Multi-Team: A Multi-attention, Multi-decoder Approach to Morphological Analysis.

Ahmet Üstün **Rob van der Goot** **Gosse Bouma** **Gertjan van Noord**
University of Groningen
{a.ustun, r.van.der.goot, g.bouma, g.j.m.van.noord}@rug.nl

Abstract

This paper describes our submission to SIGMORPHON 2019 Task 2: Morphological analysis and lemmatization in context. Our model is a multi-task sequence to sequence neural network, which jointly learns morphological tagging and lemmatization. On the encoding side, we exploit character-level as well as contextual information. We introduce a multi-attention decoder to selectively focus on different parts of character and word sequences. To further improve the model, we train on multiple datasets simultaneously and use external embeddings for initialization. Our final model reaches an average morphological tagging F1 score of 94.54 and a lemma accuracy of 93.91 on the test data, ranking respectively 3rd and 6th out of 13 teams in the SIGMORPHON 2019 shared task.

1 Introduction

This paper presents our model for the SIGMORPHON 2019 Task 2 on morphological analysis and lemmatization in context (McCarthy et al., 2019). The task is to generate a lemma and a sequence of morphological tags, which are called morphosyntactic descriptions (MSD), for each word in a given sentence. This task is important because it can be used to improve several downstream NLP applications such as grammatical error correction (Ng et al., 2014), machine translation (Conforti et al., 2018) and multilingual parsing (Zeman et al., 2018). Table 1 shows the lemma and morphological tags of: *Johnny likes cats*.

The first sub-task, Lemmatization, is to transform an inflected word form to its lemma which is its base-form (or dictionary form), as in the example of *likes* to *like*. The second sub-task, morphological tagging, is to predict morphological properties of words as a sequence of tags, including a part of speech tag. These morphological tags specify the inflections encoded in word-forms. In the example sentence, the word *likes* is annotated with a morphological tag set of {*V,SG,3,IND,PRS*}. Both tasks are dependent on context. For example, while *walking* is annotated with the lemma *walk* and tag set {*N,SG*} in the sentence: *The beach is within walking distance*; it is annotated with *walking* and {*V.PTCP;PRS;V*} in: *I was walking*.

Orig	Johnny	likes	cats	.
Lemma	Johnny	like	cat	.
MSD	PROPN;SG	V;SG;3;IND;PRS	N;PL	_

Table 1: Example sentence, annotated with lemmas and morphological tags.

These two tasks have a clear relation; in most languages the categories found in the morphological tags indicate how the lemma of the word was inflected to the word-form. In other words, syntactic inflections have a strong correlation with the morphological properties of the words.

Our approach to solve both of these tasks consists of an encoder and two separate decoders within a multi-task architecture based on a sequence-to-sequence network. The shared encoder reads words and sentences to learn character-level and word-level representations. The decoders then separately generate lemmas and morphological tags using these representations by using multiple attention mechanisms. Our contributions are threefold:

- We introduce the use of multiple attention mechanisms that selectively focus character and word sequences in the sentence context.
- We evaluate the effect of a variety of types of external embeddings for lemmatization and morphological tagging.
- We evaluate the effect of combining annotated datasets from related languages for both tasks

Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 35–49
Florence, Italy. August 2, 2019 ©2019 Association for Computational Linguistics

using dataset embeddings.

2 Related work

Our system is based on three main approaches which are heavily studied in existing literature. These are sequence-to-sequence learning, multi-task learning and multi-lingual learning.

Recent work on computational morphology showed that neural sequence-to-sequence (seq2seq) models (Sutskever et al., 2014; Bahdanau et al., 2014) have yielded new state-of-the-art performance on various tasks including morphological reinflection and lemmatization (Cotterell et al., 2016, 2017, 2018). Building on this, Dayanık et al. (2018) utilize different levels of representations such as character-level, word-level and sentence-level in the encoder of their seq2seq architecture based on previous work (Heigold et al., 2017).

Multi-task learning approaches for jointly learning related tasks have been successfully employed on syntactic and semantic tasks (Søgaard and Goldberg, 2016; Plank et al., 2016). In the context of morphological analysis, this has been used by Kementchedjhieva et al. (2018), who jointly learn morphosyntactic tags and inflections for a word in a given context, and use a shared encoder within a multi-task architecture consisting of multiple decoder similar to our model.

Multi-lingual learning approaches which benefit from joint learning for multiple languages is also studied on various tasks with different architectures. Ammar et al. (2016) uses a language embedding that contains information considering the language, word-order properties and typological properties for dependency parsing. In multilingual neural machine translation, Johnson et al. (2017) use a special token to indicate the target language. In this work, our model uses the approach of Smith et al. (2018) who introduce the treebank embedding approach to combine several treebanks for a single language or closely related languages.

Most similar to our model, Kondratyuk et al. (2018) use a joint decoder approach for morphological tagging and lemmatization. However, our model differs from theirs in substantial ways. Our model employs an encoder-decoder architecture which utilizes different levels of attention components with a multi-lingual/multi-dataset signal. Moreover, our model solves the tagging problem as a sequential prediction task instead of multi-layer classification so that we can use the same architecture for both lemmatization and tagging which are described in Section 3.2 and 3.3.

