Seq2Seq or Perceptrons for robust Lemmatization. An empirical examination.

Tobias Pütz, Daniël De Kok, Sebastian Pütz, Erhard Hinrichs

SFB833 A3, University of Tübingen

{tobias.puetz, daniel.de-kok, erhard.hinrichs}@uni-tuebingen.de

sebastian.puetz@student.uni-tuebingen.de,

ABSTRACT

We propose a morphologically-informed neural Sequence to Sequence (Seq2Seq) architecture for lemmatization. We evaluate the architecture on German and compare it to a log-linear state-of-the-art lemmatizer based on edit trees. We provide a type-based evaluation with an emphasis on robustness against noisy input and uncover irregularities in the training data. We find that our Seq2Seq variant achieves state-of-the-art performance and provide insight in advantages and disadvantages of the approach. Specifically, we find that the log-linear model has an advantage when dealing with misspelled words, whereas the Seq2Seq model generalizes better to unknown words.

KEYWORDS: Lemmatization, German, Error Analysis, Sequence2Sequence.

1 Introduction

Lemmatization is the process of mapping a form to its dictionary entry. Lemmas are a requirement to associate any inflected form with a lexical resource. In Natural Language Processing, lemmas are common features for a wide range of tasks and have been shown to improve results in parsing (Dozat and Manning, 2018) and machine translation (Sennrich and Haddow, 2016).

Besides being useful as features for statistical classifiers, lemmas are also of importance for other areas of linguistics. A common task in distributional semantics, for instance, is to algorithmically obtain a vector representing a certain word. A simple approach, which obtains discrete representations, computes the pointwise mutual information (PMI) between a word and its cooccurents. Word embeddings, in contrast to PMI, are continuous. They can be obtained through various methods, such as maximizing the similarity between word and context vectors (Mikolov et al., 2013). Both approaches are known to produce bad representations for rare words. This problem is especially relevant for morphologically-rich languages where the occurrences of rare words are divided between their possibly numerous different inflections. Building the word representations based on lemmas leads to less sparse representations, as the inflections of a word are seen as the same symbol, combining their occurrences. The sparsity issue also applies when querying a treebank of a morphologically-rich language. Here, a researcher might be interested in the usage of a certain word. Without the lemma as the common feature, every inflection of a word needs to be spelled out to obtain all usages. As a consequence, most treebanks contain lemmas as an annotation layer. As manual annotation is expensive and new methods require more data, we observe the rise of big web-corpora with automated annotation, which subsequently leads to a growing need for performant and robust lemmatization, fit for noisy web text.

In this work, we propose a morphologically-informed variant of the recently successful Sequence to Sequence (Seq2Seq) architecture (Sutskever et al., 2014) for lemmatization (*Oh-Morph*) and provide an in-depth comparison with the Lemming system (Müller et al., 2015) on two German treebanks: TüBa-D/Z (Telljohann et al., 2004) and NoSta-D (Dipper et al., 2013). In contrast to other recent work, we train and evaluate on types such that there is no overlap between training and test set. By type we denote unique combinations of form, lemma, POS and morphological tags. Table 1 specifies the input and expected output of the lemmatizer:

Input			**Output**
Form	*POS*	*Morphological tags*	*Lemma*
folgenden	NN	case:dative|number:singular|gender:neuter	folgendes
folgenden	ADJA	case:genitive|number:plural|gender:feminine	folgend

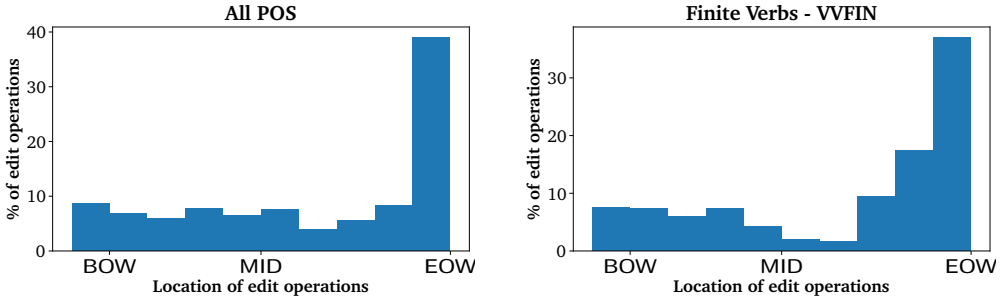
Table 1: Example of input and output of the lemmatizer.

The POS and morphological tags incorporate the necessary context information to lemmatize ambiguous forms. The type-based evaluation gives us the opportunity to perform a common model comparison but also to highlight problematic edge cases that would have been obscured under a token-based evaluation. To further our insight into the robustness of the models against noisy input, we use automatic morphological annotations instead of a gold standard both at training and test time. Moreover, we pay close attention to how the models deal with misspelled and unknown words. Given the usually well-formed newspaper texts in most treebanks, this is an aspect of the evaluation that is often overlooked.

