with respect to Nouns, Adjectives and Verbs - around 2 definitions per lemma, and with respect to Adverbs - 1 definition.

4 Quality Control

This section discusses two perspectives on quality control: inter-annotator agreement and cross-resource control.

4.1 Inter-annotator Agreement

4.1.1 Naively-measured agreement vs. Case-based evaluation

An initial evaluation of inter-annotator agreement was carried out on a limited sample of sense-annotated sentences. The sample file contains 905 sentences, with one lemma of interest per sentence (the lemmas being only nouns starting with the letter A in this particular file). Out of these, 190 sentences present cases where

the two annotators have made different choices in the selection of word senses. Naively-measured inter-annotator agreement is 79%.

An examination of the entries where annotator choices differ, however, reveals that agreement is actually higher. This is due to the nature of the task and the availability, or rather the lack of, comprehensive lexical resources. Because of that, the range of selectable senses for each lemma does not always exhaust all possible meanings; and sometimes one and the same sense is expressed through two or more definitions (which usually come from different resources). Additionally, there is the issue of word senses overlapping in terms of meaning, or of word senses that are in a relation of hyponymy/hypernymy with one another. Therefore the annotators' task is not just that of making the relevant choices, but also of enriching the sets of possible definitions. Below we give an overview of these issues.

4.1.2 Word senses introduced by the annotators

There are many cases where a relevant word sense is missing from the array of available choices and the annotator therefore needs to supply additional ones (from external lexical resources, or construct them on their own). Alternatively, there could be a presupplied word sense that is suitable in a specific case, but it somehow falls short of best expressing the meaning in question, so once again the annotators are faced with the task of introducing an additional definition. Thus even if the two annotators have chosen imported word senses that convey roughly the same meaning, these constitute formally different choices. For instance, the word 'avtomat' (automaton) is tagged with three different senses among the two annotators, but these three are very close in meaning to one another; the three definitions roughly translate as: 1) an apparatus that performs certain actions on its own, 2) a device or mechanism that carries out the same task repeatedly, without any immediate human participation, 3) an apparatus with a mechanism that performs certain actions on its own, mimicking humans. In this case there is no actual disagreement between annotators, as they have both selected word senses that are very close to one another. In many cases only one annotator has added a word sense; the added definition outlines the same idea as a pressuplied word sense, but more clearly and precisely. These cases too do not constitute real disagreement, they are rather an artifact of the annotation methodology. With some entries, of course, the annotator-introduced definitions do not actually map correctly to the contextual meaning of the words in question.

Out of the 380 items for annotation that were manually evaluated (2x190, i.e. the two identical sets of sentences), 113 constitute situations where an annotator has introduced a word sense not present in the presupplied options, i.e. 30%. The number is high, but it should be noted that some of the added definitions are introduced multiple times across items, as the same lemmas are annotated in different sentences.

4.1.3 Equivalent word senses coming from different resources

A similar source of apparent disagreement is when the presupplied word sense sets include very similar definitions, most often coming from different lexical resources. For instance, consider the word 'avans', when it appears in sports contexts (it translates to 'a lead (by points/goals/etc)'). One of the annotators has selected just one of the available senses, while the other has selected two of the options, the second being very close to the first one, though more economical in its wording. Such cases will be at some point normalized and made consistent with a common ontology. They do not constitute actual clashes between the annotators' choices.

4.1.4 Word senses with different granularity

Another reason for disagreement is differences at the level of concept granularity. In one sentence the word 'avtor' (author) is tagged with the more general sense of 'creator of a product' by one annotator and with the more specific sense of 'writer' by the other annotator. In another sentence, one annotator has chosen the more general sense, while the other has selected both the more general and the more specific, apparently unsure about the right answer. Sometimes one of the annotators has selected the more specific definition, while the other has selected only the more general; in the cases where context allows only the broader sense, this constitutes genuine inter-annotator disagreement. Some of these cases usually require a broader than sentence-scope context to disambiguate the meaning. In most cases looking at the paragraph containing the sentence is enough to solve the problem. Lemmas as the above-described one present difficulties when the more specific sense is not applicable, as well. For example, the author of a report cannot in Bulgarian be referred to as a writer (the Bulgarian analogue 'pisatel' is used chiefly to refer to authors of literature). In several places this has been the source of disagreement between the two annotators. Identifying such mismatches at the level of conceptual granularity is in itself a difficult task. A cursory examination revealed 13 such items, i.e. around 7% of all the manually examined sentences (or around 1% of the total number of sentences in the file).

4.1.5 Disagreement due to lack of context

There are occasions when even paragraph-scope context fails to support the process of word sense disambiguation. Discourse and world knowledge are needed to provide the relevant information. Consider a sentence which translates in English roughly as 'The three story building with two big *workshops* on 'Luben Karavelov' 15 will be turned into a Center for culture and debate called 'The Red House'.' The Bulgarian word used for 'workshop' ('atelie') has two related but different senses: one is of a small working room where artisans of all kinds carry out their work, the other is of the working space of a painter, sculptor, photographer (i.e. an artistic person). In this case one annotator has chosen the more general sense, while the other has selected the more specific. The paragraph where the sentence comes from

does not provide information sufficient to clarify which of the word senses is more appropriate. Reference to online sources reveals that the building was home to a famous sculptor who donated it to the state. This extra-textual piece of knowledge helps carry out the disambiguation. There are several other similar cases, although they are not many compared to the overall number of entries.

4.1.6 Complete disagreement between annotators

Last come the sources of difference where one annotator has simply made the better decision. Such cases present challenges of varying levels of difficulty to the annotators. For instance, one sentence where the annotators' choices differ contains the noun 'avtoritet' (authority). One option is to annotate it with the sense of 'the quality of having authority'; an alternative sense is 'a person commanding authority'. The right choice is fairly straightforward in the context of the sentence and the mistake of one of the annotators may probably be attributed to a momentary lapse of focus.

In other cases, however, the task is more complicated. For example, there is a sentence that could be translated in two different ways: 1) 'In this way the airport will meet the highest *aviation* standards, said Noev.' and 2) 'In this way the airport will meet the highest standards of *the national aviation*, said Noev.' That is, it is left unclear whether by 'aviatsiya' (aviation) is meant aviation as an occupation and field of activity (with its attendant international standards) or some local aviation organization (in this case the corresponding Bulgarian entity). This calls for a more subtle disambiguation thought-process.

4.1.7 Summary of the evaluation task

Out of the 190 sentences that are annotated differently by the annotators, 95 have been been classified as instances of full inter-annotator disagreement. In 42 sentences there is partial conceptual overlap between the two annotators' choices, i.e. at least one annotator has selected more than one word sense and at least one word sense option has been selected in both annotated versions, but there are additional differences in the selections. 52 sentences have been annotated with different word sense definitions that are however roughly equivalent in terms of meaning (differing mostly with respect to phrasing). There is 1 sentence where the annotators have both made correct choices but one has selected the more general sense and other the more specific. Thus, out of the 190 items where annotators apparently disagree with each other, full disagreement is observed in 50% of the cases, partial disagreement in 22% of the cases, and in 27% of the cases exhibiting apparent inter-annotator disagreement the annotators have actually selected (almost) identical word senses. A less strict evaluation approach thus gives 90% inter-annotator agreement. The nature of the task makes evaluation difficult but insightful with regards to how an ontology of word senses should be structured. Unfortunately, due to the above-described noisiness of the data, calculating more sophisticated

scores of inter-annotator agreement (e.g. Cohen's kappa) is very difficult, if at all possible.

4.2 Cross-resource control

Another way of ensuring the quality of the annotation is through the combination of annotation levels. Two types of such combination were carried out: of verb valency frames and sense annotations, on the one hand, and of DBPedia URIs and valency frames, on the other.

Since the participants in the treebank-driven valence dictionary were mapped to WordNet and the DOLCE ontology, we also get control over the other parts-of-speech. For example, the perfective verb 'izbroya' (count) has three valency frames from the valence dictionary. They happen to share one meaning. The frames are: to count COGNITIVE FACTS; to count PERSONS; PERSON counts PERSONS. It can be noted that the first two frames do not include restrictions on the subject, while the third one does. These frames are generalized into a frame that considers also the subject participant. It says: PERSON counts OBJECTS.

Another issue that benefits from the cross-resource control is the usage of perfective/imperfective verbs. The merging of valency frames with senses shows the preferences (based on frequencies) of the perfective and/or imperfective verb to be used with a specific frame and/or sense.

In this way all the valency frames are: 1. mapped to their senses and 2. generalized into coarse frames, if necessary. At the moment around 1050 verb frames have been processed in this way.

The DBPedia URIs annotation is considered in relation to valency frames and senses. The labels that mark the participants in the frame come from the DOLCE ontology. The senses mapped to the WordNet inherit ontological constraints from two ontologies - DOLCE and SUMO. DBPedia annotations follow the DBPedia ontology. All these ontologies need to be synchronized for the sake of resource consistency and utility.

5 Conclusions

In this paper we present the methodology behind sense annotation in BulTreeBank. The scheme relies on the annotation of verb valency frames, sense annotation of four parts-of-speech (nouns, verbs, adjectives, adverbs) and DBPedia URIs. The advantage of such a strategy is that the cross-reference of valence-to-senses (including DBPedia links) can also be used as a controlling mechanism to ensure the quality of the resulting resource, with respect to the integration of lexical semantics with valency, real world facts and ontological mappings.

A closer examination of inter-annotator agreement in fact shows higher integrity than the surface numbers appear to indicate. This is due to some quasi-problematic mismatches.

Our efforts reported in the paper are similar to the sense annotation task performed by [2] for English, but with some differences, such as: only Core WordNet with Bulgarian sense lexicon has been used for the annotation; the annotation was performed on a treebank, which provided the facility of using a derived valence lexicon and available grammatical roles; no confidence markers have been used by the human annotators - the superannotation technique and cross-resource mappings were adopted as quality assurance strategies instead; DBPedia links have been added.

After the cross-reference stage is performed, the treebank will be made publicly available for research use. The mapping of senses and DBPedia URIs to a formal ontology will provide a basis for the combination of sentence-based semantic analysis with links to world knowledge.

6 Acknowledgements

This research has received partial funding from the EC's FP7 (FP7/2007-2013) under grant agreement number 610516: "QTLeap: Quality Translation by Deep Language Engineering Approaches".

References

[1] Atanas Chanev, Kiril Simov, Petya Osenova and Svetoslav Marinov. (2007) *The BulTreeBank: Parsing and Conversion.* In: Galia Angelova et. al (eds.) Proceedings from RANLP 2007, pp. 114-120.

[2] Christiane Fellbaum, Joachim Grabowski and Shari Landes. (1998) Performance and Confidence in a Semantic Annotation Task. In: Christiane Fellbaum (editor) *WordNet. An Electronic Lexiacl Database.* The MIT Press, pp. 217-238.

[3] Masood Ghayoomi, Kiril Simov and Petya Osenova. *Constituency Parsing of Bulgarian: Word- vs Class-based Parsing.* In: Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk and Stelios Piperidis (eds.) Proceedings of LREC 2014, 26-31 May, Reykjavik, Iceland, pp. 4056-4060.

[4] Verena Henrich and Erhard Hinrichs. (2013) *Extending the TÜBa-D/Z Treebank with GermaNet Sense Annotation.* In: Iryna Gurevych, Chris Biemann, and Torsten Zesch (eds.): Language Processing and Knowledge in the Web, Lecture Notes in Computer Science, Vol. 8105, 2013, pp. 89-96.

[5] A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. (2004) *The nombank project: An interim report.* In:

A. Meyers, editor, HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation, pages 24–31, Boston, Massachusetts, USA, May 2 - May 7 2004.

[6] Simonetta Montemagni, Francesco Barsotti, Marco Battista, Nicoletta Calzolari, Ornella Corazzari, Alessandro Lenci, Vito Pirrelli, Antonio Zampolli, Francesca Fanciulli, Maria Massetani, Remo Raffaelli, Roberto Basili, Maria Teresa Pazienza, Dario Saracino, Fabio Zanzotto, Nadia Mana, Fabio Pianesi, Rodolfo Delmonte. (2003) The syntactic-semantic treebank of Italian. An overview. *Linguistica Computazionale* XVI-XVII, pp. 461-492.

[7] Petya Osenova, Kiril Simov, Laska Laskova, Stanislava Kancheva. (2012) *A Treebank-driven Creation of an OntoValence Verb lexicon for Bulgarian.* In: Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis (eds. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey. ELRA, pp. 2636-2640.

[8] Martha Palmer, Daniel Gildea, and Paul Kingsbury. (2005) The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31 (1): 71–106, March 2005.

[9] (2005) Andrew Philpot, Eduard Hovy and Patrick Pantel. *The Omega Ontology.* In: Proceedings of the IJCNLP workshop on Ontologies and Lexical Resources.

[10] Rudolf Rosa, Jan Mašek, David Marešek, Martin Popel, Daniel Zeman and Zdeněk Žabokrtský. (2014) *HamleDT 2.0: Thirty Dependency Treebanks Stanfordized.* In: Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk and Stelios Piperidis (eds.) Proceedings of LREC 2014, 26-31 May, Reykjavik, Iceland, pp. 2334-2341.

[11] Kiril Simov and Petya Osenova. (2005) *Extending the Annotation of BulTreeBank: Phase 2.* The Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005) Barcelona, 9-10 December 2005, pp. 173-184.

The Effect of Annotation Scheme Decisions on Parsing Learner Data

Marwa Ragheb and Markus Dickinson

Department of Linguistics
Indiana University
E-mail: {mragheb,md7}@indiana.edu

Abstract

We present a study on the dependency parsing of second language learner data, focusing less on the parsing techniques and more on the effect of the linguistic distinctions made in the data. In particular, we examine syntactic annotation that relies more on morphological form than on meaning. We see the effect of particular linguistic decisions by: 1) converting and transforming a training corpus with a similar annotation scheme, with transformations occurring either before or after parsing; 2) inputting different kinds of part-of-speech (POS) information; and 3) analyzing the output. While we see a general favortism for parsing with more local dependency relations, this seems to be less the case for parsing the data of lower-level learners.

1 Introduction

An increasingly popular topic in parsing is to parse non-canonical data [11], including the data of second language learners [4, 10, 13, 18]. In this paper, we add to this growing body of work, focusing less on the parsing techniques and more on the effect of the linguistic distinctions made in the data. In particular, we examine syntactic annotation (for English) that makes different assumptions than in previous work, relying more on morphological form than on context or meaning. We will see the effect of particular linguistic decisions by: 1) converting and transforming a training corpus with a similar annotation scheme (section 2), with transformations occurring either before or after parsing (section 3.1); 2) inputting different kinds of part-of-speech (POS) information (section 3.2); and 3) analyzing the output (section 3.3).

We work with a pilot version of the SALLE corpus [7, 19, 21], which has a fairly unique perspective. It is focused on morphologically-driven dependencies and prioritizes syntax—often to the exclusion of semantics—whereas other parsing

work has been more focused on connecting to the semantics of a sentence. Regardless of what one hopes to achieve with the annotation in the end (e.g., acquisition research [10, 12, 20], parsing to assist in error correction or feedback [26], etc.), noting differences in the parsing results is important to improve the parsing more generally, to see the influences of different kinds of information. As one example in our data, there are two POS tags for every position, to reflect different kinds of evidence [see also 5]. Parsing this corpus helps in the process of teasing apart what makes learner language difficult and where parsing can be improved.

2 Data

While we wish to parse one set of data (section 2.1), the training data with the closest available annotation scheme has significant differences (section 2.2). We describe the data sets, followed by how we prepared the training data to be compatible with testing (section 2.3), highlighting differences which can affect parsing.

2.1 Testing Data: Target Annotation

The testing data consists of 25 learner essays (491 sentences, 7381 tokens) gold-annotated with the SALLE scheme [7, 21].[1] The essays can be grouped into three different levels—beginner, intermediate and advanced—based on placement scores (1 (low) to 7 (high)) assigned by expert raters for the Intensive English Program at Indiana University. These essays were prompt-based, timed placement exams, and they represent a variety of first languages (L1s) [see 19, for more on the essays].

The annotation scheme [22] annotates lemmas, two POS tags reflecting potentially diverging morphosyntactic and distributional evidence (POS_M, POS_D), and dependency relations that are based on the morphosyntactic POS tags. The scheme also encodes subcategorization information. An example annotation is in figure 1, where POS tags are a simplified version of the SUSANNE scheme [25] and the syntactic relations are a modified and expanded set of CHILDES (see section 2.2).

The different layers provide access to innovative learner constructions, through the presence of mismatches between layers. In this example, for instance, the POS_M of VV0 (base form verb) for *decide* conflicts with the POS_D of VV (underspecified verbal form), and *job* subcategorizes for a determiner (<DET>), yet finds two of them. Note that, because both *which* and *my* are morphologically valid determiners, both are annotated as DET. That is, decisions are biased towards morphological information [7, 21], a point which affects parsing (see section 3.3).

Another point that may affect the parsing results is the use of underspecified labels. SALLE makes use of underspecification when there is not enough linguistic

[1] Though not huge, the size of the gold-annotated testing data is comparable to other studies directly parsing learner language [e.g., 10] and suits the purpose of investigating the effect of linguistic decisions. We are currently investigating ways to expand the gold annotated corpus by incorporating more semi-automatic steps into the otherwise manual process.

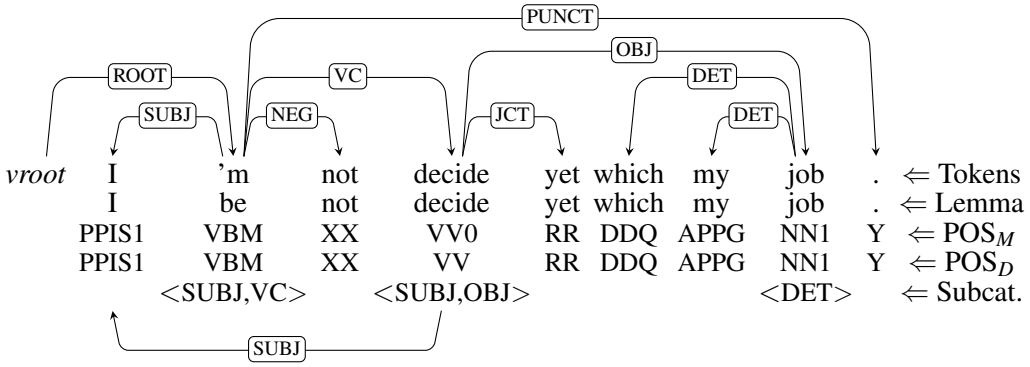


Figure 1: Example annotation using SALLE scheme

evidence to determine a full label (as is the case with the POS_D for *decide* above) or to select a certain label over another (in either the POS or dependency fields). Underspecified labels (including a complete non-specification, notated as _) are not present in the training data , a point discussed further in section 3.3.

As a side point, since the scheme aims at directly annotating language, without imposing target hypotheses or corrected versions of the learner text [20], it does not encode learner "errors". For analysis, many of the non-canonical structures in what the learners wrote can nonetheless be arrived at by examining mismatches between layers, for example, VV0 vs. VV for *decide* and the subcategorization vs. dependency realization for *job* in figure 1.

2.2 Training Data: Source Annotation

The training data is a subset of the CHILDES corpus of first language acquisition [14], from the US English data, downloaded in March 2012 [19]. This data is appropriate for training because, as a corpus of (L1) acquisitional data, it is likely to include some patterns L2 learners also use, such as overgeneralization of certain inflectional morphemes (e.g., *hitted, goed*) and similar developmental sequences (e.g., for negation) [cf., e.g., 9]. More importantly, the syntactic annotation scheme [24] formed the basis of SALLE's scheme [7, 19, 21]. The SALLE labels are based on the CHILDES labels, but there are more labels and some constructions are modified; still, the annotation schemes are relatively compatible.

This data presents a variety of challenges: 1) it is not in a format appropriate for modern trainable parsing systems, and it is occasionally inconsistent in its formatting (section 2.3.1); 2) the schemes have to be harmonized (section 2.3.2); and 3) it is for spoken data, not written data, a fact that partly accounts for lower parsing results (section 3.3). We have automatically removed some of the training data (verbless and anomalous [8] sentences), due to noise; space precludes a full discussion [see 19]. This results in 102,733 sentences and 727,297 tokens.

2.3 Data Preparation

2.3.1 Conversion

The first step to prepare the data is to convert the data from CHAT format [14] to CoNLL-X format [3], to make it appropriate for parsing. The required information for parsing is on three different tiers: the speaker tier (represented in `%flo` in the top of figure 2), the morphology tier (`%mor`) and the syntactic tier (`%xgra` or `%gra`). We want the representation in CoNLL-X format, as seen in the bottom of figure 2.

```
%flo:   what's Mamma's name ?
%mor:   pro:wh|what~aux|be&3S n:prop|Mamma~poss|s n|name ?
%xgra:  1|2|PRED 2|0|ROOT 3|5|MOD 4|5|MOD 5|2|SUBJ 6|2|PUNCT
```

1	what	what	pro:wh	_	2	PRED	_	_	_
2	's	be&3S	aux	_	0	ROOT	_	_	_
3	Mamma	Mamma	n:prop	_	5	MOD	_	_	_
4	's	s	poss	_	5	MOD	_	_	_
5	name	name	n	_	2	SUBJ	_	_	_
6	?	?	_	_	2	PUNCT	_	_	_

Figure 2: CHAT (top) and CoNLL-X (bottom) formats for *what's Mamma's name?*

When the information is properly aligned across all three tiers, the conversion process is straightforward. There are cases with misalignments between two of the three layers, however, requiring additional steps [19]. For example, in figure 2 the tilde is used to mark single `%mor` tokens which correspond to two `%xgra` (syntactic) units. In the same essay, a later instance of *mama's* contains a tilde in the `%mor` layer, but only corresponds to only one `%xgra` unit. Our conversion script covers the majority pattern for special cases (tildes, compound nouns, punctuation), and we corrected the other cases by hand, generally detectable when the CHAT layers have differing numbers of units.

One last step in the conversion process is to change the POS tags from the ones used in CHILDES to the ones used in SALLE. We automatically tag the CHILDES data with TnT [2], using the pre-built SUSANNE model, and then employ a simple mapping scheme to the SALLE labels (a subset of SUSANNE). Although this introduces some noise in training, this does not affect our current focus of determining which transformation model or which input POS results in better performance (see section 3). Additionally, the CHILDES corpus is itself mostly automatically POS-tagged [14, p.147-148], and previous experiments [19, ch. 6] using the CHILDES POS tagset showed similar trends as to what is reported in section 3.

2.3.2 Transformations

There are three main syntactic constructions that CHILDES and SALLE analyze differently: auxiliary verbs, coordination, and possessives. The data thus needs to be transformed to align with SALLE, either before training or after parsing (see section 3). We focus here on the first two areas of difference, as possessives involve a simple swapping of heads, with little effect on surrounding constructions.

The first difference, stemming from SALLE's prioritization of syntax over semantics, is the *auxiliary-as-head* analysis that SALLE adopts, whereas CHILDES considers the main verb as the head of a verbal unit (e.g., *have run*). The transformation makes the first auxiliary the head of any verbal chain and then, heuristically following SALLE attachments, keeps the following arguments as dependents of the content-ful verb, but preceding arguments as dependents of the auxiliary.

As for the second difference: while CHILDES analyzes the conjunction as the head of the coordination phrase, SALLE adopts a right-branching analysis, since this accounts better for learner innovations [6]. The transformation process likewise switches heads here, but it also has to account for the interaction between coordination and auxiliary transformations [19]. Namely, auxiliary transformations take place before coordinations, so that verbal heads are properly coordinated. Figure 3 shows the difference in coordination analyses between the schemes.

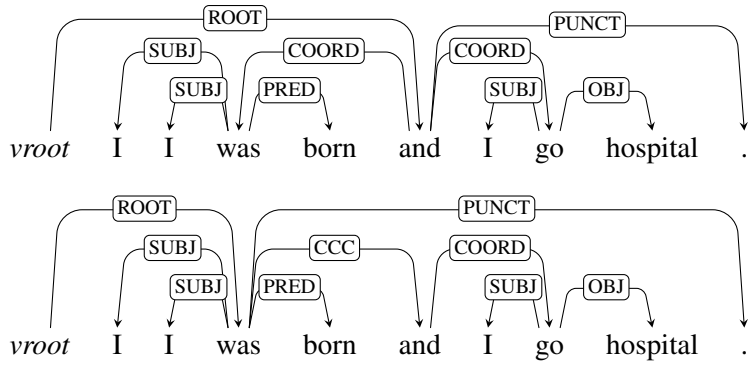


Figure 3: Coordination in CHILDES (top) and SALLE (bottom)

3 Experiments

3.1 Transformations

Given the transformations from the CHILDES annotation scheme to the SALLE one, we can ask whether it is preferable to train a parser on the CHILDES scheme and then transform the resulting output (*post*) or to first transform the training data, to learn the SALLE model directly (*pre*). As SALLE generally posits more local distinctions (section 2.3.2), and based on preliminary experiments [19] we ex-

pect to see better results with the *pre* model [cf. 16]. Note that the motivation for SALLE's scheme was not to make parsing easier, but to be more purely syntactic, which happens to be more local in English.

3.2 POS Information

The SALLE data provides two part-of-speech (POS) tags, one more driven by morphology and one by syntactic distribution. The annotation scheme used for parsing itself is more morphologically-based, but syntax by its nature has to rely on contextual information and parsing results can vary based on the POS input [e.g., 15]. Thus, we can ask which POS information works best as input to the parser: morphological (POS_M), distributional (POS_D), or both (POS_{Both})?

3.3 Results

The experiments use MaltParser [17], optimizing the parser settings with help of MaltOptimizer [1]. This chooses the *stackproj* algorithm based on the nature of the training data. Evaluation is performed using the `eval.pl` script.[2]

The results for the six models (2 transformations $\times$ 3 POS inputs) are given in table 1. We can immediately draw two conclusions: 1) the POS_M models (top row) are consistently the best (albeit, slightly); and 2) the *pre* models are consistently better than the *post* ones. This supports our hypotheses: morphologically-based POS information seems better for parsing with this scheme, and more local (i.e., more adjacent) syntactic relations are preferred over less local ones.

	pre		post	
	LAS	UAS	LAS	UAS
POS_M	62.8%	74.3%	61.4%	73.4%
POS_D	62.7%	74.2%	60.9%	73.0%
POS_{Both}	62.7%	74.2%	60.9%	73.0%

Table 1: Overall results for the six different models

Individual results The results are more complicated when examining individual files, which show great variability, as partly illustrated in table 2. For example, in this table, comparing the pre+POS_M model to the post+POS_M model, the values range from ones where the post+POS_M model has a 4.9% better LAS than the pre+POS_M model (opposite of the overall trend) to ones where the pre+POS_M is 5.8% better than the post+POS_M model. Similarly, though not in the table, the pre+POS_D model varies from having 4.5% worse LAS than the pre+POS_M model to having 2.4% better.

[2] `http://ilk.uvt.nl/conll/software/eval.pl`

Essay	Level	Origin	# of Sents	# of Words	W/S	Pre LAS	Post LAS	Diff. LAS
201	1	Taj.	2	45	22.5	41.5	46.3	-4.9
064	3	Kor.	27	261	9.7	60.0	63.7	-3.7
285	1	Saudi	8	96	12.0	65.5	67.8	-2.3
017	5	Afg.	34	472	17.8	49.7	51.6	-1.9
204	4	Thai	10	178	13.9	69.7	71.6	-1.9
267	3	Saudi	21	220	10.5	65.3	66.8	-1.5
097	4	Kor.	19	225	11.8	72.8	74.3	-1.5
021	5	Afg.	12	254	21.2	53.5	54.8	-1.3
193	7	Thai	15	356	23.7	61.5	62.1	-0.6
018	6	Afg.	17	293	17.2	62.0	62.0	0.0
250	3	Saudi	21	261	12.4	59.6	58.6	0.9
046	5	Jap.	18	329	18.3	53.8	52.7	1.0
026	4	Chi.	40	462	11.6	61.4	60.1	1.3
033	7	Jap.	25	308	22.9	60.6	59.2	1.4
036	6	Jap.	17	390	12.3	65.3	63.4	1.9
040	5	Jap.	23	317	13.8	67.1	65.0	2.1
240	6	Hun.	26	377	14.5	67.0	64.6	2.3
034	6	Jap.	16	327	20.4	63.4	61.0	2.4
052	5	Kaz.	21	359	17.1	64.6	62.1	2.5
080	2	Kor.	18	174	9.7	61.9	59.2	2.7
041	7	Jap.	16	305	19.1	73.6	69.9	3.6
016	6	Afg.	33	353	26.0	60.2	56.0	4.3
129	3	Kor.	14	364	10.7	64.0	59.3	4.7
053	5	Kaz.	18	290	16.1	61.9	56.4	5.6
020	6	Afg.	20	365	18.3	60.1	54.2	5.9
-diff	**3.7**	n/a	**16.4**	**234.1**	15.9	**59.9**	62.1	-2.2
+diff	**5.1**	n/a	**21.7**	**332.1**	16.2	**63.0**	60.1	2.8
overall	4.6	n/a	19.6	295.2	16.1	61.8	60.9	0.9

Table 2: LAS (in %) for POS_M models, with *Pre* and *Post* transformations, organized by the difference in LAS ($Pre - Post$), and including number of words, sentences, and average words per sentence; (macro-)averaged values are presented for the different variables, grouped by essays which had a better *Post* model (*-diff*, i.e., averaging the first nine rows) or a better *Pre* model (*+diff*)

Focusing in on the effect of transformations before training (*pre*) or after parsing (*post*), as in table 2, we can observe a major trend, as outlined in the summary statistics at the bottom of the table. Scoring learners on a 1–7 level (see section 2.1), learners who are at a lower level (3.7 on average vs. 5.1) tend to have worse overall parsing performance (e.g., 41.5%/46.3% LAS for learner 201). Correspondingly,

the models which perform transformations after parsing (*post*) tend to work better for such learners. We believe that this may stem from less canonical (and perhaps less local) structures in such data, allowing for noise in the transformation process to have a bigger effect on the results.

While higher average numbers of sentences and words may lead to more difficult parsing in the general case, this seems not to be the case with the learners in this data, at least for the pre+POS$_M$ model, where those with a better pre+POS$_M$ model had on average more sentences (21.7 vs. 16.4), more words (332.1 vs. 234.1), and much higher parsing accuracy (63.0% vs. 59.9% LAS). That is, essays with more words were actually easier to parse.

We can tentatively conclude the following from these results: 1) knowing the level of a learner before parsing may influence the choice of parsing model, in our case differentiating when one performs annotation transformations, i.e, parses with more local decisions or not; and 2) if one does not know the learner level, various features may combine to help select a parsing model (e.g., number of words and sentences). In short, the handling of linguistic decisions for parsing may need to be optimized differently for different kinds of learners.

Underspecification Turning to why the POS$_{Both}$ model performs the same as the POS$_D$ model, one effect we see is of underspecified (distributional) tags in both of these models, tags which the parser has never seen before—i.e., sparsity is a major issue. In figure 4, for instance, we see the effect of different parsing models for (1).

(1) I have exprence of living in a coutry that **have see** war and exprence of living as immagrant and so they all made my characteristics . (*Essay 021*)

In this learner sentence, neither *have* nor *see* is correctly inflected, which leads to underspecified POS$_D$ tags (VH and VV, respectively). Consequently, SALLE treats both of them as potential heads of a modifying clause (CMOD) [7]. Figure 4 shows the difference in dependency labels between the POS$_M$ and POS$_D$ parsed trees, as compared to the gold annotation in the top tree. Most clearly, *war* is correctly an OBJ when *see* is a known type of verb (VV0) in the POS$_M$ tree, whereas it becomes a MOD (a label appropriate for nominal modifiers) when the verb label is underspecified and the parser cannot recognize it. The verb *see* itself is labeled as POBJ (prepositional object) in the POS$_D$ parsed tree, a label appropriate for a noun but not a verb. With respect to the MCOORD (modificatory coordination) label applied to *see* in the POS$_M$, this most probably is the influence of training data inaccuracies [see 19, for more on that issue].[3]

[3]For more on the treatment of *of living*, see sections 3.3.1 and 4.5 of [7]. The parser makes reasonable decisions in these cases, and the proper handling of *-ing* forms is an area of active investigation.

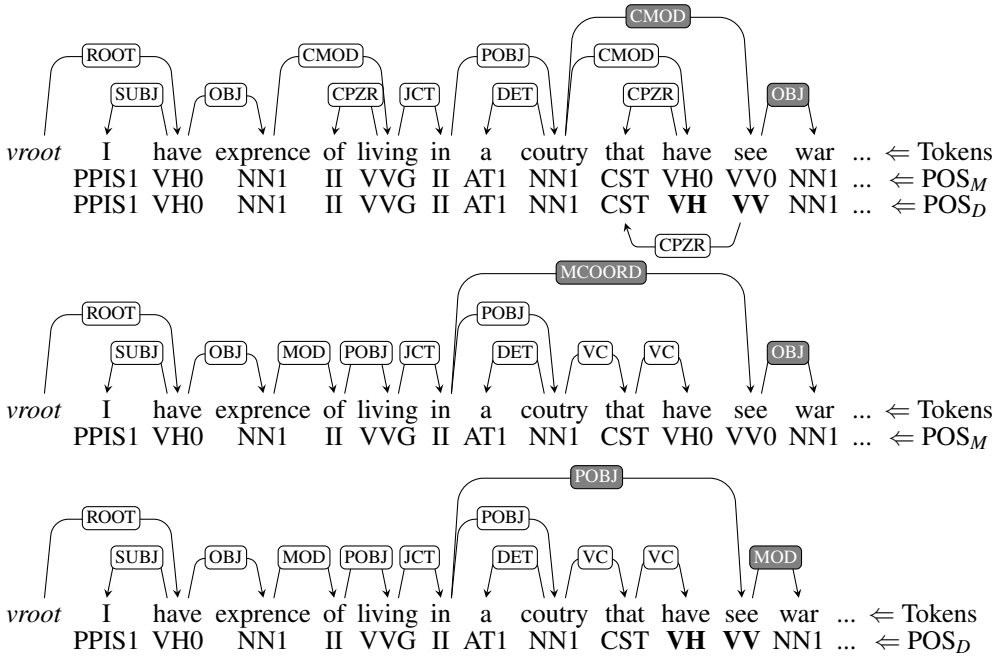


Figure 4: Comparison of POS$_M$ and POS$_D$ models: gold (top), parsed POS$_M$ (middle), and parsed POS$_D$ (bottom)

4 Conclusion and Outlook

We have experimented with parsing learner language directly, without mapping to target hypotheses or correcting errors first [23], using the SALLE scheme, the only scheme we know of which explicitly favors syntactic interpretations over semantic ones and is highly form-based. Using a training corpus with slightly different conventions, we have specifically investigated whether certain linguistic decisions in the annotation affect parsing differently, exploring: 1) different combinations of POS tags, 2) the effect of transforming annotation before or after parsing. We have seen a slight favortism for morphologically-based POS tags—with the caveat that distributionally-based POS tags suffered from unknown tags—and a clear favortism for transforming the annotation before training a parser, likely due to differences in locality. Despite this, examining individual learner files shows that parsing the data of lower-level learners may benefit from a different transformation process.

The overall results are lower than in previous parsing experiments for learner data [4, 10], and future work will have to tease apart the effect of the morphologically-driven distinctions (which should in principle be easier) and the training data (e.g., annotation errors) and transformation process. Current experiments indicate that changing the training data (from an L1 acquisition data set to a native English treebank) can yield higher parsing results. Future work will look more closely at the

constructions that are difficult for the parser to analyze correctly. An immediate goal is to increase the size of the SALLE gold-annotated learner data to see if the trend still holds, given more testing data. Developing a parser to pre-process the data should provide a faster route to obtaining more annotated data. One can also further unpack the influence of underspecified information.

Acknowledgements

We would like to thank the three anonymous reviewers for their helpful feedback.

References

[1] Miguel Ballesteros and Joakim Nivre. MaltOptimizer: An optimization tool for MaltParser. In *Proceedings of the System Demonstration Session of the Thirteenth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 23–27, Avignon, France, April 2012.

[2] Thorsten Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP 2000)*, pages 224–231, Seattle, WA, 2000.

[3] Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL-X*, pages 149–164, New York City, 2006.

[4] Aoife Cahill, Binod Gyawali, and James Bruno. Self-training for parsing learner text. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 66–73, Dublin, Ireland, August 2014. Dublin City University.

[5] Ana Díaz-Negrillo, Detmar Meurers, Salvador Valera, and Holger Wunsch. Towards interlanguage POS annotation for effective learner corpora in SLA and FLT. *Language Forum*, 36(1–2):139–154, 2010. Special Issue on New Trends in Language Teaching.

[6] Markus Dickinson and Marwa Ragheb. Dependency annotation of coordination for learner language. In *Proceedings of the International Conference on Dependency Linguistics (Depling 2011)*, pages 135–144, Barcelona, Spain, 2011.

[7] Markus Dickinson and Marwa Ragheb. Annotation for learner English guidelines, v. 0.1. Technical report, Indiana University, Bloomington, IN, June 2013. June 9, 2013.

[8] Markus Dickinson and Amber Smith. Detecting dependency parse errors with minimal resources. In *Proceedings of IWPT-11*, pages 241–252, Dublin, October 2011.

[9] Rod Ellis. *The Study of Second Language Acquisition*. Oxford University Press, Oxford, 1994.

[10] Jeroen Geertzen, Theodora Alexopoulou, and Anna Korhonen. Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge open language database (EFCAMDAT). *Proceedings of the 31st Second Language Research Forum (SLRF)*, 2013.

[11] Yoav Goldberg, Yuval Marton, Ines Rehbein, Yannick Versley, Özlem Çetinoğlu, and Joel Tetreault, editors. *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*. Dublin City University, Dublin, Ireland, August 2014.

[12] John A. Hawkins and Paula Buttery. Criterial features in learner corpora: Theory and illustrations. *English Profile Journal*, 1(1):1–23, 2010.

[13] Julia Krivanek and Detmar Meurers. Comparing rule-based and data-driven dependency parsing of learner language. In *Proceedings of the International Conference on Dependency Linguistics (Depling 2011)*, Barcelona, 2011.

[14] Brian MacWhinney. *The CHILDES Project: Tools for Analyzing Talk.* Lawrence Erlbaum Associates, Mahwah, NJ, 3rd edition, 2000. Electronic Edition, updated April 25, 2012, Part 2: the CLAN Programs: `http://childes.psy.cmu.edu/manuals/CLAN.pdf`.

[15] Wolfgang Maier, Sandra Kübler, Daniel Dakota, and Daniel Whyatt. Parsing German: How much morphology do we need? In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 1–14, Dublin, Ireland, August 2014.

[16] Jens Nilsson, Joakim Nivre, and Johan Hall. Generalizing tree transformations for inductive dependency parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 968–975, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[17] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Kubler, Svetoslav Marinov, and Erwin Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[18] Niels Ott and Ramon Ziai. Evaluating dependency parsing performance on German learner language. In *Proceedings of TLT-9*, volume 9, pages 175–186, 2010.

[19] Marwa Ragheb. *Building a Syntactically-Annotated Corpus of Learner English*. PhD thesis, Indiana University, Bloomington, IN, August 2014.

[20] Marwa Ragheb and Markus Dickinson. Avoiding the comparative fallacy in the annotation of learner corpora. In *Selected Proceedings of the 2010 Second Language Research Forum: Reconsidering SLA Research, Dimensions, and Directions*, pages 114–124, Somerville, MA, 2011. Cascadilla Proceedings Project.

[21] Marwa Ragheb and Markus Dickinson. Defining syntax for learner language annotation. In *Proceedings of the 24th International Conference on Computational Linguistics (Coling 2012), Poster Session*, pages 965–974, Mumbai, India, 2012.

[22] Marwa Ragheb and Markus Dickinson. Developing a corpus of syntactically-annotated learner language for English. In *Proceedings of the 13th International Workshop on Treebanks and Linguistic Theories (TLT13), Poster Session*, Tübingen, Germany, 2014.

[23] Ines Rehbein, Hagen Hirschmann, Anke Lüdeling, and Marc Reznicek. Better tags give better trees - or do they? *Linguistic Issues in Language Technology (LiLT)*, 7(10), 2012.

[24] Kenji Sagae, Eric Davis, Alon Lavie, and Brian MacWhinney an Shuly Wintner. Morphosyntactic annotation of CHILDES transcripts. *Journal of Child Language*, 37(3):705–729, 2010.

[25] Geoffrey Sampson. *English for the Computer: The SUSANNE Corpus and Analytic Scheme*. Clarendon Press, Oxford, 1995.

[26] Joel Tetreault, Jennifer Foster, and Martin Chodorow. Using parse features for preposition selection and error detection. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 353–358, Uppsala, Sweden, July 2010.

Metrical annotation for a verse treebank

T. M. Rainsford[1] and Olga Scrivner[2]

[1]Institut für Linguistik/Romanistik, Universität Stuttgart
[2] Indiana University
E-mail: [1]tmr740-ac@yahoo.co.uk
[2]obscrivn@indiana.edu

Abstract

We present a methodology for enriching treebanks containing verse texts with metrical annotation, and present a pilot corpus containing one Old Occitan text. Metrical annotation is based on syllable tokens, and is generated semi-automatically using two algorithms, one to divide word tokens into syllables, and a second to mark the position of each syllable in the line. Syntactic and metrical annotation is combined in a single multi-layered ANNIS corpus. Three initial findings based on the pilot corpus illustrate the close relation between syntactic and metrical structure, and hence the value of enriching treebanks in this way.

1 Introduction

The goal of the project presented here is to develop a methodology for enriching treebanks containing verse texts with detailed metrical annotation. The earliest texts preserved for many European languages, in this case Occitan, are frequently in verse, and it is therefore desirable when analysing these texts to take into consideration any possible effect of the verse form on the syntax.

In the present paper, we will outline the methodology we have adopted in producing a small pilot treebank containing a 10th-century Occitan verse text, a fragment from a verse adaptation of Boethius' *De Consolatione Philosophæ* (henceforth *Boeci*). The pilot corpus is available online at www.oldoccitancorpus.org. In working on the treebank aspects of the corpus, we have built on the work carried out for the Old Occitan *Flamenca* text by Scrivner, Kübler, Vance and Beuerlein [15].

2 Background

2.1 Why enrich treebanks with metrical annotation?

There exists a consensus among linguists that the syntax of verse texts differs from that of prose, with unusual word orders adopted to fit the constraints of the metre. For example, in introducing a study of Early Old French (12th-century) syntax, Labelle [11] feels obliged to acknowledge that "the disadvantage of concentrating on this period of time is that the available texts are in verse, and we might expect freer word order, with probably more scrambling to accommodate the rhyme." A difficult task is only made harder by the fact that data extracted from modern treebanks (e.g. the MCVF Old French treebank, Martineau et al. [12]) does not contain any metrical information, so it is not possible to establish, for example, whether a particularly unusual word order may have been adopted to place a rhyming word at the end of the line without referring back to the source edition. This problem of information loss can be resolved by adding metrical annotation to a treebank. Furthermore, it allows researchers to write combined syntactic and metrical queries, placing them in a position to demonstrate whether specific metrical constraints, especially at the end of the line (the rhyme) and at the half-line boundary (the cæsura), are in fact associated with unusual syntactic structures.

2.2 What information should be included in metrical annotation?

Corpora containing metrical annotation are relatively rare (see section 2.3 below), and there is little consensus regarding which metrical and/or prosodic features should be encoded. Indeed, it would not even be desirable for all metrical corpora to contain the same information, since different versification systems exploit different aspects of linguistic structure (e.g. the distinction between light and heavy syllables is fundamental in classical Latin verse, but irrelevant for versification in modern Romance languages). However, every metrical annotation system must take account at some level of the defining unit of a verse text: the line.[1] Beyond the line, the annotation scheme may choose to mark:

- Segments bigger than the line, e.g. stanza, poem
- Segments smaller than the line, e.g. half-line, foot, syllable, mora
- Line-linking phenomena, e.g. rhyme, assonance, alliteration

The metre of *Boeci* is typical of Old Occitan (and Old French) epic texts. The poem is written in lines of ten counted syllables, divided regularly into two half-lines by a cæsura between the fourth and fifth counted syllables (stressed syllables are underlined):

[1] The division of a verse text into such "correlatable and commensurable segments" is considered a defining feature by metricists, *cf.* Gasparov [8], p. 1.

(1) 1 2 3 4 / 5 6 7 8 9 10
Nos jo- ve om- ne / quan- dius que nos es- tam
'We young men, when young we are [...]' (l. 1)

The fourth and tenth counted syllables must bear a lexical stress (e.g. *om-*, *-tam*), while post-tonic syllables at the end of the first half-line are not counted (e.g. *-ne*). Lines are linked into *laisses* of irregular length by assonance: a simple form of rhyme, in which the final stressed vowels of lines, but not necessarily preceding or following consonants, must be similar. For instance, the first *laisse* of the poem contains lines ending with an /a/ vowel (*estam* : *parllam* : *esperam* : *annam* : *fam* : *clamam*); the third with an /o/ vowel (*fello* : *pejor* : *quastiazo*, etc.). Therefore, in order to describe the metrical structure of the poem completely, the annotation scheme should mark both properties of the *laisse* and those of the syllable in addition to the line.