3 System Description

Our model is inspired by the architecture of Dayanık et al. (2018). We employ an encoder-decoder model over the character and word sequences. Following Dayanık et al. (2018), the encoder in our model consists of two parts. First, a word encoder which runs on the character level, is used to generate embeddings for each word (Section 3.1.1). Second, a context encoder is initialized with these word embeddings, and runs on the sentence level (Section 3.1.4). We also experiment with two methods to complement the word-level embeddings (Section 3.1.2 and 3.1.3).

The representations at the different levels which are generated by the encoder are then passed into the decoders. Unlike Dayanık et al. (2018) which uses one decoder for both the lemmas and the morphological tags, we use two different decoders in a multi-task architecture. The tag decoder produces a set of morphological tags by using word representations and joint attention mechanism that one attention focuses on words and other focuses on characters (Section 3.2). The lemma decoder produces a lemma by using the same information complemented with output embeddings of the tag decoder (Section 3.3).

Multi-task Learning The decoders work jointly in a multi-task fashion and they share all internal representations of the encoder. The whole network is trained by backpropagating the sum of the losses of the decoders without any weighting:

$$\mathbf{L}(\theta) = L_{tag} + L_{lemma} \tag{1}$$

where the morphological tag loss L_{tag} and the lemma loss L_{lemma} are separately computed as the negative log likelihood loss over their softmax outputs.

Notation Given a sentence $S = w_1, ..., w_n$ and $w_i = c_1, ..., c_m$ where w denotes words and c denotes characters, our model processes S and w in encoders and jointly produces a set of morphological tags $t_i = t_{i,1}, ..., t_{i,\gamma}$ and a lemma $l_i = l_{i,1}, ..., l_{i,\phi}$ which is a sequence of characters.

3.1 Encoder

In the following subsections, we explain the different parts of the encoder. An overview of the encoder architecture is shown in Figure 1.

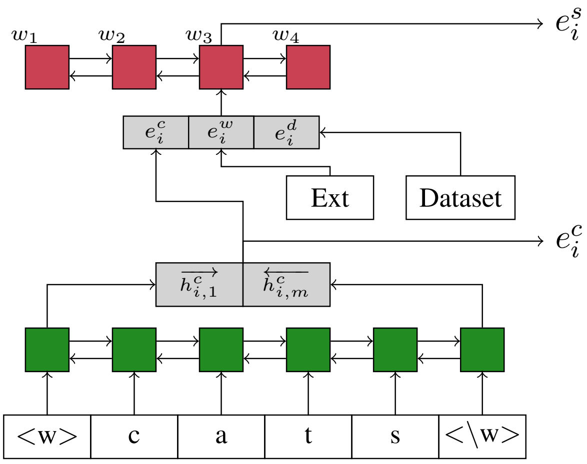

Figure 1: Overview of the encoder when processing the third word of the sentence: "Johnny likes cats .". Red:word level embeddings. Green: Character level embeddings.

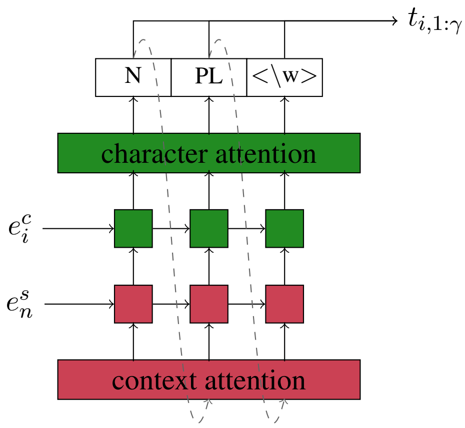

Figure 2: An overview of the morphological tag decoder.

3.1.1 Word Encoder

We use a bidirectional GRU layer (Cho et al., 2014) to encode character sequences in the word encoder. We first pass each character of a word w_i to an embedding layer to map them into the fixed dimensional character embeddings. The bi-GRUs process character embeddings in both directions and produce the hidden states $h^c_{i,1}, ..., h^c_{i,m}$. The resulting word embedding e^c_i is computed by concatenating the final states of forward and backward GRUs for the given word:

$$h^c_{i,1:m} = \text{bi-GRU}(c_{i,1:m}) \quad (2)$$

$$e^c_i = [\overrightarrow{h^c_{i,m}}; \overleftarrow{h^c_{i,1}}] \quad (3)$$

3.1.2 Word-Surface Embeddings

In addition to the character-level word embeddings, we use surface-level word embeddings which are either learned in a standalone embedding layer or taken from the pre-trained external embeddings. Word-surface embeddings are denoted by e^w_i. For the unknown words, we used a *word droupout* to overcome the sparsity issue. Following Kiperwasser and Goldberg (2016), we replace unknown tokens with a probability that is inversely proportional to the frequency of the word so that the word representation for an unknown token is learned based on infrequent words and their context.

