Evaluating Shortest Edit Script Methods for Contextual Lemmatization

Olia Toporkov, Rodrigo Agerri
HiTZ Center - Ixa, University of the Basque Country UPV/EHU
{olia.toporkov,rodrigo.agerri}@ehu.eus

Abstract

Modern contextual lemmatizers often rely on automatically induced Shortest Edit Scripts (SES), namely, the number of edit operations to transform a word form into its lemma. In fact, different methods of computing SES have been proposed as an integral component in the architecture of several state-of-the-art contextual lemmatizers currently available. However, previous work has not investigated the direct impact of SES in the final lemmatization performance. In this paper we address this issue by focusing on lemmatization as a token classification task where the only input that the model receives is the word-label pairs in context, where the labels correspond to previously induced SES. Thus, by modifying in our lemmatization system only the SES labels that the model needs to learn, we may then objectively conclude which SES representation produces the best lemmatization results. We experiment with seven languages of different morphological complexity, namely, English, Spanish, Basque, Russian, Czech, Turkish and Polish, using multilingual and language-specific pre-trained masked language encoder-only models as a backbone to build our lemmatizers. Comprehensive experimental results, both in- and out-of-domain, indicate that computing the casing and edit operations separately is beneficial overall, but much more clearly for languages with high-inflected morphology. Notably, multilingual pre-trained language models consistently outperform their language-specific counterparts in every evaluation setting.

Keywords: Contextual Lemmatization, Shortest Edit Script, Minimum Edit Distance, Deep Learning, Information Extraction

1. Introduction

Lemmatization is one of the most common basic Natural Language Processing (NLP) tasks, commonly understood as transforming an inflected wordform (e.g., *feeling, felt*) into its initial form known as lemma (e.g., *feel*), as defined by the contextual lemmatization SIGMORPHON 2019 shared task (Aiken et al., 2019).

Lemmatization remains important for morphologically rich languages as it usually plays a crucial role for information extraction systems, sentiment analysis and helps to deal with inflected named entities during named entity recognition task, especially for high-inflected languages.

Nowadays, state-of-the-art approaches to lemmatization are based on supervised contextual methods, a technique first proposed by Chrupala et al. (2008). Treating lemmatization as a supervised classification task relies on automatically inducing a set of patterns from textual corpora, encoding the minimum amount of edits needed to map the surface word to its lemma, namely, the Shortest Edit Script (SES). Ideally these SES would capture morphological patterns about word inflection making lemmatization feasible as a classification task. Thus, in Chrupala's approach, classifiers would learn previously induced SES which, at inference time, would be decoded back into their lemmas.

Modern contextual lemmatizers often rely on automatically induced Shortest Edit Scripts (SES) for optimal performance. In fact, different methods of computing SES have been proposed as an integral component in the architecture of several state-of-the-art contextual lemmatizers currently available (Malaviya et al., 2019; Straka et al., 2019; Yildiz and Tantuğ, 2019). However, previous work has not investigated the direct impact of SES in the final lemmatization performance. In order to address this issue, in this paper we compare three popular approaches to automatically induce SES (Straka et al., 2019; Yildiz and Tantuğ, 2019; Agerri et al., 2014; Agerri and Rigau, 2016) and empirically investigate which of them (if any) is the most beneficial.

In order to do so, we follow previous work by Toporkov and Agerri (2024) which demonstrates that language models can competitively perform contextual lemmatization without receiving any explicit morphological signal during training, using just the word form and its corresponding SES. This allows us to focus on lemmatization as a token classification task where the only input that the model receives is the word-label pairs in context, in other words, the labels corresponding to previously induced SES. Thus, by modifying in our lemmatization systems only the SES labels that the model needs to learn, we may then be able to objectively conclude which SES representation helps to produce the best lemmatization results.

For our experiments we pick seven languages

LREC-COLING 2024, pages 6451–6463
20-25 May, 2024.

of different morphological complexity, namely, English, Spanish, Basque, Russian, Czech, Turkish and Polish. Moreover, we use a number of multilingual and language-specific pre-trained masked language encoder-only models as a backbone to build our lemmatizers. To the best of our knowledge, this is the first systematic evaluation of the impact of the SES representations for contextual lemmatization.

Comprehensive experimental results, both in- and out-of-domain, indicate that computing the casing and edit operations separately, as proposed by UDPipe, is the best method to obtain SES overall, particularly for the languages with more complex morphology. Chrupala's approach as implemented by Agerri et al. (2014) performs as a close second, while the Morpheus method (Yildiz and Tantuğ, 2019) is the less optimal one. In addition, our results show that multilingual pre-trained language models consistently outperform their language-specific counterparts in every evaluation setting. This is consistent with previous research comparing monolingual and multilingual encoder-only models (Agerri and Agirre, 2023).

Furthermore, our experimental setting shows how to easily obtain competitive lemmatization results for the languages of our choice. Finally, we offer a word on contamination of language models, concluding that the results reported in this paper are not spuriously high due to model contamination.

Code, data and fine-tuned models are publicly available to facilitate further research on this topic and reproducibility of the results.[1]

2. Related Work

Attempts to resolve the lemmatization task started with systems based on dictionary lookup and/or finite set of rules (Karttunen et al., 1992; Oflazer, 1993; Alegria et al., 1996; Segalovich, 2003; Carreras et al., 2004; Stroppa and Yvon, 2005). These systems, apart from being language dependent, required a lot of effort, linguistic knowledge and manual intervention, especially for more complex languages with a high level of inflection. The creation of large annotated corpora, which included morpho-syntactic features and lemmas, led to the development of machine learning approaches to lemmatization in a variety of languages. Thus, initiatives such as the Universal Dependencies (Nivre et al., 2017) and the UniMorph project (McCarthy et al., 2020) allowed to gather annotated corpora in more than 118 languages, including low-resourced and endangered ones.