It should be noted that multi-layered annotation is not necessary to encode this kind of information. For example, a major corpus of historical Dutch song, the *Nederlandse Liederenbank*[2], does not annotate stanzas, lines or stressed syllables explicitly. Instead, metrical properties are given by a complex "stanza form" tag which is included in the metadata for each text. For instance, the metre of the text with incipit *Doersocht en bekent hebt ghi / Mi Heer mijn sitten mijn opstaen* is given as `3A 4B 3A 4B 3C 4D 3C 4D`: eight lines, rhyming ABABCDCD, containing alternately three and four stressed syllables. However, an approach of this kind has clear drawbacks when metrical annotation is to be combined with other annotation layers, since it provides no means of establishing correspondances at the token level.

2.3 Which corpora can serve as models?

Corpora containing metrical annotation segmenting the text into units smaller than the line are relatively rare. For syllabic verse, the Anamètre project[3] has produced a metrically annotated corpus of Classical and Modern French verse, using a series of Python scripts to mark up the text for syllable structure and to identify vowel phonemes [3]. A similar approach is adopted for the *Corpus of Czech Verse*[4], but here the metrical annotation also marks stressed and unstressed syllables, since this distinction is essential to Czech metre. While most metrical information is included in line-level tags, indicating the metre of the line as a series of "feet"[5], these tags are generated by an automated algorithm which divides the line into syllables [7]. The syllable-level representation in the database includes both a phonetic transcription of the syllable, and whether it bears a lexical stress [13]. Both corpora are intended

[2] `http://www.liederenbank.nl/`

[3] `http://www.crisco.unicaen.fr/verlaine/index.html`

[4] `http://www.versologie.cz/en/kcv_znacky.html`

[5] A fixed sequence containing one stressed and a number of unstressed syllables, e.g. iamb (unstressed–stressed), trochee (stressed–unstressed).

for the study of purely metrical phenomena: the Czech corpus, for example, has been used to establish a database of metres used in poetry and a database of rhymes.

Corpora which combine prosodic and syntactic annotation are more widespread, and share with the present corpus a need for multiple tokenization, since syntactic annotation is based on words and phrase structures, while prosodic or metrical annotation is based on syllabic structures. The Rhapsodie project has annotated a corpus of spoken French using two different base units: phonemes for prosodic structure and lexemes for syntactic structure [9]. Prosodic and syntactic annotation is organized in separate tree structures but they are interconnected by means of DAGs (directed acyclic graphs). Another method is introduced in the DIRNDL project[6]. Here, a corpus of German radio news is annotated on prosodic, syntactic and discourse levels. Each layer is presented as a separate graph that is connected to others via pipeline links [10]. However, despite some core similarities, it is important to note that the prosodic annotation of spoken language differs greatly from metrical annotation, since unlike poetry, spoken language is not designed to fit a metrical template. Metrical annotation is in this regard rather simpler, as only phenomena which are metrically relevant (e.g. syllables, stress, rhyme) need be included. Moreover, there is little need to include audio or even phonetic transcriptions, particularly when dealing with historical texts for which the precise phonology is often uncertain.

To our knowledge, the only extant corpus which combines metrical and treebank annotation is the recently-released *Greinir skáldskapar*[7] corpus of historical Icelandic verse, which combines syntactic, phonological and metrical annotation [6]. The corpus is accessible through a purpose-built online portal, queries are formulated using drop-down menus, and the interface is intended to facilitate combined syntactic and metrical queries (e.g. "find all line-initial subjects that alliterate"). However, it should be noted that Icelandic alliterative verse is organized according very different principles from the syllabic verse of Old Occitan, and thus the annotation procedure presents very different challenges.

3 Methodology

From the preceding discussion, we may identify two main challenges in enriching a treebank with metrical annotation:

1. Designing and creating a layer of metrical annotation
2. Combing metrical annotation with a treebank in such a way as to be easily searchable (ideally using existing tools)

[6] http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/dirndl.en.html

[7] http://bragi.info/greinir/

3.1 Creating metrical annotation

With regard to the first challenge, we elected to use a multi-layered approach for the metrical annotation based on both line and syllable tokens. This multimodal approach allows for overlapping and hierarchically conflicting layers of annotation that would otherwise be incompatible in a traditional inline single-level corpus [16]. The use of syllable-level tokenization is essential to create a detailed representation of the metrical structure of the text, and is crucial if the corpus is also to be used to investigate the metrical characteristics of Old Occitan texts (e.g. the extent to which stressed syllables are used to create a regular rhythm [14]). Moreover, it also allows automatic identification of the position of the cæsura, a metrical position which, like the end of the line, is likely to be associated with syntactic constituent boundaries.[8]

In order to create the annotation, we first devised a simple algorithm to divide the words in the text into syllables, a relatively straightforward task given the comparatively phonemic orthography of Old Occitan. This (i) identifies syllable nuclei (i.e. sequences of vowels), (ii) learns permitted onset and coda clusters from word-initial and word-final consonant sequences and (iii) divides sequences of consonants between vowels into coda and onset accordingly. The results produced by the algorithm were manually corrected, and the position of the lexical stress was added.[9]

The second phase of generating the annotation involved labelling each syllable according to its position in the line. This is more complex, since some syllables are subject to variable elision rules, and may not 'count' towards the ten syllables in the line. Two principal elision rules were modelled:[10]

- **Synalepha**: Word-final unstressed vowels may be elided when followed by a word-initial vowel. For example, in the sequence *El.l(a) ab* below, the final unstressed vowel of the pronoun *ella* 'she' is not counted:

  (2) 1 2 3 4 / 5 6 7 8 9 10
  El- l(a) ab Bo- e- ci / par- let ta dol- za- ment
  'She spoke so sweetly to Boethius.'

- **Syneresis**: Some vowel–vowel sequences within words may count either as one or two syllables. The most notable example is the imperfect ending *-ia*, where the two possible scansions presumably reflect two possible pronunciations: /i.a/ or /ja/.

[8] In order to create a comprehensive representation of the metre of the text, *laisse* units could also be marked. However, as we felt this was of less immediate interest for the study of the syntactic structure of the text, this layer of annotation is not currently implemented.

[9] The efficiency of this part of the workflow could be improved in future work by integrating existing syllabification algorithms, such as the finite-state syllabification method present for Middle Dutch by Bouma [2], pp. 29–31.

[10] These metrical rules are common not just in Old Occitan but in Old Romance in general (see Chambers [1], pp. 5–7).

Any syllable potentially subject to either synalepha (given the right phonological context) or to syneresis was manually tagged as such in the input to the algorithm.

Labelling each syllable according to its metrical position in the line was carried out by a 'scansion' algorithm, which operates in the following way:

1. select a variable elision rule (no elision/synalepha/syneresis);
2. apply selected variable elision rule(s);
3. apply positional elision rules (post-tonic syllables at cæsura);
4. if the line has ten counted syllables, and the fourth and the tenth bear lexical stress, mark the line as correctly scanned;
5. else, select a different variable elision rule, or a combination of rules, and return to 2;
6. if, once all possible combinations of variable rules have been applied, the line does not scan correctly, mark as 'unscannable'.

For example, when scanning the line given in (2), the algorithm begins by assuming no elision (step 1):

(2′) 1 2 3 4 / 5 6 7 8 9 10 11 12
El- la ab Bo- e- ci par- let ta dol- za- ment

This scansion is unchanged by step 3 (since the cæsura is not in the correct position) and fails at step 4. On the second pass, the algorithm elects to apply synalepha at step 1, giving the following provisional scansion at step 2:

(2″) 1 2 3 4 / 5 6 7 8 9 10 11
El- l(a) ab Bo- e- ci par- let ta dol- za- ment

Step 3 then notes the presence of a word-final unstressed syllable after the cæsura and marks it as 'uncounted', giving the following scansion:

(2) 1 2 3 4 / 5 6 7 8 9 10
El- l(a) ab Bo- e- ci par- let ta dol- za- ment

The line is then marked as correctly scanned at step 4.

No manual intervention is needed to correct the output of the algorithm: since the algorithm is supplied with all accepted rules of Occitan versification, lines marked as 'unscannable' (25 of 257) are genuine metrical exceptions in the manuscript text (see Chambers [1], p. 8–9).

3.2 Combining metrical and treebank annotation

The treebank annotation was created using automated part-of-speech tagging and parsing followed by manual correction, following the method described for the *Flamenca* text by Scrivner, Kübler, Vance and Beuerlein [15]. We elected to use

the open-source ANNIS platform as our corpus search engine, which uses PAULA-XML as its preferred input data format [5].[11] The platform's flexible architecture allows for multidimensional representations of corpora [16], while the web-based query engine is suitable not only for various overlapping annotation schemas but also for different levels of segmentation, which is impossible in most corpus tools. This is crucial for a corpus in which the two core layers of annotation rely on different tokenizations: words in the treebank, syllables in the metrical annotation. The two tokenizations are quite separate, since it is not the case that a word can be treated simply as a spanned sequence of syllable tokens. Word boundaries also occur *within* syllables: for example, of the four word boundaries in the sequence *e te m fiav' eu* 'in you I trusted' (l. 75), only two coincide with boundaries in the syllable tokens *e.tem.fi.a.veu*.

Metrical annotation was exported directly to the PAULA-XML format from the scansion module, and was combined with syntactic annotation converted to the same format. Although it permits multiple layers, PAULA-XML still requires all units to be defined based on indivisible tokens. We elected to use individual characters as tokens, defining both syllables and words using character spans. However, in the ANNIS interface, this arbitrary token level can be hidden, leaving only the relevant higher-level unit (word, syllable and line) visible to the user. At present, the user can work offline with a local version of ANNIS, or work on-line with a server-based version, which we have created.

4 Some sample findings

It is relatively straightforward to use the ANNIS query language to study the relationship between syntactic and metrical annotation layers. In Figure 1, the query finds all finite clauses (IP-MAT and IP-SUB elements) ending at the cæsura, i.e. right aligned (_r_) with the fourth syllable in the line, or with an elided syllable ("el") which immediately follows it (.). Queries of this nature have already led us to some intriguing findings.

Firstly, there is very strong correlation between the metrical structure of *Boeci* and its syntactic structure. Recall that each ten-syllable line is divided into two half-lines of four and six syllables by a cæsura. It transpires that of the 355 finite clauses in the text, *every single one* ends at a half-line boundary: 302 at the end of line, 53 at the cæsura.[12] Moreover, there is not a single line which does not end with a finite clause boundary. While this tendency is not unusual,[13] it is perhaps more surprising to see it exceptionlessly applied in our text. Moreover, it illustrates

[11] PAULA-XML must however be converted to the native relANNIS format using the the Salt-NPapper converter (`http://korpling.german.hu-berlin.de/saltnpepper/`) before the data is usable in ANNIS.

[12] We exclude 5 finite clauses which end within lines tagged as 'unscannable'.

[13] Devine and Stephens [4] note a similar pattern in Ancient Greek verse, arguing convincingly that this effect is due to the association of syntactic constituent boundaries with *prosodic* constituent boundaries in natural language, such as the intonational or the phonological phrase.

```
cat = /IP-(MAT|SUB).*/
& syll_in_line = "4"
& ( #1 _r_ #2
        | ( syll_in_line = "el"
        & #2 . #3
        & #1 _r_ #3 )
)
```

Figure 1: AQL query to identify IP-MAT and IP-SUB constituents ending at the cæsura.

the extent to which the syntax of this text is constrained by the metre: effectively, every finite clause must be four, six or ten syllables long, or contain one or more embedded finite clauses of four, six or ten syllables.

Secondly, there is a strong correlation between the length of the lexical item and its position in the line. Figure 2 shows that polysyllabic and monosyllabic words of the same part-of-speech show radically different distributional tendencies. In all cases, polysyllables are more likely to occur at the end of the line than

	polysyllabic		monosyllabic	
Part-of-speech	line-medial	line-final	line-medial	line-final
oxytonic common nouns	5	81	90	17
oxytonic past participles	7	12	14	0
oxytonic finite verbs	33	23	166	13

Figure 2: Position of oxytonic (= stress on final syllable) lexical items within the line; selected parts of speech.

monosyllables. In the case of common nouns, 94% of polysyllables are line-final but only 16% of monosyllables; in the case of finite verbs, only 7% of monosyllables are line-final; in the case of past participles, no polysyllables are line-final. Since, as we have seen, line-final position is also usually clause-final, we can conclude that in this text, polysyllabic lexical items are most likely to occur at the end of the clause. Whatever the cause of this phenomenon (and it may not necessarily be purely metrical), it is likely to have important consequences for word order in the text, and it can only be studied in a corpus which contains syllable-level annotation.

Finally, one important area of syntactic variation in Old Occitan (and Old French) is the relative order of an infinitive and its core complement. Using the treebank, we can identify 20 cases in which the infinitive and its core complement (direct object, or directional complement of a motion verb) occur together: ten with the order CV and ten with the order VC. 18 out of these 20 cases are line-final (9 CV and 9 VC). In all of the CV cases, the infinitive takes an *-ar* ending, suggesting that assonance may have played a role in the selection of one word order over

another. Since stress on the penultimate syllable (paroxytonic stress) is excluded in line-final position, paroxytonic nouns (e.g. *ri.que.za*, l. 83; *chai.ti.ve.za*, l. 88) are only found in CV orders, while the paroxytonic infinitive (*metre*, l. 22, 59) is only found with VC order. It therefore seems possible that metrical factors (final stress, assonance) contribute to this syntactic variation, and so should be taken into consideration.

5 Conclusion

Having highlighted the importance of considering metrical factors in syntactic analysis, we outline an implemented, extensible methodology for creating a layer of metrical annotation and combining it with a treebank using the ANNIS platform. Our method is not applicable to only one text, nor even just to Old Occitan epic verse in general, but can be applied with few major modifications to texts from metrical tradition based primarily on a fixed number of syllables per line.

We demonstrate some preliminary findings from our pilot corpus in order to suggest future directions for linguistic research; however these are necessarily limited by the size of the corpus. More far-reaching conclusions may be drawn in particular from corpora combining verse and prose, in which the prose texts can be used to establish a 'baseline' of frequent syntactic structures to which the verse texts can be compared. Such an approach may help us to further our general understanding of the interaction of metrical constraints and syntactic variation.

Acknowledgements

T. M. Rainsford would like to acknowledge the generous support of the British Academy, through his recent post-doctoral fellowship at the University of Oxford, in making this research collaboration possible.

References

[1] Chambers, Frank M. (1985) *An Introduction to Old Provençal Versification*, Philadelphia: American Philosophical Society.

[2] Bouma, Gosse and Hermans, Ben (2013) Syllabification of Middle Dutch. In F. Mambrini, M. Passarotti, C. Sporleder (eds.) *Proceedings of the Second Workshop on annotation of Corpora for Research in the Humanities*.

[3] Delente, Éliane and Renault, Richard (2009) Les étapes du traitement automatique d'un poème. Presentation given at "Le patrimoine à l'ère du numérique", 10–11 December 2009, Université de Caen [http://www.crisco.unicaen.fr/verlaine/ressources/patrimoine_Caen.pdf].

[4] Devine, Andrew M., and Stephens, Laurence D. (1984) *Language and Meter: Resolution, Porson's Bridge, and their Prosodic Basis*, Chico, CA: Scholars Press.

[5] Dipper, Stefanie (2005). XML-based Stand-off Representation and Exploitation of Multi-Level Linguistic Annotation. In R. Eckstein, R. Tolsdorf (eds), *Proceedings of Berliner XML Tage*, pp. 39–50.

[6] Eythórsson, Þórhallur, Karlsson, Bjarki, and Sigurðardóttir, Sigríður Sæunn (2014) Greinir skáldskapar: A diachronic corpus of Icelandic poetic texts. In *Proceedings of LREC 2014: Workshop on Language Resources and Technologies for Processing and Linking Historical Documents and Archives – Deploying Linked Open Data in Cultural Heritage*, Reykjavík, Iceland, pp. 35–41.

[7] Ibrahim, Robert and Plecháč, Petr (2011) Toward Automatic Analysis of Czech Verse. In B. P. Scherr, J. Bailey, E. V. Kazartsev (eds.) *Formal Methods in Poetics*, Lüdenscheid, RAM, pp. 295–305.

[8] Gasparov, M. L. (1996) *A History of European Versification*, tr. by G. S. Smith and Marina Tarlinskaja, ed. by G. S. Smith with Leofranc Holford-Stevens, Oxford, Clarendon Press.

[9] Gerdes, Kim, Kahane, Sylvain and Pietrandrea, Paola (2012) Intonosyntactic data structures: The rhapsodie treebank of spoken French. In *Proceedings of the 6th Linguistic Annotation Workshop*, Jeju, republic of Korea, pp.85–94.

[10] Eckart, Kerstin, Riester, Arndt and Schweitzer, Katrin (2012) A Discourse Information Radio News Database for Linguistic Analysis. In C. Chiarcos, S. Nordhoff, S Hellmann (eds) *Linked Data in Linguistics. Representing and Connecting Language Data and Language Metadata* Springer, Heidelberg, pp. 65–75.

[11] Labelle, Marie (2007) Clausal architecture in Early Old French. *Lingua* 117: 289–316.

[12] Martineau, France, Hirschbühler, Paul, Kroch, Anthony and Morin, Yves Charles (2010) *Corpus MCVF annoté syntaxiquement*, Ottawa: University of Ottawa [http://www.arts.uottawa.ca/voies/corpus_pg_en.html].

[13] Plecháč, Petr and Ibrahim, Robert (2014) Database of Czech Verse, Presentation given at "Frontiers in Comparative Metrics 2", 19–20 May 2014, Tallinn.

[14] Rainsford, Thomas M. (2010) Rhythmic Change in the Medieval Octosyllable and the Development of Group Stress. In F. Neveu, V. Muni-Toke, T. Klingler, J. Durand, L. Mondada and S. Prévost (eds) *Congrès mondial de linguistique française: CMLF 2010*, Paris, Institut de linguistique française), pp. 321–36.

[15] Scrivner, Olga, Kübler, Sandra, Vance, Barbara, and Beuerlein, Eric (2013) Le Roman de Flamenca : An annotated corpus of old Occitan. In F. Mambrini, M. Passarotti, and C. Sporleder (eds), *Proceedings of the Third Workshop on Annotation of Corpora for Research in Humanities*, pp. 85–96.

[16] Zeldes, Amir, Ritz, J., Lüdeling, Anke, and Chiarcos, Christian (2009) ANNIS: A search tool for multi-layer annotated corpora. In *Proceedings of Corpus Linguistics*, Liverpool.

Cross-lingual Dependency Transfer with Harmonized Indian Language Treebanks

Loganathan Ramasamy and Zdeněk Žabokrtský

Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Charles University in Prague
E-mail: {ramasamy, zabokrtsky}@ufal.mff.cuni.cz

Abstract

One of the most important aspect of cross-lingual dependency transfer is how different annotation styles which often negatively influence the transfer accuracy are handled. The emerging trend is that the annotation style of different language treebanks can be harmonized into one style and the cumbersome manual transformation rules thus can be avoided. In this paper, we use harmonized treebanks (POS tagsets and dependency structures of original treebanks mapped to a common style) for inducing dependencies in a cross-lingual setting. We transfer dependencies using delexicalized parsers that use harmonized version of the original treebanks. We apply this approach to five Indian languages (Hindi, Urdu, Telugu, Bengali and Tamil) and show that best performance can be obtained in delexicalized parsing when the transfer takes place from Indian language (IL) to IL treebanks.

1 Introduction

Many languages including some of the popular languages according to the number of native language speakers do not have treebanks. The reasons for the absence of treebanks for languages that have sizeable population are mainly extraneous such as lack of natural language processing (NLP) research in those languages, research funding, etc. The availability of treebanks for these languages will have tremendous practical value given the fact that treebanks are useful in many practical applications such as machine translation (MT), relation extraction, and so on.

In the case of treebanking, research in recent years focuses on developing methodologies that could reduce years of manual annotation effort by automatically inducing parsers/treebanks for resource-poor languages with varying levels of success. Those methodologies can be broadly categorized

into: (i) unsupervised methods which can parse any language data without any annotated treebank for training, (ii) semi-supervised methods which make use of small amount of treebank annotation to bootstrap the annotation further, and (iii) cross-lingual syntactic annotation transfer through projection (or simply syntactic projection) of annotation from resource-rich source languages to resource-poor target languages. Depending on the resource availability, all three frameworks mentioned above can be combined in a hybrid fashion to obtain better parsing results albeit supervised parsing is most likely to give superior performance.

In this paper, we mainly focus on cross-lingual transfer-based techniques. Recent works on cross-lingual transfer-based techniques [8], [12] and [2] directly transfer the syntax from source language parsers via *delexicalization*, ideally without using any target language resources. *Delexicalized parsing* [15] is a method of using source language parser directly to parse target language sentences. This method requires only part of speech (POS) sequences of source side training data to train delexicalized parsers. It has been shown in [8] that training a parser with POS tags alone achieves parser accuracy comparable (UAS score of 82.5% for a delexicalized parser vs. UAS score of 89.3% for a parser trained with all features for English) to that of supervised parsers. Extending that to parser transfer, delexicalized transfer parsers [8] can give surprisingly better performance than state-of-the-art unsupervised parsers given that source and target languages use the same POS tagset.

It has been shown in earlier works such as [9, 7] that mapping the annotation to common annotation style helps various tasks including POS tagging, grammar induction and cross-lingual dependency transfer. [15] showed that harmonizing POS helps to induce parsers for closely related languages. To the best of our knowledge, this approach has not been explored for Indian languages (IL) which mostly border on under-resourced category except perhaps Hindi. In this paper, we use harmonized treebanks (POS tagsets and dependency structures of original treebanks mapped to a common style) for inducing dependencies for ILs in a cross-lingual setting. We transfer dependencies using delexicalized parsers that use harmonized version of the original treebanks. We use both non-IL and IL source treebank parsers to parse target ILs. In the results section, we particularly show that parsers trained on ILs can help other ILs.

Section 2 discusses the delexicalized parsing and the harmonization of IL treebanks to a common annotation style with respect to POS and dependencies. Section 3 presents the data and experimental settings. Section 4 presents delexicalized parsing results for ILs.

2 Delexicalized Parsing

In projection based approaches [4], the projected structures (as a result of projection alone or resulting from parsers trained on projected treebanks) for the target language sentences will have some systematic differences with respect to the target treebank annotation style since the projected structures follow source treebank annotation style. One must implement some transformation rules similar to [4] to overcome the drop in evaluation accuracy due to annotation differences. Other alternative is to map both source and target annotation styles to adhere to a common standard [14, 7].

Delexicalized parsing is the method of parsing target language sentences using a parser trained on a source language treebank. The parser trained on the source treebank does not use source wordforms (hence the name delexicalized), but relies mainly on POS tags and other non-lexical features associated with the source trees. Such a parser can be used to parse target language sentences provided target language sentences too are tagged with the same POS tagset as that of the one used in the source treebank. To address the annotation differences at dependency level, we adopt the strategy of using harmonized treebanks i.e., treebanks mapped to a common annotation style. The following subsections describe the IL treebank data and provide relevant information regarding the harmonization of IL treebanks under HamleDT [14], a large scale treebank harmonization project.

2.1 POS harmonization

One of the main requirements in delexicalized parsing is a common POS tagset. Common POS tagset is essential for both source side parser training and target side parsing. At present, there are two sets of treebanks that have common annotation scheme and are easily accessible: (i) treebanks mapped to the Prague Dependency Treebank (PDT) style annotation [14] via harmonization (also known as HamleDT treebanks) and (ii) treebanks based on universal dependency annotation [7]. HamleDT [14] harmonizes treebanks by mapping POS, dependency relations and dependency structure to PDT style annotation.

Lang.	Source	# Sentences		Tagset size		
		train	test	dep.	CPOSTAG	POSTAG
Bengali (`bn`)	ICON2010 [3]	979	150	42	14	21
Hindi (`hi`)	COLING2012 [3]	12041	1233	119	36	20
Tamil (`ta`)	TamilTB 1.0 [10]	480	120	25	12	465
Telugu (`te`)	ICON2010 [3]	1300	150	41	16	24
Urdu (`ur`)	UDT [1]	2808	313	71	15	29

Table 1: IL treebank statistics

HamleDT at present contains 30 existing treebanks harmonized into PDT style annotation as well as Stanford dependencies [11]. 4 out of 5 IL treebanks[1] shown in Table 1 except Urdu (ur)[2] treebank are already POS and dependency harmonized in HamleDT. Thus, we make use of the harmonized IL treebanks in HamleDT for our delexicalized parser experiments. POS harmonization in HamleDT is done as follows: the POS tagset of original treebanks are mapped to Interset features [13]. Interset features are detailed morphological features that any POS tag(set) can be mapped into, i.e., any tag is a list of morphological features (Interset). This conversion is done via tagset drivers. Tagset drivers are written for each treebank tagset in HamleDT. Captured Interset features for each wordform are then converted into positional tags - the style PDT uses for POS annotation.

Treebank	Orig. size	PDT fine	PDT coarse
Bengali (bn)	21	29	12
Hindi (hi)	36	344	12
Tamil (ta)	465	79	10
Telugu (te)	23	58	12
Urdu (ur)	29	10	10

Table 2: POS tagset size: original vs. harmonized

For all the IL treebanks, harmonized fine-grained PDT style positional tags obtained from Interset features are larger than the original POS tagset size. Since each fine-grained positional tag is a detailed morphological tag and each position represents a particular morphological phenomenon, it is easy to extract coarse-grained tags to tailor to the needs of various NLP tasks. We use the 1^{st} position of fine-grained positional tag as coarse-grained tag.

Table 2 shows the POS tagset size of original and harmonized IL treebanks (measured only using the training section of the treebanks in Table 1). POS tagset size of the original treebanks - {bn, hi, te, ur} are relatively lower than the Tamil (ta) treebank. *AnnCorra* is a POS/chunk tagset standard [3] for Indian languages. AnnCorra POS tagset is similar to PennTagset [5] in terms of tags (most of them are reused) and its com-

[1]Throughout the paper we refer IL treebanks using language names/codes (ISO) instead of their actual treebank names. For example, when we say 'Tamil (ta)' or 'ta' treebank, it refers to the actual treebank described in the 2nd column of Table 1.

[2]We harmonized the original Urdu (ur) treebank POS tagset to Interset features. We mapped only major POS information to Interset features, at the moment, the POS harmonization does not make use of features supplied by the Urdu (ur) treebank. Coarse/fine-grained PDT style tags can be obtained from Interset features.

[3]AnnCorra: POS tagging guidelines for Indian languages - http://ltrc.iiit.ac.in/tr031/posguidelines.pdf

pactness. Treebanks {bn, hi, te, ur} use AnnCorra scheme whereas Tamil (ta) uses PDT style positional tags for the treebank POS annotation. In addition to POS tags, AnnCorra also includes chunk level tags in the annotation. These tags (such as noun chunk, verb chunk, etc.) are included in treebanks {bn,te} as coarse-grained tags (CPOSTAG column in the CoNLL data) and in hi as one of the features (through chunkId, chunkType in FEATS column) in the CoNLL data. Table 2 also shows the tagset size of treebanks after harmonization (columns 3 and 4).

Harmonized fine-grained tagset size of the Hindi (hi) treebank is larger than {bn, te} treebanks. This could be attributed to a more detailed inclusion of morphological features in the Hindi (hi) treebank annotation in comparison to {bn, te} . Moreover, the Hindi (hi) treebank has been evolving and the treebank comes from 2012 shared task whereas both {bn, te} come from 2010 ICON shared task. For all harmonized treebanks, coarse-grained tagset size is similar to universal POS tagset [9]. Our coarse-grained tag inventory contains 12 tags in total: adjective (A), numeral (C), adverb (D), interjection (I), conjunction (J), noun (N), pronoun (P), verb (V), preposition (R), particle (T), unknown (X), and punctuation (Z).

2.2 Dependency harmonization

This section describes dependency harmonization of IL treebanks in HamleDT. Dependency harmonization is important for the reason that it can minimize errors that occur due to annotation differences at structural level during delexicalized transfer. For example, if one were to parse Hindi sentences using the delexicalized source parser trained on a Czech (cs) treebank, the parser would always make postpositions the head of postpositional phrases. But the Hindi (hi) treebank makes the noun argument the head of postpositional phrases. So, the evaluation would often underestimate this type of stylistic variations due to annotation differences. This type of errors can be avoided if treebanks follow same convention regarding annotations. Dependency harmonization in HamleDT is done includes mapping dependency relations to PDT analytical functions and transforming

The delexicalized transfer approach requires both source and target IL treebanks to be harmonized at both POS and structural level. IL treebanks (Table 1) except Urdu (ur) have already been harmonized in addition to dozens of other treebanks in HamleDT.

Table 3 compares original IL treebanks and harmonized IL treebanks. Evaluation with and without punctuation are shown under columns 3 and 2 respectively. The table shows that Hindi (hi) treebank is the most affected by harmonization at structural level.

Tamil (ta) treebank did not require too much harmonization since the treebank already follows Prague dependency style annotation for most of the syntactic phenomena. Harmonized Bengali (bn) and Telugu (te) treebanks

too didn't change from their original structures. At the moment, Urdu (ur) treebank has not been harmonized at the dependency level.

Lang.	No punc	With punc
Bengali (bn)	99.9	99.1
Hindi (hi)	58.0	56.3
Tamil (ta)	100.0	99.8
Telugu (te)	100.0	99.4
Urdu (ur)	-	-

Table 3: UAS scores between original and harmonized treebanks

3 Experiments

The main goal of this experiment is to parse ILs using other language parsers (delexicalized).

We consider left and right-branching trees as one of our baseline. We also provide supervised parser results for comparison, this could in fact indicate the upper limit that other cross-lingual approaches can aim for. Indian languages are head-final in many respects and have a tendency to be left-branching. So, one can expect the left-branching accuracy to be higher than the right-branching accuracy. As a second baseline, we train supervised parsers on the training section of the treebanks in Table 1. We train our models with MSTParser [6] (version-0.5.0). We train all our parsing models with 2^{nd} order, non-projective settings. We provide supervised parser results for both POS harmonized and POS+dependency harmonized version of the IL treebanks.

We perform delexicalized transfer experiment in two settings: (i) with POS harmonization and (ii) with POS and dependency harmonization.

In *POS harmonization* experiment, we use the source structure as it is from the native treebank annotation style, but we use coarse-grained harmonized POS tags i.e., only the 1^{st} position of the fine-grained tags. We then train a regular parser on the POS harmonized data, however, without source forms. For testing, we first tag the target test data and parse the target sentences with the trained model - again by removing target word-forms before the parsing step. It must also be noted that the target tagset should match the harmonized POS tagset; in our case - PDT tagset. Target tagging step can be done in a number of ways,

1. train harmonized POS tagger and tag target sentences directly with the harmonized POS tagger.

2. train a POS tagger with the native POS tagset, and perform harmonization after tagging the target sentences with the native POS tagger.

3. obtain harmonized POS tags for target sentences in an unsupervised manner, for example, by projecting the source tags onto the target—similar to [8].

The first two methods require annotated data (from target language) for training the POS tagger. The third method requires only annotated data in source language. We do target tagging according to (2).

For this experiment, we show results for parsing five ILs (target) using delexicalized parsers trained on 30 treebanks in HamleDT (source). We use POS harmonized treebanks to train delexicalized parsers. POS harmonized treebanks are obtained from HamleDT 2.0 [14] which is both POS and dependency harmonized to PDT style annotation. This experiment requires only harmonization of POS tags of the original treebanks. So we replace original POS tags with harmonized POS tags in the original treebanks. We use coarse-grained harmonized POS tag instead of full length PDT style tag.

Four (`{bn, hi, ta, te}`) out of five treebanks mentioned in Table 1 are harmonized in the HamleDT. However, for `ta` as source, we harmonize the current version of the Tamil dependency treebank (TamilTB 1.0) and train delexicalized parser on it. HamleDT 2.0 contains harmonized version of TamilTB.v0.1. This makes sense because Tamil (`ta`) treebank mentioned in Table 1 uses TamilTB 1.0 data. Urdu (`ur`) is not part of HamleDT yet, however, we do POS harmonization separately because Urdu (`ur`) test data needs to be POS harmonized before parsing.

In *POS+dependency harmonization* experiment, we train delexicalized parsers that are both POS and dependency harmonized. This experiment is similar to the POS only harmonization experiment, but it uses fully harmonized HamleDT 2.0 treebanks. We train delexicalized parsers on HamleDT 2.0 treebanks, again by replacing forms by coarse-grained harmonized POS tags. We harmonize target test data (POS and dependency structure) before parsing with delexicalized parsers. We do not have dependency harmonization for Urdu (`ur`). So, the results section do not include Urdu (`ur`).

4 Results

We provide baseline/supervised parsing results (Table 4) and delexicalized parsing results (Table 5, with ILs as source) for five ILs. All the results in this section show unlabeled attachment scores (UAS). We also trained delexicalized parsers from non-IL treebanks (Table 6) from HamleDT, but their scores were much lower than IL treebanks as source.

The average accuracy in **bold** for ILs in Table 5 indicate that the numbers are higher than the left/right baseline in Table 4. Within ILs, `{hi, ta}` seem to act as best source for each other, and similarly `{bn, te}` too act as best source for each other. Urdu (`ur`) benefits from `{bn, te}`.

Lang.	POS harmonized				POS+dep harmonized			
	left	right	pred.	gold	left	right	pred.	gold
bn	53.6	04.6	72.1	77.7	53.6	04.6	73.0	77.8
hi	24.4	27.3	76.0	78.5	53.3	07.7	75.8	78.4
ta	50.9	09.5	57.6	67.2	50.9	09.5	58.7	68.6
te	65.8	02.4	82.6	86.2	65.8	02.4	83.1	86.2
ur	42.9	06.3	-	-	-	-	-	-

Table 4: Left/right baseline and supervised parser results: POS harmonized vs. POS+dep harmonized. *pred.* - POS tags are obtained from a supervised tagger; *gold* - gold tags.

Source	bn		hi		ta		te		ur	
	D_P	D_{PD}	D_P	D_{PD}	D_P	D_{PD}	D_P	D_{PD}	D_P	D_{PD}
bn	-	-	27.8	33.1	34.8	36.1	78.6	78.2	58.6	-
hi	57.7	55.1	-	-	41.6	46.4	67.6	68.3	48.7	-
ta	57.3	55.9	34.2	61.7	-	-	69.1	69.6	42.6	-
te	63.9	62.4	21.9	24.1	22.5	26.1	-	-	53.9	-
avg	**59.6**	**57.8**	**28.0**	39.6	33	36.2	**71.8**	**72.0**	**51.0**	-

Table 5: Delexicalized parser results in the case of ILs as source. D_P - POS harmonization; D_{PD} - POS+dependency harmonization;

When we compare delexicalized transfer of all HamleDT treebanks to ILs and IL-IL delexicalized transfer i.e., `bn-hi, bn-ta, bn-te, bn-ur, hi-bn` and so on in Table 6 and 5, there's a discernible improvment in average transfer accuracy for IL-IL transfer. There's a big gain in the average accuracy (44.1% to 59.6% for `bn`, 26.5% to 33.0% for `ta` and so on) for all ILs except Hindi (`hi`) treebank for which the average accuracy drops from 30.2% to 28.0%. Hindi (`hi`) treebank does not seem to benefit from closely related source language treebanks, in fact, language isolate `eu` as source gives the best result (49.2%). In Table 5, parsers trained from `{eu, hu, la, tr}` are overall best source to parse ILs (each source crosses the left baseline for at least 3 target ILs). `{ro, de}` source parsers did not cross the left baseline for any of the target ILs. Most of the source languages (mostly Indo-European) did not cross the left baseline for most ILs.

In terms of language relatedness - Hindi, Urdu and Bengali belong to *Indo-Aryan* family, and Tamil and Telugu belong to *Dravidian* family. But the results are quite contradictory at least in identifying which language treebank is the best source for parsing the test data. Though the original treebanks of `{hi, bn, te, ur }` follow similar annotation styles, at the data level, the treebanks are quite different. For example, the average training sentence lengths of treebanks `{bn, te}` are 6.6 and 3.9 words, respectively.

Source	Family	bn		hi		ta		te		ur	
		D_P	D_{PD}	D_P	D_{PD}	D_P	D_{PD}	D_P	D_{PD}	D_P	D_{PD}
ar	Semitic	23.1	22.4	26.5	14.4	11.7	11.9	35.4	35.9	23.4	-
bg	IE	32.1	39.5	36.9	18.1	17.7	19	50.4	59.4	21.2	-
bn	IE	-	-	27.8	33.1	34.8	36.1	78.6	78.2	58.6	-
ca	IE	49.3	53.7	26.4	14.5	14.2	15	70.3	73.9	35.1	-
cs	IE	38.1	39.3	32.9	20.2	18	18.1	53.1	53.5	27.6	-
da	IE	35	40.8	30.1	28.4	23.2	33.9	40.6	45.9	35.8	-
de	IE	41.2	45.6	22.1	27.3	30.2	31.9	56.3	58.3	32.4	-
el	IE	39.8	41.4	28.5	17.2	27.3	27.1	57.2	58.2	33.5	-
en	IE	40.9	44.7	44.4	22.8	35.5	29.5	55.1	57.2	32.4	-
es	IE	48.7	50.7	26.6	15.3	15.5	16.4	65.4	74.7	35.2	-
et	Uralic	51.1	52	27	47.6	37.2	36.3	66.4	63.6	43	-
eu	L. Isolate	54.4	56.4	49.2	39.3	47.4	47.8	66.3	67.5	48.2	-
fa	IE	43.7	43.6	27.8	23.4	15.1	16.2	63.7	65.4	28.6	-
fi	Uralic	49.6	53.6	38.8	46.8	38.6	42	59.9	65.8	37.9	-
grc	IE	54.4	55.9	21.9	29	32.5	32.3	64.1	71.5	41.3	-
hi	IE	57.7	55.1	-	-	41.6	46.4	67.6	68.3	48.7	-
hu	Uralic	53.2	58.9	26.8	50	34.7	38.1	65.9	66.3	47	-
it	IE	39.8	44.7	25.8	11.7	16.1	16.4	57.2	59.9	24.6	-
ja	Japonic	53.6	55	24.1	50.7	43.5	44.2	68.3	70.7	37.1	-
la	IE	55.4	54.6	24.7	24.7	29.1	28.1	67.5	67.5	51.3	-
nl	IE	33.7	38.9	32.7	22.5	21.3	22.9	48.7	54.3	38.1	-
pt	IE	20.9	23.7	34.2	20.7	17.9	19.5	27.2	29	13.3	-
ro	IE	31.2	31.8	22.7	8.8	7.1	7.4	50.6	50.9	17.4	-
ru	IE	36.9	37.9	31.8	25.9	17.6	18.9	56.7	58.9	29.6	-
sk	IE	35.4	38.4	34.3	21	22.5	22.4	54.5	55.5	25.4	-
sl	IE	39.5	41.4	29.5	15.5	12.9	12.5	60.4	58.5	22.9	-
sv	IE	38.9	43.6	44.4	27.4	37.6	38.4	49.4	52.6	30.2	-
ta	Dravidian	57.3	55.9	34.2	61.7	-	-	69.1	69.6	42.6	-
te	Dravidian	63.9	62.4	21.9	24.1	22.5	26.1	-	-	53.9	-
tr	Altaic	59.6	59.2	22.2	51.1	45.5	46.4	71.7	71.2	45	-
avg		44.1	46.2	30.2	28	26.5	27.6	58.5	60.8	35.4	-

Table 6: Delexicalized parser results with harmonized HamleDT treebanks as source and ILs as target. *Source* - language code of the treebank, the actual treebank sources are described in [14]; D_P - POS harmonization; D_{PD} - POS+dependency harmonization; IE - Indo-European; L. isolate - language isolate;

Whereas the average training sentence length for Hindi (hi) treebank is 22.3 words. For {ta, ur} treebanks, the average training sentence lengths are 15.1 and 13.1, respectively. That means, the treebanks {bn, te} may not have sufficient representation of structures at the syntactic level to be efficient source languages for parsing Hindi sentences. This also partly explains why the treebanks {hi, ta} are best source for each other. But this reasoning does not apply well for Urdu (ur) because {bn, te} act as best sources than {hi, ta}.

5 Conclusion

In this paper we tried to induce dependencies for five Indian languages (not so resource-rich) using the cross-lingual approach known as delexicalized parsing. The main difference between the methodology we used in the paper and earlier approaches is in the use of harmonized treebanks. Harmonized treebanks not only facilitate the use of other language tools and resources in a cross-lingual setting (without the need to write transformation rules on a case-by-case basis), but will also help parsing evaluation based on a consistent annotation style. We compared our results mainly with left/right baseline owing to the strong left-branching preference of IL treebanks. The results show that best overall performance can be obtained in delexicalized parsing when the transfer takes place from IL to IL treebanks.

Acknowledgments

This work has been using language resources developed and/or stored and/or distributed by the LINDAT/CLARIN project of the Ministry of Education of the Czech Republic (project LM2010013).

References

[1] Riyaz Ahmad Bhat and Dipti Misra Sharma. Dependency treebank of urdu and its evaluation. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 157–165, Jeju, Republic of Korea, July 2012. Association for Computational Linguistics. 3

[2] Greg Durrett, Adam Pauls, and Dan Klein. Syntactic transfer using a bilingual lexicon. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11, Jeju Island, Korea, July 2012. Association for Computational Linguistics. 2

[3] Samar Husain, Prashanth Mannem, Bharat Ambati, and Phani Gadde. The ICON-2010 tools contest on Indian language dependency parsing.

In *Proceedings of ICON-2010 Tools Contest on Indian Language Dependency Parsing*, Kharagpur, India, 2010. 3

[4] Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325, September 2005. 3

[5] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: the penn treebank. *Comput. Linguist.*, 19:313–330, June 1993. 4

[6] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 91–98, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. 6

[7] Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. 2, 3

[8] Ryan McDonald, Slav Petrov, and Keith Hall. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 62–72, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. 2, 7

[9] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). 2, 5

[10] Loganathan Ramasamy and Zdeněk Žabokrtský. Tamil Dependency Parsing: Results Using Rule Based and Corpus Based Approaches. In *Proceedings of the 12th international conference on Computational linguistics and intelligent text processing - Volume Part I*, CICLing'11, pages 82–95, Berlin, Heidelberg, 2011. Springer-Verlag. 3

[11] Rudolf Rosa, Jan Mašek, David Mareček, Martin Popel, Daniel Zeman, and Zděněk Žabokrtský. Hamledt 2.0: Thirty dependency treebanks stanfordized. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA). 4

[12] Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487, Montréal, Canada, June 2012. Association for Computational Linguistics. 2

[13] Daniel Zeman. Reusable tagset conversion using tagset drivers. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Language Resources and Evaluation Conference, LREC 2008*, pages 28–30, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). 4

[14] Daniel Zeman, David Mareček, Martin Popel, Loganathan Ramasamy, Jan Štěpánek, Zděněk Žabokrtský, and Jan Hajič. Hamledt: To parse or not to parse? In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). 3, 7, 9

[15] Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *IJCNLP 2008 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India, 2008. Asian Federation of Natural Language Processing, International Institute of Information Technology. 2

POS Tagset Refinement for Linguistic Analysis and the Impact on Statistical Parsing

Ines Rehbein and Hagen Hirschmann

German Department
Potsdam University
E-mail: rehbein@uni-potsdam.de

German Studies and Linguistics
Humboldt University Berlin
hirschhx@hu-berlin.de

1 Introduction

The annotation of parts of speech (POS) in linguistically annotated corpora is a fundamental annotation layer which provides the basis for further syntactic analyses, and many NLP tools rely on POS information as input. However, most POS annotation schemes have been developed with written (newspaper) text in mind and thus do not carry over well to text from other domains and genres. Recent discussions have concentrated on the shortcomings of present POS annotation schemes with regard to their applicability to data from domains other than newspaper text.

For German, ongoing efforts [18, 14, 17, 5, 20, 19] discuss the restructuring of the Stuttgart-Tübingen Tagset (STTS) [15], which is a quasi standard for German POS tagging. While the discussion so far has focussed on the extension of the STTS to non-canonical data such as spoken language or user-generated content from the web, we put our attention on a different, but related matter, namely the restructuring of the tagset in order to provide us with a more detailed linguistic analysis of modification. At the same time, we are interested in the impact of the tagset refinements on the accuracy of NLP tools which rely on POS as input, in particular of syntactic parsers.

Recent work investigating the impact of POS annotation schemes on parsing has yielded mixed results [8, 3, 7, 9, 13]. While a preliminary study on providing the parser with more fine-grained *gold* annotations gave proof-of-concept that, at least for German dependency parsing, a linguistically motivated distinction of modifier tags can indeed improve parsing results [13], it still has to be shown that these results carry over to larger data sets and to a real-world scenario where the parser is provided with automatically predicted POS.

In the paper, we fill this gap and present experiments on dependency and constituency parsing of German with a more fine-grained and syntactically motivated tagset for modifier relations. We explore whether the new modifier distinctions can be automatically predicted with an accuracy that is good enough to increase pars-

ing accuracy. Our results show a modest, but statictically significant improvement when training the parsers on the modified tagset.