We find that the log-linear *Lemming* outperforms *Oh-Morph* by a slight margin. *Lemming*, being able to leverage a word list, works best on lemmatizing non-standard language, like dialect

variants or misspelled words. It should be noted, though, that non-standard spelling is still one of the biggest error sources. *Oh-Morph*, in contrast, generalizes better to unknown words as we find *Lemming's* performance to deteriorate on out-of-vocabulary items. We conclude that both systems have their advantages and believe that further improvements can be made by improving their robustness against non-standard spelling of input words.

2 German Morphology



(a) macro-averaged edit operations

(b) finite verb edit operations

Figure 1: Locations of edit operations to transform forms to their lemma in TüBa-D/Z.

Inflectional patterns. As shown in Figure 1, inflections in German are largely limited to the suffix. However, certain verb forms deviate from this pattern and introduce prefixes as in *ge-schlossen* 'closed', the past participle of *schließen* 'close'. Another important exception are separable verbs, like *abschließen*. Such verbs have a prefix (e.g. *ab-* in *abschließen*) that appears separately when the verb is used as a finite verb in a declarative main clause. In such clauses, the verb is in the left bracket and the separable prefix appears in the right bracket, as demonstrated in (1).

(1) Als er erfuhr, dass er die Tür **abschließen** soll, **schloss** er sie **ab**.
When he heard, that he the door lock should, locked he it (prefix-ab).
'When he heard that the door should be locked, he locked it.'

Separable verbs have special infinitives and participles. They introduce the infix *-zu-*, a German infinitive marker, as in *ab-zu-schließen*, or *-ge-*, a participle marker, as in *ab-ge-schlossen*. Forms like *schließen*, where inflections change the parts of the root, are considered to be irregular. Irregular inflections are not limited to verbs but also occur, less frequently in other word classes such as adjectives, e.g. *gut* 'good', *besser* 'better' and *am besten* 'best'.

Challenges. A question that arises when dealing with separable verbs is whether their prefixes should be considered part of the lemma. Given the difference in meaning between e.g. *aufgeben* 'to give up' and *geben* 'to give' it would be very problematic to drop them. Keeping them, on the other hand, introduces the need to reattach separated prefixes which requires topological field or dependency annotations, steps usually performed after lemmatizing. A possible way out is to disregard the prefixes and reattach them once the required syntactical annotations have been made. This path, however, leads to the problem of deciding which of the possibly multiple prefixes is separable. Some can always be separated, some never, for others both separable and non-separable verb forms exist. In TüBa-D/Z (Telljohann et al., 2004),

lemmas mark separable prefixes with a '#' between prefix and stem, lemmatization of these forms then means to infix '#' for non-separated separables and to reattatch the prefix with the marker for separated ones. In the version used for this work, the prefixes have been removed in order to reach an homogeneity of annotation between both separated and non-separated separable verbs.

An additional challenge in the form of syncretism can be found in animate nouns. Some nouns with the masculine singular ending *-er* and the feminine *-in*, like *Schauspieler/-in* 'actor / actress' have no marked nominative plural for the masculine form. Others, like *Vorsitzenden* in (2), do not mark gender in nominative plural.

(2) Die Vorsitzenden trafen sich zum Krisengespräch.
The chair(wo)men met (refl) for-a crisis-meeting.
'The chairpersons met for a crisis meeting.'

Müller et al. (2015) mention that is important to know the lemma of these forms in order to assign a gender. We agree, however, in some cases the singular form can only be recognized if, possibly extra-sentential, discourse information is available. In other cases, e.g. if a plural word describes a mixed gendered group, the word cannot be reduced to a singular form since no un-gendered singular exists. As German grammar enforces gender, case and number congruency, syncretic forms are often disambiguated by accompanying determiners. While these prove to be useful in some cases, it should be noted that they display quite some ambiguity as well.

3 Background and Related Work

In the following section, we will first discuss existing lemmatization systems and then introduce methods relevant to our proposed model.

Previous work. Some early lemmatization systems employ finite state technology and solve morphological analysis and lemmatization as one task (Minnen et al., 2001; Schmid et al., 2004; Sennrich and Kunz, 2014). Given enough expert effort, these are able to achieve very good coverage. However, as their performance directly correlates with the completeness of their lexica, most transducers handle out-of-vocabulary items poorly. Moreover, as non-statistical tools they are not able to disambiguate syncretic forms. Others enrich the input with linguistic annotations and choose the correct transformation to the according lemma Chrupała (2006). Later a synergy between assigning these annotations and the lemma jointly was found (Chrupała et al., 2008; Müller et al., 2015). All these aforementioned approaches rely on sometimes language-specific sets of handcrafted features.