The hypothesis that context is beneficial in the case of unseen and ambiguous words incentivized the appearance of supervised contextual lemmatizers. One of the pioneer works in this field is the statistical contextual lemmatizer Morfette (Chrupala et al., 2008). It is based on a pipeline approach and uses a Maximum Entropy classifier to predict morphological tags and lemmas. Crucially, Chrupala et al. (2008) presents for the first time the idea of treating lemmatization as a classification task by predicting the Shortest Edit Script (SES), namely, the shortest sequence of instructions (insertions, deletions or replacements) needed to transform a reversed inflected word to its lemma. The work of Chrupala et al. (2008) inspired the development of many methods for contextual lemmatization, which most of the time included the idea of using minimum edit scripts. Among others, the IXA pipes system (Agerri et al., 2014; Agerri and Rigau, 2016) and Lemming (Müller et al., 2015) apply the same principle of edit trees, combining it with the possibility of adding external lexical information. Other examples of the systems that use the concept of SES are the works of Gesmundo and Samardžić (2012), Chakrabarty et al. (2017) and the system of Malaviya et al. (2019).

The development of supervised approaches involving deep learning algorithms and the appearance of the Transformer architecture (Vaswani et al., 2017) and Transformer-based masked language models (MLMs) such as BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) allowed to significantly improve the performance of supervised lemmatizers. Thus, in the SIGMORPHON 2019 shared task on contextual lemmatization (McCarthy et al., 2019) most of the participating systems were based on MLMs. The best overall system was UDPipe (Straka et al., 2019), which ensembled various pre-trained contextualized BERT and Flair embeddings as an additional input to a Bi-LSTM network. To perform lemmatization they classify the input words according to the set of generated lemma rules or SES. The third best model, Morpheus, proposed a two-level LSTM network (Yildiz and Tantuğ, 2019) which used vector-based representations of words, morphological tags and SES as input. The output of the system results in a corresponding morphological labels and SES representing the lemma class which is later decoded into its lemma form.

However, while many of these top performing systems included different methods to compute SES as an integral component in their lemmatization models, there has not been an attempt to compare and establish which of the existing methods is the optimal one for the task. In this paper we pick three of the most popular SES approaches (according to performance on the SIGMORPHON 2019 lemmatization benchmark) and make a systematic comparison with the aim of clarifying this issue. We

[1] https://github.com/hitz-zentroa/ses-lemma

believe that this could benefit the development of future lemmatizers which may include SES as an integral component of their systems.

3. Data

To train and evaluate our models we used the datasets developed for the SIGMORPHON 2019 shared task on contextual lemmatization (McCarthy et al., 2019). These datasets are annotated according to the Unimorph schema guidelines (McCarthy et al., 2020). For in-domain evaluation we chose one corpus per language with standard train and development partitions. Additionally, we also provide out-of-domain evaluation results, as this is the setting in which lemmatizers are usually deployed. As most languages are represented in the SIGMORPHON 2019 by more than one dataset, for out-of-domain evaluation we picked the test sets of datasets different from the ones selected for in-domain evaluation. The exception was Basque, for which we selected a dataset external to the UniMorph SIGMORPHON data.

With respect to in-domain, in the case of Spanish and Russian we used GSD data, which consists of Wikipedia and news articles, texts from blogs and reviews. As the lemmas of these two corpora were originally lower-cased and giving the fact that the methods of generating the Shortest Edit Scripts (SES) are case dependent, we performed a simple adjustment by changing the lemmas of the proper nouns to their upper-cased version. For the rest of the languages, there was no need of performing such adjustment, as the lemmas for the proper nouns in the corresponding corpora were correctly upper-cased by default.

For English we chose English Web Treebank (EWT) (Silveira et al., 2014) composed using different Web sources, such as several media, blog articles, reviews, e-mails and Yahoo! answers.

For Basque we used the Basque Dependency Treebank (BDT) (Aldezabal et al., 2008) made of literary and journalistic texts.

With respect to Turkish we used ITU-METU-Sabanci Treebank (IMST) (Türk et al., 2019), a corpus formed by sentences from a wide range of domains, such as non-fiction and news.

The Czech data correspond to the CAC corpus (Hladká et al., 2008), based on the Czech Academic Corpus 2.0, containing mostly full-length articles from different media sources, such as newspapers, magazines and transcripts of spoken language from radio and TV programs.

Finally, for Polish we chose the LFG corpus (Przepiórkowski and Patejuk, 2018), derived from a corpus of LFG (Lexical Functional Grammar) syntactic structures, and consisting mostly of sentences from fiction, news and non-fiction genres, as well as the texts from the Internet sources.

	train	dev	test	test(OOD)
es	345,545	42,545	43,497	54,449
ru	79,989	9,526	9,874	109,855
en	204,857	24,470	25,527	8,189
eu	97,336	12,206	11,901	299,206
tr	46,417	5,708	5,734	1,795
cz	395,043	50,087	49,253	1,930
pl	104,730	13,161	13,076	8,511

Table 1: Number of tokens in the train, development, in-domain (test) and out-of-domain (test(OOD)) test sets.

Out-of-domain evaluation was performed using the AnCora corpus (Taulé et al., 2008) in the case of Spanish, which consists mainly of the news texts. For Russian we chose SynTagRus (Lyashevkaya et al., 2016), a corpus that is formed using texts from popular science, news and journal articles and contemporary fiction. Regarding English we used the Georgetown University Multilayer (GUM) corpus (Zeldes, 2017), a collection of a wide range of Web texts from Wikipedia, Reddit and Wikinet. As for the Basque language, due to unavailability of the alternative corpora provided in the shared task data, we chose the Armiarma corpus (arm, 2000), which was created using literary reviews. To evaluate Turkish and Czech languages we used PUD data, a part of the Parallel Universal Dependencies treebanks created for the CoNLL 2017 shared task (Zeman et al., 2017). The corpora include sentences from such domains as Wikipedia and news, annotated in total for 18 languages. Table 1 provides details about the size of the datasets.