The paper is structured as follows. In Section 3, we briefly describe the new tag distinction and report on our efforts to improve POS tagging results on the new POS tags. In Section 4, we present parsing experiments investigating the impact of the different POS distinctions on the accuracy of statistical parsers. We discuss our results and put them into context in Section 5, and finally conclude in Section 6.

2 Tagset Refinements

The classification in the standard part of speech tagset for German, the STTS, is based on very heterogeneous criteria – some definitions refer to the word's inflectional status, some to its syntactic status, some to semantic or to purely lexical classes. The open word class ADV (adverb) can be described as a residual category where adverbs are defined as modifiers of verbs, adjectives, adverbs, or clauses, which are not derived from adjectives (STTS guidelines, p. 56). Since there are other parts of speech that can also modify each of these heads (e.g. modal particles, regular particles, pronominal adverbs, and ordinals), this definition is not sufficient.

We thus propose a more fine-grained subcategorisation of the residual class ADV in the STTS tagset which distinguishes between a) "real" adverbs (ADV), b) modal particles (MODP), c) focus particles (PTKFO), d) intensifiers (PTKINT), and e) lexical particles (PTKLEX). These classes are defined from a *functional syntactic* perspective, which does not include semantic classes like temporal or manner adverbs that are specific semantic subcategories of the class ADV. Furthermore, we redefine the dissociation of adverbs (ADV) and adjectives (ADJD), which –according to the STTS– is based on the criterium of *inflectibility*, in favour of a syntactically motivated notion of lexical modifiers (for a more detailed discussion of the new tag distinctions see [6, 13]).

As an example, consider Sentence 145 from the TIGER treebank [2] (Figure 1). In the original Tiger POS annotation (ORIG) which follows the STTS, the four lexical modifiers "etwa" (for instance), "so" (as), "stark" (strong), "allgemein" (generally) are described by the tags ADV (adverb: "etwa", "so") and ADJD (predicative adjective: "stark", "allgemein"). This distinction is motivated morphologically – "etwa" and "so" cannot be inflected in German, whereas "stark" and "allgemein" can. The POS tags for "etwa", "so", "stark", and "allgemein" do not express any syntactic differences between the words. Furthermore, most grammarians will question the analysis of "etwa" and "so" as adverbs in this particular context.

Compare the original tags to the new tag distinctions (NEW) in Figure 1 which show a (more) syntactically motivated POS analysis of the same lexical items. In the case of "etwa" and "so", new POS tags (PTKFO and PTKINT) have been introduced which reflect the syntactic status of the respective words. The POS tag

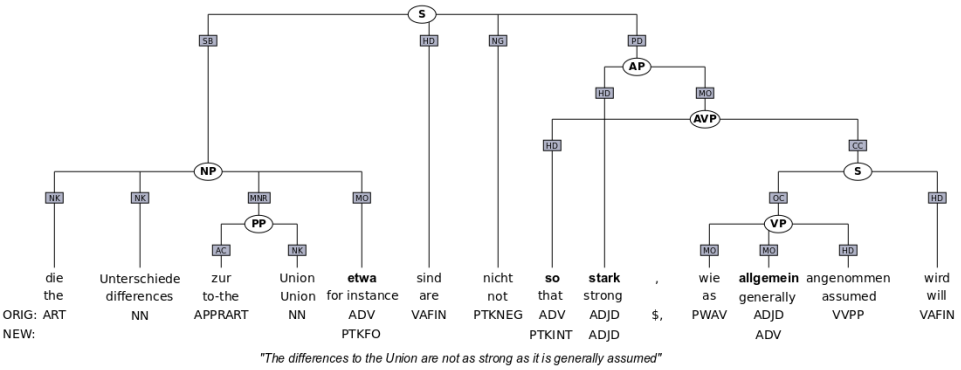


Figure 1: Example tree from TIGER illustrating four different modifers (PTKFO, PTKINT, ADJD, ADV) in the original STTS and in the new classification scheme

ADJD for "stark" is the same as in the original STTS anlysis, because "stark" is used as a predicative adjective in the given context. The POS tag for "allgemein", however, deviates from the original STTS analysis because of the adverbial function of "allgemein" in this context, which is reflected by assigning it the ADV tag in the new classification scheme.

In conclusion, our new classification provides (four) new POS tags and redefines (two) existing categories of the original STTS in order to restructure the part of speech analysis of modifying words in the STTS in a principled, syntactically motivated way. Table 1 gives an overview over the categories that are conceptually altered or newly implemented in the new tagset.

3 POS tagging experiments

3.1 Data

The data we use in our experiments are the first 10.000 sentences from the TIGER treebank [2], reannotated according to the new modifier classification. As these distinctions are sometimes ambiguous, we cannot expect the same inter-annotator agreement as for the original STTS modifier tags.[1] However, as shown in [13], even POS annotations with a lower IAA are able to provide a statictical parser with useful information. The question which remains to be answered, is whether these tag distinctions can also be learned reliably by NLP tools to achieve the same effect.

After annotating a goldstandard with 1.000 sentences randomly extracted from TIGER, the first 10.000 sentences of the treebank have been relabelled semi-automatically, using manually defined patters which make use of the syntactic information in the treebank as well as the STTS tags and the lexical forms. The automatically predicted new tags were then checked manually by one expert annotator.

[1] In an annotation study with two human coders, we obtained an inter-annotator agreement of 0.838 (Fleiss' κ) on newspaper text for the new modifier distinctions [13].

Tag	Meaning	Description	Restriction/Test
ADV	Adverb in a syntactic sense	Modifier of verbs or clauses	Can appear in the prefield
Example:			*Sie läuft* **schnell** – *Sie läuft* **glücklicherweise** she runs quickly – she runs fortunately "She runs quickly" – "Fortunately, she runs"
ADJD	Predicative adjective	Complement of a copula verb	Can appear in the prefield
Example:			*Die Läuferin ist* **schnell** the runner is quick "The runner is quick"
MODP	Modal particle	Clausal modifier	Cannot appear in the prefield
Example:			*Sie läuft* **ja** *bereits* she runs PTC already "She is already running" **[as it is well known]**
PTKFO	Focus particle	Associated with a focus element, modifying a set of alternatives	Cannot appear on their own in the prefield
Example:			**Auch** *sie läuft schnell* also she runs quickly "She runs quickly, too"
PTKINT	Intensifier	Intensifying or quantifying a gradable expression	Cannot appear on their own in the prefield
Example:			*Sie läuft* **sehr** *schnell* she runs very quickly "She runs very quickly"
PTKLEX	Part of a multiword expression	Particle which cannot be analysed compositionally	cannot appear on their own in the prefield
Example:			*Sie läuft* **immer** *noch* she runs always still "She is still running"

Table 1: Overview of the new tag distinctions and examples for each tag

Please note that we evaluate POS accuracy on the 1.000 sentences which have been reannotated from scratch for POS by two annotators, while we use the whole 10.000 sentences in a 10-fold cross-validation setting for evaluating parsing accuracy. As we do not include the POS in the evaluation of parsing results, potential POS errors in the semi-automatically annotated data should not influence the parser evaluation.

3.2 Setup

Our POS tagger is similar to the FLORS tagger [16] and makes use of 4 different feature types: a) shape features, b) prefix/suffix features, c) context features, and d) distributional features. We also use the linear L2-regularized L2-loss SVM implementation provided by LIBLINEAR [4] to train a one-vs-all classifier for each POS in the training set.[2] In contrast to [16], we also include POS context features from POS tags predicted by the Hunpos tagger trained on the original STTS tags.[3]

3.3 POS tagging baseline

As our baseline, we train the Hunpos tagger on sentences 1-9.500 from TIGER (excluding the sentences which are part of the goldstandard; the last 500 sentences have been held out as development data) and evaluate on the randomly extracted, manually annotated goldstandard (1.000 sentences). Table 2 shows results for different taggers on the original STTS (orig) and on the new classification (new).

tagger	**setting**	**acc.**
baseline1 (Hunpos)	orig tags	96.11
baseline2 (Hunpos)	new tags	94.78
own tagger (w/o POS context)	new tags	94.91
own tagger (with POS context)	new tags	**96.68**

Table 2: POS tagging accuracy for the Hunpos tagger and for our own tagger (without and with POS context features) on the gold standard (500 sentences)

Not surprisingly, the baseline tagging accuracy on the new tagset is lower than the one on the more coarse-grained STTS distinctions. The Hunpos tagger, trained on the original tags, achieves an accuracy of 96.11% while the accuracy of the same tagger on the new tagset is more than 1% lower at 94.78%. Our own tagger achieves a slightly higher accuracy of 94.91% on the new tags when using word-form context only, and a considerably higher accuracy of 96.68% when also using context features based on the original STTS tags predicted by Hunpos.

Table 3 shows results for individual tags, evaluated on the larger data set (10.000 sentences) in a 10-fold cross-validation setting. We can see that precision for most

[2]This amounts to 52 POS for the original STTS and to 56 POS for the modified annotation scheme.

[3]The Hunpos tagger is an open source reimplementation of the TnT tagger (`https://code.google.com/p/Hunpos`)

of the tags is considerably higher than recall. Exceptions are the two most frequent classes ADV and PTKFO. This is not surprising as ML methods are known to have a bias towards the more frequent classes and tend to overuse them.

TAG	prec.	rec.	f-score	row counts
ADV	87.25	89.76	88.49	(6276/6992)
PTKFO	82.32	84.73	83.51	(1243/1467)
ADJD	81.10	75.46	78.18	(824/1092)
PTKINT	80.81	72.87	76.63	(779/1069)
MODP	85.37	69.31	76.51	(70/101)
PTKLEX	81.87	67.62	74.07	(307/454)

Table 3: POS tagging accuracy on the larger data set (10.000 sentences)

The overall accuracy of our tagger on all 58 tag distinctions on the larger data set is 97.0%. This is slightly higher than the accuracy of the Hunpos tagger on the same data (96.4%), using the original STTS tag distinctions.

Whether or not a precision in the range of 80-87% and f-scores between 74% and 88% are good enough to be used in linguistic analyses is hard to answer and certainly depends on the research question. We would like to argue that the additional information provided by the new tag distinctions is useful, and that the new tags, even if not perfect, can at least be used to extract candidates for linguistic analysis. Furthermore, the new classes MODP, PTKFO, PTKINT and PTKLEX can easily be subsumed under the ADV class, so no information is lost.

4 Impact of modifier distinctions on statistical parsing

To find out whether the more fine-grained modifier distinctions are able to improve parsing accuracy when predicted automatically, we use the new tags as input for training two statictical parsers. The first parser is the Berkeley parser [10], a PCFG-LA constituency parser, and the second one the MATE parser [1], a transition-based dependency parser. Both systems are language-agnostic.

4.1 Impact on constituency parsing

Table 4 gives results for constituency parsing when training the parser on the original STTS tags (orig) and on the new tags (new). We can see a modest improvement of 0.3% f-score for the new tag distinctions over all folds. When including the grammatical function labels in the evaluation, the average improvement is 0.2.

These results are for letting the parser assign its own POS tags. When providing it with the POS tags assigned by our tagger, results are similar with an average f-score of 75.26 for the original STTS tags and a slightly higher f-score of 75.54 (excluding grammatical functions) for the new tag distinctions. The difference in f-scores between the original STTS POS tags and the new tags is statictically

significant with $p = 0.025$.[4] While the improvements are small, they do show that our new tag distinctions do not hurt parsing accuracy and might even have the potential to improve it.

	original STTS			*new tags*		
	rec	**prec**	**fscore**	**rec**	**prec**	**fscore**
fold 1-10 (avg.)	75.38	75.12	75.25	**75.45**	**75.64**	**75.55**

Table 4: Parseval results (10fold cross-validation excluding grammatical functions)

One drawback of using the Berkeley parser in our experiments is that even when provided with "gold" POS tags, the parser, when it cannot find a good analysis for the prelabelled tags, takes the liberty to reject them and reassing its own POS. Also, the Berkeley parser does not take the tags as they are but, during training, refines the annotations by applying merging and splitting operations to the nodes in the tree, and only keeps those labels which have been shown to be useful during training. By just looking at the parsing results, we do not know what the internal representation used by the parser after the training cycles looked like.

In the next section, we turn to dependency parsing, which provides us with a more straight-forward way to compare the influence of different POS tagset distinctions on syntactic parsing.

4.2 Impact on dependency parsing

In the next experiments, we use the CoNLL conversion of the same 10.000 TIGER sentences to train the MATE parser. First, we replicate the experiment of [13] on a larger data set and use the gold tags (original STTS and new classification) for training. We find a small improvement of around 0.3 (UAS) and around 0.4 (LAS) when providing the parser with the new tags (Table 5). The results are consistent with the ones of [13] obtained on a smaller data set.

fold 1-10 (avg.)	*UAS orig.*	*new*	*LAS orig.*	*new*
gold POS	91.88	**92.23**	90.02	**90.46**
pred POS	89.68	**89.81**	86.94	**87.13**

Table 5: Unlabelled and labelled attachment scores for gold / predicted POS

Next, we test the parser on automatically predicted POS tags. The training data was annotated using 10-fold jackknifing. For the original STTS we used POS tags predicted by the MATE tagger, for the new classification we provided the parser with the tags predicted by the POS tagger described in Section 3.2.

While the improvements are smaller than for the gold tags, the difference is still statictically significant with $p = 0.002$ (LAS). When looking at f-scores for

[4]For significance testing, we used Dan Bikel's Randomized Parsing Evaluation Comparator with 10.000 iterations.

identifying specific dependencies and their attachments,[5] we observe improved results for 23 (out of 40) dependencies, the same results on 3 dependencies, and lower scores on the remaining 14 dependency relations (Table 6). Amongst the ones where we obtain improved results are not only the most frequent modifier relations (MO, MNR), but also the core grammatical functions (SB: subject, +0.7%; OA: direct object, +0.5%; DA: indirect object, +2.5%). We thus argue that, despite the improvements in f-score being small, the parse trees we obtain when training the parser on the new tag distinctions are of a higher quality as we achieve higher f-scores for identifying the arguments of a sentence.

As expected, we can see that the parser benefits from the new tag distinctions when parsing modifier relations. Figure 2 shows the parser output tree for the MATE parser when provided with the original STTS tags, and the parse tree triggered by the new tags. The POS tagger correctly predicted the two more fine-grained new tags for ”auch“ (also) (PTKFO) and ”gegenwärtig“ (at present) (ADV), which helped the parser trained on the new tags to correctly identify the low attachment for PTKFO, while the original STTS tag ADV incorrectly triggers high attachment for ”auch“ (dotted-red arrow). For ”gegenwärtig“, the redefined ADV tag in the new scheme again results in the correct attachment decision, while the same parser trained on the original STTS is only provided with the underspecified ADV tag and thus again produces the wrong analysis.

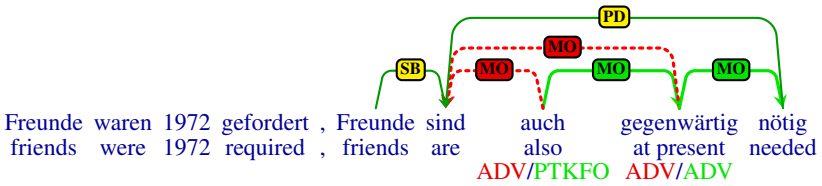


Figure 2: Parser output tree for orig (dotted-red) and new tags (green)

For predicate (PD) dependencies, however, f-scores for identifying the correct dependency label and attachment site are below the ones obtained by a parser trained on the original STTS tags. This is due to the low accuracy of the POS tagger on the ADJD part of speech tag. The distinction between adverbially used adjectives and predicative adjectives in the new tagset is difficult for the tagger which has only access to local information. For the STTS distinctions, the tagger can rely on word form information which, in many cases, in enough to identify the ”correct“ tag.

When providing the parser with gold POS tags, we observe an improvement in f-score of 1.7% for the PD dependency, from 76.7% for the original STTS tags up to 78.4% f-score when training the parser on the new modifier tags. In future work, we will try to improve the tagger by adding linguistically motivated features which might help to increase the tagging accuracy especially for predicative adjectives.

[5]For the evaluation we used a slightly modified version of the CoNLL07 evaluation script provided by `http://pauillac.inria.fr/~seddah/eval07.pl`.

DEPREL	freq.	orig f-score	new f-score	+ (%)	= (%)	- (%)
AC	172	78.3	79.0	0.7		
ADC	4	75.0	85.7	10.7		
AG	4929	92.9	93.0	0.1		
AMS	113	76.2	73.6			-2.6
APP	796	55.9	56.5	0.6		
AVC	11	50.0	42.9			-7.1
CC	446	61.0	59.3			-1.7
CD	3991	85.1	85.1		0.0	
CJ	5668	80.6	80.6		0.0	
CM	480	77.1	75.2			-1.9
CP	1759	90.8	92.9	2.1		
CVC	172	52.8	48.5			-4.3
DA	1045	57.1	59.6	2.5		
DM	16	55.2	66.7	11.5		
EP	377	77.9	78.6	0.7		
JU	320	86.4	87.0	0.6		
MNR	5227	67.0	67.5	0.5		
MO	22378	74.9	75.2	0.3		
NG	1097	76.0	76.1	0.1		
NK	55439	97.4	97.5	0.1		
NMC	552	96.4	96.1			-0.3
OA	6678	79.7	80.2	0.5		
OA2	5	0.0	0.0		0.0	
OC	8133	86.9	88.1	1.2		
OG	29	5.0	0.0			-5.0
OP	1597	51.5	51.7	0.2		
PAR	456	44.1	43.6			-0.5
PD	1888	73.8	73.2			-0.6
PG	613	80.1	79.3			-0.8
PH	21	38.7	36.4			-2.3
PM	989	98.9	98.8			-0.1
PNC	2217	89.8	89.7			-0.1
RC	1395	66.4	68.4	2.0		
RE	597	68.1	68.8	0.7		
RS	78	21.8	29.1	7.3		
SB	13065	86.4	87.1	0.7		
SBP	359	76.5	74.7			-1.8
SVP	1045	91.7	90.6			-1.1
UC	71	11.0	12.9	1.9		
VO	15	0.0	11.1	11.1		

Table 6: Results for specific dependency relations + attachment

5 Discussion

Our experiments have shown that more fine-grained, linguistically motivated POS distinctions can, at least to a small extent, improve results of a data-driven statistical parser.

There is related work by Maier et al. [9] who compare the impact of three different POS tagsets on constituency parsing results. The tagsets they use are the coarse-grained Universal Tagset (UTS) [11] which distinguishes 12 tags, the STTS (54 tags), and a fine-grained tagset enriched with morphological information ($>$ 700 tags). They also use the Berkeley parser in their experiments, which always obtained best results when trained on the STTS tagset, no matter if the POS tags were i) gold tags, ii) predicted by a HMM tagger, or iii) assigned by the parser itself. Surprisingly, the results for using the coarse-grained UTS were only slightly lower when provided as gold tags or learned by the parser. The fine-grained morphological tagset, however, proved to be too sparse and resulted in a substantial decrease in f-score. Maier et al. [9] did not modify the STTS, and only report results for constituency parsing. It would be interesting to see the impact of the UTS on dependency parsing, as it might be the case that the Berkeley parser can cope with the underspecified tags only because it applies its own refinement techniques to the annotation scheme during training.

Another relevant study is the work by Plank et al. [12] who discuss the problem of ambiguity caused by unreliable POS annotations by human coders. They show that incorporating annotator disagreements into the loss function of the POS tagger can improve results for POS tagging as well as increase the accuracy of a syntactic chunker that uses the POS tags as input. Their study can be described as complementary to ours. While we try to reduce the ambiguity in the data by refining the tagset and augmenting it with new information, their approach is to incorporate the ambiguity directly in the tagging model. In future work, it might be interesting to combine both approaches.

6 Conclusions

In the paper, we argued for a new classification of modifier distinctions in the STTS, the standard German POS tagset, which overcomes the drawbacks of the residual category ADV in the original tagset by providing more fine-grained and syntactically well motivated tag distinctions. The new tagset not only supports a more detailed linguistic analysis, it also has the potential to improve the accuracy of statistical parsers. We showed that even for automatically predicted POS tags we obtained a small, but significant improvement over the original STTS. As these improvements concern the core grammatical functions, we argue that the new modifier classification not only leads to modest improvements in parsing accuracy but, more importantly, also to a qualitatively improved syntactic analysis.

References

[1] Bernd Bohnet. Top accuracy and fast dependency parsing is not a contradiction. In *The 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China, 2010.

[2] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories*, pages 24–42, 2002.

[3] Markus Dickinson. An investigation into improving part-of-speech tagging. In *Proceedings of the Third Midwest Computational Linguistics Colloquium (MCLC-06)*, Urbana-Champaign, IL, 2006.

[4] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[5] Ulrich Heid und Kathrin Beck Heike Zinsmeister. Das Stuttgart-Tübingen Tagset – Stand und Perspektiven. 28(1), 2013.

[6] Hagen Hirschmann. Richtlinien zur Wortartenannotation von Adverb- und Partikelklassen – eine Granularisierung des STTS im Bereich von Modifikatoren. Technical report, Humboldt-Universität zu Berlin, 2014.

[7] Sandra Kübler and Wolfgang Maier. Über den Einfluss von Part-of-Speech-Tags auf Parsing-Ergebnisse. *Journal for Language Technology and Computational Linguistics*, 1(28):17–44, 2014.

[8] Andrew MacKinlay and Timothy Baldwin. Pos tagging with a more informative tagset. In *Proceedings of the Australasian Language Technology Workshop 2005*, pages 40–48, Sydney, Australia, 2005.

[9] Wolfgang Maier, Sandra Kübler, Daniel Dakota, and Daniel Whyatt. Parsing German: How much morphology do we need? In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, Dublin, Ireland, 2014.

[10] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *The 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL-COLING)*, 2006.

[11] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *The 8th International Conference on Language Resources and Evaluation (LREC-12)*, May 2012.

[12] Barbara Plank, Dirk Hovy, and Anders Søgaard. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden, 2014.

[13] Ines Rehbein and Hagen Hirschmann. Towards a syntactically motivated analysis of modifiers in german. In *Conference on Natural Language Processing (KONVENS)*, Hildesheim, Germany, 2014.

[14] Ines Rehbein and Sören Schalowski. STTS goes Kiez – Experiments on Annotating and Tagging Urban Youth Language. *Journal for Language Technology and Computational Linguistics*, 1(28):199–227, 2014.

[15] Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart, Universität Tübingen, 1999.

[16] Tobias Schnabel and Hinrich Schütze. FLORS: Fast and Simple Domain Adaptation for Part-of-Speech Tagging. *Transactions of the Association for Computational Linguistics (TACL)*, 2:15–26, February 2014.

[17] Thomas Schmidt Swantje Westpfahl. POS für(s) FOLK – Part of Speech Tagging des Forschungs- und Lehrkorpus Gesprochenes Deutsch. *JLCL*, 28(1):139–153, 2013.

[18] Angelika Storrer Thomas Bartz, Michael Beißwenger. Optimierung des Stuttgart-Tübingen-Tagset für die linguistische Annotation von Korpora zur internetbasierten Kommunikation: Phänomene, Herausforderungen, Erweiterungsvorschläge. *JLCL*, 28(1):157–198, 2013.

[19] Swantje Westpfahl. STTS 2.0? Improving the Tagset for the Part-of-Speech-Tagging of German Spoken Data. In *Proceedings of LAW VIII - The 8th Linguistic Annotation Workshop*, pages 1–10, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University.

[20] Heike Zinsmeister, Ulrich Heid, and Kathrin Beck. Adapting a part-of-speech tagset to non-standard text: The case of STTS. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pages 4097–4104, 2014.

Semi-Automatic Deep Syntactic Annotations of the French Treebank

Corentin Ribeyre° Marie Candito° Djamé Seddah◇
°Univ. Paris Diderot, Sorbonne Paris Cité, Alpage, INRIA
◇ Université Paris Sorbonne, Alpage, INRIA
firstname.lastname@inria.fr

Abstract

We describe and evaluate the semi-automatic addition of a deep syntactic layer to the French Treebank (Abeillé and Barrier [1]), using an existing scheme (Candito et al. [6]). While some rare or highly ambiguous deep phenomena are handled manually, the remainings are derived using a graph-rewriting system (Ribeyre et al. [22]). Although not manually corrected, we think the resulting Deep Representations can pave the way for the emergence of deep syntactic parsers for French.

Introduction

Syntactic parsing has been the focus of intensive international research over the last decades, leading to current state-of-the-art parsers to provide quite high performance (on well-formed English text at least). However, extracting more semantically-oriented information from syntactic parses, be them of high quality, is as not straightforward, given the abundant syntax-semantic divergences and the idiosyncratic nature of syntax itself. "Deep syntax" is generally intended as an intermediate representation between what is actually observable (surface syntax) and a semantic representation, which abstracts away from *syntactic variation*, such as diathesis alternations or non-canonical word order, and which can thus serve as an easier basis for semantic analysis. Such view forms, for example, the basis of the Meaning-Text Theory, MTT, (Melčuk [16]).

Several initiatives have been proposed to obtain "deep" syntactic treebanks, with various meanings attached to the term "deep". For instance for Spanish, the AnCORA-UPF multi-layer corpus (Mille et al. [17]) includes a deep syntactic layer, inspired by the MTT. For English, the Penn Treebank (PTB, (Marcus et al. [15])) contains a certain amount of "deep" annotations (such as traces for subjects of infinitives, long-distance dependencies and so on). Initially encoded with traces and co-indexes through constituent structures, the processing and recovery of these phenomena entailed complicated algorithms and methods. Nevertheless, the emergence of various conversion algorithms and enrichment processes from

the PTB phrase structures to deep syntax representation (e.g LFG F-Structures as in (Cahill et al. [4]), HPSG feature structures (Miyao et al. [18]), or CCG complex lexical types and derivations (Hockenmaier and Steedman [13])) have made these complex syntactic phenomenon more straightforwardly available.

Recently, more semantically oriented "Deep" treebanks have been made available (Čmejrek et al. [7], Flickinger et al. [10]) and their use was popularized through the Semeval 2014 broad semantic shared task (Oepen et al. [20]) which simplified these data set by providing mostly graph-based predicate-argument structure instances of these treebanks (Miyao et al. [19]). It worth noting that providing access to different representation layers of the same source, each having a different degree of granularity in term of syntax-to-semantic interface was, among others such as the MTT, formalized through the Prague Dependency Bank line of work (Böhmová et al. [3], Hajic et al. [12]). Inspired by the LFG theory, the various Stanford dependency schemes (De Marneffe and Manning [8], de Marneffe et al. [9]) are also a milestone in making deep syntax structures easily processable for further downstream semantic processing.

For French, which is the language we focus on, the annotations of the largest treebank available for French (the French Treebank (Abeillé and Barrier [1]), hereafter FTB) are surface-only. However, earlier attempts at deriving deeper representations were carried out by Schluter and Van Genabith [25] within a treebank-based LFG framework, using an heavily modified subset of the initial FTB release. Focusing on delivering a free data set based on structures as close as possible from the current FTB, Candito et al. [6] have defined a deep dependency syntactic annotation scheme for French, and added a deep syntactic layer to the Sequoia Treebank (Candito and Seddah [5]), a freely available corpus, made of 3,099 sentences . Although this resource can be used to train statistical deep parsers for French, we anticipate that its size will be insufficient to train accurate models given the additional complexity of deep syntax with respect to surface syntax.[1] We have thus undertaken to semi-automatically annotate the FTB with deep syntactic annotations, leading to a "silver" deep treebank of 18,500 sentences.

In the following, we start by describing the Deep Syntactic Representations (hereafter DSRs) of (Candito et al. [6]) in section 1, and the methodology used to obtain such representations for the sentences of the FTB. Section 3 is devoted to the tool we designed to convert surface dependency trees into such deep syntactic representations: we describe both the graph-rewriting system (section 3.1) and the hand-craft rules (section 3.2). We provide an evaluation of the DSRs obtained using this tool in section 4, and conclude.

[1] As shown by the mixed level of performance obtained by Ballesteros et al. [2] on a comparable parsing task for Spanish.

1 Target Deep Representations

In order to describe the Deep Syntactic Representations (DSRs) that we target, we sum up their description by Candito et al. [6]. As mentioned in the introduction, deep representations are intended to be an intermediary step between surface syntax and semantic representations. The DSRs make explicit three major types of information with respect to the surface representations:

- First, DSRs make explicit the deep syntactic arguments of verbs and adjectives and "subjects" of adjectives (predicates with other part-of-speech are left for future work). The deep syntactic arguments include those arguments that are syntactically dependent of another head (e.g. the subject of infinitival verbs) or that appear as the surface governor of the predicate (e.g. in the case of an attributive participle: *des personnes parlant italien ((some) people speaking italian)*).
- Second, following Relational Grammar (Perlmutter [21]), predicates are taken to subcategorize for dependents with certain *canonical* grammatical functions, potentially different from their effective *final* functions. The deep arcs are thus labeled with both canonical and final functions (at least for the grammatical functions that can be involved in syntactic alternations). For instance in Figure 1, while *Jean* is both the final and canonical subject of *semble*, it is the final subject and canonical object of the passive form *respecté* (written with a *suj:obj* label).
- Third, the semantically-void tokens are discarded, and dependencies coming in or out from these tokens are shifted to semantically full tokens (e.g. semantically void prepositions or complementizers are bypassed, auxiliaries are discarded and replaced by features on full verbs).

In order to capture syntactic alternations, DSRs make use of the distinction between *canonical* grammatical function (canonical GF) and *final* grammatical function (final GF)[2], and between *canonical subcategorization frames* (canonical SF) and *final subcategorization frames* (final SF). The final SF of an occurrence of a verb is defined as the list of GFs associated to its expressed arguments, plus the GFs that would be associated with the linguistic expressions that would appear as argument, if the verb were used in finite mode and in a non elliptical construction. This formulation accounts for the subject of infinitives, the subject of coordinated verbs or more generally any argument shared by several predicates. For instance, in Figure 1, the final SF both for *compter (to matter)* and for *respecté (respected)* is [subject=*Jean*].

The *deep* syntactic arguments of a verb are defined as the set of linguistic expressions that bear a final GF with respect to that verb, and that are not semantically empty. Syntactic alternations are viewed as redistributions of the grammatical functions associated to the syntactic arguments. Following Relational Gram-

[2]We use the term *canonical* instead of the Relational Grammar term *initial*.

mar (Perlmutter [21]), the final SF is considered as resulting from the application of 0 to n redistributions to a canonical SF. A simple case is for instance a passive occurrence of a transitive verb: the final SF is [subject, (by-object)] while the corresponding canonical SF is [objet, (subject)]. So for instance in Figure 1, the canonical SF of *respecté* is [object=*Jean*]. This is shown in the figure with double labels on the arcs of the form *final_function:canonical_function* (hence the label *suj:obj* between *Jean* and *respecté*).

Candito et al. [6] only considered redistributions that are morpho-syntactically marked (for instance with an auxiliary for passives, or a void reflexive clitic *se* for middle or neuter alternations). Unmarked redistributions are not accounted for, because disambiguating them resorts to semantic analysis. For instance, for the verb *couler ('to sink')*, the non-marked causative/inchoative alternation gives rise to two canonical SFs: the two constructions *X coule Y (X sinks Y)* and *Y coule (Y sinks)* are not related in the deep syntactic representation. They get the two distinct canonical SF [subject, object] and [subject] respectively, and for both occurrences, the canonical SF is identical to the final SF. On the contrary, the neuter and middle alternations, which are marked by a void reflexive clitic *se*, are represented using redistributions. For instance, for both *(Paul cassa la vase) Paul broke the vase* and *le vase se brisa* (litt. *the vase SE broke* for *the vase broke*), *vase* is canonical object.

From the formal point of view, the DSRs are graphs, whose nodes are the non-void tokens of the sentence. The arcs are labeled using the canonical functions (hence for instance, the *suj:obj* arc between *Jean* and *respecté* is labeled with *obj* only in the DSR). The DSRs may contain cycles, for instance in the case of an adjective or participle modifying a noun : the modifier dependency is retained in the deep representation, and an inverse arc is added (the noun is a deep syntactic argument of the modifying adjective or participle).

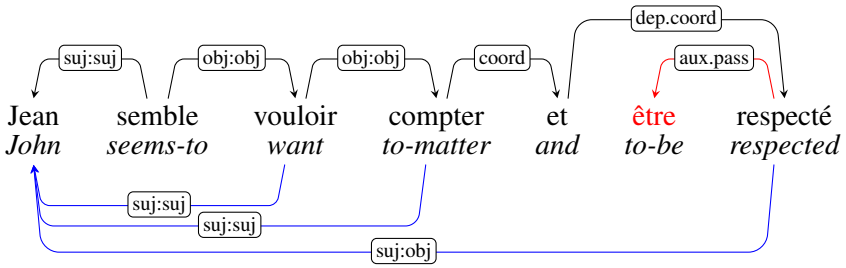


Figure 1: A dependency graph containing both the surface syntactic tree and the deep syntactic representation. The black arcs belong to both representations. The red arc belongs to the surface representation only. The blue arcs below the sentence are "deep-only": they belong to the DSR only.

2 Methodology to obtain more DSRs

In order to obtain pseudo-gold DSRs for the FTB, we used as starting point the surface dependency version of the FTB, as released for the SPMRL Shared Task (Seddah et al. [26]), which contains 18,535 newspaper sentences. To obtain DSRs for these sentences, we partially re-used the annotation methodology of the Deep Sequoia Treebank, which consisted of three steps (Candito et al. [6]):

(i) Manual annotation of certain phenomena,

(ii) Automatic pre-annotation using two independently designed graph-rewriting systems (Grew (Guillaume et al. [11]) and OGRE (Ribeyre et al. [22]))

(iii) Double manual correction plus adjudication of the divergences.

We applied this methodology on the FTB, but we skipped the last step, which currently seems out of reach given the corpus' size. We retained the dichotomy between manual annotation of certain phenomena (step (i)) and automatic annotation (step (ii)), this time using OGRE only.

The focus of this paper is on the evaluation of step (ii) mainly, so we only briefly list the phenomena that were manually annotated during step (i)[3]: long-distance dependencies, impersonal subjects, causative alternations, cleft sentences, and finally the status of the *se* clitic, which can either be part of a lexicalized pronominal verb (like *s'apercevoir (to realize)*), or mark a reflexive construction (as in *Anna se soigne tout seule (Anna cures herself on her own)*), or mark a middle diathesis alternation (*Ces livres se vendent bien (These books sell well)*) or a neuter diathesis alternation (*Le vase s'est rompu (The vase broke)*). All these phenomena are either highly ambiguous (clitic *se*) and/or rare (long-distance dependencies, causatives, cleft sentences), and their disambiguation resorts to semantic properties that are notoriously difficult to capture in a rule-based approach. By contrast, phenomena which exhibit more systematic syntactic properties, such as raising and control or the passive alternation, are handled automatically at step (ii).

We can now turn to the description of the graph rewriting system and the handcraft rules used at step (ii).

3 Surface to deep tool

3.1 OGRE

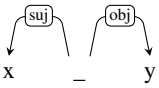
OGRE (for Optimized Graph Rewriting System) is a two-stage graph rewriting system (Ribeyre et al. [22]). The first stage is based on the Single Pushout approach described at length in (Rozenberg [24]) while the second has his roots in the constraint programming paradigm (Rossi et al. [23]).

OGRE uses a set of rules, applied in two stages. A rule is defined by a subgraph pattern called a match, a set of rewriting commands (performed at first stage)

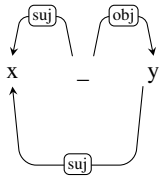
[3]The manual annotations were mainly performed by the second author of this paper, and other colleagues. We hope to be able to describe the manual annotations in another publication.

```
rule add_suj_edge{
    match{
        x <-[label:"suj"]- [] -[label:"obj"]-> y[cat:"VINF"]
    }
    negative{
        y -[]-> x
    }
    commands{
        add_edge(y, x, [label:"suj"]
    }
}
```

(a) Textual form of the rule.



(b) Subgraph pattern (match).



(c) Transformed subgraph.

Figure 2: Example of rule which adds a *suj* edge, in text format (a), with graphical format for the match pattern (b) and for the resulting graph after application (c). The rule contains a Negative Application Condition (NAC), which blocks the application if the x node depends on the y node.

and/or a set of "triggers"[4] (instantiated at first stage, but activated in the second stage). In addition, a rule may contain negative application conditions (NAC defined in Lambers et al. [14]) which block matches based on certain conditions. An example of rule is given in Figure 2.

During the first stage, rules are applied sequentially. The rewriting commands can add, modify and remove properties on nodes (token, features, POS, ...) and edges (labels, surfacic or deep status), and remove or add edges. In the surface-to-deep rules, removal of edges or features is not used though. Importantly, the match is always performed on the input graph only, independently of the added arcs/features, so the order of the rules does not matter. The set of rewriting commands is applied to the input graph and triggers are instanciated on pairs of nodes, to be used during the second stage.

In the second stage, triggers instantiated at first stage apply until no more edges are added. In the surface-to-deep rules, we only use one of the several trigger types available in OGRE, namely the share triggers[5], which add edges. A share trigger *share(l)* is defined for a pair of nodes (y,z) and a label l. It states that if a $y \xrightarrow{l} x$ arc belongs to the current graph (i.e. if it either belongs to the modified graph from the first stage, or was added by a share trigger), then the arc $z \xrightarrow{l} x$ should be

[4] In (Ribeyre et al. [22]), the term "constraints" was used instead of "triggers".

[5] Formerly defined as share constraints.

added to it. For example in Figure 3, three share triggers *share(suj)* sequentially add the orange, purple and green edges, in that order, each new edge triggering the applicability of the subsequent trigger. As will be seen in the next section, this system allows to express in a compact way general linguistic constraints such as cascaded control or raising verbs.

Termination of the second stage is guaranteed by the absence of multi-edges with the same label. Moreover, the iterative process combined with the fact that triggers can only add edges ensures the confluence of the system.

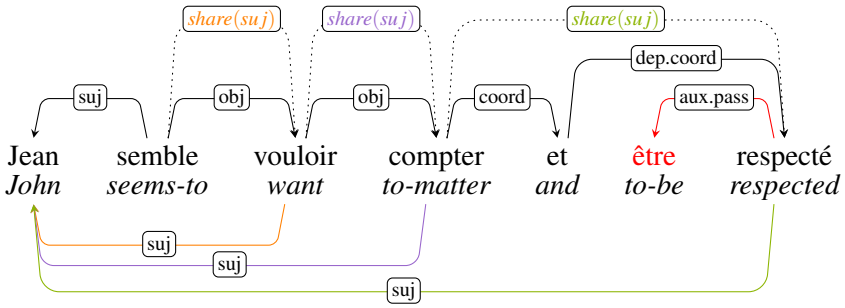


Figure 3: Share-triggers (dotted arcs) for a cascade of raising/control verbs and coordinated verbs. Each share-trigger instance adds the deep-only arc of the same color.

3.2 Rules

The rules for surface to deep syntax conversion are organized into five *modules*, namely sets of rules, designed to be applied sequentially. The rules are partitioned into modules so that arcs or features added within one module can serve as matches for the rules of a subsequent modules : while the rules within a module need not be ordered, the modules themselves are sequentially ordered.

The first module makes verbal tense and mood explicit, converting tense auxiliaries into appropriate features on the lexical verb. For instance, in example 1, the verb *respecté* is a past participle at the surface level, but it bears infinitival mood and past tense at the deep level. This normalization facilitates the writing of rules in subsequent modules.

The second module marks the final subjects of non finite verbs (and by extension, of adjectives also, whether used as predicative complements or noun modifiers). It uses the constraint propagation system of OGRE to handle embeddings involving cascades of predicates and/or coordination. For instance the rule for raising or subject control verbs introducing infinitives contains a share-constraint stating that their subject should also be the final subject of the infinitive. This constraint instantiates for two pairs of nodes in Figure 3 (the orange and purple constraint instantiations), which add *Jean* as final subject of *vouloir* and in turn as final subject of *compter*. We extracted control and raising verbs from the Dicova-

lence lexicon (van den Eynde and Mertens [27]), and subsequently extended the list during rule tuning on the DeepSequoia dev corpus.

VP coordination is handled through another constraint, stating that for two coordinated verbs, if a final subject exists for the first conjunct, then it must also be added as the final subject of the second conjunct (provided the latter does not initially have a final subject). This is displayed as the green constraint in Figure 3, which adds the final subject of *respecté* as soon as *compter* gets a final subject.

Syntactic alternations are mainly handled in the third module, which identifies canonical functions for arguments of verbs (whether these arguments were already in the surface tree, or added by the second module). For instance, a rule states that if a passive verb has a final subject, then that is its canonical object. Such a rule applies in sentence 1 to identify *Jean* as the canonical object of the passive verb *respecté*. This module interacts with some manual annotations performed at step (i) : while passive verbs are automatically identified, other highly ambiguous alternations are first manually identified at step (i) (causatives, impersonal, middle and neuter alternations), then the rules interpret the manual annotations to correctly derive the canonical functions of the arguments (including the cases of alternation interaction such as impersonal passives). A clear-cut separation between the module for final subjects and the module for syntactic alternations is not possible in particular because of control verbs specificities. The syntactic generalization applicable to control verbs mixes canonical and final functions. Indeed, a given control verb imposes which of its *canonical* argument is the *final* subject of the infinitive. For instance, the verb *condamner (to condemn)* is an object-control verb, meaning that its *canonical* object is the *final* subject of the infinitival clause it introduces. So, in *La cour a condamné Jean à être incarcéré (The court condemned Jean to be incarcerated)*, the object *Jean* is the final subject of the passive verb *incarcéré*. When *condamner* is passive, then the controler of the infinitive, it still holds that its *canonical* object (but final subject) is the final subject of the infinitive, as in *Jean a été condamné à être incarcéré (Jean was condemned to be incarcerated)*. We resolved this interaction by explicitely distinguishing rules for active and passive control verbs in the final-subject module.

A fourth module handles comparative and superlative constructions, mostly. It also adds morphological features such as definiteness in case of determiners, and identifies the clause types (interrogative, imperative...). Finally the last module exclusively deals with the removal and bypassing of semantically empty words. Incoming and outgoing edges of these words are attached to semantically full words.

To give an idea of the degree of complexity of the system, the five modules contain 19, 40, 21, 39 and 36 rules respectively, for a total of 155 rules. While being a reasonable figure, we must admit that the understanding and maintenance of this rule set requires training.

4 Evaluation

4.1 Quantative analysis

We now turn to the evaluation of the Surface-to-deep conversion module and of the quality of the DSRs we obtain for the FTB. The current version of the surface-to-deep conversion module was designed in two stages. As mentioned in section 2, the conversion module was first designed to pre-annotate the DeepSequoia, which has been subsequently manually corrected. More precisely, in order to build the DeepSequoia, the two research teams who produced the DeepSequoia (namely ours and the Sémagramme team) first manually annotated a subset of 247 sentences (called the MINIREF), and then both tuned a conversion tool on this subset. Then two pre-annotated versions of the treebank could be produced, manually corrected by each team, and finally conflicts were manually adjudicated. For the current work, we subsequently improved our conversion rules using another subset of the DeepSequoia. More precisely, we split the DeepSequoia into four parts, as shown in Table 1. We used the DEV2 set to improve the rules' coverage, while setting aside a training set for future experiments, and a test set for final evaluation. Further, in order to evaluate the quality of the DSRs obtained for the FTB, we manually annotated the first 200 sentences from the FTB development set.

The top part of Table 2 concerns the evaluation of the conversion rules on the DeepSequoia test set. We report labeled and unlabeled precision, recall and F-measure when considering either the set of deep edges (first row of Table 2) , or the set of deep-only edges (second row). Performance on this test set is rather high, although a little lower on the deep-only edges.

Sets	#Sent.	#Tokens	#Deep Tokens
TRAIN	2,202	47,415	40,792
DEV-1 (Miniref)	247	5,852	5,038
DEV-2	250	5,360	4,606
TEST	400	8,411	7,264

Table 1: Experimental Split

The bottom part of the table concerns the evaluation on the 200 sentences from the FTB. We proceeded as follows: we applied the surface-to-deep rules on the *reference surface trees*, augmented with the deep manual annotations (cf. step (i) mentioned in section 2), and obtained predicted DSRs (hereafter *Predicted Deep 1*). We manually corrected these predicted DSRs, and also manually corrected some errors in the reference surface trees. We thus obtained *corrected deep* representations and *corrected surface* trees. The line "REFERENCE vs CORRECTED SURFACE" in Table 2 shows the evaluation of the reference surface trees against the corrected surface trees. The next line provides an evaluation of the *Predicted Deep 1* representations against the corrected deep ones. It shows that the overall quality of the resulting deep syntactic corpus is rather good. It can be anticipated that the DSRs obtained for the FTB will have sufficient quality to serve as training data.