Lemming. The Lemming system (Müller et al., 2015), in the same vein as Chrupała (2006) and the Morfette system (Chrupała et al., 2008), treats lemmatization as a classification problem. During training, they derive edit-trees to transform a form into its lemma and then learn to choose the correct lemma from a set of candidates, generated by applying all possible edit-trees to a form. This simple approach achieves state-of-the-art performance on multiple languages. It greatly benefits from its ability to incorporate arbitrary global features, such as frequency counts or a word list. Müller et al. (2015) report improved performance by training Lemming and the Conditional Random Field (CRF) morphological tagger MarMot (Müller et al., 2013) jointly. However, their evaluation is bound to the token level, which we suspect to bias their evaluation towards frequent tokens, that we also expect to appear both in training and validation set.

Sequence to Sequence. The Sequence to Sequence (Seq2Seq) architecture (Sutskever et al., 2014) is a special variant of Recurrent Neural Networks (RNN). In contrast to regular RNNs, where the number of outputs is fixed, Seq2Seq enables mapping an arbitrary amount of inputs to an arbitrary number of outputs. These domain-agnostic models can be seen as feature-less and have achieved impressive results in several sequence transduction tasks, including machine translation (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2014), text summarization (See et al., 2017), constituency parsing (Vinyals et al., 2015), and closely related to lemmatization, morphological re-inflection (Cotterell et al., 2017, 2016). The common Seq2Seq architecture, also known as *encoder-decoder* network, consists of two RNNs. The encoder processes each input symbol in sequence while maintaining an internal state. The decoder is then initialized with the internal state of the encoder and predicts one symbol per step until a special end-of-sequence token is predicted. The input at every decoding step is the previously predicted token, with the first step receiving a special beginning-of-sequence token.

Attention. As the standard encoder-decoder architecture compresses its inputs into a fixed size vector, it struggles with long input sequences (Cho et al., 2014). A way to view this problem is that due to the longer input sequence, the encoder has to compress more information over a longer distance into the same dimensionality. Bahdanau et al. (2014) solve this issue by allowing the decoder to not only access the final state of the encoder but also each intermediate state, which they achieve through alignment mechanisms. Luong et al. (2015) simplified the alignment calculations by introducing the dot-product as scoring function for the attention mechanism. As both attention variants need to calculate the alignment weights for all encoder states at each decoder step, they have quadratic time complexity in $O(TU)$ where T and U are the lengths of the input and output sequence (Raffel et al., 2017). Raffel et al. (2017) reduce this to linear time with their monotonic Attention. It enforces linear alignments, where the decoder can only move forward in focusing on encoder states.

Seq2Seq lemmatization. Bergmanis and Goldwater (2018) applied the Seq2Seq architecture to lemmatization. They describe their approach as context sensitive, as the encoder processes not only the word form but also 20 characters of left and right context. In contrast to other systems, they do not require morphological or POS tags. However, as Müller et al. (2015), they evaluate on the token level. Schnober et al. (2016) compared pruned CRFs with Seq2Seq architectures and also evaluate on lemmatizing Finnish and German verbs taken from the Wiktionary Dataset (Durrett and DeNero, 2013). Besides limiting the task to verbs, they also lack a qualitative and exhaustive evaluation on the specific task of lemmatization.

4 Setup

In the following section, we will first describe the two variants of Lemming (Müller et al., 2015) that we used for comparison (*Lemming-Base*, *Lemming-List*), and then introduce our proposed model, *Ohnomore-Seq2Seq*.

4.1 Lemming

We use two variants of Lemming (Müller et al., 2015): *Lemming-Base* and *Lemming-List*. *Lemming-Base* utilizes its built-in features, including several alignment, edit-tree and lexical features. As Lemming supports the addition of arbitrary features, we also use *Lemming-List* in our experiments which adds a word list.[1] Both *Lemming-Base* and *Lemming-List* were trained using

[1]Available at `https://sourceforge.net/projects/germandict` accessed on 09.29.2018

the perceptron classifier with hashed features and morphological tags as additional features.

4.2 Ohnomore-Seq2Seq

Model. *Ohnomore-Seq2Seq* (*Oh-Morph*) (Pütz, 2018) closely resembles the classical encoder-decoder Seq2Seq architecture (Sutskever et al., 2014), extended with Luong-style monotonic Attention (Luong et al., 2015; Raffel et al., 2017). We dropped the reversing of the input, as we could not observe any differences in performance, likely due to attention which relaxes long-range dependencies. Furthermore, we concatenate the embedded morphological and POS tags with the final state of the encoder, resulting in a $d + p + (m * n)$ dimensional vector, where d is the state size of the encoder, p and m the size of morph- and POS- embeddings and n the maximal number of morphological tags encountered during training. This vector is then fed through a feed-forward layer with the SELU activation function (Klambauer et al., 2017), resulting in a vector with the dimensionality of the decoder's state size, which is the initial state of the decoder. Alternative setups, including bi-directional encoder, beam-search and a CRF-layer did not lead to improvements. It should be noted that the inclusion of a word list, as for *Lemming-List*, is not easily done, since *Oh-Morph* does not generate a candidate set ahead of time and therefore cannot include features of the lemma.