4. Methods to Induce Shortest Edit Scripts

The general idea of computing the Shortest Edit Script (SES) in contextual lemmatization is based on finding the minimum number of edit operations necessary to convert a surface word into its corresponding lemma. By edit operations we understand any change applied to the wordform, which consists in deleting, inserting or replacing letters in the surface word, as well as leaving the word unchanged in the case the inflected form and the lemma coincide (e.g. the→*the*, road→*road*). SES methods focus on codifying such minimum edits for their further application as a set of instructions to modify the surface word. In this paper we address three different approaches based on the Shortest Edit Scripts. The methods chosen are those implemented by the first and third best systems in the SIGMORPHON 2019 shared task, namely UDPipe (Straka et al., 2019) and Morpheus (Yildiz and

Tantuğ, 2019), and Chrupala's original proposal as implemented by the IXA pipes system (Agerri et al., 2014; Agerri and Rigau, 2016). [2]

4.1. UDPipe

The approach applied in the UDPipe system focuses on performing character level edits on the suffixes and prefixes of the word. They divide their script creation in two parts: (i) encoding the correct casing as a casing script and (ii) creating a sequence of character edits. Regarding the casing script, they consider both wordform and lemma as lower-cased. If the lemma contains upper-cased characters they add a rule to the casing script to uppercase such characters in the final lemma. The next step is creating a sequence of character edits by splitting the wordform into a prefix, a root (stem) and a suffix in order to process them separately. The root is obtained by finding the longest equal substring between the input word and its lemma and is kept unchanged. Then they process the prefixes and suffixes of the target word, including possible character operations such as *copy*, *add* or *delete*. The final script is produced by a concatenation of the casing and the edit scripts. The obtained SES are the complete rules which convert input words to their lemmas. When the word and lemma do not share any common parts, the word is considered irregular and is directly replaced by its lemma, skipping any possible edits.

4.2. Morpheus

Morpheus's approach is based on the prediction of minimum edits between a surface word and its lemma using four fundamental operations such as *same*, *delete*, *replace* and *insert*. *Same* and *delete* operations have only one version (the character may be left without changes (s) or deleted (d)), while *replace* and *insert* operations may vary, depending on the character they are tied to. As the minimum edit prediction decoder of Morpheus creates edit labels for each character in the word, it is only able to generate lemmas shorter or equal to the inflected forms. Still, in some languages lemmas may be longer that their corresponding wordforms. For such cases Yildiz and Tantuğ (2019) modify the standard Levenshtein distance algorithm by merging successive *insert* labels into one in the same position with multiple characters. They perform the same process for the *replace* label, combining it with the successive *insert* labels into one *replace* label and ensuring the correct lemma generation. They also consider the cases where the word is situated in the beginning of the sentence and should be lowercased, reflecting it in the Shortest Edit Script.

4.3. IXA pipes

The third method is based on the interpretation of Chrupala's technique (Chrupala et al., 2008) by Agerri et al. (2014). This approach addresses the suffixal nature of inflectional morphology where the end of the word is the most changing part and is more likely subject to modifications than the prefix or root. Chrupala et al. (2008) propose to compute the minimum edit distance between the *reversed* wordform and its lemma. They index word characters starting from the end of the string, allowing to form more coherent lemma classes and to perform lemmatization more efficiently. In the set of instructions that are generated using this technique the position of the letters that are subject to change are indicated along with the type of operation (insertion or deletion). In the adaptation of this approach it is also considered the casing of proper nouns, as well as the casing of the words that appear in the beginning of the sentence and should be lowercased for their correct lemmatization.

4.4. SES Comparison

In order to obtain a better understanding of the described methods and their core differences, we provide a brief comparison of the three minimum edit approaches, namely, UDPipe system's approach (*ses-udpipe*), Morpheus's approach (*ses-morpheus*) and IXA pipes approach (*ses-ixapipes*).

word→*lemma*	**UDP**	**IXA**	**Morph**
cats→cat	↓0;d¦-	D0s	s|s|s|d
birds→*bird*	↓0;d¦-	D0s	s|s|s|s|d
did→*do*	↓0;d¦−+o	R1ioD0d	s|r_o|d
Wolak→*Wolak*	↑0¦↓1;d¦	O	s|s|s|s|s
You→*you*	↓0;d¦	1	l|s|s

Table 2: Examples of the three types of SES patterns: UDP - ses-udpipe, IXA - ses-ixapipes, Morph - ses-morpheus.

Table 2 provides some examples of the Shortest Edit Scripts used in lemmatization for the aforementioned SES methods. For an action such as removing the last letter of the surface word (as in the case of the words 'cats' and 'birds') both *ses-udpipe* and *ses-ixapipes* apply the edit instruction to the reversed wordform, removing the last letter. Additionally, *ses-udpipe* method indicates that the word has to be lowercased. As for *ses-morpheus*, it processes each letter separately, leaving those that should remain untouched as 's' (same) and deleting the last one, marking such operation with

[2] It may be argued that the methods of Morpheus and UDPipe systems do not strictly always generate the *shortest* edit script (SES). However, we keep the SES term as it was originally coined by Chrupala et al. (2008) as a convenient acronym.

‘d’ (delete). Unlike *ses-udpipe* and *ses-ixapipes*, the scripts corresponding to the same action of deleting the last word’s letter generate two different label classes as the number of the letters in ‘cats’ and ‘birds’ is distinct.

The next lemmatization example (did→*do*), demonstrates how each of the SES approaches treats the cases where one or more letters should be inserted in order to obtain the lemma. Here *ses-ixapipes* and *ses-morpheus* methods apply similar order of minimum edits using delete and replace operations, while *ses-udpipe* first deletes the two ultimate letters of the word and only then makes the insertion of the letter ‘o’.