3.1.3 Dataset Embeddings

In order to train our model on multiple datasets at once, we use dataset embedding e^d_a for each dataset a which is mapped into a fixed dimensional vector in an embedding layer. The idea of dataset embeddings is introduced by Smith et al. (2018). These embeddings enable us to combine multiple datasets without losing their monolingual and heterogeneous characters. The strategy that we use to pick and combine datasets is described in Section 4.2

3.1.4 Context Encoder

In order to encode sentence level contextual information, we use another bidirectional GRU layer. For a given sentence, we first concatenate the output of the word encoder e^c_i, the word-surface embedding e^w_i and the dataset embedding e^d_a, for each word in the sentence. The resulting embedding sequence $e^{in}_1, ..., e^{in}_n$ is then passed into the bi-GRU. The output of the bi-GRU is a sequence of embeddings $e^s_1, ..., e^s_n$ each representing a word in the sentence:

$$e^{in}_i = [e^c_i; e^w_i; e^d_a] \quad (4)$$

$$e^s_{1:n} = \text{bi-GRU}(e^{in}_{1:n}) \quad (5)$$

3.2 Tag Decoder

As the tag decoder shows in Figure 2, we use a 2 layer stacked bidirectional GRU as the tag decoder to generate morphological tags $t_i = t_{i,1}, ..., t_{i,\gamma}$ for the target word w_i in a given sentence. In order to utilize both character-level representations and contextual representations during decoding, we initialize the first layer of the decoder with the context-level word embedding e^s_i and the second layer of the decoder with the character-level word embedding e^c_i after passing them through a *relu* layer. The decoder outputs the morphological tags over a softmax layer based on the final hidden states $\widetilde{h}_t$, which are computed in a joint attention mechanism described in the following section.

$$\widetilde{h}_t = \text{decoder}(h_t, c^c_t, c^s_t) \quad (6)$$

$$p(t_{i,t}|\widetilde{h}_t) = \text{softmax}(\widetilde{h}_t) \quad (7)$$

3.2.1 Joint Context and Character Attention

We employ two different attention mechanisms to allow the decoder to focus on multiple parts of the sentence and the target word at the same time. We use the attention mechanism introduced by Bahdanau et al. (2014) for the context attention layer. In the context attention, the alignment vector a^s_t, which consists of weights for each word in the sentence, is computed based on the previous hidden state h_{t-1} at the top layer of the stacked bi-GRU and context-level embeddings e^s of words by using the *concat* function described in Luong et al. (2015). The sentence-level context vector c^s_t which is calculated as a weighted average over word embeddings, is then passed into a simple concatenation layer W^s_c to produce the new hidden state h_t through the stacked bi-GRU:

$$a^s_t(i) = \text{align}^s(h_{t-1}, e^s_i) \quad (8)$$

$$c^s_t = \sum_i a^s_t e^s_i \quad (9)$$

$$h_t = \text{bi-GRU}(W^s_c[c^s_t; h_{t-1}], h_{t-1}) \quad (10)$$

Together with the context attention, we also employ a character-level attention model to take into account the entire output of the word encoder. We use the global attention mechanism with the *general* score function for alignment vectors (Luong et al., 2015), for the character attention. The source-side character-level attention vector c^c_t is computed as a weighted average of the outputs of the word encoder, each denoted by $h^c_{i,j}$. The resulting output state $\widetilde{h}_t$ of the tag decoder is then generated by concatenating the current hidden state at the top of the stacked bi-GRU h_t and the context vector c^c_t in a concatenation layer which has a *tanh* activation:

$$a^c_t(j) = \text{align}^c(h_t, h^c_{i,j}) \quad (11)$$

$$c^c_t = \sum_j a^c_t h^c_{i,j} \quad (12)$$

$$\widetilde{h}_t = \tanh(W^c_c[c^c_t; h_t]) \quad (13)$$

3.3 Lemma Decoder

The lemma decoder (Figure 3) produces one character at a time to sequentially form a lemma $l_i = l_{i,1}, ..., l_{i,\phi}$ for a target word w_i. Similar to the tag encoder, we use a 2 layer stacked bi-GRU as lemma decoder. The initial states of the decoder layers are taken from the word encoder output e^c_i and the context encoder output e^s_i through a *relu* layer similarly as in the tag decoder. The output of the lemma decoder $l_{i,t}$ is conditioned on the current state of the decoder h_t, the character attention c^c_t and the morphological tags $t_{i,1:\gamma}$ of the target word. The probability of the output lemma characters are then predicted through a softmax layer.

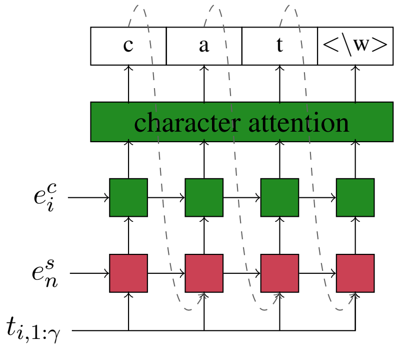

Figure 3: An overview of the lemma decoder.

$$\widetilde{h}_t = \text{decoder}(h_t, c^c_t, t_{i,1:\gamma}) \quad (14)$$

$$p(l_{i,t}|\widetilde{h}_t) = \text{softmax}(\widetilde{h}_t) \quad (15)$$

In order to exploit morphological features during lemmatization, we give the morphological tags $t_{i:\gamma}$ which are predicted by the tag decoder, as part of input to the lemma decoder. Independent of their order, the entire set of the tags are encoded by a simple *feed-forward* layer as described in the baseline model (Malaviya et al., 2019) and the resulting vector is concatenated with the input embeddings for each target word.