Hyperparameters. For training, we use mini-batches of 2,048 examples and discard forms longer than 60 characters. We use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.03 and clip gradients with a norm bigger than 5. The character embeddings have 100 dimensions, POS embeddings 50 and morph embeddings 30. We apply a dropout of 0.8 on the input embeddings. As recurrent cells in the encoder and decoder we use LSTMs (Hochreiter and Schmidhuber, 1997) with a recurrent dropout of 0.5. We train for 10,000 steps and then stop after 15 epochs without an improvement.

5 Evaluation and Data

Corpus	# Tokens	# Types
TüBa-D/Z	1.8M	213,705
NoSta-D	39,504	5,643
NoSta-D w/o. TüBa-D/Z	34,504	4,010

Table 2: TüBa-D/Z, NoSta-D and NoSta-D without the TüBa-D/Z sub-corpus with token and type count.

Evaluation. In contrast to other recent work in lemmatization like Müller et al. (2015) or Bergmanis and Goldwater (2018), we decided to evaluate on types instead of tokens. We did so because we suspect that token-based evaluation is biased towards getting frequent tokens right. Moreover, it is to be expected that a token which ends up both in the training and validation set will be predicted right, simplifying the task. We perform 10-fold cross validation on our in-domain data and average the accuracy of the 10 models on the out-of-domain data.

Data. Table 2 reports token and type counts on our data sets: TüBa-D/Z (Telljohann et al., 2004), a treebank containing articles of the German newspaper Taz, and as out-of-domain data NoSta-D (Dipper et al., 2013), a corpus containing non-standard variations of German. To account for a realistic setting with potentially erroneous morphological tags, we used the state-of-the-art CRF morphological tagger MarMot (Müller et al., 2013) to annotate TüBa-D/Z and NoSta-D using 5-fold jackknifing. After tagging we filter duplicates, such that every combination of

form, *lemma*, *POS* and *morphological tags* is unique. We further remove irregular forms and closed class words,[2] as we consider it as impossible to infer the lemma when using on type level disjoint sets. Moreover, we suspect that both irregular forms and closed class words can be easily lemmatized using dictionaries. We retrieved a list with 2,039 irregular forms from Celex German (Baayen et al., 1993). Since NoSta-D's lemma column is also used for normalization on sentence level e.g. insertions of elided tokens, we filter all tokens where an appended '|' marks a continuation or where an empty form is mapped to a lemma.

6 Results and Discussion

	TüBa-D/Z	**NoSta-D**	**NoSta-D w/o. TüBa-D/Z**
Oh-Morph	97.00%	83.69%	79.41%
Lemming-Base	96.78%	83.45%	79.00%
Lemming-List	**97.02%**	**83.96%**	**79.73%**

Table 3: Accuracy on TüBa-D/Z, NoSta-D and NoSta-D without the TüBa-D/Z sub-corpus. *Lemming-List* outperforms *Oh-Morph* by a slight margin on all sets. *Lemming-Base* consistently performs the worst. Best results are bold.

	Oh-Morph	**Lemming-Base**	**Lemming-List**	**Shared**
# total	6,078	6,078	6,078	207,627
# errors	3,088	3,565	3,051	3,326
% errors	50.80%	58.65%	50.20%	1.60%

Table 4: Unique and shared predictions on TüBa-D/Z with error rates. There are 207,627 types where all models had identical predictions and 6,078 where one had a unique prediction. The error rate within the identical predictions is only 1.6%. The unique predictions have consistent error rates of more than 50%.

	TüBa-D/Z	**Falko**	**BeMaTaC**	**Anselm**	**Unicum**	**Kafka**
Oh-Morph	94.22%	89.83%	78.86%	27.19%	**75.38%**	91.07%
Lemming-Base	94.32%	90.16%	78.80%	26.44%	74.07%	90.91%
Lemming-List	**94.37%**	**90.93%**	**79.44%**	**30.33%**	74.56%	**91.08%**

Table 5: Accuracy on the different NoSta-D sub-corpora. *Lemming-List* shows the best performance across all sub-corpora apart from Unicum (online chats), here *Oh-Morph* achieves the highest accuracy. TüBa-D/Z: news paper texts, Falko: L2 learner language, BeMaTac: spoken language, Anselm: historical text, Unicum: online chats, Kafka: literary prose. Best results are bold.

Results. Table 3 provides the results on TüBa-D/Z and NoSta-D, the results on each NoSta-D sub-corpus will be discussed in the following section. We observe that *Lemming-List* shows the overall best performance. *Oh-Morph* performs slightly worse on TüBa-D/Z and by a bigger margin on NoSta-D. *Lemming-Base* shows the lowest performance across all sets. Table 4 dissects the results on TüBa-D/Z into two sets, shared and unique predictions, and presents the error rates on the respective set. The shared set contains the 207,627 types for which all three models produced the same output. The unique set consists of the 6,078 types for which at least one of

[2]Available at `http://www.sfs.uni-tuebingen.de/Elwis/stts/Wortlisten/WortFormen.html` accessed on 09.29.2018

the models produced an unique prediction. There is an approximate 50-50 split between the 3,051 to 3,565 model-unique and the 3,326 shared errors. The large number of shared errors might hint at issues within the training data like tagging errors.