Finally, the last two examples are provided in order to reflect the edit scripts that are generated in the case of proper nouns in contrast to the ordinary nouns situated in the beginning of the sentence and, therefore, starting with the capital letter. We could see that for the proper noun ‘Wolak’ *ses-udpipe* indicates that the first letter should remain uppercased, whereas the scripts of *ses-ixapipes* and *ses-morpheus* simply leave the word unchanged. As for the pronoun ‘You’ situated in the beginning of the sentence, all three SES approaches lowercase it in order to obtain the correct lemma. It is important to mention that, as with the first two examples, in the case of the longer proper nouns the UDPipe’s and IXA pipes’ scripts would remain the same, while the script of the Morpheus’s approach would vary according to the number of letters in the surface word.

5. Systems

In our experiments we apply two multilingual and seven language-specific pre-trained masked language models (MLMs). With respect to multilingual models we use multilingual BERT (mBERT) (Devlin et al., 2019), a Transformer-based masked language model pre-trained on the Wikipedias of 104 languages. mBERT was pre-trained using both masking and next sentence prediction objectives, applying a batch size of 256 and 512 sequence length for 1M steps. The second multilingual model we apply is XLM-RoBERTa (Conneau et al., 2020), pre-trained on 2.5TB of filtered CommonCrawl data for 100 languages. This model is based on the RoBERTa architecture, was trained only on the MLM task, implies dynamic mask generation and was pre-trained over 1.5M steps with a batch of 8192 and sequences of 512 length. We used both base and large versions of XLM-RoBERTa.

Regarding the language-specific models, we choose one model for each of the target languages. For Spanish we use the cased version of BETO (Cañete et al., 2020). It is a BERT-base language model trained on a large Spanish corpus including all Spanish Wikipedia as well as the Spanish part of the OPUS project (Tiedemann, 2012) in a total size of around 3 billion words. BETO is an upgraded version of the initial BERT-base model with the application of the dynamic masking technique, introduced in RoBERTa. It was trained with the total of 2M steps in two stages: 900K steps with a batch size of 2048 and maximum sequence length of 128, and the rest of the steps using batch size of 256 and maximum sequence length of 512.

For the Czech language we apply slavicBERT (Arkhipov et al., 2019), developed by continuing the training of multilingual BERT on Russian news and the Wikipedias of Russian, Bulgarian, Czech and Polish. The vocabulary of subword tokens was also rebuilt with the use of the subword-nmt repository.[3]

For Russian we choose RuBERT (Kuratov and Arkhipov, 2019), which was developed similarly to slavicBERT, with the difference of having only Russian as the target language. The system was trained using the Russian Wikipedia and news. The authors obtain a new subword vocabulary with longer Russian words and subwords from subword-nmt.

In the case of English we train RoBERTa-base (Liu et al., 2019), an optimized version of the BERT model. This model was obtained using more than 160GB of uncompressed text, including, apart from the standard BERT datasets, the CC-news dataset with English news articles published between January 2017 and December 2019.

For the Polish language we apply the base version of HerBERT (Mroczkowski et al., 2021). This model is based on the original BERT architecture and achieves state-of-the-art results on several downstream tasks, obtaining the best overall scores for Polish language understanding on the KLEJ Benchmark. HerBERT was trained on two datasets merged from six corpora such as NKJP, Wikipedia, Wolne Lektury, CCNet and Open Subtitles. Its base version outperformed the base version of Polish RoBERTa despite being trained with a smaller batch size (2560 vs 8000) and for a fewer number of steps (50k vs 125k).

In the case of Turkish we use BERTurk.[4] It is a cased BERT-base model, trained on 35GB of data, including Wikipedia, various OPUS corpora (Tiedemann, 2016), data provided by Kemal Oflazer and the version of the Turkish OSCAR corpus (Ortiz Suárez et al., 2019) which was previously filtered and sentence segmented.

Finally, for Basque we use BERTeus (Agerri et al., 2020), a BERT-base model trained on the BMC Basque corpus, which consists of news articles

[3] https://github.com/rsennrich/subword-nmt/

[4] https://github.com/stefan-it/turkish-bert

from online newspapers and the Basque Wikipedia. The authors also perform the subword tokenization, which is closer to linguistically interpretable strings in Basque. BERTeus outperforms mBERT and XLM-RoBERTa in several NLP tasks including named entity recognition, POS tagging, sentiment analysis and topic modelling.

6. Experimental Setup

In order to compare the three different approaches to generate the Shortest Edit Scripts (SES) described in Section 4, we fine-tuned the multilingual and language-specific pre-trained masked language models for each language in a token classification task, where the labels to be predicted correspond to the automatically induced SES. The MLMs were fine-tuned by adding a single linear classification layer on top. We performed a grid search of hyperparameters to select the best batch size (8, 16), weight decay (0.01, 0.1), learning rate (2e-5, 3e-5, 5e-5) and epochs (5, 10, 15, 20). We conduct both in-domain and out-of-domain evaluation of the models. By out-of-domain evaluation we understand evaluating on a data distribution different from the one that was used for training (in the in-domain setting). For each type of SES we chose the best model on the development set among the four MLMs in terms of word accuracy and loss. For all the languages the highest accuracy was achieved using XLM-RoBERTa large model, being the only exception the *ses-morpheus* method in the case of Russian, where the best accuracy was achieved using mBERT. Thus, every result reported in the next subsections is obtained using XLM-RoBERTa-large as a backbone. Finally, apart from calculating word and sentence accuracy scores, we also report the statistical significance across the three SES methods using the McNemar test (Dietterich, 1998).