The last part of the lemma decoder is the attention network which is the same character-level attention model as in the tag decoder. The character attention mechanism allows the lemma decoder to compute an attention vector c^c_t based on the output states of the word encoder. The attention vector is then passed into a concatenation layer to generate the output state $\widetilde{h}_t$ of the decoder for each lemma character $l_{i,t}$.

$$a^c_t(j) = \text{align}^c(h_t, h^c_{i,j}) \quad (16)$$

$$c^c_t = \sum_j a^c_t h^c_{i,j} \quad (17)$$

$$\widetilde{h}_t = \tanh(W^c_c[c^c_t; h_t]) \quad (18)$$

Parameter	Val.	Parameter	Val.
teacher forcing ratio	0.5	dataset embbedding size (e_a^d)	32
dropout	0.25	word enc. hidden size (h_i^c)	1,024
patience	4	context enc. hidden size (h_i^s)	1,024
word enc. input size	128	dec. input size	128
word embedding size (e_i^w)	256	dec. hidden size (h_t)	1,024

Table 2: Default hyperparameter settings. Encoder and decoder are denoted by *enc* and *dec* respectively.

4 Setup

In this section we will give the details regarding our experimental setup. The hyperparameters we used in our experiments are shown in Table 2. These hyperparameters have been tuned on the datasets described in Section 5.1. For the training, we used ADAM (Kingma and Ba, 2014) and we applied an early stopping strategy with a minimum number of 100 epochs. We stop training if there is no improvement in the development set for 4 consecutive epochs (patience).

4.1 External Embeddings

Because of time-constraints and the large number of languages in the dataset, we used out-of-the-box embeddings. We compared the performance of three well-known pre-trained embedding repositories for different training methods. We use two word-based embeddings: Polyglot embeddings (Al-Rfou et al., 2013), and FastText embeddings (Grave et al., 2018). For FastText, two sets of pre-trained embeddings are available: one is trained only on Wikipedia (Bojanowski et al., 2017), whereas the newer versions are also trained on CommonCrawl (Grave et al., 2018). Whenever available, we pick the newer embeddings, but for many low-resource languages we fall back to the older, smaller version. We also experiment with context-based embeddings, namely ELMo embeddings (Peters et al., 2018), we use the pre-trained models from Che et al. (2018).

All of these embeddings have been trained using the default settings for the embedding type, hence their dimensions are substantially different (Polyglot; 64, FastText:300, ELMo:1,024) . We decided not to transform these, as their default dimensions are tuned towards their training algorithm and we want to provide a fair comparison for all out-of-the-box settings.

4.2 Dataset Embeddings

For the dataset embeddings, we only consider combining pairs of two for efficiency reasons. To ensure that we match datasets which are informative, we use word overlap (excluding numberals and punctuation). As this method is expected to be most benficial for small datasets, we searched for datasets which are closest (ie. have a large word overlap) to the 50 smallest datasets. The final pairs of datasets can be found in Appendix A.

5 Experiments

In this section, we will describe the data used in our experiments as well as evaluate the effectiveness of our external embeddings setup and the dataset embeddings with in a variety of settings. In all experiments we use +E and -E to indicate the model with and without external embeddings, and +D and -D for dataset embeddings.

5.1 Data

The test data of SIGMORPHON 2019 task 2 consists of a collection of datasets released in the Universal Dependencies project (Nivre et al., 2018), which are automatically converted to the UniMorph Schema (McCarthy et al., 2018). In total, we evaluate our model on 107 datasets, covering 66 languages.

After empirically looking at the trade-off between data-size and training time, we decided to limit each dataset to its first 250,000 tokens for all experiments. This speeded up the training considerably, with almost no loss in performance.

For the tuning of our model, we selected a sub-set of datasets from the main benchmark. More specifically, we aimed to get a diversion of language-family, size, and morphological richness (here proxied by the average amount of morphological tags per word). To ensure we do not overfit on a specific dataset/annotation, we selected two datasets for each of these languages. The selected datasets are shown in Table 3.

5.2 Baseline

The baseline consists of two separate parts: a morphological tagger and a lemmatizer. The morphological tagger, which predicts a set of morphological features (as one tag) for each word, is a biLSTM model with character level layers. The k-best predicted morphological tags are then used

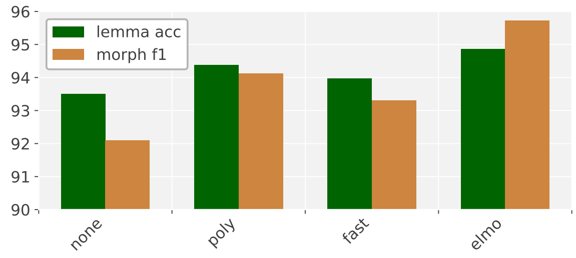

Figure 4: Results of our model when using a variety of types of external embeddings.

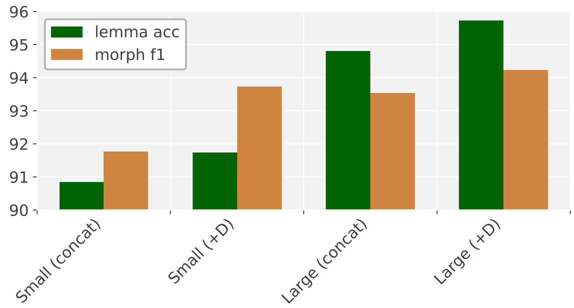

Figure 5: Average results of our model when using simple dataset concatenation versus using dataset embeddings (+D) on 4 small datasets and 4 large datasets

as extra information to improve the lemmatization. The lemmatizer, which is based on Wu et al. (2018), uses a hard attention mechanism within an encoder-decoder model. Unlike the previous models, the morphological tags are explicitly given to the lemmatizer to indicate the morpho-syntactic features of words. The lemmatizer combines the given morphological tags with a character encoding to predict the lemma.