NoSta-D sub-corpora. NoSta-D consists of six diverse sub-corpora, ranging from online chats over learner language to historic texts. Table 5 reports the accuracy on each subcorporpus. We find that both Lemming models show a better performance on L2 learner language. *Oh-Morph* seems to operate well on online chats whereas, *Lemming-List* shows the best results with spoken language and by a clear margin on historic German text. The overall performance on the historic text is bad, most likely due to spelling variations that are not common anymore and very far from their canonical spelling, like the form *vrouwe* with the standard spelling *Frau* 'woman'.

6.1 Error Analysis

In the following section we will work out model specific strengths and provide some insight in anormalities in the training data.

6.1.1 TüBa-D/Z

Analysis. We sampled 22,007 types from TüBa-D/Z and classified the incorrect portions of the predictions of the three models into 7 error classes described in the following section. Types for which all models made the same predictions will be discussed separately. For 600 at least one model had a unique prediction. For 21,407 the predictions were identical.

Error classes. For our analysis we assign the following 7 error classes:

1. **Unsolvable:** cases that fail due to annotation errors within the lemma, like spelling mistakes; and cases where sentential information is needed, e.g. a truncated form that is part of an enumeration that receives a full lemma.
2. **Spelling:** misspelled forms which the lemmatizer did not correct.
3. **NE:** errors connected to named entities.
4. **Separable:** verbs with a separable prefix where the model retained the prefix or other verbs where a prefix was mistakenly cut off.
5. **Wr. morph** cases where the predicted lemma corresponds to wrong morphological tags, e.g. a plural noun which was tagged as nominative singular where the inflected form was returned as the lemma.
6. **Ign. morph:** cases where correct morphological tags have been ignored, e.g. a form was changed where the morphological tags imply that the form is the lemma.
7. **Solvable** all cases we consider solvable that are not captured by the other classes. For example, predictions where a superfluous character remains as a suffix.

Apart from assigning these fine-grained classes, we also group the errors by whether we consider them solvable or not. The unsolvable group is already described by the **Unsolvable** class. As borderline solvable we consider:

- **Spelling** and **NE**, as they might require world knowledge;
- **Separable**, since we believe that these are hard to pick up when training on types; and
- **Wr. morph**, as correct morphological tags are a requirement to lemmatize syncretic forms.

The solvable group counts the two members:

- **Ign. morph** as the necessary information is present; and

- the **Solvable** class.

	Correct	Solvable	Ign. morph	Wr. morph	Separable	NE	Spelling	Unsolvable
Oh-Morph	49.50%	19.17%	4.33%	5.83%	2.00%	7.67%	9.33%	2.17%
Lemming-Base	42.50%	21.33%	6.00%	8.33%	5.83%	5.67%	8.17%	2.16%
Lemming-List	50.33%	18.50%	6.17%	7.50%	5.17%	5.67%	3.50%	3.16%

Table 6: Result of the analysis of 600 of the non-identical sampled predictions from *Oh-Morph*, *Lemming-Base* and *Lemming-List*. Spelling is the biggest single error cause for both list-less models. *Lemming-List* shows the most issues with morphological tags. **Correct** corresponds to the error rates presented in Table 4. Abbreviations: *wr.*: wrong, *ign.*: ignored, *NE*: named entity.

Result. The results of our analysis are presented in Table 6. We see that *Lemming-Base* and *Lemming-List* show more problems connected to morphological tags. *Oh-Morph* might benefit from its capability to form a fine-grained character-based morphological representation in addition to the morphological tags, enabling a decision whether a form-tag combination is valid or not. Linear classifiers, like the perceptron used by both Lemming variants, in contrast, cannot capture these feature interactions. Further, we find that *Oh-Morph* outperforms both *Lemming-List* and *Lemming-Base* on separable verbs. With named entities *Oh-Morph* displays issues, we especially find problems with word final *-s*. The name of the sports brand *Adidas*, for example, got reduced to *Adida*, as the *-s* was recognized as a genitive marker. These errors seem to stem from an uncertainty whether a genitive ending in *-s* is syncretic with the nominative or inflectional. With spelling errors *Lemming-List* shows the least errors. Given the clear margin between it and *Lemming-Base* we believe that the word list provides crucial information whether a generated candidate lemma is well-formed or not.

Intersection. In 3,326 cases all three models made identical errors. The analysis of 340 errors is provided in Table 7. We find that the biggest cause of shared errors is an inability to correct spelling errors (37.65%). Further outstanding are erroneous morphological tags (14.41%) and errors connected to named entities (13.82%).