Yet, while the evaluation of *Predicted Deep 1* (penultimate row) provides an es-

timation of the quality of the full set of predicted DSRs for the whole FTB, it mixes errors due to the rules, and errors in the reference surface trees. In order to evaluate the former more precisely, we applied the conversion rules on the *corrected surface* trees, and obtained a second version of predicted DSRs (hereafter *Predicted Deep 2*). The results are shown in the last row of Table 2. We obtain no drop in performance with respect to the evaluation of the DeepSequoia, which indicates that our rule set has a good coverage, and generalizes well to other corpora.

DeepSequoia (test set)	# gold edges	LP	LR	LF	UP	UR	UF
DEEP EDGES	8259	99.5	99.2	99.4	99.5	99.3	99.4
DEEP ONLY EDGES	1806	98.1	97.3	97.7	98.3	97.5	97.9
FTB (200 sent. dev.)	# gold edges	LP	LR	LF	UP	UR	UF
REFERENCE VS CORRECTED SURFACE	6170	98.7	98.0	98.4	100.0	99.4	99.7
PREDICTED DEEP 1	6012	97.5	97.1	97.3	98.9	98.4	98.7
PREDICTED DEEP 2	6012	99.5	99.3	99.4	99.6	99.4	99.5

Table 2: Rules' evaluation (Labelled/Unlabelled recall, precision, F-measure).

4.2 Qualitative analysis

We checked the errors on the 200 sentences from the FTB. A qualitative evaluation reveals that some phenomena are not (properly) handled by the rules, because of their complexity and ambiguity. For example, nominal predicative complements in sentences such as *C'est une femme Capitaine.* (*It's a female captain.*), where an *arg* edge should be added between *femme* and *Capitaine*, are not automatically annotated. Elliptic coordination is another unhandled phenomena, in particular head gapping and argument clusters.

Finally, automatic annotation of infinitive subjects leads to the highest rate of errors. We can distinguish two types: (i) **Control or raising verbs** not present in our lexical resources: *annoncer (to announce)* or *continuer de (to continue to)* are two examples (*continuer* was present, but with preposition *à*). The same goes, for "control nouns". For instance, the noun *idée (idea)* was missing in the rules, which thus fail to assign the possessive as subject of the infinitive verb in *D'où son idée de calmer le jeu. (Hence his idea to calm things down).* (ii) **Arbitrary control**, for certain modifying prepositions introducing infinitive clauses, the rules arbitrarily choose the subject of the main verb as subject of the infinitive, though it is clear that such a simple and systematic rule will fail in some cases. For instance, in *Ils ont re cu les élèves pour visiter le fournil (they received the pupils to visit the bakery)*, the subject of *visiter (to visit)* is not properly found.

Conclusion

In this paper, we described the methodology we used to add a deep syntax annotation layer to the French Treebank. Based on the work carried out by Candito et al. [6] to develop and the DeepSequoia treebank, we enhanced the conversion process

from surface trees to obtain state-of-the-art results in term of expected quality as shown by our evaluation on a small gold standard we built from the FTB. Furthermore, we manually corrected a reduced set of difficult constructions. This evaluation suggests that the resulting new data set, a deep syntax version of the FTB, can be used as pseudo-gold data to train deep syntactic parsers, or to extract syntactic lexicons augmented with quantitative information. The Deep French Treebank will be released with the paper (following the original license).

References

[1] Anne Abeillé and Nicolas Barrier. Enriching a French treebank. In *Proc. of LREC*, Lisbon, Portugal, 2004.

[2] Miguel Ballesteros, Bernd Bohnet, Simon Mille, and Leo Wanner. Deep-syntactic parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1402–1413, Dublin, Ireland, August 2014.

[3] Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. The prague dependency treebank. In *Treebanks*, pages 103–127. Springer, 2003.

[4] Aoife Cahill, Michael Burke, Ruth O'Donovan, Josef van Genabith, and Andy Way. Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In *Proc. of ACL*, pages 320–327, Barcelona, Spain, 2004.

[5] Marie Candito and Djamé Seddah. Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical. In *Proc. of TALN*, Grenoble, France, 2012.

[6] Marie Candito, Guy Perrier, Bruno Guillaume, Corentin Ribeyre, Karën Fort, Djamé Seddah, and Éric De La Clergerie. Deep Syntax Annotation of the Sequoia French Treebank. In *International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Islande, 2014.

[7] Martin Čmejrek, Jan Hajič, and Vladislav Kuboň. Prague czech-english dependency treebank: Syntactically annotated resources for machine translation. In *In Proceedings of EAMT 10th Annual Conference*. Citeseer, 2004.

[8] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Stanford University, 2008.

[9] Marie-Catherine de Marneffe, Miriam Connor, Natalia Silveira, Samuel R Bowman, Timothy Dozat, and Christopher D Manning. More constructions, more genres: Extending stanford dependencies. *DepLing 2013*, page 187, 2013.

[10] Daniel Flickinger, Yi Zhang, and Valia Kordoni. DeepBank: a dynamically annotated treebank of the wall street journal. In *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories*, pages 85–96, 2012.

[11] Bruno Guillaume, Guillaume Bonfante, Paul Masson, Mathieu Morey, and Guy Perrier. Grew : un outil de réécriture de graphes pour le TAL. In *Proc. of TALN*, Grenoble, France, 2012. URL http://hal.inria.fr/hal-00760637.

[12] Jan Hajic, Jarmila Panevová, Eva Hajicová, Petr Sgall, Petr Pajas, Jan Štepánek, Jiří Havelka, Marie Mikulová, Zdenek Zabokrtskỳ, and Magda Ševcıková Razımová. Prague dependency treebank 2.0. *CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia*, 98, 2006.

[13] Julia Hockenmaier and Mark Steedman. Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396, 2007.

[14] Leen Lambers, Hartmut Ehrig, and Fernando Orejas. Conflict detection for graph transformation with negative application conditions. In *Proceedings of the Third International Conference on Graph Transformations*, ICGT'06, pages 61–76, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-38870-2, 978-3-540-38870-8.

[15] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: annotating predicate argument structure. In *Proc. of the workshop on Human Language Technology*, pages 114–119, Stroudsburg, USA, 1994. ISBN 1-55860-357-3. doi: 10.3115/1075812.1075835. URL http://dx.doi.org/10.3115/1075812.1075835.

[16] Igor Melčuk. *Dependency syntax: theory and practice*. State University Press of New York, 1988.

[17] Simon Mille, Alicia Burga, and Leo Wanner. AnCora-UPF: A Multi-Level Annotation of Spanish. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 217–226, Prague, Czech Republic, August 2013.

[18] Yusuke Miyao, Takashi Ninomiya, and Junichi Tsujii. Corpus-oriented grammar development for acquiring a head-driven phrase structure grammar from the penn treebank. In *Lectures notes in Computer Sciences (proc. of IJCNLP 2004)*, volume 3248, pages 684–693. Springer, 2005.

[19] Yusuke Miyao, Stephan Oepen, and Daniel Zeman. In-house: An ensemble of pre-existing off-the-shelf parsers. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 335–340, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL http://www.aclweb.org/anthology/S14-2056.

[20] Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL http://www.aclweb.org/anthology/S14-2008.

[21] D.M. Perlmutter. *Studies in Relational Grammar 1*. Studies in Relational Grammar. University of Chicago Press, 1983. ISBN 9780226660509. URL http://books.google.ca/books?id=EcBbfP1zSLMC.

[22] Corentin Ribeyre, Djamé Seddah, and Éric Villemonte De La Clergerie. A Linguistically-motivated 2-stage Tree to Graph Transformation. In Chung-Hye Han and Giorgio Satta, editors, *Proc. of TAG+11*, Paris, France, 2012. INRIA.

[23] Francesca Rossi, Peter van Beek, and Toby Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006. ISBN 0444527265.

[24] Grzegorz Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformation: Volume I. Foundations*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1997. ISBN 98-102288-48.

[25] Natalie Schluter and Josef Van Genabith. Dependency parsing resources for french: Converting acquired lexical functional grammar f-structure annotations and parsing f-structures directly. In *Proc. of NODALIDA 2009*, Odense, Denmark, 2009.

[26] Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. Overview of the spmrl 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proc. of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*, Seattle, WA, 2013.

[27] Karel van den Eynde and Piet Mertens. *Le dictionnaire de valence DICOVALENCE : manuel d'utilisation*, 2006. URL http://bach.arts.kuleuven.be/dicovalence/manuel 061117.pdf.

Formalizing MultiWords as Catenae in a Treebank and in a Lexicon

Kiril Simov and Petya Osenova

Linguistic Modeling Department
Institute of Information and Communication Technologies, BAS
{kivs|petya}@bultreebank.org

Abstract

The paper presents formalization of multiwords as catenae in a treebank and in a lexicon. We view catenae as a dependency subtree, which reflects non-constituents and non-standard dependencies. Since the multiword classifications vary to great extent, starting from very narrow ones and proliferating to extended ones which include also valences, the focus in the paper is not on the multiword typology per se, but on the general formalization of multiwords.

1 Introduction

Multiwords (or Multiword Expressions (MWEs)) have been approached from various perspectives. It seems that most efforts go into introducing various classifications with respect to various NLP tasks, such as annotation, parsing, etc. Since there is no broadly accepted standard for Multiwords (see about the various classifications in [4]), we adopt the Multiword classification, presented in the seminal work of [8]. The authors divide multiwords into two groups: *lexicalized phrases* and *institutionalized phrases*. The former are further subdivided into *fixed-expressions*, *semi-fixed expressions* and *syntactically-flexible expressions*. *Fixed expressions* are said to be fully lexicalized and undergoing neither morphosyntactic variation nor internal modification. *Semi-fixed expressions* have a fixed word order, but "undergo some degree of lexical variation, e.g. in the form of inflection, variation in reflexive form, and determiner selection" (non-decomposable idioms, proper names). *Syntactically-flexible expressions* show more variation in their word order (light verb constructions, decomposable idioms). The latter group handles semantically and syntactically compositional, but statistically idiosyncratic phrases (such as, traffic lights).

We follow the understanding of [6] that multiwords have their internal syntactic structure which needs to be represented in the lexicon as well as in the sentence analysis. Such a mapping would provide a mechanism for accessing the literal

meaning of multiwords (when existing together with the idiomatic one). Thus, in this paper we focus on the formal representation of multiwords as catenae in the treebank as well as in the lexicon. Also, examples of formalization are provided for the most frequent multiword types in BulTreeBank.

The paper is structured in the following way: in Section 2 some previous works on catena are presented; Section 3 discusses the most frequent specific multiword types in the treebank; Section 4 outlines the formal definition of catena; Section 6 demonstrates the encoding of catenae in a dependency treebank; Section 6 shows the encoding of the catena in the lexicon; Section 7 concludes the paper.

2 Previous Work on Catenae

The notion of catena (chain) was introduced in [6] as a mechanism for representing the syntactic structure of idioms. He showed that for this task there is a need for a definition of syntactic patterns that do not coincide with constituents. He defines the catena in the following way: The words A, B, and C (order irrelevant) form a chain if and only if A immediately dominates B and C, or if and only if A immediately dominates B and B immediately dominates C. In recent years the notion of catena revived again and it was applied also to dependency representations. Catena is used successfully for modelling of problematic language phenomena. [2] presents the problems in syntax and morphology that have led to the introduction of the subconstituent catena level. Constituency-based analysis faces non-constituent structures in ellipsis, idioms, verb complexes. In [3] the author again advocated his approach on providing a surface-based account of the non-constituent phenomena via the contribution of catena. Here the author introduces a notion at the morphological level — morph catena.

Apart from the linguistic modeling of language phenomena, catena was used in a number of NLP applications. [5], for example, presents an approach to Information retrieval based on catenae. The authors consider catena as a mechanism for semantic encoding which overcomes the problems of long-distance paths and elliptical sentences. The employment of catena in NLP applications is additional motivation for us to use it in the modeling of an interface between the treebank and the lexicon.

As part of the morphemic analysis of compounds, catena is also a good candidate for mapping the elements of the syntactic paraphrase of the compound to its morphemic analysis as shown in [7]. In this paper we focus on the formal representation of multiwords in the treebank and lexicon from the perspective of the syntactic relations among their elements. Thus, irrespectively of the multiword classifications, the challenging issue remains the representation of multiwords with syntactic variability in the syntactic resource and the lexicon.

3 Multiwords from the Perspective of Syntactic Relations

Here we outline some frequent multiword types with respect to the syntactic relations (adjunction and complementation) among their elements. In the next sections also their modeling is presented with examples in the treebank and the lexicon. The adjunction and complementation types do not affect the formalization, which generalizes over both of them. However, it shows differences in the syntax-lexical interface.

The adjunction is expressed in the following multiword types:

1. Noun phrases of type Adjective - Noun

   вътрешен министър, 'interior minister' (Minister for Internal Affairs) снежен човек, 'snow man' (snowman)

   These patterns allow inflection in both elements for number. The first element can get a definite article. The noun phrase can be further modified: 'our interior minister'; 'a nice snow man', etc. Semantically, the first phrase is a metonymical synthetic form of the phrase 'Minister for Internal Affairs'. The second phrase conveys its literal meaning of: (1) a man-like sculpture from snow or (2) hypothetical man leaving in Himalayas or some other regions.

2. Noun phrases of type Noun - Prepositional Phrase

   срещата на върха, 'meeting-the at peak-the' (summit)

   Here 'meeting' can inflect in all its forms and allows for some modifications: 'past meetings', etc.

The complentation is expressed in the following multiword type:

1. Verb phrases of type Verb-Complement

   знае си работата, 'knows-he his business-the' (one knows one's business); затварям си очите, 'close own eys-the' (to hide from the facts);

   Here 'business' allows for only various possessive forms (one *knows* their business), but the nominal phrase always has to be definite, singular. The verb 'know' can vary in all its word forms and it allows for modification: one knows his business *well*.

4 Formal Definition of Catena

Here we follow the definition of catena (originally called chain, but later changed to catena, because of the ambiguity of the term chain) provided by [6] and [2]: a **catena** is a word or a combination of words directly connected in the dominance dimension. In reality this definition of catena for dependency trees is equivalent to

a subtree definition. We prefer to use the notion of catena to that of dependency subtree, because its high usage in modeling MultiWord Expressions. However, we have to utilize the notion of catena for two purposes: for annotation of MultiWord Expressions in the actual trees expressing the analysis of sentences as well as for representation of MultiWord Expressions in the lexicon.

Let us have the sets: LA — a set of POS tags, LE — a set of lemmas, WF — a set of word forms and a set D of dependency tags ($ROOT \in D$). Let us have a sentence $x = w_1, ..., w_n$. A **tagged dependency tree** is a directed tree $T = (V, A, \pi, \lambda, \omega, \delta)$ where:

1. $V = \{0, 1, ..., n\}$ is an ordered set of nodes, that corresponds to an enumeration of the words in the sentence (the root of the tree has index 0);
2. $A \subseteq V \times V$ is a set of arcs;
3. $\pi : V - \{0\} \rightarrow LA$ is a total labeling function from nodes to POS tags. π is not defined for the root;
4. $\lambda : V - \{0\} \rightarrow LE$ is a total labeling function from nodes to lemmas. λ is not defined for the root;
5. $\omega : V - \{0\} \rightarrow WF$ is a total labeling function from nodes to word forms. ω is not defined for the root;
6. $\delta : A \rightarrow D$ is a total labeling function for arcs;
7. 0 is the root of the tree.

We will hereafter refer to this structure as a parse tree for the sentence x. Let $T = (V, A, \pi, \lambda, \omega, \delta)$ be a tagged dependency tree.

A directed tree $G = (V_G, A_G, \pi_G, \lambda_G, \omega_G, \delta_G)$ is called **dependency catena of** T if and only if:

1. G is a connected directed tree with root $CatR$ ($CatR \in V_G$);
2. $\psi : V_G \rightarrow V$, there is a mapping from the nodes V_G into $V - \{0\}$. V_G is the set of nodes of G;
3. $A_G \subseteq A$, the set of arcs of G;
4. $\pi_G \subseteq \pi$ is a partial labeling function from nodes of G to POS tags;
5. $\lambda_G \subseteq \lambda$ is a partial labeling function from nodes to lemmas;
6. $\omega_G \subseteq \omega$ is a partial labeling function from nodes to word forms;
7. $\delta_G \subseteq \delta$ is a partial labeling function for arcs.

A directed tree $G = (V_G, A_G, \pi_G, \lambda_G, \omega_G, \delta_G)$ is a **dependency catena** if and only if there exists a dependency tree T such that G is a dependency catena of T.

Having partial functions for assigning POS tags, dependency labels, word form and lemmas allows us to construct arbitrary abstractions over the structure of catena. The mapping ψ parameterizes the catena with respect to different dependency trees. Using the mapping there is a possibility to use different word orders of the nodes of the catena, for example. Also catena could be underspecified for some of the node labels like grammatical features, lemmas and also some dependency labels.

The image (mapping) of a catena in a given dependency tree we will call **realization of the catena in the tree**. We consider the realization of the catena as fully specified subtree including all node and arc labels. For example, the catena for "to spill the beans" will allow for any realization of the verb form like in: "they spilled the beans" and "he spills the beans". Thus, the catena in the lexicon will be underspecified with respect to the grammatical features and word form for the verb.

Two catenae G_1 and G_2 could have the same set of realizations. In this case, we will say that G_1 and G_2 are **equivalent**. Representing the nodes via paths in the dependency tree from root to the corresponding node and imposing a linear order over this representation of nodes facilitates the selection of a unique representative of each equivalent class of catenae. Thus, in the rest of the paper we assume that each catena is representative for its class of equivalence.

5 Encoding of Multiword Valency in a Treebank

In the rest of the paper we represent dependency trees in CoNLL 2006 shared task format with the necessary changes. This format is a table format where each node in the dependency tree (except the root node 0) is represented as a row, the cells in a row are separated by a tabulation symbol. The fields are: Number, WordForm, Lemma, POS, ExtendedPOS, GrammaticalFeatures (in a form of attribute value pairs, attr=v, separated by a vertical bar), parent node, and dependency relation. In the paper we do not use columns 9 and 10 as they were used in the CoNLL 2006 format. Here column 9 is used for annotation of the node as being part of a catena or not. The rows that represent the nodes belonging to a catena are marked with the same identifier. If a node is not part of a catena, column 9 of the corresponding line contains an underscore symbol. Since a sentence might contain more than one catena, each one is numbered in different way. We do not allow any catena overlapping.

Let $T = (V, A, \pi, \lambda, \omega, \delta)$ be a tagged dependency tree:

1. The nodes of $V - \{0\}$ are represented in the first cell of each row in the table;
2. For each arc $\langle d, h \rangle \in A$, the head node h is represented in cell 7 of the row for node d;

3. For each node $n \in V - \{0\}$, the value $\pi(n)$ is represented in cells 4, 5, and 6 of the row for node n;

4. For each node $n \in V - \{0\}$ the value $\lambda(n)$ is represented in cell 3 of the row for node n;

5. For each node $n \in V - \{0\}$ the value $\omega(n)$ is represented in cell 2 of the row for node n;

6. For each arc $\langle d,h \rangle \in A$ the label $\delta(\langle d,h \rangle)$ is represent in cell 8 of the row for node d.

7. the root 0 is not represented in the table.

The following is an example for the sentence: Те си затварят очите пред истината (they run away from the truth):

No	Wf	Le	POS	ExPOS	GramFeat	Head	Rel	Catena
1	Те	те	P	Pp	number=pl| case=nom	3	subj	_
2	си	си	P	Pp	form=possesive	3	clitic	c_1
3	затварят	затварям	V	Vpi	number=pl| person=3	0	Root	c_1
4	очите	око	N	Nc	number=pl| definiteness=y	3	obj	c_1
5	пред	пред	R	R	_	3	indobj	_
6	истината	истина	N	Nc	number=sg| definiteness=y	5	prepobj	_

In the table it can be seen that three elements are part of the catena: си затварят очите 'their close eyes' (they close their eyes). In this way, the idiomatic meaning of the expression is ensured. Thus, each MWE in a dependency tree is represented via its realization.

This representation of MWEs is convenient for dependency trees in dependency treebanks on analytical (or surface) level of dependency analysis. Here we will not discuss the role of catena in deep level dependency analysis (e.g. the tectogrammatical level in the Prague dependency treebank).

In order to model the behavior in a better way we need to add semantics to the dependency representation. We will not be able to do this in full in this paper. In order to represent the MWEs in the lexicon, we assume a semantic analysis based on Minimal Recursion Semantics (see [1]). For dependency analyzes the MRS structure are constructed in a way similar to the one presented in [9]. In this work, the root of a subtree of a given dependency tree is associated with the MRS structure corresponding to the whole subtree. This means that for the semantic interpretation of MWEs we will use the root of the corresponding catena. In the dependency tree for the corresponding sentence the catena root will provide the interpretation of the MWE and its dependent elements, if any. In the lexicon we will provide the corresponding structure to model the idiosyncratic semantic content of MWE.

6 Encoding of Multiword Valency in a Lexicon

The lexical entry of a MWE consists of a **form**, a **catena**, **semantics** and **valency**. The form is represented in its canonical form which corresponds to one of its realizations. The catena for the multiwords is stored in the CoNLL format as described above. The semantics part of a lexical entry specifies the list of elementary predicates for the MRS analysis. When the MWE allows for some modification (also adjunction) of its elements - i.e. modifiers of a noun, the lexical entry in the lexicon needs to specify the role of these modifiers.

For example, the multiword from the above example затварям си очите is represented as follows:

[**form:** < затварям си очите >
catena:

No	Wf	Le	POS	ExPOS	GramFeat	Head	Rel
1	_	затварям	V	Vpi	_	0	CRoot
2	си	си	P	Pp	form=possesive	1	clitic
3	очите	око	N	Nc	number=pl| definiteness=y	1	obj

semantics:
No1: { run-away-from_rel(e,x_0,x_1), fact(x_1), [1](x_1) }
valency:
No1: < :indobj: x/Prep :prepobj: y/N[1] || x ∈ { пред, за } >
]

The lexical entry shows that the catena includes the elements 'shut my eyes' in the sense of 'run away from facts', which is presented in the semantics part as a set of elementary relations. In this case we have the relation run-away-from_rel(e,x_0,x_1) which determines that the multiword expression is denoting an event with two main participants denoted by the subject (x_0) and the indirect object (x_1). In the lexical entry we represent the restriction on the indirect object which has to be a fact. The actual fact in this part is indicated via a structure-sharing mechanism with a valency part — [1]. This is necessary, because in the valency part of the lexical entry the noun within the subcategorized PP by the catena 'shut my eyes' reproduces some fact from the world.

The valency information is presented by a dependency path. The arc labels are given between column marks, the node information is given after the arc information and could include a variable for the word (we also plan to add lemma information) and grammatical features. The structure-sharing identifier [1] denotes the semantics of the noun phrase that is indirect object. Its main variable is made equal to the variable for indirect object in the semantic representation of MWE — x_1. This ensures that the expected noun phrase has to denote a fact. Additionally, if one or more (but small amount of) words are possible for a node, they can be given as a set. In the example only two prepositions are possible for node x.

In many languages the elements represented in the valency are not realized. This is the case for Bulgarian — the objects and indirect objects of a verb could

be unexpressed. In such cases the semantics is assumed to be empty, expressed via the most general predicate like *everything*(x) which will agree with any other predicate. In this way the predicate assigned to the structure-sharing identifier [1] above will ensure a correct interpretation of the semantics expressed in the lexical entry for the multiword expression.

In the catena representation cell 9 is empty and this is why it is not given in the lexicon. The semantics and the valency information is attached to the corresponding nodes in the catena representation. In the example above only the information for the root node of the catena is given (node number 1 — No1). In cases when other parts of the catena allow modification, the information for the corresponding nodes will be given.

For example, the multiword среща на върха (summit) allows for modification not only of the whole catena, but also of the noun within the prepositional phrase. The lexical entry from the lexicon is given as follows:

[**form:** < среща на върха >
catena:

No	Wf	Le	POS	ExPOS	GramFeat	Head	Rel
1	_	среща	N	Nc	_	0	CRoot
2	на	на	R	R	_	1	mod
3	върха	върх	N	Nc	number=sg | definiteness=y	2	prepobj

semantics:
No1: { meeting_rel(e, *x*), member(*y*,*x*), head-of-a-country(*y*,*z*), country(*z*), [1](*z*)) }
valency:
No3: < :mod: x/Adj[1] >
]

This lexical entry allows modifications like ‘европейски’ (European) — среща на европейския връх (meeting of the European top). This catena allows also modification of the head word.

The last example presented here is for the multiword ‘снежен човек’, meaning “a man-like sculpture from snow”. It does not allow any modification of the dependent node снежен (snow), but it allows for modifications of the root like “large snow man” etc. The lexical entry from the lexicon is given as follows:

[**form:** < снежен човек >
catena:

No	Wf	Le	POS	ExPOS	GramFeat	Head	Rel
1	_	снежен	A	A	_	2	mod
2	_	човек	N	Nc	definiteness=n	0	CRoot

semantics:
No2: { snowman_rel(*x*) }
valency:
]

The grammatical features for the head noun (definiteness=n) restricts its possible form. In this way singular and plural forms are allowed. The empty valency

ensures that the dependent adjective can not be modified except for morphological variants like singular and plural forms, but also definite or indefinite forms depending on the usage of the phrase. The possible modifiers of the multiword expression are determined by the represented semantics. The relation snowman_rel(x) is taken from an appropriate ontology where it conceptual definition is given.

These three examples demonstrate the power of the combination of catenae, MRS structures and valency representation to model multiword expressions in the lexicon. The catena is appropriate for representation of syntactic structure and variation on morphological level, the semantic part represents the idiosyncratic semantics of the MWE and determines the possible semantic modification, and the valency part determines the syntactic behavior of MWE. One missing element of the lexical entry is a representation of constraints over the word order of the nodes of the catena. We envisage addition of such constraints as future work.

7 Conclusion

In this paper a formalization of the multiwords as catenae was presented. The focus was on their modeling in the treebank and in the lexicon. Although the catenae approach provided a good apparatus for this, there are specificities in the syntax and lexical representation that had to be reflected. The common perspective for the syntax-lexical interface of multiwords lies in the syntactic relations among their elements (adjunction and complementation).

Sag et. al (2002) [8] enumerated several problems for MWEs representation which we hope our representation of MWEs in the lexicon solves to a great extent. The **overgeneration problem** is solved by an appropriate combination of syntactic, morphological, semantic and valency constraints. They are enough to rule out the impossible realizations of the multiword expressions. The **idiomaticity problem** is also solved because any peculiarities on these levels can be expressed in the lexical entry. The **flexibility problem** is solved by the definition of catena which allows for different realizations in the actual dependency trees. The **lexical proliferation problem** is manageable by using the valency constraints. In this way we can incorporate semantic constraints on the dependents. In the case of light verb, for example, the semantic of the verb in most cases is very general, but the actual semantic is coming from the direct object.

In future we will develop a lexicon for the MWEs appearing in the Bulgarian treebank. Then we will develop a mechanism to use the created multiword lexicon in parsing and generation processing. In addition, the formalization represented above need to be extended with word order constraints and statistical information for institutionalized phrases.

8 Acknowledgements

This research has received partial funding from the EC's FP7 (FP7/2007-2013) under grant agreement number 610516: "QTLeap: Quality Translation by Deep Language Engineering Approaches".

References

[1] Copestake, A., Flickinger, D., Pollard, C., and Sag, I. (2005) Minimal Recursion Semantics: an Introduction. *Research on Language and Computation*, 3(4).

[2] Thomas Gross (2010) *Chains in syntax and morphology.* In Otoguro, Ishikawa, Umemoto, Yoshimoto, and Harada, editors, PACLIC, pages 143–152. Institute for Digital Enhancement of Cognitive Development, Waseda University.

[3] Thomas Gross (2011) *Transformational grammarians and other paradoxes.* In Igor Boguslavsky and Leo Wanner, editors, 5th International Conference on Meaning-Text Theory, pages 88–97.

[4] Aline Villavicencio and Valia Kordoni (2012) *There's light at the end of the tunnel: Multiword Expressions in Theory and Practice*, course materials. Technical report, Erasmus Mundus European Masters Program in Language and Communication Technologies (LCT).

[5] K. Tamsin Maxwell, Jon Oberlander, and W. Bruce Croft (2013) *Feature-based selection of dependency paths in ad hoc information retrieval.* In Proceedings of the 51st Annual Meeting of the ACL, pages 507–516, Sofia, Bulgaria.

[6] William O'Grady (1998) The syntax of idioms. *Natural Language and Linguistic Theory*, 16:279–312.

[7] Petya Osenova and Kiril Simov (2014) *Treatment of Multiword Expressions and Compounds in Bulgarian.* In Proceedings of the Workshop on Computational, Cognitive, and Linguistic Approaches to the Analysis of Complex Words and Collocations (CCLCC 2014), ESSLLI, Tuebingen, Germany, pages 41–46.

[8] Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger (2001) *Multiword expressions: A pain in the neck for NLP.* In In Proc. of the CICLing-2002, pages 1–15.

[9] Simov, K., and Osenova, P. (2011) *Towards Minimal Recursion Semantics over Bulgarian Dependency Parsing.* In Proc. of the RANLP 2011.

Estimating the Utility of Simplified Discriminants in Grammar-Based Treebanking

Arne Skjærholt and Stephan Oepen

Language Technology Group, Department of Informatics
University of Oslo
E-mail: {arnskj,oe}@ifi.uio.no

Abstract

We investigate different types of discriminants in grammar-based treebanking, grounded in a review of earlier work; with an eye towards annotation by non-experts, we propose different simplifications of common discriminant types and quantify their 'discriminative power' as well as efficacy in the preparation of training data for a discriminative parse ranker.

1 Introduction

So-called *discriminants*, first suggested by Carter [2], are a vital concept in grammar-driven treebanking (van der Beek et al. [1], Oepen et al. [7], Rosén et al. [9]), enabling annotators to easily select the correct parse from a parse forest of hundreds or even millions of candidate parses. This power stems from the fact that discriminants represent localized 'atomic' ambiguities (individual factors in combinatorial explosion) and, thus, allow the annotator to perform what is essentially a binary search over the parse forest, requiring only $O(\log_2 n)$ decisions to fully disambiguate a forest of n trees; for example, disambiguating a million trees can be accomplished through only approximately 20 decisions.

The first application of discriminants to treebanking is the work of Carter [2], whose TreeBanker presents several kinds of discriminant to the user: word senses (for example *serve* in the sense of *fly to* vs. *provide*), labelled phrase structure spans and sentence type (the special case of a labelled span covering the entire input), semantic triples of two word senses connected by a preposition or conjunction, and specific grammar rules used to build a constituent.

The next application of discriminants for treebanking is the Alpino Dependency Treebank of Dutch (van der Beek et al. [1]), which is couched in the framework of HPSG. In this case, annotators could choose between lexical (coarse-grained PoS tags), unlabelled constituent, and dependency path discriminants. The Alpino dependency paths are paths from the root of the tree to either a word or a

phrase, and these discriminants are additionally pruned to only show the shortest paths. That is, if two discriminants decide between the exact same sets of trees, the discriminant that has the shorter path will be preferred.

At roughly the same time and also working in HPSG, Oepen et al. [7] identified four types of discriminants in building the LinGO Redwoods treebank: the lexical type of a token (a fine-grained PoS tag including information about argument structure), the identity of the HPSG construction applied to a span, the semantic predicate associated with a constituent, and simplified constituent labels in terms of 'traditional' phrase structure categories. In more recent Redwoods development, only the first two types were used. Later, a third type of discriminant was added: predicate–argument triples extracted from the underspecified logical forms generated by the grammar (Oepen and Lønning [6]).

Finally, discriminants have been applied to treebanking using LFG grammars by Rosén et al. [9]. They identify four discriminant types: lexical (corresponding to PoS ambiguities), morphological (corresponding to homographs and morphological syncretism), c-structure (ordinary phrase structure), and f-structure (corresponding to discriminating values in syntactico-semantic feature structures).

There is clearly considerable framework-specific variation in the details of discriminant-based annotation, but nevertheless discriminants can be grouped into four broad categories: lexical information, syntactic constituents (either labelled or unlabelled), syntactic dependencies, and semantic predicate–argument information. PoS information can be considered a special case of syntactic constituents of one word, but considering them a separate class is beneficial for the annotators as ambiguities involving a single word are usually very easy to decide (van der Beek et al. [1], Rosén et al. [9]).

However all of these applications have in common that they are intended for relatively well-trained annotators, with the goal of efficiently finding a single gold-standard tree among the trees in the parse forest.[1] In this paper, with an eye towards reducing annotation costs, we investigate the potential of only using only simpler discriminants. While these discriminants do not, in the general case, allow an annotator to recover a single correct parse, they do allow an annotator to decide important classes of ambiguity. In return for this loss of precision, we get an annotation problem that is significantly simplified, allowing us to tap a wider pool of annotators.

2 Simplified HPSG Discriminants

We take as our point of departure the LinGO Redwoods syntactic discriminants. As mentioned above, there are two predominant types of these discriminants: lexical

[1] While some discriminant-based annotation tools in fact operate directly over the packed parse forest, others actually require extracting a (possibly partial) list of full parses prior to discrimnant extraction and annotation. Although important technically and conceptually, this distinction has no immediate consequences for our experiments.

types of individual words, and grammatical constructions applied to spans. Figure 1 shows what is known as the *derivation trees* of both analyses licensed by the (1212 version of the) LinGO English Resource Grammar (ERG; Flickinger [3]). Here preterminal nodes are labelled with lexical types and the remaining internal nodes contain the construction applied at that constituent. Together with a copy of the grammar used to parse the sentence, this information enables us to reconstruct the full HPSG feature structure corresponding to that particular parse.

In Figure 1, nodes that correspond to discriminants are highlighted in bold face. In total there are 11 such spans, 4 in the topmost tree (which is the gold tree in the treebank), and 7 in the bottom one. These discriminants are both very specific and very general. The lexical types are highly specialised, encoding not only part of speech, but information such as argument selection (for example, v_np*_le in Figure 1 designates a verb that takes an optional nominal complement); the LinGO ERG contains some 1200 different lexical types. The syntactic rules however, as a consequence of HPSG being a highly lexicalised theory, are in the main comprised of general construction types such as the subject–head and head–complement rules (sb-hd_mc_c and hd-cmp_u_c) at the top of the tree in Figure 1; the ERG contains some 220 such constructions.

In this paper we consider a number of different simplified discriminants, derived from the standard types. The first two types are lexical in nature. An obvious first choice here is the lexical types of the grammar. We do not consider these particularly useful for a wider pool of annotators however, and rather we study this type to see how it compares with a simplified set of lexical types where all additional information (argument preferences, etc.) is stripped, yielding a coarse-grained part-of-speech tagset similar to that of Petrov et al. [8]. These simplified tags are capable of deciding between important classes of ambiguity, such as the noun vs. verb ambiguity of the word *saw*, but not the lemma ambiguity of the same word between the present tense of *saw* and the past tense of *see*.[2]

A slightly more complex kind of discriminant is phrasal discriminants. We consider three discriminants in this class. The first of these is simply unlabelled spans, i.e. bracketing a sequence of tokens as a constituent (of an arbitrary category). While clearly not able to handle all classes of ambiguity, important cases such as PP and other modifier attachments can be disambiguated using such discriminants. For example, whether "the man in the park" is a constituent or not decides between high and low attachment in the case of "I saw a man in the park".

A slightly more complex discriminant type is labelled spans. In this case, the labels are not individual constructions of the grammar, but rather a simplified set of phrase structure labels like S, VP, NP, etc. This is clearly a more powerful type of discriminants, as the distinction between a modifier PP and a selected-for PP is not discernible without bracket labels. The third and final type of phrasal

[2]LFG morphological discriminants do distinguish the two possible lemmas; however these are not directly portable to HPSG as inflectional morphology is handled by unary rules in the lower layers of the tree. In Figure 1, the rule v_3s-fin_olr corresponds to the present tense inflection of *plays*.

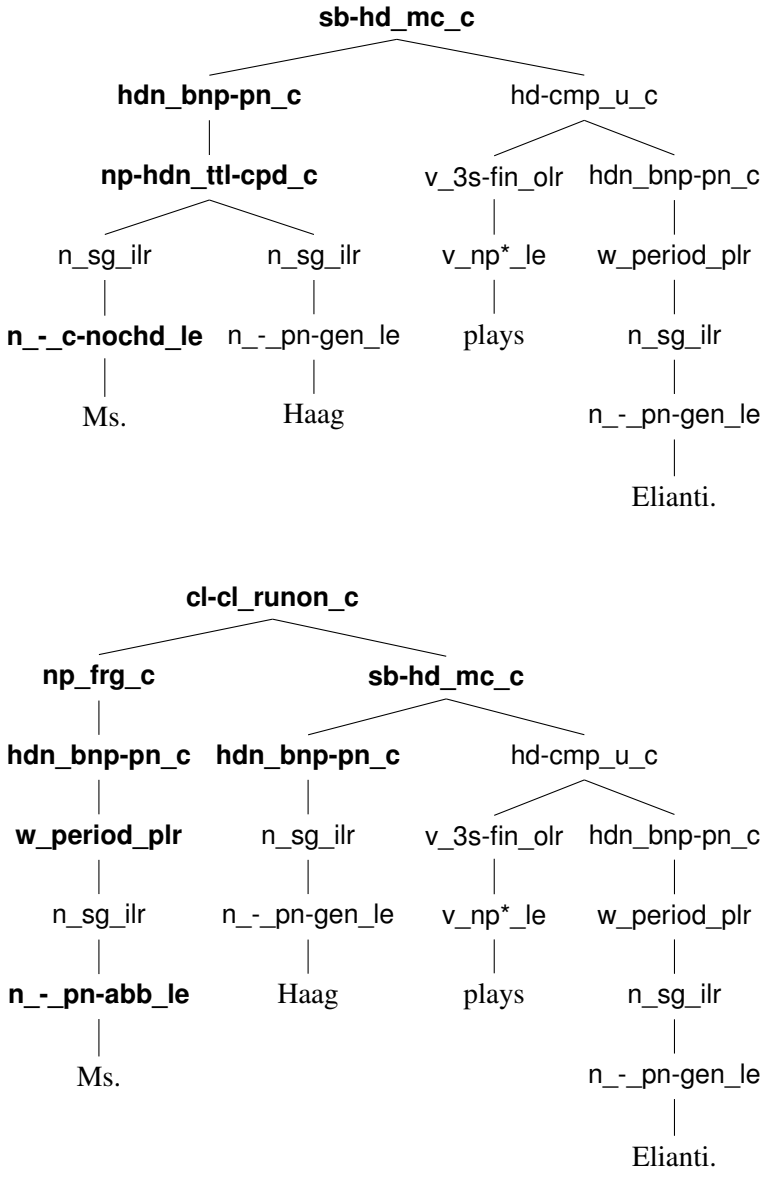


Figure 1: ERG derivation trees for our running example.

discriminant is also labelled, but with a slightly simplified label set compared to that just described. In the topicalized variant "In the park, I see a man." the phrase "I see a man" receives the label S/PP (denoting a sentence containing a 'gap' for the extracted PP). In this final type we strip the trailing slash category,

3 Experimental Protocol

We will evaluate different types of simplified discriminants both intrinsically and extrinsically, using the DeepBank (Flickinger et al. [4]) reannotation of the venerable Wall Street Journal corpus in the LinGO Redwoods framework.

For the intrinsic evaluation we will compute what we term the *discriminative potential* for each type of discriminant. Using gold-standard DeepBank annotations, we can construct an oracle to decide whether a discriminant is good (the correct tree has this property) or bad (the correct tree does not have this property). Then, for each sentence in the corpus we can compute the ratio of the number of trees removed in the presence of the oracle (r) to the number of non-gold analyses generated by the grammar (g). 'Strong' discriminant types will score higher, as they are able to prune away a larger fraction of non-gold trees than the less powerful discriminants. We will then evaluate the discriminative potential of a discriminant type as the mean r/g over the DeepBank corpus. Additionally we will take note of the number of sentences that can be fully disambiguated by a discriminant and the number of sentences where no distinctions can be made.

Our extrinsic evaluation metric will be the performance of parse rankers trained on partially disambiguated data. In the LinGO ecosystem, a presumed correct parse is selected from the parse forest generated by the grammar by a discriminative maximum entropy ranker, as described by Toutanova et al. [11]. Normally the parse ranker is trained on fully disambiguated sentences, but it is equally possible to train on a partially disambiguated forest. Partially disambiguated training data will obviously make available to the ranker less information, but it will nevertheless convey important information about preferred vs. dispreferred parse types, especially for discriminant types that are able to prune away large parts of the forest.

We will create the partial forests using essentially the same technique as we use to compute the discriminative potential of a discriminant type, marking parses that are excluded by the discriminant oracle as dispreferred and leaving the remainder of the parses as preferred. We will then use the resulting modified treebanks, DeepBank Sections 00 through 20, to train parse rankers, and evaluate them on Section 21 using common metrics for this problem, the fraction of sentences where the correct parse is ranked the highest (sentence accuracy), and the mean ParseEval score when comparing the top-ranked parse with the gold parse from the treebank.

Type	Mean (%)	Median (%)	Complete	None
Labelled span	96.8	99.4	6 745	312
Simple labelled span	96.3	99.2	5 724	367
Unlabelled span	90.6	96.6	1 458	898
Simple lexical	53.0	57.7	410	2 930
Lexical type	86.3	92.6	2 323	397

Table 1: Discrimination rates on WSJ00–19

4 Results

The results of our intrinsic evaluation are shown in Table 1; to avoid artificially inflating the values, we do not count sentences where all trees licensed by the grammar are marked as gold. This leaves us with a total of 33650 out of 34105 sentences in the first 20 sections of DeepBank. The distribution of the values themselves are not terribly surprising: the more information, the better the discrimination rate. As shown by the median values, the distributions are clearly not normal, with a small peak caused by the sentences where no disambiguation is possible.

There is also a dramatic drop when going from the very detailed lexical types of the ERG to the simplified PoS tagset, from an average 86% for the full lexical types to 53% for the simple tagset. Still structural knowledge is more powerful, with the unlabelled spans outperforming the full lexical types by some 5 percentage points. Structure is still more important to syntax than detailed lexical information. Also of some interest is the difference (or lack thereof) between the full and simple labelled span types; there is some benefit from the slashes in the labels, but the drop in mean discrimination is only about half a percentage point. Still, the difference in fully disambiguated sentences is 1000, about 3% of the corpus.

The results of the extrinsic evaluation are shown in Table 2, with the correlations between discrimination rate and ranker performance shown in Figure 2. There is a very marked drop going from the 'baseline' ranker, trained on the fully disambiguated treebank, to even the ranker trained on data disambiguated by the labelled span discriminant. Once again, the simple and full labelled span discriminant are neck and neck in performance, and likewise the unlabelled spans and full lexical types being relatively similar. The simple lexical types are, as expected, quite a ways behind the other types.

It appears that much of the information required for high ranker performance may be in the very fine distinctions discernible only in a fully disambiguated treebank, but in contrast to the 'baseline' ranker we have yet to tune the hyper-parameters of the models trained on partially disambiguated treebanks.[3] One possi-

[3]For experiments on the scale reported here, exploring the space of plausible hyper-parameters in the discriminative learning set-up is computationally rather costly. For the current results, we merely applied the hyper-parameters found by Zhang et al. [12] for training on fully disambiguated data.

Type	SA (%)	PE (%)
Baseline	39.5	96.8
Labelled span	16.9	86.6
Simple labelled span	15.4	86.2
Unlabelled span	10.4	81.7
Simple lexical	5.45	64.5
Lexical type	9.61	72.9

Table 2: Extrinsic evaluation results. Sentence accuracy (SA) and ParsEval (PE) scores.

ble interpretation of this is that the information required to eliminate clearly wrong interpretations of a sentence are relatively easy to acquire. The finer distinctions on the other hand, such as choosing between high and low attachment for prepositional phrases is far harder to come by. This tendency is reflected in the correlation curve, where better training data has relatively little impact on performance, until the critical point of about 95% discrimination is reached, at which point ranker performance sky-rockets.

5 Conclusions and Future Work

In our estimation, simplified discriminants clearly have the potential to be a useful tool in grammar-driven treebanking, enabling the use of annotators without years of experience in syntactic theory and the particular grammar used. Furthermore, knowing the relative strengths of the different kinds of discriminant should have implications in the design of treebanking tools. To our knowledge, there have been no formal studies of the impact of user interface on annotation efficiency, but just like preprocessing quality can have an important impact on speed (cf. Fort and Sagot [5] for morphological annotation and Skjærholt [10] for syntax) it should be possible to leverage this information in order to make grammar-based annotation more efficient. And while our experiments are grounded in the LinGO ecosystem of HPSG tools, we believe these results should generalise well to other formalisms.

The parse ranker results are less satisfying so far. While we did hypothesise a non-linear correlation between discrimination, the extreme effects we did observe are something of a disappointment (but see Footnote 3). While there is some potential for improved results with a more tailored approach to the ranker learning, the general shape of the learning curve is not likely to change appreciably. Thus, it is not likely that partially disambiguated data alone is enough to train an adequate parse ranker. However, there is some potential for the use of partially disambiguated data as additional data in a domain adaptation setting.