5.3 External Embeddings

In Figure 4, we plotted the average performance of our model when the different types of embeddings are used to initialize the word-surface embeddings (detailed results are in Appendix B). The results show that a performance boost of approximately 2.5% can be obtained for lemmatization and 5% for morphological tagging. Especially the ELMo embeddings perform very well, and result in an improvement of 3.77 and 6.35 percentage points. The Polyglot embeddings perform surprisingly well, considering they only have an embedding size of 64. In addition to the reported settings, we also experimented with concatenating the vectors from all types of external embeddings. However, our empirical results showed that this performed worse compared to using any of the embeddings in isolation.

Because not all types of embeddings are available for all languages, we use fallback options for the test data. We choose embeddings in the following order: ELMo, Polyglot, FastText. After this selection, three languages still have no embeddings (Akkadian, Coptic and Naija), we omitted datasets in these languages from the external embedding experiments.

Dataset	Language Family	Sents	words	tag/word
en_ewt	IE,Germanic	13,297	204,857	1.95
en_pud	IE,Germanic	800	16,927	1.88
tr_imst	Turkic,Southwestern	4,508	46,417	3.58
tr_pud	Turkic,Southwestern	800	13,380	2.78
zh_cfl	Sino-Tibetan	360	5,688	1.00
zh_gsd	Sino-Tibetan	3,997	98,734	1.06
fi_pud	Uralic,Finnic	800	12,556	2.97
fi_ftb	Uralic,Finnic	14,978	127,536	3.07

Table 3: The datasets which we used to tune our models, with data properties based on the training split. IE: Indo-European

5.4 Dataset Embeddings

To test whether the dataset embeddings are necessary, we compare them with a naive approach to combine datasets: simply training on the concatenation of both datasets. The average results on 4 small datasets and 4 large datasets which are given in Table 3, are compared separately in Figure 5. In both small and large settings, using dataset embeddings improves the performance in both morphological tagging and lemmatization, however the effect of dataset embeddings is higher on small datasets, especially in the morphological tagging task. For the detailed results on our tune datasets, we refer to Appendix C.

6 Results

In this section, we will compare our final results for two settings with the baseline. In general, we compare two setups: use of external data (external embeddings, +E) and a constrained setup (-E), which only uses training data. For the dataset embeddings, we could only run for the smallest 50 datasets because of time limitations, so for the development data, we only report results for these datasets. For the test data, we used dataset embeddings for datasets for which they have shown to be beneficial on the development data. Our average results are shown in Table 7. For the results for all four settings per dataset, we refer to Appendix D; here we see that the best setting is generally to use dataset embeddings when available.

6.1 Morphological Tagging

For the morphological tagging task, external embeddings show to be more beneficial for the tagging task, whereas the dataset embeddings are particularly beneficial for lemmatization, but combining them leads to the best scores for both tasks. Furthermore, our model outperforms the baseline by a large margin. This is because, while the baseline has a separate component for morphological tagging, our model learns both tasks jointly. This approach implicitly enables the decoder to access lemma information for morphological tagging. Besides, we use a multi-attention strategy which combines word level and character level attentions which improves the tagging performance.

6.2 Lemmatization

In contrast to the results on the development data, the baseline outperforms our model on the test data (Table 7). Especially on small datasets which are not paired with another dataset, such as `UD_Akkadian-PISANDUB`, the baseline performs better with a large margin.

There are two main reasons for this performance difference. First, the baseline uses a hard attention to model alignment distribution explicitly, whereas, our model uses soft attention for both tasks. The results show that a hard attention mechanism performs better on the lemmatization, confirming the findings of Wu et al. (2018). Integrating a lemma decoder having hard attention with a morphological tag decoder which employs soft attention, could be explored in future studies. Second, as explained in the previous section, we optimize for both tasks jointly without any weighting. Although this is more elegant, as only one model is trained, it might not lead to the most optimal performance.

7 Conclusion

In this paper, we presented our model for the Sigmorphon 2019 Task 2 on morphological analysis and lemmatization. We use an encoder-decoder model by utilizing multi-task learning approach. A shared encoder runs on the character and sentence level and two separate decoders jointly learn to generate morphological tags and the lemma for each word.

	Morph. tags		Lemma	
Models	Acc	F1	Acc	Lev
dev (small)				
base	69.66	85.38	91.53	0.19
-E -D	83.16	89.45	86.75	0.29
+E -D	85.84	91.54	87.65	0.28
-E+D	85.58	91.26	89.70	0.27
+E+D	88.03	92.96	91.29	0.24
test (all)				
base	73.16	87.92	94.17	0.13
-E	89.00	93.35	93.05	0.16
+E	90.61	94.57	93.94	0.15

Table 7: Average results for all evaluation metrics for development and test data. +E: use external embeddings for initialization, +D: use dataset embedding strategy. On the development data, we report the average over the datasets where predictions of all settings were available.