	Solvable	Ign. Morph	Wr. morph	Separable	NE	Spelling	Unsolvable
Intersection	18.53%	6.47%	14.41%	2.35%	13.82%	37.65%	6.76%

Table 7: Result of the analysis of 340 errors of the 21,407 sampled identical predictions. Spelling is the biggest single cause of errors.

Vocab	Type	Oh-Morph	Lemming-Base	Lemming-List	% unknown
Train	Form	**95.74%**	95.21%	95.62%	48.97%
	Lemma	**96.32%**	96.04%	95.98%	34.09%
List	Form	**94.34%**	94.27%	94.20%	28.34%
	Lemma	**96.48%**	96.47%	95.70%	28.98%

Table 8: Average share of unknown lemmas and forms per validation-fold of TüBa-D/Z with accuracies of *Oh-Morph*, *Lemming-Base* and *Lemming-List*. *Oh-Morph* performs the best on all out-of-vocabulary items. *Train* rows report the accuracy on forms and lemmas not contained in the training data, *List* on items not contained in the word list of *Lemming-List*. Best results are bold.

Out of vocabulary. Table 8 quantifies the performance of the models on unknown forms and

lemmas.[3] We inspect how the models deal with out-of-vocabulary items with respect to two vocabularies: *Lemming-List's* word list and the respective training fold. It should be noted that neither *Lemming-Base* nor *Oh-Morph* use the word list, we include their results for the sake of comparison. For all vocabularies *Oh-Morph* seems to be suited best to deal with unknown items. *Lemming-List* performs surprisingly poorly on unknown lemmas and shows worse results than *Lemming-Base* on these items. It seems that while *Lemming-List* is able utilize its word list to deal with spelling errors, it also relies on its completeness and shows a drop in performance on unknown entities.

	Oh-Morph	**Lemming-Base**	**Lemming-List**	**Shared**
# total	171	171	171	283
% errors	85.07%	72.51%	71.93%	68.90%

Table 9: Unique and shared errors on dialect and colloquial language. None of the models is suited to deal with dialect variants.

Colloquial language. In TüBa-D/Z words from dialects and colloquial language with non-standard spelling are mapped to their standard spelling with a trailing underscore. These form-lemma pairs are often only vaguely related, sometimes through phonetic similarities as in *verschandölln-verschandeln_* 'to vandalize' or *Frollein-Fräulein_* 'miss', in other cases like *koscht-kosten_* 'to cost' they are contractions. A problem when dealing with these cases is that the borders between lemmatization and text normalization with spelling errors and dialectical variants become blurry. Some dialect forms can easily be mistaken for spelling errors, with others context might be needed to retrieve the canonical form. In total there are 454 types where a lemma has a trailing underscore. The error rates on these types are reported in Table 9. We find that none of the models is suited to deal with these types as none manages to predict the right word in more than 30% of the cases. Moreover, both Lemming models infer the right lemma twice as often as *Oh-Morph*.

Ambiguity. During the annotation process, we noticed several *form-pos-morph* combinations that were associated with more than one lemma. Further examination revealed that there were in fact 921 cases in which the training data contains contradictory examples. The vast majority of these are nouns and named entities, accounting for 803 cases. Both Lemming models have an error rate of 72% on these cases, while *Oh-Morph* fails in 66% of the cases to give the expected lemma. Most of the ambiguous examples are nominalized verbs like *(der) Gehende* 'the walker' / *(ein) Gehender* 'a walker' where for definite and indefinite a separate nominative singular exists. According to the TüBa-D/Z annotation guidelines (Telljohann et al., 2006) these forms should be lemmatized to the indefinite nominative singular. As our data is machine tagged we searched gold-standard TüBa-D/Z for ambiguous combinations and find 738 cases that cannot be lemmatized using our featurization. It remains to be explored whether some of these are disambiguated by their context or if they should be considered inconsistent.

6.1.2 NoSta-D

Analysis. For a preliminary analysis, we sampled the predictions for 4,519 types from NoSta-D. For 4,207 the predictions of the three models were identical, for 312 at least one model produced a unique lemma. As before, we will first discuss the unique errors, then the identical ones.

[3] Forms and lemmas can occur both in the training and evaluation data, as our types are tuples of *form*, *lemma*, *POS* and *morphological tags*.

Classes. During the annotation process, we noticed that the lemmatization style in NoSta-D is different from the one in TüBa-D/Z. Comparatives and superlatives, for instance, are reduced to their positive, whereas in TüBa-D/Z the correct lemma is the nominative singular of the respective degree. To account for cases where the produced lemma is correct according to the TüBa-D/Z guidelines (Telljohann et al., 2006), we also assign the correct class to these tokens, hence **Correct** does not solely reflect the share of predictions that matched the lemma but also those that we consider correct. It should be noted that this only eliminates false negatives but not false positives where a lemma matches the gold-standard which would be considered wrong in TüBa-D/Z style. For cases where the correct lemma was not certain to us we assign the **Undecided** class, this happened mostly to nominalized verbs that can only be disambiguated within context. An analysis of these errors remains for the future. To account for the prevalence of colloquial language, dialect forms and historic forms, we also introduced the class **Non-standard** in our analysis of NoSta-D.