There are several interesting avenues of further work following on this. First of all, it remains to be determined whether the trends observed in our extrinsic

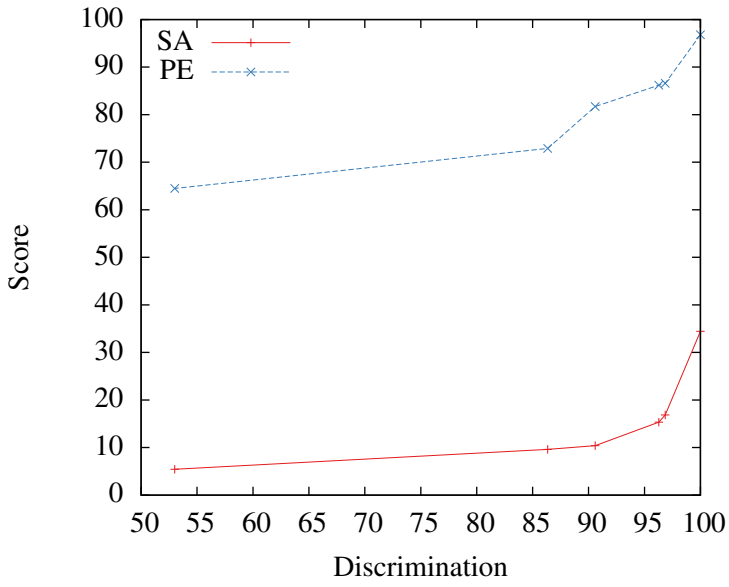


Figure 2: Correlation between disambiguation and ranker performance

evaluation remain true once we complete tuning of hyper-parameters in training from partially disambiguated treebanks. Second, it would be interesting to compare these results with similar discriminant types in other frameworks, and in particular how discriminants like LFG's morphological discriminants, not applicable in the exact same form to HPSG, compare to the types covered in this work. Third, we have not investigated the interaction of these simplified discriminants. For example, it would be very interesting to see how the combination of simplified lexical types and unlabelled spans perform. We did not perform these experiments as our experiments are computationally quite resource-intensive, and constraints on both time and available compute power necessitated a slightly limited scope. Finally, and arguably most importantly, we will seek to shed light on how easy or difficult different discrimant types are to judge reliably by non-experts, e.g. undergraduate students and ultimately crowd-sourcing workers.

Acknowledgements

We are grateful to our colleagues at the Oslo Language Technology Group and within the larger Deep Linguistic Processing with HPSG Initiative (DELPH-IN) for many fruitful discussions, as well as to three anonymous reviewers for insightful comments. In particular, we would like to thank Johan Benum Evensberget for his input, and Dan Flickinger for making available the grammar and treebanks. Large-scale experimentation and engineering is made possible through access to the ABEL high-performance computing facilities at the University of Oslo, and we

are grateful to the Scientific Computing staff at UiO, as well as to the Norwegian Metacenter for Computational Science and the Norwegian tax payer.

References

[1] Leonoor van der Beek, Gosse Bouma, Rob Malouf, and Gertjan van Noord. The Alpino dependency treebank. In Mariët Theune, Anton Nijholt, and Hendri Hondorp, editors, *Computational Linguistics in the Netherlands 2001. Selected papers from the Twelfth CLIN Meeting*. Rodopi, Amsterdam, The Netherlands, 2002.

[2] David Carter. The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, page 9 – 15, Madrid, Spain, 1997.

[3] Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6 (1):15 – 28, 2000.

[4] Dan Flickinger, Yi Zhang, and Valia Kordoni. DeepBank. A dynamically annotated treebank of the Wall Street Journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, page 85 – 96, Lisbon, Portugal, 2012. Edições Colibri.

[5] Karën Fort and Benoît Sagot. Influence of pre-annotation on pos-tagged corpus development. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 56–63, Uppsala, Sweden, 2010.

[6] Stephan Oepen and Jan Tore Lønning. Discriminant-based MRS banking. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, page 1250 – 1255, Genoa, Italy, 2006.

[7] Stephan Oepen, Daniel Flickinger, Kristina Toutanova, and Christopher D. Manning. LinGO Redwoods. A rich and dynamic treebank for HPSG. *Research on Language and Computation*, 2(4):575 – 596, 2004.

[8] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, page 2089 – 2096, Istanbul, Turkey, May 2012.

[9] Victoria Rosén, Paul Meurer, and Koenraad De Smedt. Designing and implementing discriminants for LFG grammars. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the 12th International LFG Conference*, Stanford, USA, 2007.

[10] Arne Skjærholt. Influence of preprocessing on dependency syntax annotation: Speed and agreement. In *Proceedings of the Seventh Linguistic Annotation Workshop and Interoperability with Discourse*, page 28 – 32, Sofia, Bulgaria, 2013.

[11] Kristina Toutanova, Christopher D. Manning, Dan Flickinger, and Stephan Oepen. Stochastic HPSG Parse Disambiguation using the Redwoods Corpus. *Research on Language and Computation*, 3:83 – 105, 2005.

[12] Yi Zhang, Stephan Oepen, and John Carroll. Efficiency in unification-based n-best parsing. In *Proceedings of the 10th International Conference on Parsing Technologies*, page 48 – 59, Prague, Czech Republic, July 2007.

A grammar-licensed treebank of Czech*

Tomáš Jelínek[1], Vladimír Petkevič[1], Alexandr Rosen[1],
Hana Skoumalová[1], Přemysl Vítovec[2] and Jiří Znamenáček[2]

Institute of Theoretical and Computational Linguistics
Faculty of Arts, Charles University, Prague
[1]E-mail: first.last@ff.cuni.cz
[2]E-mail: first.last@gmail.com

Abstract

We describe main features of a treebank of Czech, licensed by an HPSG-style grammar. The grammar interacts with corpus texts, preprocessed by morphological analysis and morphological disambiguation, a (largely) stochastic dependency parser and subsequent transformation into phrase-structure trees. The resulting trees, including functional and categorial information on words and phrases, are represented as typed feature structures. The grammar cooperates with a valency lexicon: the actual data are matched with surface valency frames, derived by lexical rules from predicate-argument structures in the lexicon. If the match is successful, the resulting annotation is enriched with information derived from the parsed data and from the lexicon. The paper concludes with an evaluation of the individual processing steps.

1 Introduction

There may be different opinions about the status of *langue* and *parole*, reflected both in linguistics and in NLP by the preference of rationalist or empirical methods, but the continuing coexistence of (hand-crafted) grammars as an approximation of language as a system and (real-text) corpora as an expression of language use indicates that the two notions are (indeed) two sides of a coin. We believe that a corpus, annotated by theoretically motivated and explicitly defined categories and structures, may in fact be that coin. The empirical and the theoretical sides of linguistics meet in the annotation of a corpus.

There is not a single proper way for such an annotation. Many potential users may prefer categories common in traditional grammar books but not much used in NLP, corpus linguistics or most linguistic theories, such as those treating analytical

*Work on this paper was supported by grant no. GAČR P406/13-27184S.

verbal forms as a part of morphology. We propose a solution that reconciles the paradigmatic and syntagmatic view of complex forms.

The definition of the annotation scheme and content is a de facto competence grammar. In addition to its theoretical appeal, the formal definition may support checking of both the data and the grammar, help to formulate efficient queries, offer concordances as correctly displayed structures, provide conversions to different representations, and assist grammar development.

In this paper, we focus on the way the grammar interacts with the corpus texts, pre-processed by a (largely) stochastic tool. First, the general approach is outlined and the objectives of developing a grammar-licensed treebank are specified in §2. Then the steps of processing an input text, including stochastic parsing, are briefly described in §3. Next, §4 focuses on the annotation format: a constituency-based tree structure, including categorial information in words and phrases. The core of the paper is a description of the grammar and the lexicon (§5). As an important step, grammar rules ("principles") make sure that the valency information is matched with actual data (i.e., that valency requirements are satisfied). Finally, §6 is devoted to an evaluation of several processing stages, before some issues and perspectives are discussed in §7.

2 An overview of the approach

In addition to an annotation scheme, we want the treebank to be consistent with theoretically motivated linguistic patterns. The collection of constraints on the annotation format and its content are viewed as a formal grammar. The grammar necessarily undergenerates due to *langue/parole* differences, insufficient coverage of *langue*, or simply because of errors in the data or in a previous processing step. Rather than excluding sentences failing the grammaticality test from the treebank, such sentences are flagged and reasons for the failure are examined.

However, there is no guarantee that sentences passing the test are 100% correct either. The grammar overgenerates – the space of possible language expressions is restricted by specifying phenomena-specific constraints, leaving difficult phenomena unchecked. In fact, the grammar, due to its modularity, can be set to a more or less restrictive mode. This scenario also allows for modular and incremental grammar development.[1]

The grammar augments the annotation produced by the stochastic parser and checks its consistency. It does so (i) by matching lexical information (esp. valency), obtained from external lexica, with the parsed texts and (ii) by projecting lexical information into phrasal nodes.

By matching treebank data with constraints of the grammar we can detect errors in the treebank data, the grammar and the lexical specifications, explore un-

[1] In the terminology of [7] the grammar operates in a satisfiability-based rather than validity-based setup: a string is licensed if all constraints in the grammar are satisfied. The more constraints, the fewer strings are licensed; an empty grammar licenses any string.

derstudied linguistic phenomena, and thus provide necessary feedback leading to improvements in the grammar, the lexicon and the treebank annotation.

In order to handle large data (in the order of hundreds of millions of word tokens), the treebank annotation is generated by software tools only, without any manual intervention, including (stochastic) parsing and (grammar-based) checking.

To satisfy users (and applications) of different tastes, the resulting annotation is available in different formats (e.g. a constituency-based or dependency-based one), both for viewing and export.

Many treebanks are built only by human effort and/or by applying stochastic parsers, but we are not unique in the use of hand-crafted rule-based methods either. There are a number of projects relying on hand-crafted lexica, such as *PDT-VALLEX* [2], *CCGbank* [5], *FrameNet* [12], *PropBank* [1], *TüBa-D/Z Valency Lexicon* [3], or grammars, such as *LinGO Redwoods* [11], *Alpino* [21], *Norgram* [16], *BulTreeBank* [17], or *Składnica* [23]. Our approach is different in that it combines stochastic parse of unrestricted texts with hand-crafted grammar and lexicon.

3 Processing the input text

The input text is morphologically analyzed and tagged by a combination of a rule-based disambiguation module and a stochastic tagger. It is then parsed by MaltParser, a stochastic dependency parser [10]. In order to improve the accuracy of the parser, the text is partially lexically simplified: several groups of syntactically equivalent words, e.g. personal proper nouns or numerals, are replaced by equivalent proxies, reducing lexical and formal variability of the text by ca. 20%. The parser is trained on a text simplified in this way [6]. The result, with the original forms restored, is a skeletal parse with functional annotation. Dependency trees are then converted to phrase-structure trees, where each terminal node is assigned a morphosyntactic description and a syntactic function. Finally, the morphological and syntactic annotation are transformed into a typed feature structure in the HPSG style.

For a sample sentence (1) we show its dependency structure in the PDT format [13], produced by the stochastic parser (Fig. 1), and its phrase structure (Fig. 2) after the conversion.

(1) Přemýšlel jsem o dobru a zlu.
Thought AUX-1SG about good and evil.
'I was thinking about good and evil.'

In Fig. 1, every node except for the top node is assigned a word form and a syntactic function: Pred is assigned to the verbal predicate of the sentence, conjunction *a* represents Coord(ination) of objects as conjuncts (Obj_Co), while AuxV, AuxP and AuxK denote auxiliary verb, preposition and full stop, respectively.

The phrase structure in Fig. 2 shows terminal and nonterminal nodes with their syntactic functions. Three kinds of heads are distinguished: Head, DeepHead and

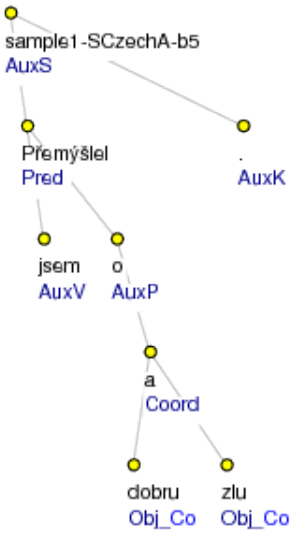


Figure 1: Input dependency structure of sentence (1)

SurfHead, where SurfHead is assigned to the auxiliary verb *jsem* and the preposition *o*, DeepHead to the VP headed by the content verb *přemýšlel* and to the coordinated structure consisting of two conjuncts (assigned the CoMemb function) and a coordinating conjunction *a* (CoConj). The prepositional phrase *o dobru a zlu* 'about good and evil' is Obj(ect) of the verb *přemýšlel*. (Syntactic annotation is discussed in §4.)

In the current setup, the parser produces a single fully disambiguated result for each sentence,[2] although the resulting functional labels may fold some cases of structural ambiguities, such as PP-attachment. In the next step (augmentation and

[2]We plan to test an n-best or voting scenario, involving several parsers.

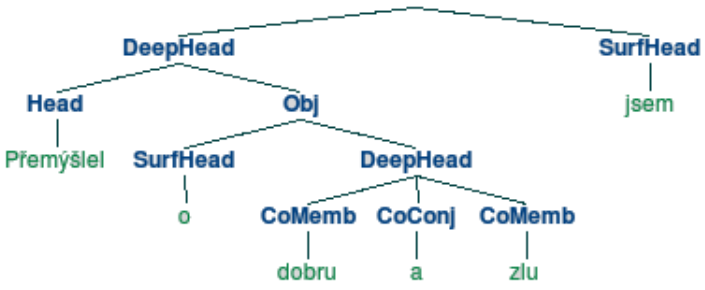


Figure 2: Output phrase-structure tree for sentence (1)

checking by the grammar), any such ambiguities are either resolved, or preserved,[3] while some additional ambiguities may occur when more information is added (e.g. with a more detailed classification of adverbials). Most of these ambiguities are represented as underspecification rather than disjunction.

4 The annotation format

Linguistic categories and structures in the data are modeled as typed feature structures using: (i) format declaration for lexical and phrasal categories as a type hierarchy with multiple inheritance; and (ii) constraints on complex relationships between the values, representing various phenomena (agreement, government, grammatical control).

Each word form is assigned appropriate cross-classifying properties of three kinds: morphological (inflectional), syntactic and semantic (lexical) (cf. [15]). Traditional parts of speech are, in fact, a mix of these three criteria. For some word classes the three criteria coincide: nouns refer to entities (semantic criterion), decline according to nominal paradigms (morphological criterion) and occur in nominal positions (syntactic criterion). In Czech, their morphological categories reflect syntactic function by case and an approximation of cardinality by grammatical number, while they show the inherent (lexical) property of grammatical gender. On the other hand, numerals and pronouns are defined by purely semantic criteria, but in other aspects cardinal numerals and personal pronouns behave like nouns, whereas ordinal numerals and possessive pronouns behave like adjectives, including the appropriate morphological categories.

In our cross-classification we also model some regular derivational relations: deverbal nouns and adjectives (inflectional classes) are derived from verbs (lexical class), while possessive adjectives are derived from nouns. An example of morphological annotation is presented in Fig. 3 below.

iNoun_lVerb_sNoun	
iLemma	***neprověření***
iNum	*pl*
iGend	*n*
iCase	*inst*
lLemma	***prověřit***
lPol	*minus*
lAspect	*perf*

Figure 3: Morphological annotation of the word form *neprověřeními* '(with) non-verifications' as a typed feature structure

[3] The latter option is not implemented yet.

In Fig. 3 the three-dimensional typed feature structure for the word form *neprověřeními* '(with) non-verifications' is shown: the form is a negative plural instrumental form of a neuter deverbal noun: *neprověřeními* (thus inflectional Number (iNum) is *pl*ural, inflectional Gender (iGend) is specified as *n*euter, which is true for all deverbal nouns, and inflectional Case (iCase) is equal to *inst*rumental). It declines as a noun ("inflectional noun" is iNoun), it is derived by a regular derivation from the lexical Verb *prověřit* 'verify' (lVerb) and it has a nominal syntactic distribution (sNoun). In a typed feature structure used for the description of linguistic objects, this triple of values is rendered as a complex type *iNoun_lVerb_sNoun*. The derivation is also reflected in two kinds of lemmas: inflectional lemma (iLemma) *neprověření*) and lexical/semantic lemma (lLemma) *prověřit*). The lexical properties of polarity (lPol) and aspect (lAspect) specify the base verb as negated and perfective.

An additional dimension is used to identify morphological categories of analytical verb forms since we want to accommodate interpretations of expressions consisting of one or more function (auxiliary) words and a content word, mainly those traditionally viewed as belonging to the domain of analytical morphology. This "analytical" word class dimension specifies categories such as tense, mood, and voice.

The phrase-structure trees use a combination of binary and flat branching. The constituents are assigned syntactic functions, such as subject, attribute, object, etc. (cf. Fig. 2). In headed structures,[4] the number and properties of non-head daughters are determined by the valency of the head daughter (assigned the Head syntactic function). Optional modifiers are added non-deterministically to the list of complements.[5] Heads can be further distinguished as surface and deep. This distinction is used in phrases including function words: e.g., in prepositional phrases, where the preposition is the surface head (SurfHead) and the noun phrase is the deep head (DeepHead). An unqualified head is both a surface and a deep head. This type of classification of heads allows for interpreting syntactic structures according to users' preferences as constituency or dependency trees, and as surface or deep structures.

5 The grammar

The grammar treats the result of the stochastic parser as a set of constraints, assumed to be true about a sentence. The data format and the constraints are checked for consistency, and additional constraints originating in the grammar and external lexica are matched with those present in the data. As a result, lexical information is projected from the leaves of the syntactic tree to phrasal categories. At the same time, morphological categories, syntactic structure and syntactic functions

[4]There are also unheaded structures: coordination, adordination, unspecified (for some multiword units and other non-standard strings).

[5]This is a solution commonly used in HPSG grammars, cf. e.g. [14].

proposed by the parser are checked.

Information added from external lexica concerns mainly valency. Valency frames actually used in the treebank are derived from more general specifications in the external lexica by a set of lexical rules. The grammar and the lexical rules are specified and implemented in Trale, a formalism for implementing HPSG grammars.[6]

5.1 Valency lexicon

At the moment, two external valency dictionaries are used: VALLEX [9] and PDT-VALLEX [2], including about 5,000 and 10,000 verbs, respectively, with their deep valency frames and surface morphological realizations.[7] The frames reflect the Praguian valency theory of the Functional Generative Description.[8] The frames are used to check whether valency requirements of the verbs used in the parsed sentence are met.

The frames consist of lists of arguments together with their surface realization in an active sentence. The frames are converted to attribute-value matrices (AVM's), used by lexical rules, specifying valency properties for derived forms. The rules are implemented as a Trale grammar (see §5 above). The grammar produces surface frames for various forms of a given verb, namely finite indicative/imperative, passive participle, past participle, infinitive and transgressive. Different diatheses exist for these verb forms, including reflexivization. The rules are based on the following assumptions about Czech verbs (cf. [18] and [20]): (i) non-reflexive verbs in the active voice have identical deep and surface frame; (ii) reflexive verbs and verbs in deagentive constructions add a reflexive particle slot to their surface valency frame; (iii) inherent reflexives have only the active voice; (iv) non-reflexive verbs with at least two arguments, where Actor is realized by a structural case,[9] can form periphrastic (analytical) passive; (v) non-reflexive verbs with at least one argument can form a deagentive construction; (vi) in the active voice, Actor realized by a structural case is the subject while other arguments are objects; (vii) in the passive voice and deagentive constructions, an argument different from Actor and realized by a structural case is the subject, while Actor realized by a structural case is omitted on the surface.

[6] See `http://www.sfs.uni-tuebingen.de/hpsg/archive/projects/trale/`

[7] Verbalex [4], a valency dictionary larger than VALLEX, is another candidate. It is based on a different theory and the arguments are assigned semantic roles. We plan to use this resource as well as to offer a different classification of the arguments.

[8] The FGD theory assumes deep and surface levels of syntax. On the deep 'tectogrammatical' level, every verb frame contains up to five arguments (actants), which correspond to subject and objects on the surface level, and obligatory free modifications, which correspond to adjuncts. The arguments are called Actor, Patient, Addressee, Origin and Effect.

[9] Structural case is a case which is not assigned in the lexicon, but depends on the diathesis and/or negation of the verb. Typically, direct object in an active sentence has Accusative case, but in passive sentence, it becomes Subject in Nominative case. In Czech, there also exist Genitive of Negation and Partitive Genitive.

The surface frames are used to check the saturation of valency. As the source lexica do not contain information on possible passivization, the lexical rules overgenerate for some diatheses. However, hypothetical passive forms of some verbs may never occur in the data.

5.2 Principles of grammar

As usual in HPSG grammars, the main parts, determining the syntactic skeleton of annotation, are performed by the Head Feature Principle (HFP) and the Valency Principle (ValP).[10] HFP propagates head information from lexical heads up to nonterminal nodes in the tree. ValP makes sure that saturation of valency frames is checked. If this check/match is successful, arguments receive additional information, such as deep syntactic function.

Other principles are in charge of properly distributing morphosyntactic information within lexical items. This includes mainly the sharing of values determined by the form itself and its valency requirements. The lexical rules in §5.1, handling the relation between deep and surface valency, can be seen as expressing the Linking Principle. The Surface Function Assignment Principle, the Case Principle and the Agreement Principle take care of other respective phenomena.

Due to the absence of other principles, e.g. those governing word order, the grammar overgenerates, but this will be gradually remedied as the grammar is developed further.

5.3 Analytical verb forms

Standard grammars of Czech treat analytical forms as a morphological rather than syntactic phenomenon. Tense, mood and voice are seen as morphological categories, interpreting grammatical meanings of content verbs, often requiring auxiliary forms. On the other hand, most approaches within corpus or theoretical linguistics assume that morphological categories do not span word boundaries.

To reconcile these two perspectives, content words have an additional analytical dimension with the three properties of tense, mood and voice. Their values are determined by the grammar, operating on specifications from the lexicon, rather than by any device targeting individual orthographical words. The grammar treats analytical forms as syntactic phrases, consisting of a function word as the surface head and a content word (or a phrase including a content word), as the deep head.[11] The details are encoded in the lexical specifications of function words, the rest is the task of a general valency satisfaction mechanism (ValP) and a rule projecting features of the head daughter to its phrasal mother (HFP).

[10] In addition to the definition of possible data types and their properties in the signature part of the grammar.

[11] It is the surface head which is the parallel of *head* in standard HPSG grammars. The head of the whole structure is the finite (personal) form. This is true also about analytical forms including more than one auxiliary, such as past conditional: the (surface) head of *byl bys přijel* 'you would have arrived' is the 2nd position clitic *bys* 'would'.

Other multi-word units, such as idioms, are also annotated by co-indexing the relevant lexical nodes, which do not always form a phrase (subtree). The annotation is based on a collocation lexicon, derived from [22], see [8].

6 Evaluation

Four key stages of processing an input sentence were evaluated: (i) accuracy of morphological annotation; (ii) accuracy of stochastic dependency parsing against the data from the Prague Dependency Treebank (PDT);[12] (iii) accuracy of the conversion of dependency parses to constituency-based structures, and (iv) accuracy of the conversion of the string-like morphological and POS tags to typed feature structures;

(i) **Morphological annotation**. The accuracy of morphological annotation including morphological analysis and disambiguation (the rule-based cooperating with a stochastic one) is 94.57% (cf. [19]).

(ii) **Stochastic dependency parsing**. MaltParser was tested automatically on the d-test data set of the PDT. Using the method of text simplification (see §3 above), it achieves an unlabeled accuracy of 86.76% and a labeled accuracy of 81.21% on single tokens. If counted for entire sentences, the accuracy is 44.52% (unlabeled) and 34.01% (labeled, i.e. a third of sentences is parsed correctly).

(iii) **Conversion from dependency to phrase structures**. The conversion of 400 sentences whose dependency structure was manually annotated in accordance with the PDT Annotation Manual[13] was manually checked. 21 input sentences (5.25%) were assigned incorrect dependency annotation, but we evaluated only the conversion proper, regardless of whether the input dependency structure was correctly annotated or not. There were 15 sentences (3.75%) incorrectly converted due to the following reasons: (a) subject is expressed by a prepositional phrase, (b) subject is a coordination, itself modified by a coordination of attributes, (c) a coordination of verbal predicates is combined with overt subject, (d) a reflexive particle does not form a constituent with its base verb, (e) errors in transformation rules: (e1) a verbal attribute modifies a noun rather than a verb, (e2) cardinal numerals are converted as heads rather than as surface heads. Moreover, 57 input sentences included the function label ExD (*Ex*ternal *D*ependent) attached to a node lacking its mother node (ellipsis). Such structures were correctly converted by a default rule, while ExD structures should be treated separately (see below).

The errors of type (a)–(f) can be rectified in a relatively simple way. As for ExD, there are two solutions: in most elliptical structures the mother node of a converted ExD element will have the gap feature flagged; the remaining cases, such as comparative structures (*němý jako ryba* 'mute as a fish'), will be annotated as non-elliptical.

(iv) **Conversion of the string-like morphological and POS tags to typed feature**

[12] See http://ufal.mff.cuni.cz/pdt3.0

[13] See https://ufal.mff.cuni.cz/pdt2.0/doc/manuals/cz/a-layer/html

structures. The total of 200 input word forms were manually checked. On input, every word form was assigned a lemma and a string-like morphological tag. This annotation was transformed to a typed feature structure (Fig. 3 in §4). The only conversion problems were caused by lexical lemmas (lLemma, see also §4 above) of deverbal nouns and deverbal adjectives: 5 word forms were assigned incorrect lexical lemmas. This can be remedied by corrections of conversion tables.

7 Discussion and perspectives

The annotation of language structures in the treebank of Czech and the main features of the formal grammar licensing these structures is the core of the project. Despite the limited coverage of grammatical phenomena the feasibility of the concept has already been verified on less complex syntactic structures.

The grammar is still very permissive but this will be gradually remedied by more grammatical constraints (e.g. for capturing word order phenomena). We shall also be concerned with the issue of preserving/restoring structural ambiguity in the data. However, the perspective of further development of the treebank data annotation and the grammar concern all stages of processing input data. Especially the quality of parsing should be further improved: by tuning the parser to cope with specific features of Czech and by using multiple parsers in a voting scenario. The valency lexicon will be further developed in several directions: we will continue to test valency frames in the VALLEX lexicon (and other lexica such as [4]) on real data and add frames for more verbs will be added. Moreover, the pilot search and data visualisation and export/import options will be developed to interact with the corpus data and to handle various visualization formats such constituency vs. dependency, surface and deep structure).

References

[1] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *36th Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 86–90, Montréal, 1998.

[2] Jan Hajič, Jarmila Panevová, Zdeňka Urešová, Alevtina Bémová, and Petr Pajas. PDT-VALLEX: Creating a large-coverage valency lexicon for treebank annotation. In *Proceedings of The Second Workshop on Treebanks and Linguistic Theories*, pages 57–68. Växjö University Press, 2003.

[3] Erhard W. Hinrichs and Heike Telljohann. Constructing a valence lexicon for a treebank of German. In *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories*, pages 41–52, 2009.

[4] Dana Hlaváčková and Aleš Horák. Verbalex - new comprehensive lexicon of verb valencies for Czech. In *Computer Treatment of Slavic and East European Languages*, pages 107–115. Slovenský národný korpus, Bratislava, Slovakia, 2006.

[5] Julia Hockenmaier and Mark Steedman. CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3):355–396, September 2007.

[6] Tomáš Jelínek. Improvements to dependency parsing using automatic simplification of data. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014. European Language Resources Association (ELRA).

[7] Mark Johnson. Two ways of formalizing grammars. *Linguistics and Philosophy*, 17(3):221–248, 1994.

[8] Marie Kopřivová and Milena Hnátková. From a dictionary to a corpus. In *Proceedings of Europhras 2012*, Maribor, 2012. In print.

[9] Markéta Lopatková, Zdeněk Žabokrtský, and Václava Kettnerová. *Valenční slovník českých sloves*. Univerzita Karlova v Praze, Nakladatelství Karolinum, Praha, 2008.

[10] Joakim Nivre, Johan Hall, and Jens Nilsson. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC-2006*, pages 2216–2219, Genova, 2006. ELRA.

[11] Stephan Oepen, Dan Flickinger, Kristina Toutanova, and Christoper D. Manning. LinGO Redwoods: A rich and dynamic treebank for HPSG. In *Proceedings of the Workshop on Treebanks and Linguistic Theories, September 20-21 (TLT02)*, Sozopol, Bulgaria, 2002.

[12] Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

[13] Prague Dependency Treebank, 2013. Version 3.0, http://ufal.mff.cuni.cz/pdt3.0/.

[14] Adam Przepiórkowski. On Complements and Adjuncts in Polish. In Robert D. Borsley and Adam Przepiórkowski, editors, *Slavic in HPSG*, Studies in constraint-based lexicalism, pages 183–210. CSLI Publications, Stanford, 1999.

[15] Alexandr Rosen. A 3D Taxonomy of Word Classes at Work. In *Proceedings of OLINCO 2014*, in print.

[16] Victoria Rosén, Koenraad de Smedt, and Paul Meurer. Towards a toolkit linking treebanking to grammar development. In *Proceedings of the 5th International Workshop on Treebanks and Linguistic Theories (TLT'05)*, Prague, Czech Republic, 2006.

[17] Kiril Simov, Petya Osenova, Sia Kolkovska, Elisaveta Balabanova, Dimitar Doikoff, Krassimira Ivanova, and Milen Kouylekov Alexander Simov. Building a linguistically interpreted corpus of Bulgarian: the BulTreeBank. In *Proceedings of LREC 2002*, pages 1729–1736, Canary Islands, Spain, 2002.

[18] Hana Skoumalová. *Czech syntactic lexicon*. PhD thesis, Charles University, 2001.

[19] Hana Skoumalová. Porovnání úspěšnosti tagování korpusů. In Vladimír Petkevič and Alexandr Rosen, editors, *Korpusová lingvistika Praha 2011: 3 – Gramatika a značkování korpusů*, volume 16 of *Studie z korpusové lingvistiky*, pages 199–207, Praha, 2011. Ústav Českého národního korpusu, Nakladatelství Lidové noviny.

[20] Zdeňka Urešová. *Valence sloves v Pražském závislostním korpusu*. Institute of Formal and Applied Linguistics, Prague, 2011.

[21] Leonoor Van der Beek, Gosse Bouma, Rob Malouf, and Gertjan Van Noord. The Alpino dependency treebank. *Language and Computers*, 45(1):8–22, 2002.

[22] František Čermák, Jiří Hronek, Jaroslav Machač, Renata Blatná, Miloslav Churavý, Vlasta Červená, Jan Holub, Marie Kopřivová, Libuše Kroupová, Vladimír Mejstřík, Milan Šára, and Alena Trnková. *Slovník české frazeologie a idiomatiky*, volume 1–4. LEDA, Praha, 2nd edition, 2009.

[23] Marek Świdziński and Marcin Woliński. Towards a bank of constituent parse trees for Polish. In *Proceedings of the 13th International Conference on Text, Speech and Dialogue*, TSD'10, pages 197–204, Berlin, Heidelberg, 2010. Springer-Verlag.

Evaluating Parse Error Detection across Varied Conditions

Amber Smith and Markus Dickinson

Department of Linguistics
Indiana University
E-mail: {smithamj,md7}@indiana.edu

Abstract

We investigate parse error detection methods under real-world conditions, outlining and testing different variables for evaluation and pointing to useful experimental practices. In particular, we focus on four different conversion methods, ten different training data sizes, two parsers, and three error detection methods. By comparing a set number of tokens across conditions, we measure error detection precision and revised labeled attachment scores to see the effect of each of the variables. We show the interactions between variables and the importance of accounting for parser choice and training data size (cf. initial parser quality). Importantly, we provide a framework for evaluating error detection and thus helping build large annotated corpora.

1 Introduction and Motivation

Automatic error detection of parses is potentially useful for helping build large, reliably-annotated corpora [7, 30] in a two-step process, with error detection speeding up the step of semi-automatically improving the annotation [14, 29]; for active learning, where parser training size continually increases [23]; or for parse revision [16], where the choice of base parser varies [13]. One difficulty in realizing these benefits has been in having a common framework for comparing and evaluating error detection methods, especially given the wide range of situations. Indeed, there are few studies directly comparing error detection methods, even for manual annotation [though, see 18]. We thus investigate how parse error detection methods work under varied conditions, outlining and testing different variables for evaluation and pointing to useful experimental conditions and evaluation metrics.

Training corpus size, for example, is known to affect parser accuracy [2, 10], but how does error detection interact with this? Of course, training data size is but one factor in error detection performance. To see some other factors, consider the related area of error detection for manual dependency annotation: the method in [3] identifies errors with precisions of 93% for Swedish, 60% for Czech, and

48% for German. The lower results for Czech relate to annotation scheme decisions for, e.g., coordination, and the results for German are likely due to both data size and scheme decisions. When other error detection methods [e.g., 12, 31] are considered, comparison seems nearly impossible, with: 1) different languages and annotation schemes, 2) different sizes of data, and 3) different parser choices, in the case of [31] but even more relevant for our task of parse error detection. Without accounting for these variables, one cannot properly evaluate method quality.

Our goal is to identify the variability in error detection effectiveness, as well as how evaluation metrics impact conclusions drawn from the data. We use four variables, described in Section 2 and listed in Table 1. While not exhaustive, these conditions represent variables we expect could have a big impact on the effectiveness of error detection. For error detection, we use two related methods and one unrelated one; this allows us to probe both internal development and comparison between different kinds of methods.

We have very practical aims, attempting to: 1) quantitatively measure the size of the impact of different variables and where we might expect to see one method outperform another; 2) identify metrics to indicate which method is better; 3) provide guidance in the selection of experimental settings, so that future work does not need to run 80 experiments for every new method, as we have; and 4) provide insight into corpus-building, by identifying where effort in annotation may best be spent; varying training size is especially important here.

2 Experimental Setup

Corpus Conditions An error detection method based on a larger training corpus might perform better, as it has many examples to learn from, but a better initial parser could affect the types of errors left to detect, negatively impacting detection accuracy. The combination of parser and error detection accuracy bears on the question of how much annotation effort to spend in obtaining an initial training corpus versus post-processing after parsing.

Scheme	Training Size		Parser	Method
lth07	02_{q1}	02-04	malt	high
lth08	02_{q2}	02-05	mst	b01
clear	02_{q3}	02-06		disagree
stan	02	02-07		
	02-03	02-15		

Table 1: Summary of conditions involved in the current experiments

To approach these questions, we use different sizes of training corpora. Given the number of varying conditions, we choose to use just the Wall Street Journal (WSJ) portion of the Penn Treebank [20]—which also has the benefit of having several conversion programs available (see below). We want to use very small training corpora, to match real-world conditions [cf. 11], so we break up section 02 of the WSJ: 02_{q1} for the first 25%, 02_{q2} for the first half, and 02_{q3} for the first 75%, splitting based on the nearest sentence end. We then combine section 02 with a number of following sections (e.g., *02-04* contains sections 02, 03, and 04). The

training sizes are in Table 2.[1]

We use three conversion programs to test annotation scheme impact. Stanford CoreNLP [5] uses the basic Stanford dependencies annotation scheme (*stan*), also used in a modified form by ClearNLP [*clear*, 4]. With the LTH Constituent-to-Dependency Conversion Tool [15], we use one setting imitating the annotation scheme from CoNLL 2007 [*lth07*, 24], and one from CoNLL 2008 [*lth08*, 28]. In the future, one may want to investigate specific decisions, e.g., regarding coordination [cf., e.g., 9, 26, 27].

Sec.	Sen.	Tokens
02_{q1}	508	12,003
02_{q2}	953	23,236
02_{q3}	1,479	36,110
02	1,989	48,134
02-03	3,469	83,779
02-04	5,734	138,308
02-05	7,868	190,114
02-06	9,695	234,467
02-07	11,858	285,824
02-15	27,487	656,975

Table 2: Size of training corpora

Finally, we use two dependency parsers, MSTParser [21] and MaltParser [25], with default settings. Both parsers are commonly used and readily available, and they consider different information in making parsing decisions [22]. This could make error detection methods perform differently. This complementarity also makes the parsers a good pair for error detection through parser disagreement, used below.

Error Detection Methods We use two distinct approaches to parse error detection, one threshold-based and one binary. First, we use the method outlined by [7], that of detecting anomalous parse structures (DAPS). This method is based on *n*-gram sequences of dependency structures. The method finds anomalous valency sequences, checking whether the same head category (e.g., verb) has a set of dependents similar to others of the same category. In short, the training and parsed testing corpora are first reduced to sets of rules that consist of a head and its dependents. The resulting rules from the testing corpus are then scored based on their similarity to rules for heads of the same category in the training corpus.

The scoring is based upon *n*-gram sequences within these rules, and we use two variants: 1) the high-gram method (*high*) uses all *n*-grams of length 3 or greater, ignoring bigrams when scoring, as bigrams do not encode sufficient context [see 7]; and 2) a weighted version of the all-gram method (*wall*), multiplying bigram counts by a dampening weight of 0.01, giving bigrams a small impact on the overall score. Thus, if a rule lacks a bigram, it is more likely that the rule is of poor quality than if it simply lacked a trigram [see also 16, 8].[2] Since the methods give scores for each token in testing, thresholds can be set to identify the positions most likely to be erroneous (see Section 3).

[1] *lth08* generated a few more tokens than the other conversions (6 more for 02_{q1}, 218 more for *02-15*), due to a different treatment of slashes. With only this small difference, we report size numbers for the other conversions throughout.

[2] The code here implements *high* by default and requires only minor modification for *wall*: `http://cl.indiana.edu/~md7/papers/dickinson-smith11.html`

Other error detection methods make a binary distinction: a dependency is an error or it is not [1, 31]. We implement a simple parser disagreement method (*disagree*), which compares the MST and Malt output to each other, flagging as potential errors any positions where the heads or dependency labels do not match. The method thus contrasts with the DAPS methods, as the set of flagged positions to evaluate is in some sense fixed by the method.

Metrics for Evaluation To evaluate error detection, we start with precision and recall, calculated in the standard ways. Since we want to reduce manual effort, i.e., minimize the number of false positives an annotator would have to examine, precision is a higher priority in this context. In addition, we propose evaluating results based on a set number of tokens (see Section 3), in which case recall becomes less informative as it is tied directly to precision.

While informative, these measures do not take into account the effect of baseline parser quality. Lower initial parser accuracy means more errors to identify, which generally results in higher error detection precision. For example, for a 5% segment of the testing data (see Section 3), the *clear.02.malt* setting has a baseline labeled attachment score[3] (LAS_b) of 81.9%, and error detection precision of 78.6% for the *high* condition. For *clear.02.mst*, we see an LAS_b of 83.1% and precision of 72.1%. Using MST results in a higher LAS_b but lower error detection precision.

To mitigate this problem and provide a practical measure for corpus-building, we introduce revised LAS (LAS_r) to take the baseline labeled attachment score (LAS_b) into account. LAS_r assumes that all identified corpus errors will be corrected and recalculates LAS for the additional correct tokens. While we acknowledge that annotators are unlikely to correct all identified errors, and may in fact introduce new errors, LAS_r nonetheless provides a useful estimation of the potential resulting corpus quality.

Another proposed metric measures "Accuracy Gain on Inspecting top x% edges" (AGI_x), which is the gain in LAS (LAS_r–LAS_b) divided by the percentage of the corpus examined (x) [14]. This is done to "take[] into account the human effort that goes into ... revision." While the amount of the corpus examined helps account for effort and normalize across different conditions—issues taken up in the next section—AGI_x works out to be the number of corrections divided by the number of positions examined, i.e., error detection precision.

3 Comparing Across Conditions

To evaluate error detection methods across varied scenarios, it is crucial to establish a consistent and reliable method of comparison. Such comparison is made difficult, however, by the fact that both score-based and binary error detection methods identify variable numbers of tokens across the different testing conditions, frequently with a difference of thousands of identified tokens between two conditions.

[3] LAS is the percentage of words with the correct attachment and label [17, ch. 6].

For example, setting a threshold of zero identifies as few as 150 tokens (*lth08.02-15.mst.wall*) or as high as 6080 tokens (*stan.02$_{q1}$.malt.high*). Similarly, the parser disagreement method identifies varying numbers of potential errors across the different scenarios, ranging from 6287 (*clear.02-15*) to 12,189 positions (*stan.02$_{q1}$*).

Consider the context again: one has limited annotation time and wants to optimize the number of corrections in that time. For evaluation, *time* can be approximated by specifying a pre-defined amount of the corpus as the amount of material annotators will be able to revise in that time. To represent a consistent human effort across conditions, one can evaluate results based on a set *segment* of the tokens in the testing corpus (see also the precision-at-*k* metric in information retrieval [19, ch. 8]). When using a fixed testing corpus size, these segments identify a set number of positions and therefore correspond directly to a notion of time.

The number of positions to correct, relative to the size of the testing corpus, can also have a big impact on the quality of the error detection method, but this impact tends to be relatively stable for a given percentage of the corpus. Even though we use only one testing corpus, there is in the general case a potential for different sizes of testing corpora across studies, and thus to evaluate error detection methods it may be best to set segment size by percentage for evaluation. Given this, we report results based on percentage-based segments (mainly 5%). Focusing on a relatively small portion of the data (5%) is to some extent arbitrary, but the smaller segments seem to be a reasonable amount of data for an annotator to examine, whereas larger percentages begin to be infeasible—especially for very large corpora of the kind we envision in the future (see also [8]).

With a set segment, the choice of evaluation metrics becomes simpler, as an increase in precision results in an increase in recall. Thus, we only report precision and LAS$_r$. The difference between binary and threshold-based identification also becomes relatively unimportant, providing the same number of tokens.

4 Results

The variables interact with each other, making it difficult to tease apart the contribution of each one. Still, we discuss each variable in turn.

Corpus and Segment Sizes We see scores for *clear.malt.high* in Table 3, as an example of general trends, the clearest trend being that larger corpora generally yield better parsers (LAS$_b$). While a small training corpus and a lower-quality parsing model should intuitively lead to easier-to-identify errors, here the best precision is not achieved with the smallest training corpora but with the *02-03* training set.

Corpus	P	LAS$_b$	LAS$_r$
02q1	72.3%	77.2%	80.8%
02q2	74.7%	79.6%	83.3%
02q3	77.7%	80.9%	84.8%
02	78.6%	81.9%	85.8%
02-03	**81.5%**	83.6%	87.7%
02-04	77.4%	84.7%	88.6%
02-05	77.1%	84.2%	89.1%
02-06	76.1%	85.6%	89.4%
02-07	75.5%	85.6%	89.4%
02-15	70.6%	87.0%	**90.5%**

Table 3: Precision, LAS$_b$, LAS$_r$ for *clear.malt.high*, 5% segment

In contrast, LAS_r for these same models consistently increases with training corpus size. In other words, once we balance error detection precision with the LAS_b for each model, the corrected corpus has a higher LAS_r with a larger training corpus.

With even this small sample, we see that error detection precision varies based on the size of the training corpus, but this impact is fairly predictable. Thus, just a few different training sizes should be sufficient when testing new methods.

Concerning segment size, for both threshold-based methods the precision consistently decreases as segment size increases (see Figure 1). The *disagree* method instead maintains relatively stable precision scores regardless of the segment size. Comparing the methods side-by-side, *high* outperforms *disagree* for a 5% segment but not higher segment sizes. Thus, it is important to test multiple segment sizes.

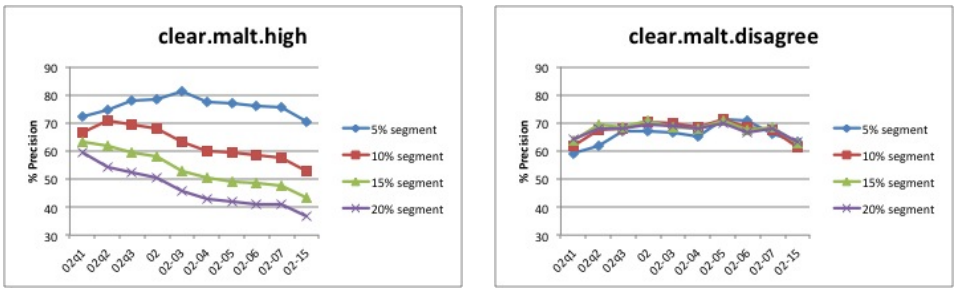


Figure 1: Precision scores for various percentage-based segment sizes.