Our system achieved an average morphological tagging F1 score of 94.57 and an average lemma accuracy score of 93.94 on the test data. The experimental analysis showed that:

- Employing a multi-task achitecture having multiple levels of attention mechanism improved the morphological tagging over the baseline strategy.
- Using the pre-trained embeddings substantially improved our scores for both tasks.
- Applying a multi-lingual/dataset strategy by learning special embeddings also improved our scores, specifically for small datasets. On 50 datasets (Table 7), the multi-dataset strategy improved the performance of our model substantially, by 2.95 (accuracy) on lemmatization and 1.81 (F1) on morphological tagging.
- Furthermore, these improvements are highly complementary: using dataset embeddings simultaneously with external embeddings leads to superior performance.

The code to re-run all experiments can be found on: https://bitbucket.org/ahmetustunn/morphology_in_context

Dataset	Lemma		Morph. tags				Dataset	Lemma		Morph. tags			
	Acc.	Lev.	F1	Acc.	+E	+D		Acc.	Lev.	F1	Acc.	+E	+D
af_afribooms	96.44	0.10	98.05	98.45	+	+	it_postwita	95.20	0.18	95.42	96.64	+	-
akk_pisandub	47.52	1.35	76.24	75.84	-	+	it_pud	97.11	0.06	93.73	96.96	-	+
am_att	98.49	0.02	87.81	91.52	-	-	ja_gsd	98.98	0.01	98.00	97.76	+	+
ar_padt	94.65	0.14	94.16	96.90	+	+	ja_modern	96.87	0.04	96.74	96.80	-	+
ar_pud	82.53	0.41	83.61	94.16	+	+	ja_pud	99.01	0.02	98.56	98.39	+	+
be_hse	90.28	0.18	82.20	91.52	+	+	kmr_mg	91.31	0.14	83.51	89.44	+	-
bg_btb	97.19	0.08	97.27	98.79	+	-	ko_gsd	90.09	0.21	95.93	95.35	+	-
bm_crb	87.47	0.21	91.42	93.77	+	-	ko_kaist	94.62	0.09	96.84	96.46	+	-
br_keb	92.24	0.18	86.57	89.50	+	+	ko_pud	99.13	0.01	92.38	95.59	+	+
bxr_bdt	87.12	0.31	83.65	86.57	+	+	kpv_ikdp	85.94	0.26	66.41	75.96	-	+
ca_ancora	99.00	0.02	97.94	99.04	+	-	kpv_lattice	81.87	0.46	69.23	82.21	+	+
cop_scriptorium	96.13	0.08	94.67	96.31	-	-	la_ittb	98.33	0.04	95.01	97.77	+	-
cs_cac	98.39	0.03	95.21	98.36	+	-	la_perseus	92.73	0.15	83.75	93.01	+	+
cs_cltt	97.60	0.29	93.30	97.59	+	+	la_proiel	96.76	0.07	90.28	96.60	+	-
cs_fictree	97.78	0.04	93.84	97.57	+	-	lt_hse	80.14	0.46	67.23	83.26	+	+
cs_pdt	97.94	0.04	94.36	97.97	+	-	lv_lvtb	95.02	0.09	92.96	96.91	+	+
cs_pud	96.84	0.05	91.19	97.21	+	+	mr_ufal	72.63	0.67	62.33	76.02	+	+
cu_proiel	95.54	0.10	88.67	95.48	-	-	nl_alpino	96.25	0.08	95.10	96.05	+	-
da_ddt	96.96	0.05	96.05	97.49	+	+	nl_lassysmall	94.30	0.12	93.45	94.26	-	-
de_gsd	95.24	0.10	84.99	93.71	+	-	no_bokmaal	97.72	0.04	95.21	97.05	+	-
el_gdt	94.64	0.11	92.79	97.47	+	+	no_nynorsk	95.86	0.08	94.05	96.27	-	-
en_ewt	98.39	0.08	96.18	97.24	+	+	no_nynorsklia	97.58	0.04	94.53	96.62	+	+
en_gum	97.85	0.04	95.95	96.95	+	+	pcm_nsc	99.48	0.02	94.79	93.01	+	+
en_lines	97.96	0.04	96.45	97.32	+	-	pl_lfg	97.06	0.06	94.55	97.76	+	+
en_partut	97.97	0.03	95.40	96.27	+	+	pl_sz	97.11	0.05	90.88	96.56	+	+
en_pud	97.20	0.04	95.44	96.85	+	+	pt_bosque	98.24	0.03	94.83	97.53	+	-
es_ancora	99.03	0.02	97.83	98.91	+	-	pt_gsd	98.14	0.10	98.24	98.37	+	-
es_gsd	98.75	0.02	94.60	97.37	-	+	ro_nonstandard	96.44	0.07	92.74	96.18	-	+
et_edt	95.07	0.11	94.51	97.24	+	-	ro_rrt	98.29	0.03	97.47	98.42	+	-
eu_bdt	96.03	0.09	90.15	95.38	+	-	ru_gsd	96.79	0.05	90.69	96.05	+	-
fa_seraji	95.20	0.23	97.76	98.23	+	-	ru_pud	94.31	0.10	87.93	95.50	+	+
fi_ftb	94.65	0.12	95.17	97.37	+	-	ru_syntagrus	96.76	0.07	95.10	97.71	+	-
fi_pud	89.35	0.28	95.24	97.51	+	+	ru_taiga	93.44	0.15	86.33	93.83	+	+
fi_tdt	93.61	0.14	95.31	97.52	+	-	sa_ufal	52.26	1.18	42.21	64.45	+	+
fo_oft	85.59	0.29	80.60	90.62	-	+	sk_snk	95.61	0.08	91.49	96.75	+	-
fr_gsd	98.12	0.04	97.31	98.43	+	-	sl_ssj	97.84	0.03	93.65	97.13	+	-
fr_partut	96.54	0.05	94.96	97.71	+	+	sl_sst	96.24	0.07	90.72	95.09	+	+
fr_sequoia	98.27	0.03	97.18	98.77	+	-	sme_giella	87.54	0.27	86.22	91.38	+	+
fr_spoken	99.52	0.01	98.15	98.18	+	+	sr_set	96.09	0.07	92.38	96.27	+	-
ga_idt	89.07	0.26	83.95	90.82	+	-	sv_lines	96.43	0.08	93.13	97.03	+	-
gl_ctg	98.31	0.03	97.80	97.59	+	-	sv_pud	94.19	0.11	94.97	97.09	+	+
gl_treegal	96.56	0.06	93.97	96.93	+	-	sv_talbanken	96.65	0.07	96.32	98.20	+	-
got_proiel	95.04	0.10	85.99	94.39	-	-	ta_ttb	88.17	0.28	81.14	91.29	+	-
grc_perseus	92.42	0.18	88.90	95.69	+	-	tl_trg	75.68	2.24	86.49	91.30	+	+
grc_proiel	96.70	0.08	91.15	97.37	+	-	tr_imst	96.09	0.07	90.79	95.52	+	+
he_htb	96.61	0.06	95.86	97.35	+	-	tr_pud	86.46	0.34	87.63	94.96	+	+
hi_hdtb	98.61	0.02	91.80	97.30	+	-	uk_iu	95.45	0.09	91.92	96.42	+	-
hr_set	94.18	0.11	89.41	96.02	+	-	ur_udtb	95.91	0.07	77.31	92.02	+	-
hsb_ufal	87.11	0.21	77.12	86.73	+	+	vi_vtb	99.20	0.03	89.55	88.18	-	+
hu_szeged	94.17	0.12	87.95	96.22	+	+	yo_ytb	98.06	0.02	92.64	93.27	-	-
hy_armtdp	92.15	0.15	84.64	91.66	+	+	yue_hk	99.29	0.01	92.32	90.23	-	+
id_gsd	99.09	0.02	89.32	93.04	-	-	zh_cfl	96.57	0.04	91.61	90.35	+	+
it_isdt	97.83	0.04	96.78	98.01	-	-	zh_gsd	99.02	0.01	94.61	94.59	+	+
it_partut	98.25	0.04	97.30	98.45	-	+	average	93.94	0.15	90.61	94.57		