	Correct	Undecided	Solvable	Ign. morph	Wr. morph	Separable	NE	Spelling	Non-standard	Unsolvable
Oh-Morph	38.14%	2.24%	17.30%	0.64%	1.60%	2.88%	0.32%	7.05%	21.47%	8.33%
Lemming-Base	37.82%	1.92%	15.06%	0.64%	6.09%	3.20%	0.32%	7.37%	21.15%	6.41%
Lemming-List	47.43%	2.88%	11.53%	0.32%	5.77%	2.88%	0.32%	4.17%	18.58%	6.08%

Table 10: Preliminary result of the analysis of 312 of the non-identical sampled predictions of the three models on NoSta-D. The biggest error sources are non-standard language and spelling, *Lemming-List* has the least of these errors. Abbreviations: *wr.*: wrong, *ign.*: ignored, *NE*: named entity.

Preliminary results. The preliminary results of the analysis of the unique predictions of NoSta-D are presented in Table 10. These results mostly confirm the model-specific findings of the analysis on TüBa-D/Z. Most likely due to their prevalence in the corpus, we find that errors connected to colloquial language, dialect forms, and non-standard spelling are the most common ones for all three models. Again, we find that *Lemming-List* produces the least errors of these classes. Moreover, we find an indication, stronger than on TüBa-D/Z, that *Oh-Morph* suffers less from erroneous morphological tags, possibly due to tagging errors being more frequent on the noisy data. There are more unsolvable cases than on TüBa-D/Z, most of these are related to the normalization of nicknames in the BeMaTaC sub-corpus of NoSta-D which we consider to be solvable only on a sentence or even extra-sentential level.

Correct	Undecided	Solvable	Ign. morph	Wr. morph	Separable	NE	Spelling	Non-standard	Unsolvable
92.42%	1.16%	0.33%	0.02%	0.09%	0.05%	0.5%	1.38%	2.41%	1.62%

Table 11: Preliminary result of the analysis of 4,207 of the identical sampled predictions on NoSta-D. Non-standard language and spelling are the biggest error sources. Abbreviations: *wr.*: wrong, *ign.*: ignored, *NE*: named entity.

Intersection. The analysis of the sampled unique predictions on NoSta-D is presented in Table 11. Since we also assigned the **Correct** class, we include it in the table. The first thing to notice is that unsolvable errors make up the second largest class. This is mostly explained by the aforementioned normalization of chat nicknames. Within the intersection we find that most errors are connected to non-standard variations like the historic forms within the Anselm sub-corpus. This confirms our finding on colloquial language in TüBa-D/Z. Further notable is again the big share of spelling related errors. We believe that the amount of errors related to the surface form of the words, e.g. non-standard spelling or spelling mistakes, overshadows the amount of other errors that could have been produced if the normalization step would not have failed already.

Out of vocabulary While we did not find a distinctive effect for out-of-vocabulary items as in TüBa-D/Z, we discover a known issue of Seq2Seq architectures, namely unseen input and output symbols. *Oh-Morph* is not equipped to deal with unknown characters and just dropped them in most cases. Lemming on the other hand, will only fail if the unknown character is part of an inflection, another advantage of its edit-trees. The issue for the Seq2Seq model might be tackled by introducing a copy item which replaces individual characters in parts where form and lemma align.

Inconsistencies. During our analysis of NoSta-D, we found several lemmas that were in fact inflected forms. A search for the lemma *Nägel* 'nails', for example, brings up 7 hits in the BeMaTaC sub-corpus. Further, we find that verbal adjectives are in some cases reduced to the verb they are derived from and in others to the nominative singular of the adjective. A more thorough examination remains for future work.

7 Conclusion

In this work, we have proposed a morphologically-informed variant of the Seq2Seq architecture for lemmatization. We evaluated its effectiveness on German and provided a detailed error analysis with an emphasis on robustness and show strengths and weaknesses of the respective models. The results show that the Seq2Seq architecture achieves competitive performance. More precisely we found that *Oh-Morph* is less prone to suffer from wrong morphological tags which might lead to a better ability to incorporate them into its predictions. *Lemming-List*, on the other hand, seems to benefit from its word list, as it indicates whether a candidate is well-formed or not. Lemming's big advantage here is that it is a classifier over a candidate set rather than a generative model. Generating the potential lemmas ahead of time allows to incorporate features of the lemma, such as spelling or it being present in a word list. A Seq2Seq system, in contrast, cannot recover from false predictions which might be a reason for its tendency to transfer spelling errors from form to lemma. Turning to out-of-vocabulary items, we find that *Lemming-List's* advantage on malformed forms leads to the worst performance on unknown lemmas, whereas *Oh-Morph* shows the best performance with both unknown forms and lemmas.