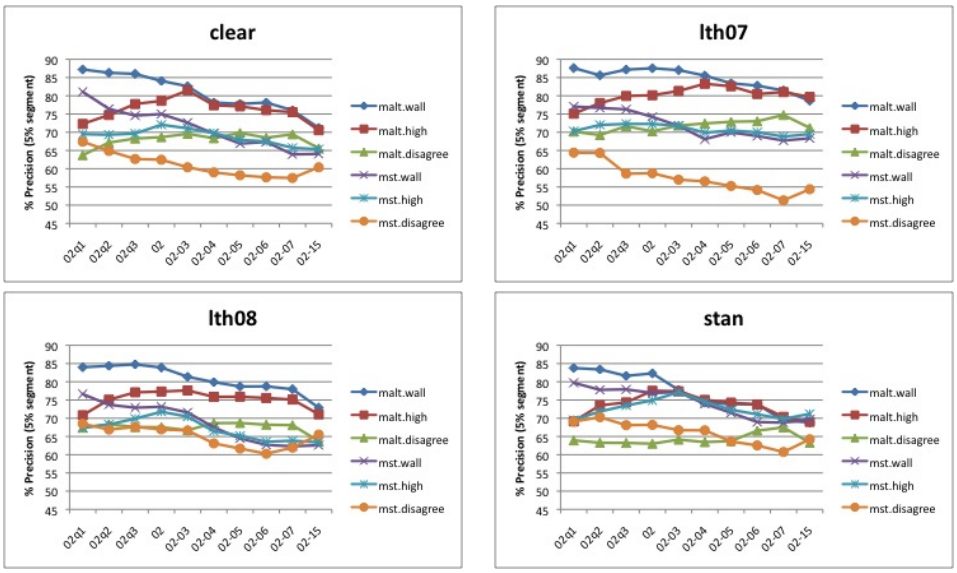


Figure 2: Precision for 5% segment of testing corpus, for two conversion schemes

Conversion Scheme Figure 2 shows the results by conversion scheme, where there is a fairly consistent shift in the precision patterns between *02* and *02-04*.

For the smallest training data sizes, error detection precision follows the pattern of *wall* > *high* > *disagree*, while for the largest training sets, we see *malt* > *mst*. In addition, for the *high* and *wall* methods, the results for a given parser are much closer for the larger training sets (e.g., *malt.wall* and *malt.high* precision scores are closer for *02-15* than for 02_{q1}). Despite this consistency, there can be sizable variation for similar conditions, such as a 12% gap at *02-07* in precision between *lth07* and *stan* for *malt.wall*, so conversion scheme cannot be ignored.

Parser The impact of parser choice can vary widely—likely due to how complementary the error detection model is to the parsing model. However, despite some highly varying error detection precision between parsers, LAS_r for the 5% segment of the *malt* and *mst* models barely differs once the other variables are fixed. In fact, the greatest difference is only 2.7% (for *lth07.02-07.high*).

Error Detection Methods While the main focus of this paper is on evaluation—and less on the merits of the error detection methods—the evaluation still reveals some interesting patterns. First, as previously mentioned, the method's impact is closely tied with the training corpus size. In Figure 2, there is a split in the methods around *02-03* and below (*wall* > *high* > *disagree*). However, from *02-03* on, the threshold-based methods have a fairly small difference. The evaluation thus provides feedback: (weighted) bigrams are useful for smaller training sets, where they make up for some data sparsity, but they cloud the more reliable longer *n*-grams of the *high* method for larger training corpora [see also 8]. In addition, *disagree* is in general not as reliable when focusing on a small portion of the testing corpus, but begins to outperform the threshold-based methods for larger segments (10–15%). In terms of real-world application, then: if there is only time to correct a small portion of the corpus, a method optimized to find a few highly likely errors may be preferable. But if time is available to edit more of the corpus, it may be beneficial to use a broader method.

5 Discussion and Recommendations

To underscore the trends we have observed, we took the 5% segment size and ran an ANOVA to obtain the sum of squares, as well as partial eta-squared values, to see the effect size of each of the four variables in a model. The sum of squares gives an indication of the percent variability explained. We ran the ANOVA with each of the two metrics as the dependent variable (Precision, LAS_r); results are shown in Table 4. The high effect sizes for the method can be attributed to having different kinds of methods. Aside from that, the choice of parser has the biggest effect, in these experiments, on the error detection precision, but very little effect when it comes to LAS_r. In this case, training corpus size has a large effect, as does the conversion scheme.

At this point, some recommendations emerge:

Variable	Precision SumSq	Precision Effect	LAS_r SumSq	LAS_r Effect
Training	1076	0.257	2339	**0.972**
Conversion	160	0.049	287	0.807
Parser	2985	0.489	12	0.144
Method	5168	**0.624**	13	0.159

Table 4: Sum of squares (*SumSq*) & Partial eta-squared (*Effect*) for the different variables; all variables significant at 0.001, except Conversion/Precision (< 0.05)

- The most important factors for error detection depend upon interactions between variables reflecting parser quality, linguistic decisions, and error detection method; thus, evaluation should incorporate a range of such variables. Even methods specific to one corpus project can vary the amount of data used, changes in the base parsing algorithm, etc.
- Error detection methods should be evaluated using parsers of varying quality, obtained via different training data sizes, choice of parsing algorithm, and so forth. Even two (complementary) parsers and both a small and large training corpus would go a long way towards conveying method strengths and weaknesses.
- A reasonable segment (or better, segments) of the testing corpus should be set for evaluation, in order to make precision comparable.
- LAS_r should also be reported, to account for the effect of a baseline parser.

Aside from recommendations of using parsers of varying quality—obtained via different training data sizes, choice of parsing algorithm, etc.—and using different segment sizes, another consideration has to do with the time spent in getting to a particular LAS value. Back in Table 3, we can note how LAS_r exceeds LAS_b for parsers trained on significantly more data. For example, if an annotator fixes 5% of the testing corpus for a parser trained on the 3,500 sentences of *02-03*, they can improve LAS by 4% (83.6% $\mapsto$ 87.7%). This is greater than the 87.0% obtained by training a parser on 7–8 times more annotated data (27,500 sentences, to get to *02-15*). This may help allocate annotator resources, indicating that for some situations time annotating could be better spent correcting a lower quality parsed corpus than in building a large high quality corpus from the outset.

6 Summary and Outlook

We have shown how parse error detection methods work under various real-world conditions, investigating the effect of: a) training data size, b) conversion scheme, c) choice of parser, and d) the error detection method itself. We saw the importance of accounting for initial parser quality—from both parser choice and training data

size—and of accounting for annotator time. We have emphasized the utility of comparing segments of the same size across conditions, so that one may focus on two evaluation metrics, precision and revised labeled attachment score (LAS_r)—the latter of which focuses the task on its actual impact on corpus building.

There are several directions to go in the future. First, we have not yet explored the effect of particular languages or domains [cf., e.g., 6], only investigating English news data. While different annotation schemes give some sense of different linguistic constructions, the effect of different languages or domains should lead to larger differences. Secondly, with these steps in place, we can of course investigate ways to improve error detection and to flesh out differences in methods that only emerge in particular experimental conditions (cf. section 4); the work in [8] is an example of using these evaluation recommendations to assist in such error detection development. Finally, we have touched on the importance of training data size and parser quality, but we have not broached how to delineate where annotation time should be spent. Namely, how much effort should be spent in building a well-annotated corpus to train a parser vs. in post-processing, and, relatedly, how well do parsers trained on silver standards perform?

Acknowledgements

We would like to thank Sandra Kübler, the participants of the IU computational linguistics colloquium, and the three anonymous reviewers for their useful feedback, as well as Stephanie Dickinson at the Indiana Statistical Consulting Center (ISCC) for help with statistical analysis.

References

[1] Bhasha Agrawal, Rahul Agarwal, Samar Husain, and Dipti M. Sharma. An automatic approach to treebank error detection using a dependency parser. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing, 14th International Conference, CICLing 2013, Proceedings, Part I*, Lecture Notes in Computer Science 7816, pages 294–303. Springer, 2013.

[2] Miguel Ballesteros, Jesús Herrera, Virginia Francisco, and Pablo Gervás. Are the existing training corpora unnecessarily large? *Revista Española para el Procesamiento del Lenguaje Natural*, 48, 2012.

[3] Adriane Boyd, Markus Dickinson, and Detmar Meurers. On detecting errors in dependency treebanks. *Research on Language and Computation*, 6(2):113–137, 2008.

[4] Jinho D. Choi and Martha Palmer. Robust constituent-to-dependency conversion for english. In *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories (TLT-9)*, pages 55–66, Tartu, Estonia, 2010.

[5] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *LREC 2006*, 2006.

[6] Felice Dell'Orletta, Giulia Venturi, and Simo netta Montemagni. Ulisse: an unsupervised algorithm for detecting reliable dependen cy parses. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 115–124, Portland, OR, June 2011.

[7] Markus Dickinson and Amber Smith. Detecting dependency parse errors with minimal resources. In *Proceedings of IWPT-11*, pages 241–252, Dublin, October 2011.

[8] Markus Dickinson and Amber Smith. Finding parse errors in the midst of parse errors. In *Proceedings of the 13th International Workshop on Treebanks and Linguistic Theories (TLT13)*, Tübingen, Germany, 2014.

[9] Jakob Elming, Anders Johannsen, Sigrid Klerke, Emanuele Lapponi, Hector Martinez Alonso, and Anders Søgaard. Down-stream effects of tree-to-dependency conversions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 617–626, Atlanta, Georgia, June 2013.

[10] Kilian A. Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. Because size does matter: The hamburg dependency treebank. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014.

[11] Dan Garrette and Jason Baldridge. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 138–147, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[12] Katri Haverinen, Filip Ginter, Veronika Laippala, Samuel Kohonen, Timo Viljanen, Jenna Nyblom, and Tapio Salakoski. A dependency-based analysis of treebank annotation errors. In *Proceedings of the International Conference on Dependency Linguistics (Depling'11), Barcelona, Spain*, pages 115–124, 2011.

[13] Enrique Henestroza Anguiano and Marie Candito. Parse correction with specialized models for difficult attachment types. In *Proceedings of EMNLP-11*, pages 1222–1233, Edinburgh, 2011.

[14] Naman Jain, Sambhav Jain, and Dipti Misra Sharma. Effective parsing for human aided nlp systems. In *Proceedings of The 13th International Conference on Parsing Technologies (IWPT-2013)*, pages 141–146, Nara, Japan, 2013.

[15] Richard Johansson and Pierre Nugues. Extended constituent-to-dependency conversion for english. In *Proceedings of NODALIDA 2007*, Tartu, Estonia, 2007.

[16] Mohammad Khan, Markus Dickinson, and Sandra Kübler. Does size matter? text and grammar revision for parsing social media data. In *Proceedings of the Workshop on Language Analysis in Social Media*, Atlanta, GA USA, 2013.

[17] Sandra Kübler, Ryan McDonald, and Joakim Nivre. *Dependency Parsing*. Morgan & Claypool Publishers, 2009.

[18] Hrafn Loftsson. Correcting a POS-tagged corpus using three complementary methods. In *Proceedings of EACL-09*, pages 523–531, Athens, Greece, March 2009.

[19] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. CUP, 2008.

[20] M. Marcus, Beatrice Santorini, and M. A. Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[21] Ryan McDonald, Kevin Lerman, and Fernando Pereira. Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of CoNLL-X*, pages 216–220, New York City, June 2006.

[22] Ryan McDonald and Joakim Nivre. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[23] Seyed Abolghasem Mirroshandel and Alexis Nasr. Active learning for dependency parsing using partially annotated sentences. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 140–149, Dublin, Ireland, October 2011. Association for Computational Linguistics.

[24] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonaldand Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The conll 2007 shared task on dependency parsing. In *Proceedings of EMNLP-CoNLL 2007*, Prague, 2007.

[25] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135, 2007.

[26] Lauma Pretkalniņa, Artūrs Znotiņš, Laura Rituma, and Didzis Goško. Dependency parsing representation effects on the accuracy of semantic applications

- an example of an inflective language. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may 2014.

[27] Anders Søgaard. An empirical study of differences between conversion schemes and annotation guidelines. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 298–307, Prague, Czech Republic, August 2013.

[28] Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England, August 2008. Coling 2008 Organizing Committee.

[29] Dan Tufiş and Elena Irimia. Roco-news: A hand validated journalistic corpus of romanian. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*, pages 869–872, Genoa, Italy, 2006.

[30] Gertjan van Noord and Gosse Bouma. Parsed corpora for linguistics. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 33–39, Athens, March 2009.

[31] Alexander Volokh and Günter Neumann. Automatic detection and correction of errors in dependency treebanks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 346–350, Portland, OR, 2011.

Part II

Short papers

Quantitative Comparison of Different Bi-Lexical Dependency Schemes for English

Norveig Anderssen Eskelund and Stephan Oepen

Department of Informatics
University of Oslo
E-mail: {norveiga|oe}@ifi.uio.no

Abstract

In this work, we summarize an in-depth quantitative and qualitative analysis of three dependency schemes for English, in order to acquire and document contrastive insights into how these schemes relate to each other. We demonstrate how a simple quantitative method can reveal salient structural properties of individual schemes, as well as uncover systematic correspondences between pairs of schemes. We use this method to estimate and compare the expressiveness of three dependency schemes and we identifiy several linguistic phenomena that have a different syntactic analysis in two of these schemes.

1 Introduction & Background

Bi-lexical syntactic dependencies have gained great popularity in the past decade, both as a comparatively theory- and language-neutral framework for syntactic annotation, and as the interface representation in syntactic parsing. For English (and maybe a handful of other languages), there exist different conventions for representing syntactic structure in terms of bi-lexical dependencies; variations on existing schemes as well as new ones emerge continuously. Despite great community interest in dependency syntax, there is relatively little documented knowledge about similarities and differences between the various annotation schemes (Ivanova et al. [4]; Zeman et al. [9]; Popel et al. [8]).

In this work, we summarize an in-depth quantitative and qualitative analysis of three dependency schemes for English, in order to acquire and document contrastive insights into how these schemes relate to each other (Eskelund [2]). We anticipate that such knowledge can (a) further our linguistic understanding of competing syntactic analyses; (b) support conversion across schemes and enable cross-framework evaluation; and (c) inform the design of future dependency schemes and treebanking initiatives.

We consider two common dependency schemes that are both derived from the syntactic analyses of the venerable Penn Treebank (Marcus et al. [6]), viz. Stanford Basic Dependencies (SB) (De Marneffe et al. [1]) and CoNLL Syntactic Dependencies (CD) (Johansson et al. [5]). As a more independent point of comparison, we include as a third annotation scheme what Ivanova et al. [4] term DELPH-IN Syntactic Derivation Trees (DT), based on a fresh annotation of the same PTB Wall Street Journal Text in the linguistic framework of Head-Driven Phrase Structure Grammar (Flickinger et al. [3];Pollard et al. [7]).

Our goals in this study are two-fold, viz. (a) to determine relative degrees of (dis-)similarity across different schemes and (b) to facilitate conversion between schemes, by uncovering systematic correspondences. For both objectives, we propose a quantitative, data-driven methodology, using simple descriptive statistics at various levels of detail. The data we have used for our study draws on Wall Street Journal Sections 00–17, which comprise 29,672 sentences (or 651,980 tokens) that can be aligned for tokens and PoS tags across the three schemes.

2 Comparison of Schemes and Scheme Pairs

To gain knowledge about the expressiveness of the dependency schemes, and possibly whether any of them are more expressive than the others, we will estimate the *granularity* and *variability* of each scheme. We define the granularity of a scheme as its available range of possible distinctions, which we contrast with variability, i.e. the amount of distinctions actually exercised. We presume that a scheme that is linguistically rich, will have a high degree of granularity and variability.

To estimate the granularity of the annotation schemes, we calculate the number of possible combinations of PoS tags and relation types for each scheme. According to available documentation, CD has the largest label set (69), DT the smallest (52), and SB falls in-between them (56). Assuming equal PoS tagging, CD thus has available the largest number of possible combinations of a relation label with a PoS assignment on either the head or dependent, or of course both. Conversely, we estimate the variability of the three schemes by counting labels and combinations of labels and PoS tags that are actually used in our data. The results, both in absolute and relative frequencies, are presented in Table 1.

	CD	DT	SB	CD	DT	SB
parts of speech	45	45	45			
dependency labels	62	50	49	89.9	96.2	87.5
head tag & label	546	588	677	17.6	25.1	26.9
dependent tag & label	688	690	577	22.2	29.5	22.9
dependent tag & head tag & label	3503	3541	3479	2.5	3.4	3.1

Table 1: Different tag and label combinations

We can see that only a small fraction of the possible combinations of PoS tags and labels are used. Although these proportions differ a bit between the different schemes, the numbers of combinations of dependent tags, head tags, and labels (shown in the last row) are surprisingly similar. This could indicate that the differ-

ence in variability between the schemes is trifling.

To further investigate variability across schemes, we consider the frequency distributions of the PoS tag and label combinations used, i.e. whether a small proportion of combinations covers a large number of dependencies, or whether these proportions are more evenly distributed. Table 2 shows how many tag–label combinations are required to 'cover' a given percentage of all dependencies in our treebanks. For all three schemes, the most frequently used combination of one PoS tag, for the head and dependent each, and one dependency type accounts for at least five percent of all dependencies; only a little more than 300 distinct combinations (or less than ten percent of all distinct combinations) cover at least ninety percent of the treebank. Again, these distributions are very similar across our three schemes, i.e. highly comparable in skewness and, thus, variability.

	5%	10%	25%	50%	75%	90%	95%	100%
CD	1	2	9	34	115	319	479	3503
SB	1	2	8	36	129	325	565	3479
DT	1	3	10	41	137	343	551	3541

Table 2: Tag and label combinations growth

We also investigate the tree-depth, the number of nodes in the longest path from the root to a terminal node in a sentence, in the three formats. The average (maximum) tree-depth in the SB scheme is 6.35 (20). Corresponding values are 7.75 (24) and 7.93 (25) for CD and DT, respectively. The SB scheme thus has a considerably lower tree-depth than the two other schemes.

	CD	DT	SB
VBD	43.5	38.3	35.8
VBZ	28.3	24.2	15.7
VBP	14.3	11.6	7.4
MD	8.2	7.1	0.1
CC	0.0	13.3	0.0
VBN	0.5	0.5	13.3
VB	0.7	0.6	8.2
NN	1.1	0.8	5.6
JJ	0.1	0.1	5.0
VBG	0.1	0.0	4.0
	96.8	96.5	95.1

Table 3: Most common tags of roots

Table 3 shows the 10 most common PoS tags for tokens used as roots and their percentage for each scheme. Table 3 confirms some of the facts we already know about the scheme. One of these facts is that in the SB scheme, content words are preferred as heads (De Marneffe et al. [1]). This explains that non-finite verbs forms (VB, VBG, VBN), nouns (NN) and adjectives (JJ) occur as roots far more often in the SB scheme than in the other schemes, while modal verbs (MD), in contrast to the other schemes, hardly ever are used as roots in SB. It can also explain why, as we can see from the table, there is a higher variation among PoS tags frequently used as roots in SB than in the other schemes.

Another phenomenon that Table 3 reveals, is that DT treats coordination different from SB and CD (Ivanova et al. [4]). Coordination is a well-known area of differences between dependency schemes. According to Popel et al. [8], three main models for this problematic issue are most frequently used: the Stanford parser style, the Mel'čuk style, and the Prague Dependency style. The SB scheme uses the Stanford parser style and CD makes use of the Mel'čuk style. The model used by the DT scheme is the Prague Dependency style, the only model among these three where the coordinating conjunction is considered the head of the coordination structure. From Table 3 we see that the coordinating conjunction (CC) often appears as root in DT, in the other schemes it practically never does.

We have also compared pairs of schemes, to gain knowledge about similarities and differences between them. Table 4 shows the unlabelled attachment score (*UAS*) and the unlabelled sentence accuracy (*USA*), both including and excluding punctuation tokens, as well as the share of identical roots for our scheme pairs. There is a considerably higher correspondence between roots in the DT/CD pair than in the other pairs. The UAS is quite similar for the DT/CD and the CD/SB pairs when punctuations are included, and substantially lower for the DT/SB pair.

Leaving out punctuation tokens gives a considerable increase in the similarity scores for both the SB/DT and the DT/CD pair. The scores for the CD/SB pair is nearly unchanged, showing that punctuation tokens are, in most cases, identically attached in these two schemes. The most similar schemes, when we exclude punctuation tokens, are DT and CD, even though—unlike CD and SB—DT does not derive from the original phrase structure annotations in the PTB.

	identical	w/ punctuation		w/o punctuation	
	roots	UAS	USA	UAS	USA
CD/SB	62.5	72.7	13.1	72.9	13.1
SB/DT	53.7	55.7	1.2	60.5	5.3
DT/CD	84.4	73.6	1.2	80.8	17.7

Table 4: Similarity scores for scheme pairs

3 Detecting Structural Differences between Schemes

We know that the three schemes use different approaches to coordination structures, that content words are preferred as heads in SB, that CD uses non-projective dependencies to handle discontinuous structures and that punctuations are attached differently in DT than in CD and SB. Are there other systematic differences of syntactic structure, and how can we identify them? We have explored the use of a quantitative methodology for detection of patterns of different syntactic analyses in scheme pairs.

dependent tag	DT label	DT head tag	aligned	unaligned
CC			1029	14457
		CC	12125	20373
POS			43	5868
		POS	12	5997
	NUM-N		32	6682
CD			11317	12818
		CD	2010	15143
		DT	707	2398

Table 5: Distributions of aligned vs. unaligned dependencies (without punctuation)

We count aligned and unaligned dependencies per dependent PoS tag, $scheme_1$ label, $scheme_2$ label, $scheme_1$ head PoS tag and, for unaligned dependencies, also $scheme_2$ head PoS tag. We compare these data to see if we can find combinations of PoS tags and labels that differ in a way that can imply a systematic difference between the schemes. Subsequently, we extract sample sentences that adhere to these combinations and see if we can discover patterns of systematic differences. Finally, we rewrite the structures in $scheme_1$ to the $scheme_2$ pattern and recalculate the unlabelled attachment score.

We explore the use of this methodology, by applying it to the DT ($scheme_1$) and

CD ($scheme_2$) pair. We will concentrate on investigations of structures that constitute the unaligned dependencies with the dependent PoS tags, source format labels and/or head PoS tags shown in Table 5. The large number of unaligned dependencies involving CC (coordinating conjunction) heads and/or CC dependents, are caused by the use of different coordination models in the two schemes. Rewriting these structures in the DT data so that they adhere to the Mel'čuk style, increases the UAS for the DT/CD pair by 4.5.

A closer investigation of dependencies involving tokens tagged POS (*possessive ending*) reveals the information shown in Table 6. We see that most POS tokens are differently attached in DT and CD and the greater part of these are attached by a dependency relation labelled SP-HD in DT. The counts also reveal that most dependents attached to a POS head in DT, have a different head in CD. The greater part of these dependents, as well, are attached by arcs labelled SP-HD in DT.

dependent tag	DT label	DT head tag	aligned	unaligned
POS			43	5868
POS	SP-HD		13	5516
		POS	12	6031
	SP-HD	POS	12	5621

Table 6: Aligned vs. unaligned possessives

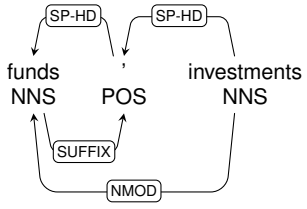


Figure 1: Example dependencies involving possessive endings (DT top, CD bottom)

Examining sample sentences, we find that the pattern illustrated in Figure 1 is common. While CD attaches the possessive ending to its noun (the possessor), DT treats it more like a two-place relation. Rewriting these structures to adhere to the CD pattern, increases the UAS for the DT/CD pair by 1.7 (punctuations are included in this score). We investigate the other groups of unaligned dependencies in the same manner. A closer look at NUM-N labels reveals that most of the dependents attached by an edge labelled NUM-N in DT, are attached to a different head in CD. We also find that most of these dependents are attached to a CD node in DT and that the greater part of them are tagged NN or NNS.

We have a closer look at some of these sentences and find the common pattern exemplified in Figure 2. DT seems to assign the amount to be head of the measure unit, while CD takes the opposite stance and chooses the measure unit as head of the amount. Rewriting these structures in the DT data, increases the UAS of the scheme pair by 2.0.

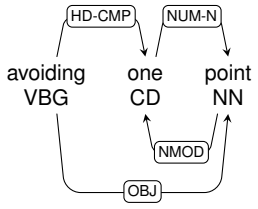


Figure 2: Example dependencies with NUM-N in DT

Examining unaligned dependencies involving CD (cardinal number) tokens further, shows us that there is a considerable number CD tokens that are attached to another CD token in our DT data, but attached to a '$' token (dollar sign) in the CD scheme. All these tokens are attached by an arc labelled SP-HD in DT. These

structures, i.e. '$ 212 million', contain both a cardinal number and a numeral, both tagged CD. While DT attaches the numeral to the dollar sign and the cardinal number to the numeral, CD assigns the dollar sign as head of both the numeral and the cardinal number. Rewriting these structures increases the UAS by 0.5.

Further investigation of the remaining unaligned dependencies involving heads tagged CD in DT, reveals another pattern of systematic differences between the two schemes. In constuctions like 'until Dec. 31', DT assigns the day number as the head of the month. In CD the day number and month are not directly connected. Rewriting these structures increases the UAS by 0.4.

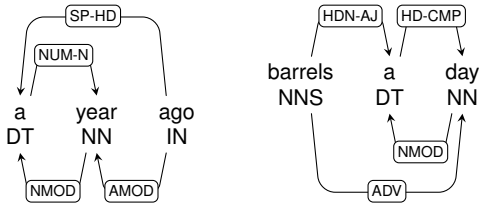


Figure 3: Examples of dependencies with noun dependents and determiner head in the DT data

We have also investigated the comparatively modest number of unaligned constructions that have determiner heads in DT. We find that more than half of these dependents are nouns. A closer look at them reveals several patterns, where DT assigns the DT node as head of the noun, while CD treats it as an ordinary determiner and considers it a dependent of the noun. Two patterns are shown in Figure 3. In the first of these examples, *'a'* is actually a cardinal number (meaning *'one'*), not a determiner. In the second graph *'a'* works as a preposition and should ideally have been tagged IN. Rewriting these structures increases the UAS for the format pair by 0.4.

4 Conclusions

We have demonstrated how a simple quantitative method can reveal salient structural properties of individual schemes, as well as uncover systematic correspondences between pairs of schemes. We have used this method to estimate and compare the expressiveness of three dependency schemes. We have also identified and documented several linguistic phenomena that have a different syntactic analysis in CD and DT. These and similar findings, together with our discovery procedure based on simple contrastive statistics, was applied by Eskelund [2] in the development of a heuristic converter from the DT to the CD scheme.

References

[1] De Marneffe, Marie-Catherine and Manning, Christopher D (2008) Stanford typed dependencies manual. (URL: http://nlp. stanford. edu/software/dependencies_manual. pdf).

[2] Eskelund, Norveig Anderssen (2014) Dependency Interconversion. Master's thesis. University of Oslo.

[3] Flickinger, Dan, Zhang, Yi and Kordoni, Valia (2012) DeepBank: A Dynamically Annotated Treebank of the Wall Street Journal. In *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories*, pp. 85–96. Edições Colibri.

[4] Ivanova, Angelina, Oepen, Stephan, Øvrelid, Lilja and Flickinger, Dan (2012) Who did what to whom?: a contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pp. 2–11.

[5] Johansson, Richard and Nugues, Pierre (2007) Extended constituent-to-dependency conversion for English. In *Proc. of the 16th Nordic Conference on Computational Linguistics (NODALIDA)*, pp. 105–112.

[6] Marcus, Mitchell P, Marcinkiewicz, Mary Ann and Santorini, Beatrice (1993) Building a large annotated corpus of English: The Penn Treebank. In *Computational linguistics* 19.2, pp. 313–330. MIT Press.

[7] Pollard, Carl and Sag, Ivan A (1994) *Head-driven phrase structure grammar*. University of Chicago Press.

[8] Popel, Martin, Marecek, David, Štepanek, Jan and Zeman, Daniel and others (2013) Coordination Structures in Dependency Treebanks. In *ACL (1)*, pp. 517–527.

[9] Zeman, Daniel, Marecek, David, Popel, Martin, Ramasamy, Loganathan, Štepanek, Jan, Zabokrtsky, Zdenek and Hajic, Jan (2012) HamleDT: To Parse or Not to Parse? In *LREC*, pp. 2735–2741.

The definition of tokens in relation to words and annotation tasks

Fabian Barteld, Renata Szczepaniak, and Heike Zinsmeister

Institut für Germanistik
Universität Hamburg
E-mail: firstname.lastname@uni-hamburg.de

Abstract

Tokens are the basic units of annotations. When working with corpora of non-standardized texts, tokenization is often problematic, as the usage of whitespace can vary. We show examples of how decisions in the tokenization process can influence an annotation and argue that the principles underlying the tokenization should be grounded in theoretical concepts selected on the basis of the annotation task. We present a corpus of Early New High German texts in which the annotation layers reference two different concepts of words: syntactic words and graphematic words. Consequently, we use two kinds of tokens: graphic tokens and syntactic tokens.

1 Introduction

This paper concerns tokens, which are the basic units of annotations. The tokenization process is therefore decisive for all further annotation, including the assignment of part-of-speech (PoS) tags and syntactic analysis. For languages with alphabetic scripts, tokens are roughly defined by the appearance of whitespace (cf. Schmid [10]). In the modern, standardized, written variants of English and German, tokens defined in this way generally coincide with syntactic words. However, even in texts that are close to standard, there are cases in which the boundaries of tokens as defined by whitespaces and those of syntactic words do not align (cf. Grefenstette / Tapanainen [6]). This leads to divergent tokenizations in different corpus projects, which impacts the resulting annotations. Using the example of contractions, we show the consequences of tokenization choices for the creation of treebanks and the assignment of PoS tags. From this, we conclude that tokenizations should be determined on the basis of the annotation task at hand.We present a corpus of Early New High German (ENHG) protocols of witch interrogations as a case study for the implementation of multiple tokenizations that are each motivated by specific annotation tasks.[1]

[1]This corpus is created as part of the project "Development of Sentence-internal Capitalization in German" (SIGS), funded by the German Research Foundation (DFG). Two of the three authors

2 Different tokenizations – different annotations

A typical case in which deviations between whitespaces and syntactic words emerge is the use of non-standard contractions in direct speech in newspaper texts. The segmentation rules applied in TüBa-D/Z, a treebank of Modern German newspapers (Telljohann et al. [11]), distinguish between cliticized verb-pronoun combinations marked by an apostrophe, such as *gibt's* (< *gibt* + *es* 'it exists'), and non-standard graphic contractions without apostrophes, such as *glaubense* (< *glauben* + *sie* 'you believe'). Only in the first case is the contraction split into two tokens. Because the token is the basic unit of the syntactic annotation in the treebank, the syntax tree shows no trace of the merged subject pronoun in *glauben=se*. Consequently, two similar constructions are treated differently when graphic (rather than syntactic) rules are used for the tokenization. This shows that graphic tokenization rules can lead to inconsistencies in the subsequent syntactic analysis.

Because TüBa-D/Z, which consists of 1.5 million tokens, contains only 56 instances in which a token includes more than one syntactic word, the problem is negligible.[2] In contrast, in corpora of non-standardized texts (such as internet-based communication and historical documents), such cases are much more common. Therefore, the tokenization must be more carefully conceptualized and applied. A comparison of different adaptations of the German PoS tagset STTS (Schiller et al. [9]) to non-standardized texts shows that the definition of tokens can have far-reaching consequences for subsequent annotation layers.

For internet-based communication, Bartz et al. [1] propose to treat contracted forms such as *machste* (< *machst* + *du* '*you make*'), as one token. To this end, a new tag class for contractions is introduced. The tags in this class consist of the prefix KTR and parts identifying the constituents of the contraction (for the above contractions of verb and pronoun, KTRVVPPER).

Contracted forms are also frequent in historical texts. However, the adaptation of STTS to historical texts, HiTS (Dipper et al. [3]), does not make use of specific tags for contracted forms because the written texts are first "normalized" – that is, they are segmented into units that approximate the tokens of modern, standardized, written German. By means of this procedure, contracted forms such as *machste* are separated. The HiTS tagset itself is only applied to the units of the normalized tokenization.

These examples make it clear that different tokenizations yield different annotations, thus, emphasizing the importance of the careful definition of tokens, especially when dealing with non-standardized texts. We argue that this definition should not be based on graphic conventions; instead, tokenizations should be motivated by linguistic concepts of the unit word, which in turn must be defined with respect to a specific linguistic level (cf. Zwicky [13], Fuhrhop [5]). Furthermore,

of this paper, Fabian Barteld and Renata Szczepaniak, are collaborators in this joint project with the University of Münster (Klaus-Michael Köpcke and Marc Schutzeichel).

[2] Here we are referring to TüBa-D/Z Release 9 (11.12.2013) in which these cases are marked at the lemma level.

different annotations or applications may call for different tokenizations (cf. Chiarcos et al. [2]). As a consequence, when multi-layer annotations are being created, multiple tokenizations may be necessary.

In the following section, we will present the tokenization approach used in the SIGS project, which is in the process of creating a corpus of Early New High German texts. SIGS defines two levels of tokenization: one based on a graphematic concept of words, and the other on a syntactic concept.

3 Case study: Graphic and syntactic tokens in a corpus of ENHG texts

In the SIGS project, a corpus of protocols of witch interrogations, written between 1570 and 1670, (Macha et al. [7]) is being annotated to analyze the spread of word-initial capitalization in Early New High German (ENHG). Word-initial capitalization in German started as a pragmatic marker (emphasizing certain words); it later developed into a syntactic marker (marking the head of a nominal phrase). In this project, we analyze how multiple factors (e.g., the syntactic function, the semantic role and the animacy of a noun's referent) interacted in this development. To this end, the corpus is annotated in multiple layers, one of which is the syntactic constituency. As noted above, using graphic rules to segment a text into tokens can result in similar structures being annotated differently. This is further illustrated by example (1),[3] which shows two types of mismatches between whitespaces and syntactic words: preposition-article contractions and the spelling of compounds as separate words.

(1) a. auff=m Teufel-ß dantz
at=[the]DAT devil-LE dance[DAT]
at the devil's dance
(Alme 1630)

b. in=s teufel-ß Nahme-n
in=[the]GEN devil-GEN name-DAT
in the name of the devil
(Alme 1630)[4]

In both parts of example (1), the units that are separated by whitespaces would be analyzed with the same parts of speech according to STTS – namely, APPRART, NN, and NN. However, these two graphically identical structures are syntactically different. (1a) includes the separated compound *Teufelß dantz* ('devil's dance'). Hence, syntactically speaking, the phrase contains only one noun, whereas (1b) contains two nouns. Furthermore, the preposition-article contraction *auffm* in (1a)

[3]The glosses in the examples follow the Leipzig Glossing rules (`http://www.eva.mpg.de/lingua/resources/glossing-rules.php`. LE here means "linking element".

[4] Place and Year of the protocoll reference the text in the edition of Macha et al. [7].

includes an article that agrees in case (dative) with the head noun *Teufels dantz*. In contrast to Modern German, ENHG also allows contractions when the article is part of an embedded genitive phrase. In example (1b) the article contained in the contraction *ins* agrees in case (genitive) with *teufelß*, which modifies *Nahmen*.

These syntactic differences can easily be seen in Fig. 1 in which the graphic units *auffm* and *ins* are treated as separate tokens and the two graphic units *Teufelß* and *dantz* are merged.

Figure 1: Syntactic annotation based on syntactic tokens

This example also indicates the relevance of the syntactic structure for capitalization. In both phrases, the only capitalized graphic unit is the beginning of the head of the noun phrase immediatly dominated by the PP. However, if we only tokenized in this way (which is similar to the normalization process in Dipper et al. [3]), we would lose the possibility of referring to simple alphabetic strings that are delimeted by whitespaces (i.e., to the unit that can be capitalized). Examples (2) and (3) show two cases in which compounds (indicated by brackets) are graphically separated and the second part is capitalized.

(2) drey stuckh [rindt Viech]
three pieces cattle cattle
three pieces of cattle
(Baden-Baden 1628)

(3) der zu geordtneten [Gerichts Schöpffenn]
the to allocated court jury
the allocated jury
(Georgenthal 1597)

In the SIGS project, we use two tokenizations, which will be described in the following two sections. The different layers of the annotation can then be based on either of the two tokenizations. Using the ANNIS interface, the two tokenizations can be queried and combined (cf. Zeldes [12]).

3.1 Graphematic words and graphic tokens

When investigating capitalization, the fundamental units need to be based on the graphic form. The relevant word concept related to the graphic form is defined as the "graphematic word". Fuhrhop [5] proposes a formal definition of the graphematic word in German: Basically, a graphematic word stands between two spaces and does not itself contain spaces. This definition is similar to many definitions of tokens. However, her concept also entails differences in relation to common approaches of tokenization. For example, Fuhrhop suggests that sentence final punctuation marks should be viewed as belonging to a graphematic word. She therefore treats a form like <*denken.*> ('to think') as a distributional variant of the form <*denken*>, similar to intonational differences that can be found in phonological words depending on their context.

The graphematic word, however, cannot be directly used as the basic unit for the study of the development of capital letters. Here, even units smaller than graphematic words are relevant. In the SIGS corpus, a number of instances similar to example (4) can be found.

(4) ge- <linebreak> Antworrt
PTCP answer
answered
(Erkelenz 1598)

In this example, we see a graphematic word divided after the participle prefix *ge* at the end of a line. The hyphen indicates that the two parts form one graphematic word (cf. Fuhrhop [5]); the problem is that the second part of the graphematic word (*Antwortt*) starts with an uppercase letter. In order to investigate such capitalizations, the second part of the graphematic word must be annotated on its own. Hence, initial capitalization can be related to units smaller than graphematic words. The relevant units should be defined as non-whitespace characters surrounded by whitespace characters (which include the linebreak). The SIGS project uses this definition for the graphic tokens. For the annotation of such graphic tokens with PoS tags, we need tags for contractions such as those defined by Bartz et al. [1], as well as tags for parts smaller than the units that are normally annotated in PoS tagging – e.g., for the prefix *ge-* in example (4).

3.2 Syntactic words and syntactic tokens

Syntactic words can be defined as the basic units of a sentence (cf. Fuhrhop [5]) and therefore represent a good basis for the tokens in syntactic annotations, as illustrated in example (1). In the SIGS corpus, there are nine different types of mismatches between graphic tokens and syntactic words: (i) words split at the end of a line, (ii) verb particles graphically separated from the verb, (iii) compounds written as separate words, (iv) the infinitive particle *zu* ('to') merged with

the verb, (v) pronominal adverbs such as *davon* written as separate words, (vi) clitics and contractions, (vii) univerbations such as *allweg* ('always' < 'all' + 'ways') written as separate words, (viii) words that have been deleted (e.g., by means of strikethroughs), and (ix) catchwords.[5]

Most of these types are also mentioned in Dipper et al. [3]. The two interesting cases that are not mentioned there are deleted words and catchwords. In both cases, graphic tokens exist that should not appear at the level of the syntactic annotation, as they would be superfluous. However, at the level of the graphic token, it is important to retain them, as they can be relevant with regard to capitalization (see Fig. 2, in which the two instantiations of *vnnd* differ in terms of capitalization).

Text	Inn einem stotzen gepracht, vnnd <pagebreak> Vnnd zu getragenn,								
Graph. Token	Inn	einem	stotzen	gepracht	,	vnnd	Vnnd	zu	getragenn	,
Synt. Token	Inn	einem	stotzen	gepracht	,	vnnd		zugetragenn		,
Translation	In	a	drinking vessel	brought	,	and		brought		,

Figure 2: Tokenizations of an ENHG example containing a catchword (Georgenthal 1597)

The SIGS corpus is still under construction. At the time of writing (November 2014), a pre-final tokenization of 18 protocols exists, which thus far consists of 26,709 annotated graphic tokens and 26,158 syntactic tokens. In 24,893 cases, the graphic and syntactic tokens are equivalent, but 1,816 (6.8%) graphic tokens and 1,265 (4.8%) syntactic tokens deviate from each other.

4 Conclusion and outlook

In this paper, we have illustrated how different tokenizations can lead to different annotations. Consequently, the choices behind the tokenization should be based on theoretical assumptions relevant to the specific annotation task and should be made explicit in the annotation guidelines. This is especially important when creating a corpus of non-standardized texts, as there can be substantial variation in the usage of whitespace.

Tokens are often intended to resemble words. However, because the boundaries of words differ on different linguistic levels, the specific concept of "word" underlying the tokenization process must be selected on the basis of the annotation or application. We have given two examples of the definition of tokens based on graphematic and syntactic words. Furthermore, we have shown the need for multiple tokenizations in a project examining the multiple factors behind the development of sentence-internal capitalization in German, as the different annotation layers reference different concepts of words. Depending on the corpus and the aims of the annotation, other types of tokens might be useful, such as tokens based on phonological words in corpora of spoken language (cf. Eckart et al. [4], Rehbein / Schalowski [8]).

[5]Catchwords are repetitions of the first word on a page at the bottom of the previous page, which were used as an aid for binding the pages in the right order.

5 Acknowledgements

We would like to thank our annotators in the SIGS project: Annemarie Bischoff, Lisa Dücker, Eleonore Schmitt and Annika Vieregge in Hamburg, and Julia Hübner, Johanna Legrum and Katja Politt in Münster.

We would also like to thank Claire Bacher for improving our English. All remaining errors are ours.

References

[1] Bartz, T., Beißwenger, M. and Storrer, A. (2013) Optimierung des Stuttgart-Tübingen-Tagset für die linguistische Annotation von Korpora zur internetbasierten Kommunikation: Phänomene, Herausforderungen, Erweiterungsvorschläge. *JLCL* 28(1), pp. 157–198.

[2] Chiarcos, C., Ritz, J. and Stede, M. (2009) By all these lovely tokens... Merging conflicting tokenizations. In *Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP 2009*, pp. 35–43. Suntec, Singapore.

[3] Dipper, S., Donhauser, K., Klein, T., Linde, S., Müller, S. and Wegera, K.-P (2013) HiTS: Ein Tagset für historische Sprachstufen des Deutschen. *JLCL* 28(1), pp. 85–137.

[4] Eckart, K., Riester, A., and Schweitzer, K. (2012) A Discourse Information Radio News Database for Linguistic Analysis. In Chiarcos, C., Nordhoff, S., and Hellmann, S. (eds.) *Linked Data in Linguistics. Representing and Connecting Language Data and Language Metadata*, pp. 65–76. Heidelberg: Springer.

[5] Fuhrhop, N. (2008) Das graphematische Wort (im Deutschen): Eine erste Annäherung. *Zeitschrift für Sprachwissenschaft* 27(2), pp. 189–228.

[6] Grefenstette, G. and Tapanainen P. (1994) What is a word, what is a sentence? Problems of tokenization. In *Proceedings of the 3rd Conference on Computational Lexicography and Text Research*, pp. 79–87. Budapest, Hungary.

[7] Macha, J., Topalović, E., Hille, I., Nolting, U. and Wilke, A. (eds.) (2005) Deutsche Kanzleisprache in Hexenverhörprotokollen der Frühen Neuzeit. Berlin / New York: de Gruyter.

[8] Rehbein, I. and Schalowski, S. (2013) STTS goes Kiez: Experiments on Annotating and Tagging Urban Youth Language. *JLCL* 28(1), pp. 199–227.

[9] Schiller, A., Teufel, S., Stöckert, C. and Thielen, C. (1999) Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). University of Stuttgart and University of Tübingen, Germany. (URL:

http://www.ims.uni-stuttgart.de/forschung/ressourcen/lexika/TagSets/stts-1999.pdf).

[10] Schmid, H. (2008) Tokenizing and Part-of-Speech Tagging. In Lüdeling, A. and Kytö, M. (eds.) *Corpus Linguistics. An International Handbook, Vol. 1*, pp. 527–551. Berlin, New York: Mouton de Gruyter.

[11] Telljohann, H., Hinrichs, E., Kübler, S., Zinsmeister, H. and Beck, K. (2012) Stylebook for the Tübingen treebank of written German (TüBa-D/Z). Department of Linguistics, University of Tübingen, Germany. (URL: http://www.sfs.uni-tuebingen.de/fileadmin/static/ascl/resources/tuebadz-stylebook-1201.pdf).

[12] Zeldes, A. (2013) ANNIS3 – Multiple Segmentation Corpora Guide. Version: 2013-6-15a. SFB 632 Information Structure / D1 Linguistic Database, Humboldt University of Berlin and University of Potsdam, Germany. (URL: http://www.sfb632.uni-potsdam.de/annis/download/ANNIS3_multiseg_guide_2013-6.pdf).

[13] Zwicky, A. (1990) Syntactic words and morphological words, simple and composite. In Booij, G. and Van Marle, J. (eds.) *Yearbook of Morphology 3*, pp. 201–216. Dordrecht: Foris.

From <tiger2/> to ISOTiger – Community Driven Developments for Syntax Annotation in SynAF

Sonja Bosch, University of South Africa (UNISA) – Kerstin Eckart, University of Stuttgart – Gertrud Faaß, University of Hildesheim – Ulrich Heid, University of Hildesheim – Kiyong Lee, Korea University – Antonio Pareja-Lora, Universidad Complutense de Madrid – Laurette Pretorius, University of South Africa (UNISA) – Laurent Romary, Humboldt-Universität zu Berlin – Andreas Witt, Institut für Deutsche Sprache – Amir Zeldes, Georgetown University – Florian Zipser, Humboldt-Universität zu Berlin

E-mail: kerstin.eckart@ims.uni-stuttgart.de, laurent.romary@inria.fr

Abstract

In 2010, ISO published a standard for syntactic annotation, ISO 24615:2010 (SynAF). Back then, the document specified a comprehensive reference model for the representation of syntactic annotations, but no accompanying XML serialisation. ISO's subcommittee on language resource management (ISO TC 37/SC 4) is working on making the SynAF serialisation ISOTiger an additional part of the standard. This contribution addresses the current state of development of ISOTiger, along with a number of open issues on which we are seeking community feedback in order to ensure that ISOTiger becomes a useful extension to the SynAF reference model.