Table 6: All four evaluation metrics for the test data of our best system. E: use of external embeddings. D: use of dataset embeddings. Results might be different compared to the ones in the overview paper, as we did not have enough time to run all experiments before the deadline. +E: whether external embeddings were used. +D: whether dataset embeddings were used.

Acknowledgements

We would like to thank all our colleagues and the anonymous reviewers. Furthermore, we would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Many languages, one parser. *Transactions of the Association for Computational Linguistics*, 4:431–444.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64, Brussels, Belgium. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Costanza Conforti, Matthias Huck, and Alexander Fraser. 2018. Neural morphological tagging of lemma sequences for machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, volume 1, pages 39–53.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, et al. 2018. The conll–sigmorphon 2018 shared task: Universal morphological reinflection. *arXiv preprint arXiv:1810.07125*.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, et al. 2017. Conll-sigmorphon 2017 shared task: Universal morphological reinflection in 52 languages. *arXiv preprint arXiv:1706.09031*.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The sigmorphon 2016 shared taskmorphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22.

Erenay Dayanık, Ekin Akyürek, and Deniz Yuret. 2018. Morphnet: A sequence-to-sequence model that combines morphological analysis and disambiguation. *arXiv preprint arXiv:1805.07946*.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Georg Heigold, Guenter Neumann, and Josef van Genabith. 2017. An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 505–513.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Googles multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Yova Kementchedjhieva, Johannes Bjerva, and Isabelle Augenstein. 2018. Copenhagen at conll–sigmorphon 2018: Multilingual inflection in context with explicit morphosyntactic decoding. *Proceedings of the CoNLL SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 93–98.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Daniel Kondratyuk, Tomáš Gavenčiak, Milan Straka, and Jan Hajič. 2018. Lemmatag: Jointly tagging and lemmatizing for morphologically rich languages with brnns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4921–4928.

Thang Luong, Hieu Pham, and Christopher D Manning. 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.

Chaitanya Malaviya, Shijie Wu, and Ryan Cotterell. 2019. A simple joint model for improved contextual neural lemmatization. *arXiv preprint arXiv:1904.02306*.