6.1. Results

Table 4 reports the best overall word accuracy results for in-domain and out-of-domain settings. We could see that among the three SES types *ses-morpheus* is the least optimal. Since its functioning principle implies that the same edit operation may generate various labels depending on the word's total number of characters (as demonstrated in Table 2 with the examples of the words 'cats' and 'birds'), this approach creates the highest amount of unique labels for 5 out of 7 languages of our survey (as illustrated by Table 3). This might be one of the possible reasons that leads to the lower performance of this SES method, as in this case the range of the SES classes is wider, which could difficult the learning and generalization processes of the model.

	ses-udpipe	ses-ixapipes	ses-morpheus
es	444	670	1,213
ru	1,157	2,390	3,208
en	286	445	891
eu	2,247	5,324	3,710
tr	236	4,147	799
cz	1,020	2,345	3,033
pl	947	1,920	2,692

Table 3: The amount of unique labels for each SES type.

	ses-udpipe		ses-ixapipes		ses-morpheus	
	IND	OOD	IND	OOD	IND	OOD
es	0.983	0.971	**0.983**	**0.972***	0.975	0.963
ru	**0.973**	**0.945***	0.970	0.941	0.927	0.885
en	0.991	0.939	**0.991**	**0.941**	0.979	0.916
eu	**0.969***	**0.890***	0.966	0.885	0.952	0.857
tr	**0.964***	**0.853***	0.915	0.827	0.938	0.804
cz	**0.994***	0.947	0.991	**0.951**	0.987	0.924
pl	**0.982***	**0.952**	0.980	0.950	0.943	0.917

Table 4: Word accuracy results for the 3 SES types for in-domain (IND) and out-of-domain (OOD) settings. In **bold**: best overall results across systems and SES types. *:results, that are statistically significant at $\alpha = .05$.

	ses-udpipe		ses-ixapipes		ses-morpheus	
	IND	OOD	IND	OOD	IND	OOD
es	0.703	0.489	**0.708**	**0.505***	0.582	0.397
ru	**0.614**	**0.426***	0.604	0.401	0.314	0.187
en	**0.890**	0.425	0.888	**0.439**	0.773	0.305
eu	**0.684**	**0.203***	0.663	0.195	0.551	0.150
tr	**0.707***	**0.080***	0.496	0.010	0.583	0.050
cz	**0.896***	0.430	0.855	**0.500**	0.796	0.320
pl	**0.876***	**0.656**	0.861	0.657	0.675	0.519

Table 5: Sentence accuracy results for the 3 SES types for in-domain (IND) and out-of-domain (OOD) settings. In **bold**: best overall results across systems and SES types. *:results, that are statistically significant at $\alpha = .05$.

With respect to the other two methods, we could observe that for 5 out of 7 languages (namely, for Russian, Basque, Turkish, Czech and Polish) the highest word accuracy in-domain is achieved using *ses-udpipe* approach (4 out of 5 of these results are statistically significant). However, in the case of Spanish and English the results are almost identical for both *ses-udpipe* and *ses-ixapipes* methods. Regarding out-of-domain, in 4 out of 7 cases *ses-udpipe* is the optimal choice as well (3 statistically significant), while *ses-ixapipes* benefits the Czech language and performs similar to the UDPipe's method for English and Spanish.

Still, the differences in word accuracy results for *ses-udpipe* and *ses-ixapipes* are very small, which makes it difficult to distinguish between approaches. In order to obtain a clearer picture in the methods' performance we decided to additionally compute the sentence accuracy metric as proposed for POS tagging by Manning (2011).

As demonstrated in Table 5, sentence accuracy allows us to better distinguish between the models' performance. First, it confirms the results regarding *ses-morpheus* approach, achieving much lower accuracy for all the languages. Second, the almost equivalent results in word accuracy for English and Spanish using both *ses-udpipe* and *ses-ixapipes* methods are now noticeably different when evaluated using sentence accuracy. While in the case of Spanish the approach of IXA pipes seems to be more beneficial both in-and out-of-domain, for English it allows to achieve 1.4 points better in sentence accuracy out-of-domain. The same phenomenon can be observed in the case of the Czech language, with 7 points better in sentence accuracy out-of-domain for *ses-ixapipes* method with respect to *ses-udpipe*. The results for the rest of the languages follow the tendency obtained with the word accuracy metric, where the *ses-udpipe* method scores the highest.

Although sentence accuracy results provide a clearer picture, we would like to establish whether the differences are in fact statistically significant. Thus, we perform the McNemar test to determine whether the scores obtained by *ses-udpipe* and *ses-ixapipes* are statistically significant or not (null hypothesis). When evaluating word accuracy the test shows that the differences in performance of the two SES approaches mentioned above are statistically significant ($\alpha = .05$) in *ses-udpipe* favor for the agglutinative languages (Basque and Turkish) both in-domain (with p-value < 0.02 for Basque and p-value < 0.001 for Turkish) and out-of-domain (with p-value < 0.001 for both Basque and Turkish); for Czech and Polish languages in-domain (p-value < 0.001 for Czech and p-value < 0.005 for Polish) and for Russian out-of-domain (p-value < 0.001). Such small p-value results indicate that the differences in performance of the models trained with different minimum edit approaches is noticeable. The test results also suggest that in the case of lemmatizing using *ses-ixapipes* method the model commits a larger percentage of the errors respect to *ses-udpipe*. As for *ses-ixapipes*, the results are statistically significant only for Spanish in the out-of-domain setting (p-value < 0.002). For sentence accuracy the McNemar test results reflect the same tendency as for word accuracy. Therefore, the McNemar test indicates that *ses-udpipe* approach is more beneficial in the generation of the Shortest Edit Scripts that the other two methods, at least in the proposed spectrum of languages.