1 Introduction

In 2010 an ISO[1] standard on the syntactic annotation framework SynAF was published, ISO 24615:2010. Even though this ISO standard specified a comprehensive reference model for the representation of syntactic annotations, it did not provide an accompanying XML serialisation for this type of annotations [1].

[1] thus presented <tiger2/>, an XML serialisation for SynAF, enhancing the existing TIGER-XML format [8] from the TIGER treebank [2] to meet the specifications of the SynAF model, such as being able to handle not only constituency-based representations but also dependency analyses and others which make use of

[1] International Organization for Standardization, http://www.iso.org

extensible types of nodes and edges. [1] described the <tiger2/> format and presented examples of its use in the modelling of linguistic constructions, including e.g. contractions, elliptic subjects or compound sentences as they appear in Zulu.

In the meantime, ISO's subcommittee on language resource management (ISO TC 37/SC 4) is working on making the serialisation an additional part of the standard. For this reason, it was agreed in 2014 to rename the standard to ISO 24615-1 *Language resource management – Syntactic annotation framework (SynAF) – Part 1: Syntactic model* and start a new standard project for *Part 2: XML serialization (ISOTiger)*[2].

The SynAF serialisation ISOTiger is the continuation of <tiger2/>, pursuing two objectives: i) including feedback from the community, cf. [1], and ii) aligning SynAF even more closely with other existing standards such as the *Linguistic annotation framework* (LAF) [7], the *Morpho-syntactic annotation framework* (MAF) [6] and the combined ISO and TEI standards on *feature structures* (FSR, FSD) [4, 5].

The main purpose of this contribution is to explore how the two objectives of ISOTiger are met in a consistent, non-contradicting way. Section 2 briefly describes the SynAF reference model, Section 3 addresses the current state of development of ISOTiger and Section 4 discusses some open issues on which we are seeking community feedback, in order to ensure that ISOTiger becomes a useful extension to the SynAF reference model.

2 SynAF components

The SynAF – Part 1 metamodel specifies syntactic annotations as consisting of *SyntacticNodes*, *SyntacticEdges* and their corresponding *Annotations*. The model distinguishes between terminal nodes (*T_node*) for morpho-syntactically annotated word forms (or empty elements when appropriate) and non-terminal nodes (*NT_node*), which can be annotated with syntactic categories from the phrasal, clausal and sentential level. Edges can be established between (both terminal and non-terminal) nodes and can also be annotated. While this metamodel can be implemented on its own, it is recommended to express morpho-syntactically annotated terminal nodes following the MAF standard [6] and to apply a data category registry [3] to specify the syntactic categories that are part of the annotation.

3 XML serialisation

Figure 1 shows an excerpt of an XML-encoded syntactic annotation example[3]. The `<annotation>` element of the header makes reference to an external annotation

[2]SynAF – Part 2 is currently at the stage of a committee draft (ISO/CD 24615-2). For an overview of the stages in the development of ISO standards see:
`http://www.iso.org/iso/home/standards_development/resources-for-technical-work/support-for-developing-standards.htm`

[3]For more elaborate examples in different languages see [1].

```
<head>
  <!-- ... -->
  <annotation>
    <external corresp="annot_decl.xml"
  </annotation>
</head>
<body>
  <s xml:id = "s1">
    <graph xml:id="s1_g1">
      <terminals>
        <t xml:id="s1_t1" tiger2:corresp ="m1.maf#wf1"/> <!-- we -->
        <t xml:id="s1_t2" tiger2:corresp ="m1.maf#wf2"/> <!-- can -->
        <t xml:id="s1_t3" tiger2:corresp ="m1.maf#wf3"> <!-- see -->
          <edge tiger2:type ="dep" label="nsubj" tiger2:target ="#s1_t1"/>
          <edge tiger2:type ="dep" label="aux" tiger2:target ="#s1_t2"/>
        </t>
      </terminals>
      <nonterminals>
        <nt xml:id="s1_nt1" cat="NP">
          <edge tiger2:type ="prim" label="HD" tiger2:target ="#s1_t1"/>
        </nt>
        <nt xml:id="s1_nt2" cat="VP">
          <edge tiger2:type ="prim" label="HD" tiger2:target ="#s1_t3"/>
        </nt>
        <nt xml:id="s1_nt3" cat="VP">
          <edge tiger2:type ="prim" label="--" tiger2:target ="#s1_nt2"/>
          <edge tiger2:type ="prim" label="HD" tiger2:target ="#s1_t2"/>
        </nt>
        <nt xml:id="s1_nt4" cat="S">
          <edge tiger2:type ="prim" label="SBJ" tiger2:target ="#s1_nt1"/>
          <edge tiger2:type ="prim" label="--" tiger2:target ="#s1_nt3"/>
        </nt>
      </nonterminals>
    </graph>
  </s>
</body>
```

Figure 1: Excerpt from an example encoded in <tiger2/> (version V2.0.5).

declaration, cf. Figure 2. Furthermore, the example utilizes a standoff notation where the `terminals` refer to `wordForms` from a MAF document, cf. Figure 3. While there will be changes on the transition from <tiger2/> to ISOTiger, it is planned to still allow for inline notation in terminal nodes.

The example shows some main characteristics of the current format.[4] This format includes both a header (to describe the tags utilized in the annotations) and a body. In the body, the `<s>` element denotes a segment of the primary data, which is a more generic version of the respective TIGER-XML element denoting a sentence. A segment can contain several `<graph>` elements for syntactic graph structures, and a graph may include terminal nodes (`<t>`), non-terminal nodes (`<nt>`) and

[4]There are also additional features, such as corpus structuring and corpus metadata elements.

edges (<edge>). Nodes and edges can be typed and can be annotated by generic attribute-value-pairs defined in the <annotation> element of the header. Terminal nodes refer to a textual segment or to a word form in a morpho-syntactically annotated corpus (the latter is shown in the example), thus implementing *T_Node* from the SynAF reference model. Non-terminal nodes implement SynAF's *NT_node* and help represent hierarchical structures. <edge> elements are embedded in the element that denotes their start node, and they specify their target node by means of the @target attribute. The start node of an edge may not only be a non-terminal node, as stipulated in TIGER-XML, but also a terminal node, thus implementing the *SyntacticEdge* from the SynAF reference model. This allows representing e.g. constituency trees as well as dependency relations such as in Figure 1. The @type attribute distinguishes between different kinds of nodes and edges, e.g. *dep* vs. *prim* for dependency and constituency edges respectively in Figure 1.

Typing nodes and edges also allows to define specific attribute-value-pairs for the different node and edge types. The attributes @domain and @type of the feature element in the annotation declaration specify if the respective annotation can be applied to a terminal node, a non-terminal node or an edge (@domain), and, if applicable, to which user defined type of these (@type). Hence, the feature name *label* in the above <tiger2/> example can have different value sets for dependency and constituency edges, cf. Figure 2. Since annotations are user-defined attribute-value pairs, there are also no restrictions with respect to specific linguistic theories; however, the semantics of the annotations needs to be specified. Accordingly, every feature and feature value can be linked to a specific data category, which in the ISO setup should come from a data category registry compliant to ISO 12620:2009 [3], e.g. ISOcat[5] (see the feature value definition for *NP* in Figure 2).

To inspect more <tiger2/> examples one can also make use of a web service client[6] described by [9] that generates MAF and <tiger2/> encoded analyses for Spanish sentences.[7]

4 Open issues

The current state of the SynAF XML serialisation is still closely related to the original TIGER-XML format. This closeness was a main concern in the development of <tiger2/>. In this way, an already utilized and accepted treebank format was taken into account and enhanced, instead of inventing a completely new format.

However, considering the new flexibility of treebank annotation possibilities that is offered by the current format, the annotation declarations, such as shown in Figure 2[8], fall short in two respects: the generic attribute-value-pairs neither offer

[5] www.isocat.org

[6] http://quijote.fdi.ucm.es:8084/ClienteFreeLing/

[7] The annotations themselves are generated by means of FreeLing (http://nlp.lsi.upc.edu/freeling/demo/demo.php), a multilingual part-of-speech tagger and a parser for both phrase structure and dependency analyses.

[8] This example is based on <tiger2/>, but already includes the dcr namespace.

```
<annotation>
  <feature  name="cat" domain="nt"
     dcr:datcat ="http://www.isocat.org/datcat/DC-1506">
    <value  name="NP" dcr:datcat="http://www.isocat.org/datcat/DC-2256"/>
    <value  name="S" dcr:datcat="http://www.isocat.org/datcat/DC-2295"/>
    <value  name="VP" dcr:datcat="http://www.isocat.org/datcat/DC-2255"/>
  </feature>
  <feature  name="label" domain="edge" type="prim"
     dcr:datcat ="http://www.isocat.org/datcat/DC-5596">
    <value  name="HD" dcr:datcat="http://www.isocat.org/datcat/DC-2306"/>
    <value  name="SBJ" dcr:datcat="http://www.isocat.org/datcat/DC-2261"/>
    <value  name="--"/>
  </feature>
  <feature  name="label" domain="edge" type="dep"
     dcr:datcat ="http://www.isocat.org/datcat/DC-2304">
    <value  name="nsubj">nominal subject</value>
    <value  name="aux" dcr:datcat="http://www.isocat.org/datcat/DC-2262"/>
  </feature>
</annotation>
```

Figure 2: Document annot_decl.xml containing external annotation declarations.

```
<wordForm xml:id="wf1" lemma="we" tokens="#t1"/>
<wordForm xml:id="wf2" lemma="can" tokens="#t2"/>
<wordForm xml:id="wf3" lemma="see" tokens="#t3"/>
```

Figure 3: Excerpt from a MAF document (m1.maf).

the full descriptive power of feature structures as defined in standards from ISO and TEI [4, 5], nor do they match the standard representation. Utilizing the FSR and FSD standards as in MAF (ISO 24611 - sections 7.2 and 7.4)[9] would however go far beyond the original TIGER-XML format.

Figure 4 shows the *NP* node and an outgoing edge, where the annotations are encoded as feature structures. On the one hand, we would no longer have to deal with generic XML attributes for nodes and edges, and the <tiger2/> elements `<feature>` and `<value>` would no longer be needed. On the other hand, we would (i) introduce structured annotations, which might not be completely mappable onto formats with non-structured annotations and (ii) introduce a slightly more verbose representation. However, utilizing FSR would of course also allow for the use of libraries, which could be declared centrally (or externally) and be referred to by a new ISOTiger attribute of nodes and edges. Furthermore, applying the ISO and TEI standards on feature structures fosters an integration of the different standardization approaches. A standoff notation making reference to external feature structure declarations could also allow for structured annotations as an option, while still keeping the possibility of specifying simple attribute-value-pairs.

The second aspect under discussion is a reference mechanism to primary data,

[9] Section 7.4 in MAF states how to declare and reuse FSR libraries and Section 7.2 defines how to actually annotate word forms with feature structures.

```
<nt xml:id="s1_nt1">
  <fs>
    <f name="cat" dcr:datcat="http://www.isocat.org/datcat/DC-1506">
      <symbol value="NP" dcr:datcat="http://www.isocat.org/datcat/DC-2256"/>
    </f>
  </fs>
  <edge xml:id="s1_e3" type="prim" target="#s1_t1">
    <fs>
      <f name="label" dcr:datcat="http://www.isocat.org/datcat/DC-5596">
        <symbol value="HD" dcr:datcat="http://www.isocat.org/datcat/DC-2306"/>
      </f>
    </fs>
  </edge>
</nt>
```

Figure 4: Open issue: feature structures in ISOTiger

```
Locations in the document:
|w|e|  |c|a|n|  |s|e|e|
0 1 2 3 4 5 6 7 8 9 10

<terminals>
  <t xml:id="s1_t1" from="0" to="2"/> <!-- we -->
  <t xml:id="s1_t2" from="3" to="6"/> <!-- can -->
  <t xml:id="s1_t3" from="7" to="10"/> <!-- see -->
</terminals>
```

Figure 5: Open issue: reference mechanism to primary data in ISOTiger

for cases where there is no morpho-syntactic annotation, yet SynAF terminals are required to be represented in a standoff way. Therefore, for such cases, ISOTiger could refer to LAF [7], where the generic reference mechanism introduces virtual anchors in between base units of the primary data representation (e.g. characters), which can be referenced to select a region from the primary data. Figure 5 includes an example utilizing possible new ISOTiger attributes `@from` and `@to`, together with the idea of the virtual anchors. A related representation has been proposed in MAF [6]. However according to the SynAF – Part 1 metamodel, terminals in SynAF are equivalent to word forms, and can thus for example also be defined over multiple spans. Furthermore, pointing directly from a terminal node to the primary data might hide the essential distinction between tokens and word forms. Therefore a direct reference from terminals to primary data would only be allowed in exceptional cases.

It should be noted that the two ISOTiger examples in Figure 4 and Figure 5 only provide suggestions for further developments to transform <tiger2/> into the ISOTiger standard, and are likely to undergo changes before the standardization process is complete. A discussion in the community on these open issues, as well as on the current state of ISOTiger, would help to meet the requirements of the users in this ongoing standardisation work.

References

[1] Sonja Bosch, Key-Sun Choi, Éric La de Clergerie, Alex Chengyu Fang, Gertrud Faaß, Kiyong Lee, Antonio Pareja-Lora, Laurent Romary, Andreas Witt, Amir Zeldes, and Florian Zipser. <tiger2/> as a standardised serialisation for ISO 24615 – SynAF. In Iris Hendrickx, Sandra Kübler, and Kiril Simov, editors, *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories (TLT11)*, pages 37–60, Lisbon, Portugal, 2012. Edições Colibri, Lisboa.

[2] Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620, 2004.

[3] ISO 12620:2009 Terminology and other language and content resources – Specification of data categories and management of a Data Category Registry for language resources.

[4] ISO 24610-1:2006 Language resource management – Feature structures – Part 1: Feature structure representation.

[5] ISO 24610-2:2011 Language resource management – Feature structures – Part 2: Feature system declaration.

[6] ISO 24611:2012 Language resource management – Morpho-syntactic annotation framework (MAF).

[7] ISO 24612:2012 Language resource management – Linguistic annotation framework (LAF).

[8] Esther König, Wolfgang Lezius, and Holger Voormann. *TIGERSearch 2.1 User's Manual. Chapter V - The TIGER-XML treebank encoding format*. IMS, Universität Stuttgart, 2003.

[9] Antonio Pareja-Lora, Guillermo Cárcamo-Escorza, and Alicia Ballesteros-Calvo. Standardisation and interoperation of morphosyntactic and syntactic annotation tools for spanish and their annotations. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA).

Challenges in Enhancing the *Index Thomisticus* Treebank with Semantic and Pragmatic Annotation

Berta González Saavedra and Marco Passarotti

Università Cattolica del Sacro Cuore, Milan, Italy
E-mail: {berta.gonzalezsaavedra; marco.passarotti}@unicatt.it

Abstract

Building treebanks for ancient languages, like Ancient Greek and Latin, raises a number of challenges that have restricted so far the enhancement of the available treebanks for Classical languages with higher levels of analysis, like semantics and pragmatics. By detailing the semi-automatic annotation procedures and the treatment of two specific constructions of Latin, this paper presents the first steps towards the semantic and pragmatic annotation of a Medieval Latin treebank, the *Index Thomisticus* Treebank.

1 Introduction

When working with ancient/dead languages, like Ancient Greek and Latin, a number of specific aspects must be considered that affect the construction of Language Resources (LRs) like treebanks. First, there are not native speakers (and, actually, no speakers at all), which is not a trivial matter, since more than one interpretation of the same text is often possible, stemming from two millennia of philological work. Interpretation can be difficult also because most of the extant texts belong to a high register, which in turn makes the corpora for Classical languages poorly representative. Finally, building a LR for a Classical language requires a close collaboration between scholars from (often conservative areas in) the Humanities and computational linguists, which is not yet widespread in the research community.

These features raise a number of challenges for those scholars who want to build new LRs for Classical languages, especially when higher levels of analysis (like semantics and pragmatics) are concerned, since they depend heavily on deep textual interpretation. So far, this has restricted the enhancement of the available treebanks for Classical languages with such levels of annotation[1]. However, the times are mature enough also for such treebanks to get out of the cradle of surface syntactic analysis and to finally include semantic information. This paper presents the first steps towards the semantic and pragmatic annotation of a Medieval Latin treebank, the *Index Thomisticus* Treebank (IT-TB).

[1] Some semantic annotation of Classical languages is available in the PROIEL corpus [1].

2 From Analytical to Tectogrammatical Analysis

The IT-TB is a dependency-based treebank consisting of the texts of Thomas Aquinas and designed in accordance with the Prague Dependency Treebank (PDT) annotation style [3]. The PDT is based on Functional Generative Description (FGD), a theoretical framework developed in Prague, which motivates the three-layer analysis of sentences provided by the PDT [5]: (a) a morphological layer, consisting of lemmatization and morphological analysis; (b) a surface syntax layer (called "analytical"); (c) a semantic and pragmatic layer (called "tectogrammatical").

The development of each layer requires the availability of the previous one(s). Both the analytical and the tectogrammatical layers describe the sentence structure with dependency tree-graphs, respectively named Analytical Tree Structures (ATSs) and Tectogrammatical Tree Structures (TGTSs).

In ATSs every word and punctuation mark of the sentence is represented by a node of a rooted dependency tree. The edges of the tree correspond to dependency relations that are labelled with (surface) syntactic functions called "analytical functions" (like Subject, Object etc.).

TGTSs describe the underlying syntactic structure of the sentence, conceived as the semantically relevant counterpart of the grammatical means of expression (described by ATSs). The nodes of TGTSs represent autosemantic words only, while function words and punctuation marks are left out. The nodes are labelled with semantic role tags called "functors". These are divided into two classes according to valency: (a) arguments, called "inner participants", i.e. obligatory complementations of verbs, nouns, adjectives and adverbs: Actor, Patient, Addressee, Effect and Origin; (b) adjuncts, called "free modifications": different kinds of adverbials, like Place, Time, Manner etc. TGTSs feature two dimensions that represent respectively the syntactic structure of the sentence (the vertical dimension) and its information structure ("topic-focus articulation"), based on the underlying word order (the horizontal dimension). Also ellipsis resolution and coreferential analysis are performed at the tectogrammatical layer and are represented in TGTSs through newly added nodes (ellipsis) and arrows (coreference).

The first two layers of annotation are already available for the IT-TB, while the tectogrammatical annotation of data has just been started. The present size of the IT-TB is 249,271 nodes, in 14,447 sentences. So far, the first 600 sentences of *Summa Contra Gentiles* (SCG) have been annotated at tectogrammatical layer (8,910 nodes). The annotation guidelines used are those for the tectogrammatical layer of the PDT [2].

2.1 Annotation Procedures

The workflow for tectogrammatical annotation in the IT-TB is based on TGTSs automatically converted from ATSs. The TGTSs that result from the conversion are then checked and refined manually by two annotators. The conversion is performed by adapting to Latin a number of ATS-to-TGTS conversion modules provided by

the NLP framework *Treex* [4]. Relying on ATSs, the basic functions of these modules are: (a) to collapse ATSs nodes of function words and punctuation marks, as they no longer receive a node for themselves in TGTSs, but are included into the nodes for autosemantic words; (b) to assign "grammatemes", i.e. semantic counterparts of morphological categories (for instance, *pluralia tantum* are tagged with the number grammateme "singular"); (c) to resolve grammatical coreferences, i.e. coreferences in which it is possible to pinpoint the coreferred expression on the basis of grammatical rules (mostly with relative pronouns); (d) to assign functors.

Tasks (a) and (b) are quite simple and the application of the modules that are responsible for them results in good accuracy on average.

Collapsing nodes for not autosemantic words and punctuations relies on the structure of the ATSs given in input: in this respect, Latin does not feature any specific property to require for modifications of the ATS-to-TGTS conversion procedures already available in *Treex* and already applied to other languages.

Assigning grammates is a task strictly related with the lexical properties of the nodes in TGTSs. Thus, we are in the process of populating the modules that assign grammatemes with lists of words (lemmas) that are regularly assigned the same grammatemes.

The automatic processing of task (c) is just at the beginning. So far, the modules are able to resolve only those grammatical coreferences that show the simplest possible construction occurring in ATSs, i.e. that featuring an occurrence of a relative pronoun (*qui* in Latin) direcly depending on the main predicate of the relative clause. However, this construction is the most frequent for relative clauses in the IT-TB: among the 326 occurrences of *qui* in our data, 176 present this construction and are correctly assigned their grammatical coreference by the conversion modules. The remaining 150 occurrences either lack grammatical coreference or do occur in more complex constructions.

In order to assign functors automatically (task (d)), we rely both on analytical functions and on lexical properties of the ATSs nodes. For instance, all the nodes with analytical function Sb (Subject) that depend on an active verb are assigned functor ACT (Actor), and all the main predicates of subclauses introduced by the subordinating conjunction *si* (*if*) are assigned functor COND (Condition). Table 1 reports the number of nodes occurring in the TGTSs of the first 600 sentences of SCG automatically produced by the modules (column "Parsed") and in the same ones manually checked and modified (column "Gold"). The column "Correct" reports the number of nodes that are assigned the correct functor in the automatically parsed data[2]. Precision, recall and F-score of automatic functor assignment are provided [6].

The overall accuracy of the automatic assignment of functors (provided by the F-score) is around 66%. However, since the accuracy varies heavily by functor,

[2]The nodes that are newly added in TGTSs (for ellipsis resolution purposes) are not considered in table 1, since no reconstructed node is supplied in the TGTSs built automatically by the conversion modules. The automatically parsed data include 101 nodes more than the gold standard; these nodes are those that were manually collapsed and included into others.

Parsed	Gold	Correct	Precision	Recall	F-Score
6620	6519	4318	65.23	66.24	65.73

Table 1: Evaluation of automatic functor assignment

table 2 reports the evaluation of the automatic assignment for the ten most frequent functors in the gold standard that occur at least once also in the automatically parsed data[3]. Precision, recall and F-score are reported for each functor.

Functor	Parsed	Gold	Correct	Precision	Recall	F-Score
PAT	1249	1307	964	77.18	73.76	75.43
RSTR	2752	1124	1052	38.23	93.59	54.28
ACT	774	858	628	81.14	73.19	76.96
PRED	515	503	447	86.8	88.87	87.82
PREC	220	266	215	97.73	80.83	88.48
CONJ	256	255	238	92.97	93.33	93.15
RHEM	231	239	221	95.67	92.47	94.04
MEANS	99	211	91	91.92	43.13	58.71
APP	65	208	62	95.38	29.81	45.42
MANN	82	207	54	65.85	26.09	37.37

Table 2: Evaluation of automatic functor assignment by single functors

5,785 out of the 6,519 not newly added nodes in the gold standard are assigned a functor that is present at least once also in the automatically parsed data. The 734 nodes remaining are those that receive a functor that the modules for automatic conversion from ATSs to TGTSs have never assigned. Among these, the most frequent are the locative functors DIR1, DIR2, DIR3 and LOC (respectively, From, Which way, To and Where: 204 cases), REG (Regard: 101), CRIT (Criterion: 61), CPR (Comparison: 59) and ADDR (Addressee: 58).

The results reported in table 2 show that the modules for automatic conversion generally achieve high precision (always higher than 80% but for PAT and MANN), while recall shows lower values. In particular, recall is always lower than precision but for PRED and CONJ (where the two values are very close). The functor RSTR must be evaluated separately, since it is the functor that is assigned by default in those cases where no rule is available in the modules to assign a functor. This motivates its very low precision and, conversely, its high recall.

[3]ACT: Actor; APP: Appurtenance; CONJ: (paratactic) Conjunction; MANN: Manner; MEANS: Means; PAT: Patient; PREC: reference to Preceding text; PRED: Predicate of the main clause; RHEM: Rhematizer; RSTR: Restrictor. For more details about functors, see [2].

2.2 Modifications to the PDT Manual

Performing tectogrammatical annotation of Latin texts has required a number of modifications to the rules stated in the PDT manual. In the following, we discuss two of such modifications, one dealing with a typical Latin construction (passive periphrastics), the other with the semantics of one specific subordinating conjunction (*ne*).

The passive periphrastic construction in Latin expresses the idea of obligation. It consists of one form of the verb *sum* (*to be*) and of a gerundive, a mood for verbal adjectives (always bearing a passive meaning). In the analytical layer, the gerundive is treated as the predicate nominal depending on the node for *sum*.

In TGTSs, the node for a modal verb headings an infinitive (e.g. *debeo dicere*, *I must say*) is collapsed and included into the node for the infinitive and its meaning (e.g. obligation for *debeo*) is reported in a specific grammateme assigned to the infinitive ("deontmod": deontic modality). We treat the passive periphrastic construction in Latin consistently. Although the node for the verb *sum* heads this construction in ATSs, it still acts as an auxiliary verb for the gerundive; thus, in TGTSs the node for *sum* in passive periphrastics is collapsed and included into the node of the gerundive, which becomes the head of the construction. This implies that the values of all the grammatemes of *sum* are assigned to the gerundive. Among the grammatemes, deontmod is assigned the value for obligation ("hrt"). The functor of *sum* is assigned to the gerundive, and all the nodes depending on *sum* are made dependent on the gerundive. According to the passive meaning of the gerundive, the subject of *sum* in the ATS is assigned the functor PAT in the TGTS.

For instance, in the clause *quae de deo* [...] *consideranda sunt* (*those things about God that must be considered*; SCG, 1.9), the node for *sum* (lemma of *sunt*) is included into that for *considero* (lemma of *consideranda*), which is assigned the value "hrt" for the grammateme deontmod. All the nodes depending on *sum* in the ATS are made dependent on *considero* in the TGTS and the node for *qui* (lemma of *quae*), which is the subject of *sunt* in the ATS, is assigned the functor PAT.

For what concerns *ne* (*in order not to*), it is a subordinating conjunction that introduces clauses expressing a negative purpose, or a negative imperative. The meaning of *ne* is, thus, composite: negative + purpose/imperative. Like for all the subordinating conjunctions in TGTSs, the node for *ne* is collapsed and included into the node for the head-verb of the clause introduced by *ne*. Given the composite nature of the meaning carried by *ne*, this makes the semantic value of negation of *ne* to be lost in the TGTS. We solve this loss by adding in the TGTS a new node with the technical lemma "#Neg" depending on the head-verb of the clause.

For instance, in the clause *ne te inferas in illud secretum* (*do not get into that secret*; SCG, 1.8), the node for *ne* is included into that for *infero* (lemma of *inferas)* and a new node with lemma "#Neg" is added depending on *infero*.

3 Conclusions

Moving from analytical to tectogrammatical annotation concerns the long debated topic of the relations holding between syntax and semantics.

On one side, several aspects of tectogrammatical annotation can be automatically induced from ATSs. In our work, this is done by applying to Latin a number of ATS-to-TGTS conversion modules already used for other (modern) languages, thus opening research questions in diachronic comparative linguistics.

On the other side, starting the tectogrammatical annotation of a treebank that includes texts in a dead language, which lacks advanced NLP tools able to process semantics, demands a significant amount of manual work. In fact, so far ellipsis resolution, topic-focus articulation and textual coreference (i.e. coreference realized not only by grammatical means, but also via context, mostly with non-relative pronouns) are performed fully manually in the IT-TB.

In the near future, we have to both increase the recall of the already available rules for functor assignment and to build new ones for the automatic processing of both ellipsis resolution and textual coreference. Further, once a sufficient amount of annotated data will be available, we shall start to train stochastic NLP tools to perform semi-automatic annotation.

References

[1] Haug Dag and Jøhndal Marius. Creating a Parallel Treebank of the Old Indo-European Bible Translations. In *Proceedings of LaTeCH 2008*, pages 27-34, Marrakech, 2008.

[2] Mikulová Marie et alii. *Annotation on the Tectogrammatical Layer in the Prague Dependency Treebank*. Institute of Formal and Applied Linguistics, Prague, 2006. Available at `http://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/tlayer/html/index.html`.

[3] Passarotti Marco. Theory and Practice of Corpus Annotation in the *Index Thomisticus* Treebank. *Lexis*, 27, pages 5-23, 2009.

[4] Popel Martin and Žabokrtský Zdeněk. TectoMT: Modular NLP Framework. In *Proceedings of IceTAL, 7th International Conference on Natural Language Processing*, pages 293-304, Reykjavík, 2010.

[5] Sgall Petr, Hajicová Eva, and Panevová Jarmila. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Reidel, Dordrecht, 1986.

[6] Van Rijsbergen Cornelis Joost. *Information Retrieval*. Butterworths, London, 1979.

TüBa-D/W: a large dependency treebank for German

Daniël de Kok

Seminar für Sprachwissenschaft
University of Tübingen
E-mail: daniel.de-kok@uni-tuebingen.de

Abstract

We introduce a large, automatically annotated treebank, based on the German Wikipedia. The treebank contains part-of-speech, lemma, morphological, and dependency annotations for the German Wikipedia (615 million tokens). The treebank follows common annotation standards for the annotation of German text, such as the STTS part-of-speech tag set, TIGER morphology and TüBa-D/Z dependency structure.

1 Introduction

In this paper we introduce the automatically annotated TüBa-D/W dependency treebank. Our goal with TüBa-D/W is to provide a large treebank of modern written German, that follows common annotation standards and is freely available under a permissive license. The TüBa-D/W is based on Wikipedia text, consists of 36.1 million sentences (615 million tokens), and is distributed under the same license as Wikipedia.[1] After discussing related work, we will describe how the material for this treebank was collected. Then we will discuss the annotation layers in the treebank and how they are constructed. Finally, we will discuss the treebank format and future work.

2 Related work

In the past two decades three major manually corrected treebanks have been developed for German: NEGRA [6], TIGER [5], and TüBa-D/Z [20]. Although these treebanks are in principle phrase structure treebanks, edges are labeled with grammatical roles. The presence of grammatical roles makes them amenable for

[1] https://creativecommons.org/licenses/by-sa/3.0/

conversion to dependency structure. Such conversions exist for both TIGER [19] and TüBa-D/Z [21].

Recent research has shown that larger automatically annotated treebanks can be a useful resource to gauge the distribution of lexical or syntactic phenomena in a language [4, 16, 10]. Although the use of automatic annotation usually implies a loss of annotation accuracy compared to manually corrected treebanks, their size makes it possible to get more fine-grained statistics and discover low-frequency phenomena. For instance, the largest of the aforementioned treebanks (TüBa-D/Z) has annotations for 1.6 million tokens, while the automatically annotated corpus used in [10] is more than two orders of magnitude larger.

Given that vast computational resources and fast parsers are now readily available, it is perhaps surprising that the number of large automatically annotated treebanks for German is small. The TüPP-D/Z [14] corpus contains partial parses for 204 million tokens from the German newspaper taz. The VISL Corpuseye provides a public search interface and syntactic analyses for Europarl (15 million tokens), Wikipedia (28.7 million tokens), and the Leipzig internetcorpus (47 million tokens). Unfortunately, the annotations do not follow common annotation standards for German and the Wikipedia material is older and substantially smaller than that in the present work. The German reference corpus (DeReKo) contains a recent version of Wikipedia, including discussion pages [7]. However, this corpus does not contain syntactic annotations.

Our contribution is a dependency treebank that is larger than the aforementioned treebanks, using annotation standards that are broadly used for German resources, using a pipeline that can be reproduced and applied to new material easily.

3 Material

For the construction of the treebank, we use a dump of the German Wikipedia that was downloaded on May 6, 2014. Since Wikipedia dumps contain MediaWiki markup, we use the Wikipedia Extractor[2] to convert the Wikipedia dump to plain text. We then convert the plain-text files to the Text Corpus Format (TCF) [9]. The conversion to TCF allows us to process Wikipedia using WebLicht[11], an environment for automatic annotation of corpora. In WebLicht users can compose annotation pipelines of annotation tools that are hosted by CLARIN centers. After composing the pipeline in WebLicht, the corpus was processed using WebLicht as a Service [8], which is a non-interactive version of WebLicht that is tailored to processing of large corpora.

Another preprocessing step that was required, was the replacement of 78 unicode characters that are problematic for many for off-the-shelf natural language processing tools. This set of characters mainly consists of quotation characters, dashes/underscores, and arithmetic operators. To this end, we developed and added

[2] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

a small annotation tool to WebLicht that performs replaced these characters with ASCII equivalents.

4 Annotations

In this section, we give an overview of the annotation layers in the treebank. For each layer we discuss the annotation standard and the tools we use for the automatic annotation.

Tokenization The tokenization and sentence splitting of the corpus is performed using the OpenNLP[3] tokenizer. We retrained the tokenizer on a detokenized version of the TüBa-D/Z treebank [20], release 9. Detokenization reverses tokenization using a set of rules, inserting special markers where the splits occurred. For instance, the tokenized sentence:

> " Gut für sie , gut für Europa " steht klein darunter .

is detokenized to:

> "<SPLIT>Gut für sie<SPLIT>, gut für Europa<SPLIT>" steht klein darunter<SPLIT>.

We used the detokenization rules that were provided by OpenNLP. However, we found that we had to add a rule to handle forward slash (/) characters. The OpenNLP tokenizer obtained an average f-score of 0.9986 in ten-fold cross-validation. For the sentence splitter, we use the model provided for German by OpenNLP.

One problem that we found in the sentence splitter is that it merges headlines with the first first sentence, because headlines usually do not end with end-of-sentence punctuation in Wikipedia. Fortunately, since new lines do not occur in running text in the plain text dump, we could segment the text by using newline characters as boundaries. We then apply the sentence splitter and tokenizer per segment.

Part-of-speech tags The treebank is tagged using the OpenNLP POS tagger, trained on TüBa-D/Z release 9. TüBa-D/Z uses the Stuttgart-Tübingen-TagSet (STTS) [17]. Two changes were made to the tag set in TüBa-D/Z before training the model to make it compatible with the tag set of the TIGER treebank [1]: (1) the pronominal adverb tag was changed from *PROP* to *PROAV* and (2) TIGER does not make the distinction between between attributive indefinite pronouns with and without determiner (*PIDAT* and *PIAT*), so we replaced all *PIDAT* tags by *PIAT*.

The OpenNLP tagger has an accuracy of 96.93% when performing ten-fold cross-validation on TüBa-D/Z with these modifications.

[3] https://opennlp.apache.org/

Morphology Morphological annotations are added using RFTagger [18], with the model for German included in RFTagger, which was trained on the TIGER treebank [5]. Morphological information that is added include gender, case, number, person, tense, and degree. We add morphology annotations because it improves the output of the dependency parser and is useful in some types of treebank queries.

The morphology and part-of-speech tag layers provide overlapping annotations. For instance, the OpenNLP tagger marks a finite verb as *VVFIN*, while RFTagger assigns the category *verb* and attributes such as the tense, person and number. Sometimes the analyses of the part-of-speech tagger and the morphological tagger diverge. In such cases, we do not perform any filtering or post-processing. The parser, which is discussed below, uses both part-of-speech tags and morphological information as features. We expect the training procedure to reduce the weights of features in cases of systematic errors.

Dependency structures The sentences are dependency parsed using the MaltParser [15]. We constructed a model that uses tokens, part-of-speech tags, and morphology as features. The feature templates were constructed using MaltOptimizer [2], using 17072 dependency structures from TüBa-D/Z release 9 as training data with cross-validation on 17071 dependency structures from TüBa-D/Z. We then trained the model using the aforementioned training instances and evaluate it on a third, held-out set of another 17070 dependency structures. In these sets, we used gold standard part-of-speech tags and the output of RFTagger for creating morphological features. The resulting model has a labeled attachment score of 89.0% (88.2% without morphology features).

Lemmatization For lemmatization, we use the SepVerb lemmatizer. This is a lemmatizer that was developed in-house to produce lemmatizations that follow TüBa-D/Z [22]. It first uses the MATE lemmatizer [3], trained on a simplified version of the TüBa-D/Z and then applies post-processing rules to obtain the canonical TüBa-D/Z lemmatization.

TüBa-D/Z lemmatization differs from standard lemmatization in the following ways: (1) the suffix *%passiv* is added to *werden* in passive constructions; (2) the suffix *%aux* is added to auxiliary and modal verbs; (3) particles are added to and marked in separable verbs, for instance *gehen* in *geht davon aus* 'to assume' is lemmatized as *aus#gehen*; (4) reflexives get the lemma *#refl*; and (5) *zu* is removed from infinitives that contain *zu*, for instance *einzufordern* becomes *ein#fordern* 'to demand'. Furthermore, (6) SepVerb uses the lemma *d* and *ein* respectively for definite and indefinite articles.

Transformations for 4-6 can be performed using rules that use the lemma and part-of-speech. However, transformations for 1-3 require syntactic information. For this reason, the SepVerb lemmatizer requires input from a parser. The transformations in SepVerb operated on constituency trees. For the construction of dependency treebanks, we extended SepVerb with rules that work on dependency

structures. The rules for 1-3 for a lemma *l* are:

1. If lemma *l* is the head of lemma *m* with dependency relation *AVZ* (separable verb prefix) and *m* is marked with part-of-speech tag *PTKVZ* (verb particle), then *l* is replaced by *l#m* and *m* is replaced by the empty lemma.

2. Else if $l = werden$ is the head of a token with the tag *VVPP* (perfect participle) with dependency relation *AUX*, *l* is replaced by *l*%passiv.

3. Else if *l* dominates a token with the dependency relation *AUX*, *l* is replaced by *l*%aux.

The lemmatizer uses a model that was trained on TüBa-D/Z release 8 and applies verb processing rules after lemmatization. The lemmatizer achieves 97.66% accuracy in 10-fold cross-validation on the TüBa-D/Z.

5 Availability and future work

TüBa-D/W is provided in the CONLL-X dependency format. Moreover, we added the treebank to the TüNDRA [12] visualization and search tool. To this end, we optimized TüNDRA to work efficiently with treebanks of this size [8].

This paper only describes the first version of TüBa-D/W. We plan to provide updates of the treebank. The initial changes will focus on making the annotations as close to TüBa-D/Z as possible. For instance, we plan to use the morphological information from RFTagger and the dependency information from MaltParser to use gender-specific lemmas (e.g. *der*, *die*, *das*) as in TüBa-D/Z. We would also like to extend the morphology layer such that it provides features in TüBa-D/Z-style in addition to the current TIGER morphology.

Statistical dependency parsing is an active field of work and state-of-the-art parsers such as TurboParser [13] provide an improvement over the MaltParser in our initial experiments with German. If performance and computing facilities permit, we might parse a future version with a parser such as TurboParser to improve the dependency annotations.

Acknowledgments

The development of this treebank was supported by CLARIN-D. We would like to thank the CLARIN-D center IMS Stuttgart for making the RFTagger available as a WebLicht web service.

References

[1] Stefanie Albert, Jan Anderssen, Regine Bader, Stephanie Becker, Tobias Bracht, Sabine Brants, Thorsten Brants, Vera Demberg, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Hagen Hirschmann, Juliane Janitzek, Carolin Kirstein, Robert Langner, Lukas Michelbacher, Oliver Plaehn, Cordula Preis, Marcus Pußel, Marco Rower, Bettina Schrader, Anne Schwartz, George Smith, and Hans Uszkoreit. TIGER Annotationsschema. Universität des Saarlandes, Universität Stuttgart and Universität Potsdam, 2003.

[2] Miguel Ballesteros and Joakim Nivre. MaltOptimizer: An optimization tool for MaltParser. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 58–62. Association for Computational Linguistics, 2012.

[3] Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 33–36. Association for Computational Linguistics, 2010.

[4] Gosse Bouma and Jennifer Spenader. The distribution of weak and strong object reflexives in Dutch. In F van Eynde, A Frank, K D Smedt, and G van Noord, editors, *Proceedings of the Seventh International Workshop on Treebanks and Linguistic Theories (TLT 7)*, pages 103–114, 2009.

[5] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories, September 20-21 (TLT02)*, Sozopol, Bulgaria, 2002.

[6] Thorsten Brants, Wojciech Skut, and Hans Uszkoreit. Syntactic annotation of a german newspaper corpus. In Anne Abeillé, editor, *Treebanks*, volume 20 of *Text, Speech and Language Technology*, pages 73–87. Springer Netherlands, 2003.

[7] Noah Bubenhofer, Stefanie Haupt, and Horst Schwinn. A comparable Wikipedia corpus: From Wiki syntax to POS tagged XML. *Multilingual Resources and Multilingual Applications*, 96B:141–144, 2011.

[8] Daniël de Kok, Dörte de Kok, and Marie Hinrichs. Build your own treebank. In *Proceedings of the CLARIN Annual Conference 2014*, 2014.

[9] Ulrich Heid, Helmut Schmid, Kerstin Eckart, and Erhard W Hinrichs. A corpus representation format for linguistic web services: The D-SPIN text corpus format and its relationship with ISO standards. In *Proceedings of LREC 2010, Malta*, 2010.

[10] Erhard Hinrichs and Kathrin Beck. Auxiliary fronting in German: A walk in the woods. In *The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*, page 61, 2013.

[11] Erhard Hinrichs, Marie Hinrichs, and Thomas Zastrow. Weblicht: Web-based LRT services for German. In *Proceedings of the ACL 2010 System Demonstrations*, pages 25–29. Association for Computational Linguistics, 2010.

[12] Scott Martens. TüNDRA: A web application for treebank search and visualization. In *The Twelfth Workshop on Treebanks and Linguistic Theories (TLT12)*, page 133, 2013.

[13] Andre Martins, Miguel Almeida, and Noah A. Smith. Turning on the Turbo: Fast third-order non-projective Turbo Parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617–622, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[14] Frank Henrik Müller. Stylebook for the Tübingen partially parsed corpus of written German (TüPP-D/Z). 2004.

[15] Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(02):95–135, 2007.

[16] Tanja Samardžić and Paola Merlo. The meaning of lexical causatives in cross-linguistic variation. *Linguistic Issues in Language Technology*, 7:1–14, 2012.

[17] Anne Schiller, Simone Teufel, and Christine Thielen. Guidelines für das Tagging deutscher Textcorpora mit STTS. *Manuscript, Universities of Stuttgart and Tübingen*, 1995.

[18] Helmut Schmid and Florian Laws. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 777–784. Association for Computational Linguistics, 2008.

[19] Wolfgang Seeker and Jonas Kuhn. Making ellipses explicit in dependency conversion for a german treebank. In *Proceedings of LREC 2012, Istanbul*, pages 3132–3139, 2012.

[20] Heike Telljohann, Erhard W Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. Stylebook for the tübingen treebank of written German (TüBa-D/Z). In *Seminar für Sprachwissenschaft, Universität Tübingen, Germany*, 2003.

[21] Yannick Versley. Parser evaluation across text types. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, 2005.

[22] Yannick Versley, Kathrin Beck, Erhard Hinrichs, and Heike Telljohann. A syntax-first approach to high-quality morphological analysis and lemma disambiguation for the TüBa-D/Z treebank. In *Ninth International Workshop on Treebanks and Linguistic Theories*, page 233, 2010.

What can linguists learn from some simple statistics on annotated treebanks

Jiří Mírovský and Eva Hajičová

Charles University in Prague
Faculty of Mathematics and Physics
E-mail: mirovsky,hajicova@ufal.mff.cuni.cz

Abstract

The goal of the present contribution is rather modest: to collect simple statistics carried out on different layers of the annotation scenario of the Prague Dependency Treebank (PDT; [1]) in order to illustrate their usefulness for linguistic research, either by supporting existing hypotheses or suggesting new research questions or new explanations of the existing ones. For this purpose, we have collected the data from the already published papers on PDT (quoted at the relevant places), adding some more recent results and drawing some more general consequences relevant for Czech grammar writers.

1 Frequency of occurrences of particular phenomena

1.1 Non-projectivity of word order

Projectivity of dependency trees representing the syntactic structure of sentences has been and still is a frequently discussed property of the trees as this property offers a possible restriction on syntactic representations. It is well known that word order in Czech is not in principle guided by grammatical rules, so that it might be expected that the instances of non-projectivities in Czech might not be frequent. A detailed analysis of non-projective constructions in Czech is given in [13]. His statistical data are based on the PDT analytical (surface structure) level comprising 73,088 non-empty sentences and 1,255,590 words (incl. punctuation marks). There are 16,920 sentences (23.2%) in the collection that contain at least one non-projectivity (i.e. including at least one node in a non-projective position). However, from the point of view of the total number of nodes in the analyzed collection, there were only 23,691 (1.9%) nodes hanging in a non-projective way. As the PDT annotation is carried out both at the surface syntactic as well as at the underlying syntactic level, it was possible to compare the two levels. The statistical findings indicate that 71.47% of non-projectivities stem from special properties of the surface syntactic level: function words separated from the lexical words they

are associated with and analytic verb forms (50.54%), split constructions such as phrasemes and noun groups (2.46%), placement of particles "outside" the sentence (17%), grammatical restrictions on surface word order (1.47%). It seems then plausible to work with the assumption that the underlying, tectogrammatical level can be characterized as projective. Moreover, the statistical data have indicated that the main cause of non-projectivities is the information structure of the sentence (e.g. in the case of split noun groups). Even here more detailed classification of the statistical data give us some guidance (see [5]).