Since both spelling errors and unknown tokens are to be expected when processing noisy web-corpora, we believe that good performance on noisy input and unknown tokens should not be a contradiction. For future work we plan to tackle the issue with spelling errors, as we saw that almost 40% of the shared errors were due to these cases. Possible approaches include incorporating a word list or more global optimization algorithms like Minimum Risk Training (Shen et al., 2016) or MIXER (Ranzato et al., 2015). Work in this direction should explore the intersection of lemmatization and text normalization, possibly in a joint training scenario. Given the influence of wrong morphological tags we are also confident that improving on morphological tagging will yield better results. Here it could be worthwhile to explore whether jointly assigning morphological tags and lemmas yields the same improvements as Müller et al. (2015) and Chrupała (2006) report. Another possibility that should be explored, pointed out by an anonymous reviewer, is the investigation of the effect of frequency by training on tokens.

As we have found that *Lemming-List* and *Oh-Morph* have somewhat complementary strengths, other future work should look into a possible ensemble consisting of an edit-tree classifier and a Seq2Seq model. A first naive approach could add the Seq2Seq lemmas to the candidate set of Lemming. A natural follow-up would then explore neural classifiers for edit scripts, with

a potentially simpler architecture than that of a fully fledged Seq2Seq model. The idea is compelling as it would allow to include arbitrary features, including lemma features, while keeping the flexibility of a character based encoder, with, in contrast to the log-linear Lemming, feature interactions that come with neural networks.

Acknowledgments

Financial support for the research reported in this paper was provided by the German Research Foundation (DFG) as part of the Collaborative Research Center "The Construction of Meaning" (SFB 833), project A3. Moreover, we would like to thank Patricia Fischer for her extensive and helpful comments on an early version of this paper.

References

Baayen, R. H., Piepenbrock, R., and van H, R. (1993). The CELEX lexical data base on CD-ROM.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bergmanis, T. and Goldwater, S. (2018). Context sensitive neural lemmatization with lematus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1391–1400.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. *Syntax, Semantics and Structure in Statistical Translation*, page 103.

Chrupała, G. (2006). Simple data-driven context-sensitive lemmatization. *Procesamiento del lenguaje natural, n^o 37 (sept. 2006), pp. 121-127*.

Chrupała, G., Dinu, G., and Van Genabith, J. (2008). Learning morphology with morfette.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqui, M., Kübler, S., Yarowsky, D., Eisner, J., et al. (2017). CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., and Hulden, M. (2016). The SIGMORPHON 2016 shared task—morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22.

Dipper, S., Lüdeling, A., and Reznicek, M. (2013). NoSta-D: A corpus of German non-standard varieties. *Non-Standard Data Sources in Corpus-Based Research*, (5):69–76.

Dozat, T. and Manning, C. D. (2018). Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490. Association for Computational Linguistics.

Durrett, G. and DeNero, J. (2013). Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *CoRR*, abs/1706.02515.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Minnen, G., Caroll, J., and Pearce, D. (2001). Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Müller, T., Cotterell, R., Fraser, A., and Schütze, H. (2015). Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274.

Müller, T., Schmid, H., and Schütze, H. (2013). Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332.

Pütz, T. (2018). Neural Sequence to Sequence Lemmatization. B.A. thesis, Eberhard Karls Universität Tübingen. `https://uni-tuebingen.de/en/34984`.

Raffel, C., Luong, M.-T., Liu, P. J., Weiss, R. J., and Eck, D. (2017). Online and linear-time attention by enforcing monotonic alignments. In *International Conference on Machine Learning*, pages 2837–2846.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2015). Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Schmid, H., Fitschen, A., and Heid, U. (2004). SMOR: A German computational morphology covering derivation, composition and inflection. In *LREC*, pages 1–263. Lisbon.

Schnober, C., Eger, S., Dinh, E.-L. D., and Gurevych, I. (2016). Still not there? comparing traditional sequence-to-sequence models to encoder-decoder neural networks on monotone string translation tasks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1703–1714. The COLING 2016 Organizing Committee.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1073–1083.

Sennrich, R. and Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, volume 1, pages 83–91.

Sennrich, R. and Kunz, B. (2014). Zmorge: A German morphological lexicon extracted from Wiktionary. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1683–1692.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Telljohann, H., Hinrichs, E., Kübler, S., and Kübler, R. (2004). The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004*. Citeseer.

Telljohann, H., Hinrichs, E. W., Kübler, S., Zinsmeister, H., and Beck, K. (2006). Stylebook for the Tübingen treebank of written German (TüBa-D/Z). In *Seminar fur Sprachwissenschaft, Universitat Tubingen, Tubingen, Germany*.

Vinyals, O., Kaiser, Ł., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2015). Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.