7. Discussion

In order to make the comparison of the three Shortest Edit Script methods more complete we discuss the following points. First, we analyze the performance of the pre-trained masked language models on in-vocabulary and out-of vocabulary words. The aim of such analysis is to understand which SES approach contributes better to the generalization capabilities of the MLMs. Second, we conduct a brief error analysis in order to understand what makes UDPipe's method more successful that its two other counterparts. Finally, we discuss model contamination issues.

Generalization on out-of-vocabulary words Pre-trained masked language models, in particular XLM-RoBERTa, demonstrate good generalization abilities and are capable of achieving competitive results lemmatizing the words they did not see during the training process (Toporkov and Agerri, 2024). In order to check which SES approach benefits such capabilities more, we calculate word accuracy on in-vocabulary and out-of-vocabulary words, comparing how the model performs on the words it has seen during the training respect to the words it sees for the first time. Table 6 reports the results.

Interestingly, all three SES approaches perform equally well on in-vocabulary words in-domain and obtain very similar results out-of-domain. Things start changing when we analyze the out-of-vocabulary performance. We can see the significant drop in the generalization capability of the models using *ses-morpheus* approach, which confirms the word and sentence accuracy results. We also could see that for Spanish, English and Czech the results are better using *ses-ixapipes* method, the point that reinforces the sentence accuracy results. There is also a strong correlation between the results where the differences between *ses-udpipe* and *ses-ixapipes* are statistically significant and how these approaches perform on unseen words.

In any case, from an overall perspective *ses-udpipe* demonstrates stronger performance, achieving the highest accuracy in-domain for 5 out of 7 languages and out-of-domain for 4 out of 7 languages both for in-vocabulary and out-of-vocabulary words. Table 7 in Appendix A provides more detailed results on out-of-vocabulary statistics with respect to lemmas and SES. Thus, the overall better performance of *ses-udpipe* is reinforced by having the lowest percentage rate of SES that have not been seen during training. This data indicates that the *ses-udpipe* approach has better generalization capabilities.

In conclusion, the results of our experiments show that the *ses-udpipe* method is more beneficial for the lemmatization task, especially in the case

		ses-udpipe		ses-ixapipes		ses-morpheus	
		INV	OOV	INV	OOV	INV	OOV
es	ind	0.989	0.906	**0.989**	**0.912**	0.989	0.816
	ood	0.976	0.904	**0.977**	**0.917***	0.975	0.807
ru	ind	**0.995**	**0.908**	0.994	0.900	0.991	0.741
	ood	**0.972**	**0.878***	0.972	0.865	0.967	0.686
en	ind	**0.995**	**0.931**	0.994	0.927	0.993	0.751
	ood	**0.954**	0.833	0.953	**0.849**	0.954	0.631
eu	ind	**0.990**	**0.852***	**0.990**	0.832	0.989	0.748
	ood	**0.926**	**0.777***	**0.926**	0.757	0.926	0.645
tr	ind	0.991	**0.882***	0.991	0.685	**0.992**	0.775
	ood	**0.946**	**0.693***	0.945	0.625	0.944	0.564
cz	ind	**0.998**	**0.955***	**0.998**	0.923	**0.998**	0.876
	ood	0.987	0.807	**0.988**	**0.821**	0.987	0.703
pl	ind	**0.998**	**0.919***	0.997	0.909	0.992	0.742
	ood	**0.981**	**0.816**	0.981	0.808	0.974	0.650

Table 6: Word accuracy for in-vocabulary (INV) and out-of-vocabulary (OOV) words for in-domain (ind) and out-of-domain (ood) results. In **bold**: best results per SES and per language; *:results, that are statistically significant at $\alpha = .05$.

of the languages with more complex morphology. To analyze what makes this method better than its close counterpart *ses-ixapipes*, we conduct a brief error analysis in an attempt to identify the most important factors that may influence performance.

Error Analysis The first noticeable advantage that is perceived in the structure of the *ses-udpipe* patterns is the absence of indexing. While *ses-ixapipes* misplaces some indexes, wrongly annotating them to the letters that should be deleted or replaced, *ses-udpipe* approach simplifies this process by only indicating the positions of the letters that should be modified without having to map it with the corresponding index. Such misplacements usually affect the complex words that need a lot of edit operations in order to be lemmatized.

Another important difference is how to deal with non-Latin alphabet and some language-specific letters. In the cases of such languages as Russian and Turkish these letters may cause a certain confusion during minimum edit generations as it happens to *ses-ixapipes*, which sometimes assigns to the final SES pattern the letters that do not appear neither in the surface word, nor in the lemma.

The third interesting observation is encountered mostly in the lemmatization of agglutinative languages (Basque and Turkish) and has to do with their suffixal nature. Whereas the *ses-udpipe* method processes the parts of the words separately, *ses-ixapipes* does not take into account this issue. Thus, *ses-ixapipes* focuses on indexing the correct letters without considering if its the part of the suffix or of the root. As a result, this approach may create an alternative minimum edit script, which may map correctly, but that does not coincide with the gold standard SES. For example, when lemmatizing the Basque word *folklorearen* ('of folklore', lemma *folklore*), the gold standard SES would be D5rD4eD3aD0n, while in one of the predictions *ses-ixapipes* offered an alternative version of SES, which is D4eD3aD2rD0n. Applying both sets of edits will deliver the same result, but as the goal of the classification task is to correctly assign the corresponding SES to its surface word, such cases are considered incorrect. In order to check whether this could be crucial in evaluating the overall SES performance, we calculate the total number of occurrences where the SES distinct from the gold standard delivers the correct lemma for the Basque language. Our results show that for *ses-udpipe* approach there are 9 out of 11901 cases where an alternative SES leads to the same lemma (in-domain), while in the case of *ses-ixapipes* the number of such occurrences is 17 out of 11901 respectively. This data indicates, that although such cases could appear, their influence on the overall result is insignificant.

Finally, it also seems beneficial to encode the casing script as implemented in the *ses-udpipe* method and which is only partially implemented in both *ses-ixapipes* and *ses-morpheus* approaches.