1.2 Information structure annotation of the Czech corpus (TFA)

In the theoretical account of topic-focus articulation (TFA) within the framework of the Functional Generative Description, the dichotomy of topic (what is the sentence about) and focus (what it says about the topic) is understood as based on the primary notion of contextual boundness. Every node of the tectogrammatical dependency tree carries an index of contextual boundness: a node can be either contextually bound (*t*, or, in case of contrast, *c*) or non-bound (*f*). For the identification of the dichotomy of topic and focus on the basis of contextual boundness, a rather strong hypothesis was formulated, namely that the topic-focus distinction can be made depending on the status of the main verb (i.e. the root) of the sentence and its immediate dependents.

To test this hypothesis, an implementation of the algorithm was applied to the whole PDT data. The results reported in detail in [4] can be summarized as follows: focus consisting of a contextually non-bound verb and its contextually non-bound subtrees occurred in 85.7%; focus consisting only of the contextually non-bound elements depending on the contextually bound verb together with the subtrees depending on them: 8.58%. There occurred about 4.47% of special cases and an ambiguous partition was found in 1.14% of cases. No focus was identified in 0.11% of cases.

The results indicate that a clear division of the sentence into topic and focus according to the hypothesized rules has been achieved in 94.28% of sentences to which the procedure has been applied; the real problem of the algorithm then rests with the case of ambiguous partition (1.14%) and cases where no focus was recognized (0.11%). The results achieved by the automatic procedure were then compared to the judgements of Czech speakers ([14]). The annotators were instructed to mark – according to their intuition – every single word in the sentence as belonging to topic or focus and, at the same time, they were supposed to mark which part of the sentence they understand as topic and which part as focus. It is interesting to note that the annotators' agreement in the assignments of individual words in the sentences to topic or to focus was much higher (about 75% in both the three and six parallel analyses compared to 36% of the assignments of the topic and focus as a whole) than the assignments of the topic-focus boundary.

The work on this step is still in progress. It is a matter of course that the variability of manual solutions must be taken into considerations; we are aware of

the fact that while we get only a single, unambiguous result from the automatic procedure, more ways of interpretation could be correct.

The empirical study of Czech texts has led to the assumption that the ordering of the elements in the focus part of the sentence is primarily given by the type of the complementation of the verb. A hypothesis called systemic ordering of the elements in the focus of the sentence was formulated and empirically tested pairwise (i.e. successively for two of the complementation types) and supported also by several psycholinguistic experiments. Though the hypothesis was based on the examination of hundreds of examples, the material of the PDT offers a far richer material. The statistical findings support the following assumptions: (a) the sentential character of a complementation is a very important factor in that there is a tendency of a contextually non-bound element expressed by a clause to follow the non-sentential element, (b) the influence of the form of the complementation: e.g. the assumed order Manner – Patient is more frequent if the complementation of Manner is expressed by an adverb and the complementation of Patient by a nominal group; also the outer form of the Actor plays an important role: if the Actor is expressed by infinitive, Patient precedes Actor, while the hypothesized order Actor – Patient is attested if both complementations are expressed by nominal groups; (c) with some pairs, such as Patient and Means, there was a balance between the frequency of the two possible orders, which may indicate that for some particular complementations more than a single complementation occupy one position on the scale ([10]).

In some cases the decisions of the annotators are not the only possible ones and this fact has to be taken into consideration when drawing conclusions. This observation is confirmed also by the data on annotators' agreement/disagreement, see also [12].

1.3 Annotation of discourse relations

The discourse annotation in PDT 3.0 was based on a narrowly specified category of language expressions commonly known as connectives. However, it soon has become clear that such an annotation would miss some important discourse relations that are expressed by other means. The importance of this broader view is supported by the comparison of the number of relations expressed by connectives and those expressed by some alternative way (called AltLexes):

	all	intra-sentential	inter-sentential
AltLex:	726	272 (2.1%)	454 (7.7%)
connective:	17,983	12,523 (97.9%)	5,460 (92.3%)
total:	18,709	12,795 (100%)	5,914 (100%)

The numbers indicate that AltLexes express mostly inter-sentential discourse relations. Among them, they form almost 8% of all explicitly expressed relations, which makes them an indispensable part of the analysis of discourse (see [11]).

The largest proportion of occurrences within a single (complex) sentence is documented for the relations of purpose (100%), condition (99%), and disjunctive alternative (95%). These relations only rarely occur between two independent sentences (0, 1, 5%, respectively). On the basis of these observations, a preliminary hypothesis can be formulated that the semantic content expressed by the arguments of the above relations are more closely bound together than with the other relations. Also the relatively high position of conjunction (81%) is surprising as one would expect a more balanced distribution, perhaps similar to that found with opposition (43%).

The measuring of the ratio between the number of sentences and the number of discourse relations in individual genres has led to the observation ([8]) that in the PDT journalistic data, explicit connectives are most frequently used in genres with a high degree of subjectivity, i.e. where opinions, desires, evaluations, beliefs etc. are expressed. With the exception of sport, the first eight positions are represented by genres in which a certain degree of subjectivity often plays an important role, while the "objective" genres gathered consistently lower in the connective frequency scale. On the other hand, program or captions are typical in containing only a minimum of connectives since they are either very short (captions) or they are often represented by verbless phrases only (both genres).

2 Annotators' agreement

One of the interesting issues that can be observed when following the data on annotators' agreement as categorized according to the linguistic levels of description is the increasing number of disagreements if one proceeds from the POS or morphological level (which is the closest one to the outer linguistic form) to the level of underlying syntax and discourse.

Morphology: Agreement in PDT on choosing the correct morphological tag (5 thousand different tags): 97% ([3]). For German – in Negra (54 tags): 98.57% ([2]).

Surface syntax: No numbers for PDT; in Negra: (F-measure) for the unlabelled structural annotation: 92.43%, and for the labelled structural annotation (labelled nodes with 25 phrase types and labelled edges with 45 grammatical functions): 88.53% ([2]).

Deep syntax (tectogrammatics): In PDT, the agreement on establishing the correct dependency between pairs of nodes was 91%. The agreement on assigning the correct type to the dependency relation (67 possible values of the tectogrammatical functor) was 84% ([6]).

Topic-focus articulation: The agreement on assigning the correct value to individual nodes in the annotation of contextual boundness (i.e. the assignment of the values 'contextually bound' or 'contextually non-bound') was 82% ([12]).

Discourse phenomena: The agreement on the recognition of a discourse relation (connective-based F1-measure) was 83%. The agreement on the recognition

of a textual coreference or a bridging anaphora (chain-based F1-measure) was 72% and 46%, respectively. The agreement on the type of the relations in cases where the annotators recognized the same relation (a simple ratio) was 77% (Cohen's κ 71%) for discourse, 90% (Cohen's κ 73%) for textual coreference, and 92% (Cohen's κ 89%) for bridging anaphora ([9]). Sometimes even a small amount of annotated data can reveal important facts. In a small probe of annotating implicit discourse relations, the task proved to be highly challenging – the annotator's agreement on setting the type of implicit discourse relation between adjacent sentences was less than 60%.

The numbers of agreement for the different tasks cannot be directly compared (as they measure different phenomena, use different methods of evaluation and sometimes annotate different (type of) data), however, they seem to support the hypothesis that the deeper we go in the abstraction of the language description, the more difficult it is to achieve high values of the inter-annotator agreement. The above data also support the view (doubted by some linguists in the past) that it is easier to assign the structure (in other terms, the relation of dependency: the status of the governor and that of the dependent) than the value (type) of the dependency relations. This observation is also supported by the data on the Prague Czech-English Dependency Treebank (PCEDT) where the agreement on establishing the correct dependency between pairs of nodes was 88% while the agreement on assigning the correct type to the dependency relation was 85.5% ([7]).

3 Conclusion

We have collected some observations related to different layers of corpus annotation to demonstrate that even simple frequency data may give a linguist an important guidance for his/her deeper analysis of different linguistic phenomena. The prescribed length of the paper has allowed us just to summarize these observations; a more detailed statistics as well as analysis of the data can be found in the papers referred to.

Acknowledgements

We gratefully acknowledge support from the Grant Agency of the Czech Republic (project n. P406/12/0658). This work has been using language resources developed, stored and distributed by the LINDAT/CLARIN project of the Ministry of Education of the Czech Republic (project LM2010013).

References

[1] Bejček, E., Hajičová, E., Hajič, J. et al. (2013) *Prague Dependency Treebank 3.0*. Data/software, Charles University in Prague, Czech Republic.

[2] Brants, T. (2000) Inter-Annotator Agreement for a German Newspaper Corpus. In: *Proceedings of the Second LREC*, Athens, Greece.

[3] Hajič, J. (2005) Complex corpus annotation: The Prague dependency treebank. In *Insight into the Slovak and Czech Corpus Linguistics 2005*, 54.

[4] Hajičová, E., Havelka, J., Veselá, K. (2005) Corpus Evidence of Contextual Boundness and Focus. In: *Proceedings of the Corpus Linguistics Conference Series*, U. of Birmingham, pp. 1–9.

[5] Hajičová, E., Havelka, J., Sgall, P. et al. (2004) Issues of Projectivity in the Prague Dependency Treebank. In *The Prague Bulletin of Mathematical Linguistics, 81*, Charles University in Prague, pp. 5–22.

[6] Hajičová, E., Pajas, P., Veselá, K. (2002) Corpus Annotation on the Tectogrammatical Layer: Summarizing the First Stages of Evaluations. In *The Prague Bulletin of Mathematical Linguistics, 77*, Charles University in Prague, pp. 5–18.

[7] Mikulová, M., Štěpánek, J. (2010) Ways of Evaluation of the Annotators in Building the Prague Czech-English Dependency Treebank. In: *Proceedings of the 7th LREC*, Valletta, Malta, pp. 1836–1839.

[8] Poláková, L., Jínová, P., Mírovský, J. (2014) Genres in the Prague Discourse Treebank. In: *Proceedings of the 9th LREC*, Reykjavík, Iceland, pp. 1320–1326.

[9] Poláková, L., Mírovský, J., Nedoluzhko, A. et al. (2013) Introducing the Prague Discourse Treebank 1.0. In: *Proceedings of the 6th IJCNLP*, Nagoya, Japan, pp. 91–99.

[10] Rysová, K. (2013) *On Word Order from the Communicative Point of View*. PhD Thesis at Faculty of Arts, Charles University in Prague.

[11] Rysová, M. (2012) Alternative Lexicalizations of Discourse Connectives in Czech. In: *Proceedings of the 8th LREC*, İstanbul, Turkey, pp. 2800–2807.

[12] Veselá, K., Havelka, J., Hajičová, E. (2004) Annotators' Agreement: The Case of Topic-Focus Articulation. In: *Proceedings of the 4th LREC*, Lisboa, Portugal, pp. 2191–2194.

[13] Zeman, D. (2004) *Parsing with a Statistical Dependency Model*. PhD thesis, Univerzita Karlova v Praze, Praha.

[14] Zikánová, Š., Týnovský, M., Havelka, J. (2007) Identification of Topic and Focus in Czech: Evaluation of Manual Parallel Annotations. In: *The Prague Bulletin of Mathematical Linguistics, 87*, Charles University in Prague, pp. 61–70.

Estonian Dependency Treebank and its annotation scheme

Kadri Muischnek[1], Kaili Müürisep[1], Tiina Puolakainen[1], Eleri Aedmaa[2], Riin Kirt[1], and Dage Särg[2]

[1]Institute of Computer Science
[2]Institute of Estonian and General Linguistics
University of Tartu
E-mail: {kadri.muischnek, kaili.muurisep, tiina.puolakainen}@ut.ee

Abstract

In this article, we present Estonian Dependency Treebank, an ongoing corpus annotation project. The size of the treebank, once finished, will be ca 400,000 words. The treebank annotation consists of three layers: morphology, syntactic functions and dependency relations. For each layer, an overview of the labels and the annotation scheme is given.

As for the actual treebank creation, each text is annotated by two independent annotators, plus a super-annotator, whose task is to solve the discrepancies. The article also gives a short overview of the most frequent sources of dissensions between the annotators.

1 Introduction

The Estonian Dependency Treebank (EDT) is an ongoing annotation project which aims at creating a 400,000-word corpus annotated for dependency syntactic structures by the end of the year 2014. By the end of November 2014, we had completed the annotation of all texts, and were working on comparing annotated versions and solving discrepancies.

In the past, few attempts have been made to create an Estonian treebank. The first syntactic analyser of Estonian, a surface-syntactic Constraint Grammar Parser, was finished by 2001 [6]. Simultaneously, a corpus annotated in this framework was created. In order to convert this corpus into a treebank, phrase structure rules were applied and the output was checked manually [2]; this work resulted in a smallish treebank (Arborest).

These experiments showed clearly that although a phrase structure grammar suits well for representing an Estonian noun phrase, Estonian as a typical non-configurational language has no proper verb phrase and components of a multiword

verb may be separated from each other by intervening constituents. The word order (constituent order) is determined mainly by information structure; the main word order rule being that the finite verb occupies the second position in the clause. The word order inside a noun phrase, on the other hand, is fixed. It is generally believed that dependency representation is more suitable for free word order languages [8].

While creating the EDT, every text in the treebank is labelled by two annotators, and a super-annotator compares the versions and solves the discrepancies. No special software is used for annotation, but we have scripts for converting the treebank into CoNLL data format, that enables to use MaltParser tools for detecting formal errors in annotation, e.g. cycles, missing or redundant root-node [9, 1]. In addition, we can thus use MaltEval visualization tool [7].

2 Annotation

The annotation has separate layers for morphology, surface syntax and dependency relations. In the following subsections, we will provide a more detailed discussion of these layers.

2.1 Morphological tagset and syntactic labels

The morphological annotation layer contains information about lemma, part of speech and grammatical categories (e.g. case and number for nominals; mood, tense, person and number for verbs) for every word-form in the text[1]. EDT morphological annotation scheme is somewhat different from Universal Dependencies Scheme[2] (UDS): EDT lacks POS tag for determiners (substituted by pronouns) and also some universal features like gender, animacy, aspect, definiteness or state.

Surface-syntactic layer contains the syntactic function labels. According to our annotation scheme, the members of the verbal chain can be finite or infinite main verbs (FMV, IMV), and finite or infinite auxiliaries (FCV, ICV). Also, we distinguish particles as parts of particle verb (VPart), and verb negators (NEG). The arguments of the verb are labelled as subject (SUBJ), object (OBJ), predicative (PRD) or adverbial (ADVL); the adjuncts also get the adverbial label. In addition, the attributes of a nominal are tagged according to their part-of-speech (AN, NN, KN, PN, DN etc). We distinguish the nouns governed by an adposition with a special label (<P or P>) and also nouns governed by a quantor (<Q or Q>). In contrast to the UDS, we analyse adpositions and quantors as heads and the heads of their nominal dependants get special surface-syntactic tag. There is a special symbol for indicating whether the word form is a pre- or postmodifier (<NN or NN> for example). Also, we label conjunctions (J) and interjections (I).

[1] A table containing all the morphological tags can be found here: http://www.cl.ut.ee/korpused/morfliides/seletus.php?lang=en

[2] http://universaldependencies.github.io/docs/

The main shortcoming of our annotation scheme is that we do not distinguish between adverbial modifiers and adverbial complements. The syntactic layer is shallow, meaning that no virtual nodes are postulated. EDT does not have equivalents for various other core relations described in UDS, for example indirect object (iobj), relations indicating passive (nsubjpass, auxpass) or clausal counterparts (csubj, ccomp) and also loose joining relations as the list, remnant and different kinds of clausal modifier tags. In order to express these relations, more general tags are used in EDT.

Dependency layer gives information about the governor of every word form in the text.

```
"<Hoolimata>"                    % despite
   "hoolimata" L0 K pre cap <el> @ADVL #1->11
"<köhast>"                       % cough
   "köha" Lst S com sg el @<P  #2->1
"<ja>"                            % and
   "ja" L0 J crd @J #3->5
"<kõrgest>"                      % high
   "kõrge" Lst A pos sg el @AN> #4->5
"<palavikust>"                   % fever
   "palavik" Lst S com sg el @<P #5->2
"<ei>"                           % not
   "ei" L0 V aux neg @NEG #6->7
"<saanud>"                       % could
   "saa" Lnud V mod indic impf ps neg <NGP-P> @FCV #7->11
"<ta>"                           % he
   "tema" L0 P pers ps3 sg nom @SUBJ #8->11
"<ettekannet>"                   % presentation
   "ette_kanne" Lt S com sg part @OBJ #9->11
"<ära>"                          % particle
   "ära" L0 D @Vpart #10->11
"<jätta>"                        % cancel
   "jät" La V main inf <NGP-P> <PhVerb> <ära> @IMV #11->0
"<.>"
   "." Z Fst @Z #12->11
```

Figure 1: Sample annotated sentence “In spite of cough and high fever he could not cancel the presentation”.

An example in Fig. 1 demonstrates the use of tags in EDT format. The word forms are in separate rows following their morphological and syntactic description. The description consists of the lemma, ending, POS, morphological information, valency information (between angle brackets), syntactic label (starting with @) and dependency information (starting with #). The first word form *Hoolimata* is

adposition (K), preposition (pre), starting with capital letter (cap), its dependants should be in elative case (<el>), it is functioning as an adverbial and depends on the word form in the position 11 (#1->11). The second word form *köhast* is substantive (S), singular (sg), elative (el), it belongs to the preposition phrase (@<P) as a head of a noun phrase and it depends on the word form in the position 1 (#2->1).

2.2 Annotation scheme

In general, our annotation scheme is quite coarse for annotating intra-clausal phenomena, and comparable to the Stanford annotation scheme [3]. It should also be kept in mind that a lot of information that the Stanford tagset presents explicitly in the form of syntactic labels, we present as a combination of morphological and syntactic labels. For example, we do not distinguish between coordinating and subordinating conjunctions on the level of syntactic labels, but this information is presented at the morphological level with two different POS-labels: *J crd* and *J sub*. But while annotating the dependency relations that hold between the clauses, we neither distinguish clausal subjects, clausal complements, nor clausal modifiers, and we only state that there is a dependency relation between the clauses.

As McDonald et al [4] have pointed out, common divergences among dependency treebanks are found in the analysis of coordination, verbal chains, subordinate clauses and multiword expressions.

We annotate all coordinated sentence elements using the same syntactic function label. As for dependencies, we annotate each following coordinated element as a dependant of the previous one, and the coordinating conjunction as the dependant of the coordinated element following the conjunction.

While annotating the verb group consisting of a finite auxiliary verb and an infinite lexical verb, there are two possible solutions. Firstly, one can handle the finite auxiliary verb form as the governor of the verb phrase. The other possibility is to treat the infinite lexical verb form as the governor. In our work, we have chosen the second option, since the lexical verb determines the presence and coding of the arguments in the clause. This is consistent with the principle of primacy of content words in UDS.

We connect subordinate clauses to the main clause by attaching the governing verb of the subordinate clause to the governing verb of the main clause. As for coordinated clauses, we follow the overall principles for annotating coordination and annotate them in the same way. We do not actually distinguish between subordination and coordination at the clause level, apart from one exception: the relative clauses are governed by the noun they are modifying.

In EDT, we have treated multiword names as head-final noun phrases that consist of nouns having the part-of-speech tag of a proper noun. So, again, in our annotation scheme the information is spread between the syntactic and morphology layers. As for other types of multiword expressions, we are currently recognizing only particle verbs, a frequent phenomenon in Estonian.

2.3 Description of the annotation process

The texts of the Estonian Dependency Treebank form a subset of the Estonian Reference Corpus[3] and belong to the text classes of fiction, newspapers and science.

First, selected texts undergo morphological analysis and disambiguation, and are then parsed with the latest version of the Estonian CG dependency parser. The output is provided to two independent human annotators who manually check and disambiguate the parser output. After that, the texts are further compared and discrepancies resolved by a third human expert (superannotator). In practice, the latter task includes creating a linguistic description for several syntactic constructions of Estonian that have been unnoticed or labelled as "peripheral" in earlier descriptive grammars of Estonian.

As for the interannotator agreement, the disagreement rate (punctuation excluded) is 2.49% for morphological description, 6.79% for syntactic description and 8.56% for dependency relations.

While checking for the correctness of morphological description, annotators most often disagree over the lemma form of an unknown proper name. For example, if the genitive word form *Honti* occurs only once in the text, it is difficult to decide whether its lemma is *Hont* or *Honti*. Somewhat surprisingly, annotators also tend to disagree over the case label of a homonymous word form having several possible case readings, especially those of nominative, genitive and partitive cases. Most such ambiguous word forms are parts of idiomatic multiword expressions, but also, there are cases where even a mothertongue speaker with linguistic training is not able to choose between genitive and partitive reading of a word form. For instance, in example (1) word form *arvu* 'number' remains ambiguous.

(1) Karude arvu saab välja selgitada DNA näidete järgi
bear number-GEN/PART can out find DNA samples by
'The number of bears can be identified by samples of DNA.'

As for common sources of disagreement over syntactic functions, one can name quantors as governors of quantor phrases that can be ambiguous between object or predicative vs degree adverbial readings, e.g. (2).

(2) Jüri maksis talle selle eest 300 (OBJ vs ADVL) eurot
Jüri paid him that for 300 euros
'Jüri paid him 300 euros for that.'

The word order of Estonian is free, the most important rule being that the finite verb occupies the second position in the clause (V2), and information structure is one of the most important factors deciding the word order. Due to these factors, while annotating a copular clause, the annotators often disagree which noun should be labelled as subject and which one as predicative, e.g. (3).

[3] http://www.cl.ut.ee/korpused/segakorpus/index.php?lang=en

(3) Kõige olulisem ülesanne on haiglavõrgu korrastamine.
Most important task is hospital network arrangement
'Most important task is an arrangement of the network of hospitals.'

Common sources of disagreement on dependency relations are particles (e.g. *ikka* 'still, again', *veel* 'yet, again', *just* 'just', *muidugi* 'of course' etc) that can function both as sentence and phrase adverbials. Deciding on their proper governor often depends on semantics or even on the information structure of the sentence and that of the neighbouring sentences.

Also, in some cases it is difficult to decide upon the exact governor of a modifier in a long noun phrase, e.g. in example (4) it is hard to decide whether the participle *kujunenud* 'evolved' modifies the word form *liitude* 'unions' or *süsteem* 'system', did the unions or the system evolve during the last decades.

(4) Vana sajandi viimastel aastakümnetel kujunenud liitude süsteem ei
Old century last decades evolved unions system not
olnud veel lõplik.
was yet final
'The system of unions that evolved during the last decades of the past century was not final yet.'

3 Conclusions and further developments

The Estonian Treebank has been a long sought resource, and although it has been implemented during the last two years, many preliminary ideas for its creation existed before. The construction of a treebank has instigated many discussions, and a number of disputable issues have surfaced about the structure of Estonian and its representation in dependency format. We have tried to keep the annotation in such a format that it could be semi-automatically converted, if needed.

We have already used the beta version of our treebank for parser development. It has been used for improving an existing rule-based parser, training MaltParser and experimenting with various ways to combine those two [5].

After completing the first version of EDT, we plan to continue our efforts to harmonize and elaborate its annotation. Our immediate goals include re-labelling dependency links between subclauses and introducing distinct labels for sentence and phrasal adverbials.

Acknowledgements

This work was supported by Estonian Ministry of Education and Research (grant IUT20-56 "Eesti keele arvutimudelid / Computational models for Estonian“) and the European Regional Development Fund through the Estonian Centre of Excellence in Computer Science (EXCS).

References

[1] M. Ballesteros, J. Nivre. MaltOptimizer: An Optimization Tool for MaltParser. In *Proc. of the System Demonstration Session of the EACL 2012*. Avignon, France, 23-27 April 2012.

[2] E. Bick, H. Uibo, K. Müürisep. Arborest - a VISL-Style Treebank Derived from Estonian Constraint Grammar Corpus. In *Proc. of the Third Workshop on Treebanks and Linguistic Theories (TLT 2004)*. Tübingen, Germany, Dec 10-11, 2004.

[3] M. de Marneffe, C. D. Manning. The Stanford typed dependencies representation. In *COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation*. 2008.

[4] R. T. Mcdonald, J. Nivre, Y. Quirmbach-Brundage, Y. Goldberg, D. Das, K. Ganchev, K. B. Hall, S. Petrov, H. Zhang, O. Täckström, C. Bedini, N. Bertomeu, Castelló, J. Lee. Universal dependency annotation for multilingual parsing. - *Proc. of ACL '13*, pp 92-97. 2013.

[5] K. Muischnek, K. Müürisep, T. Puolakainen. Dependency Parsing of Estonian: Statistical and Rule-based Approaches. In *Human Language Technologies - The Baltic Perspective*. Frontiers in Artificial Intelligence and Applications Vol 268. IOS Press, Amsterdam, 2014, pp. 111-118.

[6] K. Müürisep, T. Puolakainen, K. Muischnek, M. Koit, T. Roosmaa, H. Uibo. A New Language for Constraint Grammar: Estonian. In *Proc. of International Conference Recent Advances in Natural Language Processing*. Borovets, Bulgaria, 2003, pp. 304-310.

[7] J. Nilsson, J. Nivre. MaltEval: an Evaluation and Visualization Tool for Dependency Parsing. *LREC*, 2008

[8] J. Nivre. Dependency parsing. *Language and Linguistics Compass*, 4(3):138-152, 2010.

[9] J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, E. Marsi. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, Cambridge University Press, 2007, 13, 95-135

Developing a Corpus of Syntactically-Annotated Learner Language for English

Marwa Ragheb and Markus Dickinson

Department of Linguistics
Indiana University
E-mail: {mragheb,md7}@indiana.edu

Abstract

Syntactic annotation for learner language has received some attention in recent years. We review the SALLE annotation scheme for syntactically annotating learner English, the main effort we are aware of that thoroughly investigates the linguistic categories for annotating syntax, presenting an overview of its development. Specifically, we focus on what is entailed in designing and implementing such a scheme with respect to: 1) interpretation, 2) defining the syntactic dependency layers, and 3) handling challenging cases.

1 Introduction

Syntactic annotation of data for second language learners is in its infancy, with only a handful of projects considering it and most focusing on improving automatic analysis [1, 6, 10, 14, 19, 20]. We focus on the SALLE (Syntactically Annotating the Language of Learner English) project, the main effort we are aware of that investigates the linguistic categories for annotating syntax. The SALLE project had its first publication at TLT in 2009 [3], and we review what has transpired since then [15], pointing towards further syntactic analysis of learner data.

As learner data can diverge from canonical language use, the annotation scheme splits annotation into separate layers, one for each piece of linguistic evidence. This can be illustrated with part-of-speech (POS) annotation [2]: in the phrase *for almost every **jobs** nowadays* [2], the word *jobs* is distributionally in a singular noun slot, but has the English plural marker. SALLE thus annotates two different POS.

This under-committal to an analysis is argued to be appropriate for second language research [16] and is applicable for canonical or non-canonical constructions, but there is a question of the degree of interpretation needed to annotate learner data [21]. In this short paper, we sketch: a) how interpretation is handled; b) how the syntactic dependency layers are defined; and c) examples revealing difficulty in annotating. The goal is not to present a large corpus or argue for particular analyses, but to outline some crucial decisions we found in designing and implementing a

syntactic annotation scheme for learner language. We can only make the reader aware of the major issues here; many more details are available in our previous publications; in a beta version of the annotation guidelines [5]; and in a recently-completed dissertation thesis [15]. A take-home point is an old one [cf., e.g., 22]: compared to the quirks of the text itself, the ease or difficulty of annotation lies as much, if not more, in the clarity and ease of the annotation scheme, in the quality of the guidelines, and in providing a decision procedure for "corner cases."

2 Interpretation

To discuss interpreting learner data, our perspective needs to be established. In SALLE, the goal is to be able to annotate any level of learner from any native language (L1) for any type of text. This means that little is assumed about the learner or the context in which something was written [compare to 13]. This leads to a second point: the annotation tries to avoid intended meaning, a point which fits with the goal of annotating categories in the learner data that benefit second language research [16]. Indeed, the annotation is not focused on errors or target hypotheses [12]—although, mismatches arising from different linguistic layers may point towards non-canonical structures (see section 3).

Additionally, while dependencies are often used to index meaning, the goal for SALLE is to annotate as much about the syntax as possible; when the syntactic form does not correspond to a likely semantic interpretation, one nonetheless annotates the apparent syntactic properties (more below). Finally, the project subscribes to the notion that all annotation is interpretation [11], and a thorough set of guidelines are used to adjudicate complicated cases, e.g., whether to "attach high" (sec. 4.1 of [5]), how to handle sentences lacking a copula (sec. 5.1.1 of [5]), etc.

The principles laid out in the annotator guidelines (of which we focus on the first two) shed light on how the different goals play out. The first principle is 'Give the learner the benefit of the doubt' and the second is to 'Assume as little as possible about the intended meaning of the learner' (p. 3 of [5]).

Benefit of the Doubt Giving the benefit of the doubt means assuming that the text is more well-formed than not. More specifically, this means trying to fit the sentence into the context, if possible, and, if not possible, to annotate as if the sentence were as syntactically well-formed as it can be, possibly ignoring meaning.

Consider the sentence *Cause all over the nation use it .*, occurring after *English skill is important*. In this case, the intended meaning may be something along the lines of *(because) all over the nation people use it*, but to annotate this way requires positing a missing word (*people*). Aside from the context not providing enough evidence of one intended meaning or another, this analysis presumes an error. The fallback strategy is to annotate the sentence in a more well-formed way, staying relatively within the bounds of the context. We see in figure 1 that this means treating *all* as the subject (SUBJ). (In either analysis, *cause* is an un-

selected complementizer (CPZR)—i.e., the subcategorization frame of *use* is not <CPZR,SUBJ,OBJ>—an issue stemming from sentence segmentation.) Crucially, the analysis based on a preferred intended form is ignored.

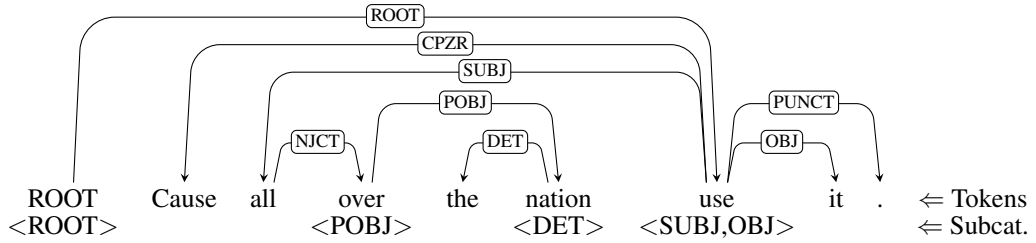


Figure 1: Giving the learner the benefit of the doubt

Semantics Eschewing (intended) semantic form was illustrated above, but to take a perhaps clearer example: regarding *then , I **study** law **dgree** .*, one does not usually study degrees. However, *study* takes a noun object, so *dgree* (cf. *degree*) serves as the object, and the only oddity is in requiring a determiner (see section 3 for subcategorization). The point is that, across a variety of contexts and learners, meaning cannot always be determined, whereas syntactic properties often can.

Being less concerned about intended meaning also means that we ignore some properties that may be deemed non-nativelike. If there is an unusual word choice or a pragmatically-odd construction, for example, where the sentence is syntactically well-formed, it will be annotated as if there are no problems (cf. the principle of minimal interaction [8]). In (1), for instance, the phrase *in each one of the spaces* may sound odd, but syntactically it is a valid prepositional phrase and is annotated as such. Thus, an annotator does not need to determine the source of the unacceptability of a sentence.

(1) In this moment of my life, I have differents goals in each one of the spaces.

3 Distinct but Intertwined Layers

The original intention of SALLE was to provide multiple syntactic dependency layers, corresponding to different kinds of evidence [4, 15, 17]: 1) subcategorization, 2) morphological dependencies, and 3) distributional dependencies. Subcategorization and morphologically-based dependencies, however, require some degree of context to define, and thus make distributional dependencies rather redundant [17]. That is, while the motivation for learner language is to keep the layers somewhat distinct, they cannot be kept totally distinct if annotation is to be practical (cf. also, [9]). We walk through the arguments from [17].

Subcategorization Although subcategorization is not often a part of syntactic annotation, it helps capture argument structure innovations [4]. Consider *we moved*

again to other house, where *house* requires a determiner. SALLE captures this by having *house* subcategorize for a determiner (<DET>), despite none being present.

For words with more than one possible subcategorization frame (e.g., plural nouns), all of them could be annotated, but this would not capture cases where a reading is prohibited. Thus, context is used to annotate only one frame, i.e., *distributional information* disambiguates subcategorization. In fact, subcategorization annotation often is preferred over distributional dependencies. Consider (2).

(2) I wondered what success **to be**.

Morphologically, *to be* has non-finite marking and the clause is thus a non-finite complement (XCOMP), as in the left side of figure 2. With a subcategorization for a finite complement (COMP), there is a mismatch. In a distributional tree, COMP is the label, but the subtree is unclear (see right side). If *to be* is in a finite distributional position, is *to* a finite auxiliary with a verbal complement? Is *be* a finite verb with an extraneous *to*? Subcategorization does not force an internal analysis.

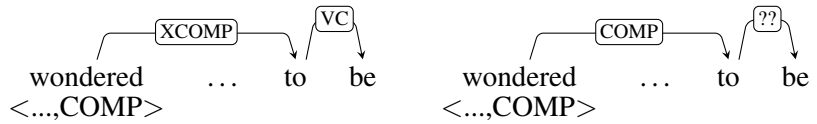


Figure 2: Morph. (left) and dist. (right) trees for a complement mismatch

Likewise, for missing arguments, annotating subcategorization information captures differences that (distributional) dependencies alone cannot capture.

Morphosyntax Consider annotating dependency trees based strictly on morphological forms. For (2) and depending on the annotation scheme, this would mean annotating based on *wondered* being either a past tense verb (VVD) or a past participle (VVN), *what* as a determiner (DDQ) or pronoun (also DDQ, but a different syntactic function), and *to* as a preposition (II) or a infinitive marker (TO). This would lead to 8 (=$2 \times 2 \times 2$) different trees, many of which are hard to define (e.g., *to* as II) or completely irrelevant for what was produced (e.g., *wondered* as VVN).

As with subcategorization, we annotate the closest fit to the context, where the "closest fit" generally leads to the most well-formed tree (section 2). For example, in *the step of my walked*, *walked* is marked as VVN, not VVD, as VVN is a tag for adjectival uses, and thus, in this nominal context, leads to a better tree. The ultimate decision in SALLE is to annotate morphosyntactic dependencies and subcategorization, but not dependencies rooted mainly in distributional evidence. Essentially, while other information is used (e.g., context), the annotation is primarily based on *form*. This allows the annotation to be applied to data from different kinds of learners and texts: one may use unspecified annotations at times (e.g., for extraneous words), but it is relatively clear which POS categories one starts with. Additionally, from the NLP perspective, there are some initial indications that basing the annotation on form could lead to better parsing results [see 19].

4 Some Challenging Cases

Consider (3), where the words *heart* and *accoss* present challenges in interpreting what words are being used. SALLE has a guideline of treating a word like its intended spelling when a misspelling is "phonetically or typographically-driven (but not semantically-driven)" [5, p. 20]. While there are debatable cases, the general idea is to balance the idea of giving the learner the benefit of the doubt—when it is clear to do so—and using the evidence (i.e., form) at hand.

(3) I can **heart** the sound of stream **accoss** the stone.

In this case, the fact that *accoss* is similar to *across* allows it to be treated like the preposition. As for *heart*, however, due to the fact that *heart* is an actual word, SALLE deems it too unclear as to whether it is a misspelling to treat it like *hear*. Thus, one winds up with a tree where the noun *heart* is an unspecified dependent of *can*, with an unspecified dependent *sound*, as in figure 3. The "minor" judgment of what constitutes an acceptable misspelling turns out to have big consequences [see 15, p.214 for more discussion of this example].

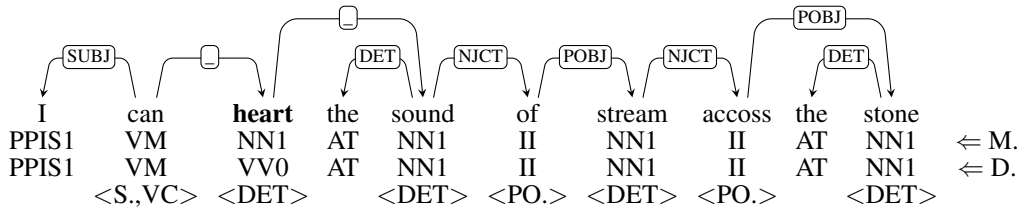


Figure 3: Example of a problematic lemma

In addition to revealing the impact of deciding on which word is present, the SALLE scheme has also provided some insight into how to treat coordination [4] and promises to provide other insights into dependencies for learner data. More to the point for this paper, the scheme allows one to provide informative annotation without guessing at the exact intention. For example, the meaning is unclear in figure 4. Nonetheless, giving the learner the benefit of the doubt, most of the syntactic properties can be determined, as shown. Aside from an unspecified (_) relation between *felt* and *me*, the tree is well-formed (see sec. 6.6 of [5] for more).

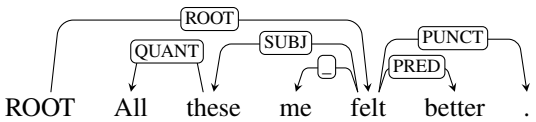


Figure 4: Unclear meaning/intention with syntactic annotation

5 Conclusion and Outlook

We have reviewed the SALLE annotation scheme for syntactically annotating learner English, an annotation effort that has aimed to thoroughly investigate the linguistic categories for annotating syntax. Specifically, we have focused on the design and implementation of the scheme with respect to: 1) interpretation, 2) definition of the syntactic dependency layers, and 3) the treatment of challenging cases. While many challenges remain for syntactic annotation, we have uncovered many issues facing linguistic annotation for second language data, such as: the importance of thoroughly defining a word, the need for separation between morphology and distribution and yet the interrelatedness of the two, and the lasting effect of what seem like simple heuristics onto all aspects of the annotation.

One major benefit of the SALLE scheme, as the first of its kind, stems from the fact that it is thoroughly documented; as we state on our website[1]: "The decisions we have made (certainly needing refinement in some cases) point out many of the essential questions that need to be addressed for linguistically annotating learner data, and we hope they can stimulate discussion." Outlining our decisions and the reasons for them should help pave the way for future work, where the decisions researchers make may be quite different. Given that annotation provides linguistic interpretation, a user of the annotation is able to understand what it means and what it does not mean. Indeed, an inter-annotator agreement study covering the different annotation layers reported fairly high inter-annotator agreement [18]. The difficulty of the text—whether stemming from complicated linguistic patterns or from innovative constructions—had an impact on agreement statistics, but the scheme has been successfully applied to learners of different levels and native languages.

Aside from continuing to apply the annotation to new and varied data, there are many routes to take this work: 1) automatically parse more data, determining how parsing can be improved [19]; 2) extract information relevant for second language acquisition (SLA) investigations; 3) thoroughly compare the positives and negatives of this scheme to more semantically-oriented annotation schemes [e.g., 6]; 4) continue to unpack specific linguistic constructions (e.g., coordination [4]), to see which aspects of the annotation are learner-specific or not; and 5) connect the work to other non-canonical data, such as historical texts and web data [7].

Acknowledgements

For fruitful discussion, we would like to thank the participants of previous conferences and workshops, the participants of the CL colloquium at IU, our student annotators, various people who have provided advice (e.g., Kathleen Bardovi-Harlig, Stuart Davis, Sandra Kübler, Detmar Meurers, Rex Sprouse, David Stringer), and the three anonymous reviewers here.

[1] http://cl.indiana.edu/~salle/

References

[1] Aoife Cahill, Binod Gyawali, and James Bruno. Self-training for parsing learner text. In *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*, pages 66–73, Dublin, Ireland, August 2014. Dublin City University.

[2] Ana Díaz-Negrillo, Detmar Meurers, Salvador Valera, and Holger Wunsch. Towards interlanguage POS annotation for effective learner corpora in SLA and FLT. *Language Forum*, 36(1–2):139–154, 2010. Special Issue on New Trends in Language Teaching.

[3] Markus Dickinson and Marwa Ragheb. Dependency annotation for learner corpora. In *Proceedings of the Eighth Workshop on Treebanks and Linguistic Theories (TLT-8)*, pages 59–70, Milan, Italy, 2009.

[4] Markus Dickinson and Marwa Ragheb. Dependency annotation of coordination for learner language. In *Proceedings of the International Conference on Dependency Linguistics (Depling 2011)*, pages 135–144, Barcelona, Spain, 2011.

[5] Markus Dickinson and Marwa Ragheb. Annotation for learner English guidelines, v. 0.1. Technical report, Indiana University, Bloomington, IN, June 2013. June 9, 2013.

[6] Jeroen Geertzen, Theodora Alexopoulou, and Anna Korhonen. Automatic linguistic annotation of large scale L2 databases: The EF-Cambridge open language database (EFCAMDAT). *Proceedings of the 31st Second Language Research Forum (SLRF)*, 2013.

[7] Yoav Goldberg, Yuval Marton, Ines Rehbein, Yannick Versley, Özlem Çetinoğlu, and Joel Tetreault, editors. *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*. Dublin City University, Dublin, Ireland, August 2014.

[8] Jirka Hana, Alexandr Rosen, Svatava Škodová, and Barbora Štindlová. Error-tagged learner corpus of Czech. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 11–19, Uppsala, Sweden, 2010. Association for Computational Linguistics.

[9] Leoš Hejl. Evolution of the conception of parts of speech. Diplomová práce, Univerzita Palackého v Olomouci, Filozofická fakulta, 2014.

[10] Hagen Hirschmann, Anke Lüdeling, Ines Rehbein, Marc Reznicek, and Amir Zeldes. Syntactic overuse and underuse: A study of a parsed learner corpus

and its target hypothesis. Talk given at the Ninth Workshop on Treebanks and Linguistic Theory, December 2010.

[11] Geoffrey Leech. Adding linguistic annotation. In Martin Wynne, editor, *Developing Linguistic Corpora: a Guide to Good Practice*, pages 17–29. Oxbow Books, Oxford, 2004.

[12] Anke Lüdeling, Maik Walter, Emil Kroymann, and Peter Adolphs. Multi-level error annotation in learner corpora. In *Proceedings of Corpus Linguistics 2005*, Birmingham, 2005.

[13] Detmar Meurers, Niels Ott, and Ramon Ziai. Compiling a task-based corpus for the analysis of learner language in context. In *Proceedings of Linguistic Evidence 2010*, pages 214–217, Tübingen, 2010.

[14] Niels Ott and Ramon Ziai. Evaluating dependency parsing performance on German learner language. In *Proceedings of TLT-9*, volume 9, pages 175–186, 2010.

[15] Marwa Ragheb. *Building a Syntactically-Annotated Corpus of Learner English*. PhD thesis, Indiana University, Bloomington, IN, August 2014.

[16] Marwa Ragheb and Markus Dickinson. Avoiding the comparative fallacy in the annotation of learner corpora. In *Selected Proceedings of the 2010 Second Language Research Forum: Reconsidering SLA Research, Dimensions, and Directions*, pages 114–124, Somerville, MA, 2011. Cascadilla Proceedings Project.

[17] Marwa Ragheb and Markus Dickinson. Defining syntax for learner language annotation. In *Proceedings of the 24th International Conference on Computational Linguistics (Coling 2012), Poster Session*, pages 965–974, Mumbai, India, 2012.

[18] Marwa Ragheb and Markus Dickinson. Inter-annotator agreement for dependency annotation of learner language. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 169–179, Atlanta, GA, 2013.

[19] Marwa Ragheb and Markus Dickinson. The effect of annotation scheme decisions on parsing learner data. In *Proceedings of the 13th International Workshop on Treebanks and Linguistic Theories (TLT13)*, Tübingen, Germany, 2014.

[20] Ines Rehbein, Hagen Hirschmann, Anke Lüdeling, and Marc Reznicek. Better tags give better trees - or do they? *Linguistic Issues in Language Technology (LiLT)*, 7(10), 2012.

[21] Victoria Rosén and Koenraad De Smedt. Syntactic annotation of learner corpora. In Hilde Johansen, Anne Golden, Jon Erik Hagen, and Ann-Kristin Helland, editors, *Systematisk, variert, men ikke tilfeldig. Antologi om norsk som andrespråk i anledning Kari Tenfjords 60-årsdag [Systematic, varied, but not arbitrary. Anthology about Norwegian as a second language on the occasion of Kari Tenfjord's 60th birthday]*, pages 120–132. Novus forlag, Oslo, 2010.

[22] Atro Voutilainen and Timo Järvinen. Specifying a shallow grammatical representation for parsing purposes. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics (EACL-95)*, pages 210–214, Dublin, Ireland, 1995.