Regarding the other two minimum edit approaches, namely, *ses-ixapipes* and *ses-morpheus*, a brief error inspection shows that in the case of *ses-ixapipes* most of the errors are of suffixal and root nature, more precisely, in the incorrect indexing or letter misplacement. Furthermore, the performance of *ses-morpheus* is mainly affected by the large number of generated SES classes, which makes the classification task much more difficult. The cases where lemma is longer than wordform, and, therefore the edit operations are applied jointly, constitute between 5 and 15 of the total error rate across the inspected languages, and is another source of possible low performance of this method with respect to the other two.

A word on model contamination We would like to finish by offering a word on model contamination. More specifically, we would like to discuss whether the performance of a MLM such as XLM-RoBERTa has been spuriously high because the model already saw the datasets we are experimenting with during pre-training, namely, whether XLM-RoBERTa has been contaminated. In order to address this, we would like to clarify that CC-100, the corpus used to train XLM-RoBERTa, was constructed with CommonCrawl snapshots from between January and December 2018. Moreover, the SIGMORPHON data was released in 2019[5] with the test data including gold standard lemma and UniMorph annotations not being released until

[5] First GitHub commit December 19, 2018.

April 2019. Finally, and most importantly, XLM-RoBERTa does not see the lemmas themselves during training or inference, but the SES classes we automatically generate in an ad-hoc manner for the experimentation. The datasets containing both the words and the SES classes used have not been yet made publicly available. Therefore, we can conclude that XLM-RoBERTa seems to generalize over unseen words and that its performance is not justified by any form of language model contamination.

8. Conclusion

In this paper, we present the first detailed systematic comparison of three popular methods to compute Shortest Edit Scripts (SES), widely used in modern contextual lemmatization models. After a comprehensive battery of experiments with various evaluation metrics and statistical tests, results indicate that *ses-udpipe* is the optimal method for contextual lemmatization among the Shortest Edit Script approaches. Its main advantages consist in: (i) computing casing and edit operations separately; (ii) processing the wordform by morphemes and the absence of indexing, which allows to avoid the cases where there are the same letters in the suffix and the root (especially for agglutinative languages such as Basque and Turkish) and to create less ambiguous SES; (iii) better generalization capabilities, that result in obtaining less out-of-vocabulary SES and creating fewer SES labels, which benefits the models by having to learn a smaller amount of SES classes. Furthermore, our results indicate the following: (i) more metrics should be implemented in the analysis of the MLMs performance along with the word accuracy; (ii) out-of-domain evaluation should be considered as an important step as it allows to obtain a clearer picture of how far the task is solved.

We believe that the results of our study could be useful for the future development of contextual lemmatizers which may include SES as an integral component of their systems.

9. Acknowledgements

This work has been partially supported by the HiTZ center and the Basque Government (Research group funding IT-1805-22). We are also thankful to several MCIN/AEI/10.13039/501100011033 projects: (i) DeepKnowledge (PID2021-127777OB-C21) DeepKnowledge (PID2021-127777OB-C21) and by FEDER, EU; (ii) Disargue (TED2021-130810B-C21) and European Union NextGenerationEU/PRTR; (iii) DeepMinor (CNS2023-144375) and European Union NextGenerationEU/PRTR. Olia Toporkov is supported by a doctoral grant from the UPV/EHU "Formación de Personal Investigador" (PIF 20/186). Rodrigo Agerri was additionally funded by the RYC-2017-23647 fellowship (MCIN/AEI/10.13039/501100011033 and by ESF Investing in your future).

10. Bibliographical References

Rodrigo Agerri and Eneko Agirre. 2023. Lessons learned from the evaluation of Spanish Language Models. *Proces. del Leng. Natural*, 70:157–170.

Rodrigo Agerri, Josu Bermudez, and German Rigau. 2014. IXA pipeline: Efficient and ready to use multilingual NLP tools. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3823–3828, Reykjavik, Iceland. European Language Resources Association (ELRA).

Rodrigo Agerri and German Rigau. 2016. Robust multilingual named entity recognition with shallow semi-supervised features. *Artificial Intelligence*, 238(2):63–82.

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for Basque. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4781–4788, Marseille, France. European Language Resources Association.

Brad Aiken, Jared Kelly, Alexis Palmer, Suleyman Olcay Polat, Taraka Rama, and Rodney Nielsen. 2019. Sigmorphon 2019 task 2 system description paper: Morphological analysis in context for many languages, with supervision from only a few. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 87–94, Florence, Italy. Association for Computational Linguistics.

I. Aldezabal, M.J. Aranzabe, A. Diaz de Ilarraza, and K. Fernández. 2008. From dependencies to constituents in the reference corpus for the processing of Basque. *Procesamiento del Lenguaje Natural*, (41):147–154.

Iñaki Alegria, Xabier Artola, Kepa Sarasola, and Miriam Urkia. 1996. Automatic morphological analysis of basque. *Literary and Linguistic Computing*, 11:193–203.

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. In *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing*, pages 89–93, Florence, Italy. Association for Computational Linguistics.

Xavier Carreras, Isaac Chao, Lluís Padró, and Muntsa Padró. 2004. Freeling: An open-source suite of language analyzers. In *LREC*.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained BERT model and evaluation data. In *PML4DC at ICLR 2020*.

Abhisek Chakrabarty, Onkar Arun Pandit, and Utpal Garain. 2017. Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1481–1491, Vancouver, Canada. Association for Computational Linguistics.

Grzegorz Chrupala, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with Morfette. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas G. Dietterich. 1998. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

Andrea Gesmundo and Tanja Samardžić. 2012. Lemmatisation as a tagging task. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 368–372, Jeju Island, Korea. Association for Computational Linguistics.

Barbora Hladká, Jan Hajič, Jirka Hana, Jaroslava Hlaváčová, Jiří Mírovský, and Jan Raab. 2008. The Czech academic corpus 2.0 guide. *The Prague Bulletin of Mathematical Linguistics*, 89:41–96.

Lauri Karttunen, Ronald M. Kaplan, and Annie Zaenen. 1992. Two-level morphology with composition. In *COLING 1992 Volume 1: The 14th International Conference on Computational Linguistics*.

Yuri Kuratov and Mikhail Y. Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for Russian language. *CoRR*, abs/1905.07213.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Olga Lyashevkaya, Kira Droganova, Daniel Zeman, Maria Alexeeva, Tatiana Gavrilova, Nina Mustafina, and Elena Shakurova. 2016. Universal Dependencies for Russian: A new syntactic dependencies tagset. *SSRN Electronic Journal*.

Chaitanya Malaviya, Shijie Wu, and Ryan Cotterell. 2019. A simple joint model for improved contextual neural lemmatization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1517–1528, Minneapolis, Minnesota. Association for Computational Linguistics.

Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Computational Linguistics and Intelligent Text Processing*, pages 171–189, Berlin, Heidelberg. Springer.

Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020. UniMorph 3.0: Universal Morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. 2021. HerBERT: Efficiently pretrained transformer-based language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine. Association for Computational Linguistics.

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2015. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal. Association for Computational Linguistics.

Joakim Nivre, Daniel Zeman, Filip Ginter, and Francis Tyers. 2017. Universal Dependencies. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, Valencia, Spain. Association for Computational Linguistics.

Kemal Oflazer. 1993. Two-level description of Turkish morphology. In *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht, The Netherlands. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019*, pages 9–16.

Adam Przepiórkowski and Agnieszka Patejuk. 2018. From Lexical Functional Grammar to enhanced Universal Dependencies. In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 2–4, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ilya Segalovich. 2003. A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. In *MLMTA*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Milan Straka, Jana Straková, and Jan Hajic. 2019. UDPipe at SIGMORPHON 2019: Contextualized embeddings, regularization with morphological categories, corpora merging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 95–103, Florence, Italy. Association for Computational Linguistics.

Nicolas Stroppa and François Yvon. 2005. An analogical learner for morphological analysis. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 120–127, Ann Arbor, Michigan. Association for Computational Linguistics.

Mariona Taulé, M. Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel annotated corpora for Catalan and Spanish. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann. 2016. OPUS – parallel corpora for everyone. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation: Projects/Products*, Riga, Latvia. Baltic Journal of Modern Computing.

Olia Toporkov and Rodrigo Agerri. 2024. On the Role of Morphological Information for Contextual Lemmatization. *Computational Linguistics*, pages 1–35.

Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Abdullatif Köksal, Balkiz Ozturk Basaran, Tunga Gungor, and Arzucan Özgür. 2019. Turkish treebanking: Unifying and constructing efforts. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 166–177, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Eray Yildiz and A. Cüneyd Tantuğ. 2019. Morpheus: A neural network for jointly learning contextual lemmatization and morphological tagging. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 25–34, Florence, Italy. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

11. Language Resource References

2000. *Armiarma corpus*. SUSA-LITERATURA. PID https://armiarma.eus/.

Appendix A. Detailed Out-of-Vocabulary Results

				ses-udpipe		ses-ixapipes		ses-morpheus	
		oov words	oov lemmas	oov ses	oov lemmas (ses in train)	oov ses	oov lemmas (ses in train)	oov ses	oov lemmas (ses in train)
es	ind	7.85%	6.18%	**0.02%**	99.89%	0.05%	99.74%	0.11%	99.07%
	ood	7.65%	5.93%	**0.28%**	96.41%	0.43%	98.27%	**0.28%**	97.86%
ru	ind	25.50%	13.74%	**0.27%**	99.04%	0.67%	98.31%	0.78%	97.35%
	ood	29.21%	15.36%	**1.65%**	96.23%	2.45%	95.03%	2.52%	94.33%
en	ind	5.71%	4.19%	**0.08%**	99.72%	0.10%	99.81%	0.25%	97.57%
	ood	11.89%	11.45%	**1.32%**	90.41%	1.56%	91.58%	1.36%	91.15%
eu	ind	15.28%	5.07%	**0.61%**	96.52%	1.45%	94.86%	0.92%	94.69%
	ood	24.26%	11.99%	**1.13%**	95.98%	2.49%	94.68%	1.45%	94.78%
tr	ind	24.83%	5.67%	**0.12%**	99.69%	4.52%	95.69%	0.56%	97.23%
	ood	36.71%	20.72%	**0.45%**	97.85%	6.52%	91.40%	3.40%	97.85%
cz	ind	8.85%	3.19%	**0.09%**	99.11%	0.20%	98.66%	0.24%	97.90%
	ood	21.97%	11.76%	**2.33%**	99.12%	2.59%	99.12%	2.90%	98.24%
pl	ind	19.53%	7.80%	**0.28%**	99.22%	0.52%	98.82%	0.85%	97.84%
	ood	17.65%	8.72%	**0.40%**	98.65%	0.62%	97.57%	0.67%	97.57%

Table 7: The proportion (in %) of out-of-vocabulary words, lemmas and SES in the in-domain (ind) and out-of-domain (ood) test sets with respect to the train set, per language. In **bold**: lowest percentage of out-of-vocabulary (oov) SES among the three SES types.

Table 7 reports the proportion of out-of-vocabulary (oov) words, lemmas and SES, both for in-domain (ind) and out-of-domain (ood) settings for the three SES types. By out-of-vocabulary we understand words, lemmas and SES in the test sets that the system did not see during the training process. The column *'oov lemmas (ses in train)'* refers to the proportion of lemmas that the model does not see during the training (out-of-vocabulary lemmas) while their corresponding SES exist in the train set. In other words, they have been seen by the system.