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium. Association for Computational Linguistics.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Crosslinguality and context in morphology. In *Proceedings of the 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Florence, Italy. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Rogier Blokland, Victoria Bobicev, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Gülsen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Carly Dickerson, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Tomaž Erjavec, Aline Etienne, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Hüner Kasıkara, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Kamil Kopacewicz, Natalia Kotsyba, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Gustavo Mendonca, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Shinsuke Mori, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Adédayọ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Emily Pitler, Barbara Plank, Thierry Poibeau, Martin Popel, Lauma Pretkalnina, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Michael Rießler, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Muh Shohibussirri, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Yuta Takahashi, Takaaki Tanaka, Isabelle Tellier, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Sowmya Vajjala, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Seyi Williams, Mats Wirén, Tsegay Woldemariam, Tak-sum Wong, Chunxiao Yan, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, Manying Zhang, and Hanzhi Zhu. 2018. Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *arXiv preprint arXiv:1604.05529*.

Aaron Smith, Bernd Bohnet, Miryam de Lhoneux, Joakim Nivre, Yan Shao, and Sara Stymne. 2018. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 113–123, Brussels, Belgium. Association for Computational Linguistics.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Shijie Wu, Pamela Shapiro, and Ryan Cotterell. 2018. Hard non-monotonic attention for character-level transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4425–4438, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

A Matching of Datasets

Src Data	Additional	Emb. type	Src Data	Additional	Emb. type
af_afribooms	nl_alpino	poly	it_postwita	it_isdt	elmo
akk_pisandub	cs_pdt	elmo	it_pud	it_isdt	elmo
am_att			ja_gsd	ja_pud	elmo
ar_padt	ar_pud	elmo	ja_modern	ja_gsd	elmo
ar_pud	ar_padt	elmo	ja_pud	ja_gsd	elmo
be_hse	ru_syntagrus	poly	kmr_mg	es_gsd	poly
bg_btb	ru_syntagrus	elmo	ko_gsd	ko_kaist	elmo
bm_crb	cs_pdt	fast	ko_kaist	ko_gsd	elmo
br_keb	no_bokmaal	poly	ko_pud	ko_kaist	elmo
bxr_bdt	ru_syntagrus	fast	kpv_ikdp	ru_syntagrus	fast
ca_ancora	es_ancora	elmo	kpv_lattice	ru_syntagrus	fast
cop_scriptorium			la_ittb	la_proiel	elmo
cs_cac	cs_pdt	elmo	la_perseus	la_proiel	elmo
cs_cltt	cs_pdt	elmo	la_proiel	la_ittb	elmo
cs_fictree	cs_pdt	elmo	lt_hse	lv_lvtb	poly
cs_pdt	cs_cac	elmo	lv_lvtb	hr_set	elmo
cs_pud	cs_pdt	elmo	mr_ufal	hi_hdtb	poly
cu_proiel	ru_syntagrus	elmo	nl_alpino	nl_lassysmall	elmo
da_ddt	no_bokmaal	elmo	nl_lassysmall	nl_alpino	elmo
de_gsd	fr_gsd	elmo	no_bokmaal	no_nynorsk	elmo
el_gdt	grc_proiel	elmo	no_nynorsk	no_bokmaal	elmo
en_ewt	en_gum	elmo	no_nynorsklia	no_nynorsk	elmo
en_gum	en_ewt	elmo	pcm_nsc	en_ewt	elmo
en_lines	en_ewt	elmo	pl_lfg	pl_sz	elmo
en_partut	en_ewt	elmo	pl_sz	pl_lfg	elmo
en_pud	en_ewt	elmo	pt_bosque	pt_gsd	elmo
es_ancora	es_gsd	elmo	pt_gsd	pt_bosque	elmo
es_gsd	es_ancora	elmo	ro_nonstandard	ro_rrt	elmo
et_edt	cs_pdt	elmo	ro_rrt	ro_nonstandard	elmo
eu_bdt	es_ancora	elmo	ru_gsd	ru_syntagrus	elmo
fa_seraji	ur_udtb	elmo	ru_pud	ru_syntagrus	elmo
fi_ftb	fi_tdt	elmo	ru_syntagrus	ru_gsd	elmo
fi_pud	fi_tdt	elmo	ru_taiga	ru_syntagrus	elmo
fi_tdt	fi_ftb	elmo	sa_ufal	hi_hdtb	poly
fo_oft	no_nynorsk	poly	sk_snk	cs_pdt	elmo
fr_gsd	fr_sequoia	elmo	sl_ssj	hr_set	elmo
fr_partut	fr_gsd	elmo	sl_sst	sl_ssj	elmo
fr_sequoia	fr_gsd	elmo	sme_giella	no_nynorsk	poly
fr_spoken	fr_gsd	elmo	sr_set	hr_set	poly
ga_idt	cs_pdt	elmo	sv_lines	sv_talbanken	elmo
gl_ctg	es_ancora	elmo	sv_pud	sv_talbanken	elmo
gl_treegal	gl_ctg	elmo	sv_talbanken	sv_lines	elmo
got_proiel	no_nynorsk	none	ta_ttb		
grc_perseus	grc_proiel	elmo	tl_trg	es_gsd	poly
grc_proiel	grc_perseus	elmo	tr_imst	tr_pud	elmo
he_htb	ru_gsd	elmo	tr_pud	tr_imst	elmo
hi_hdtb	mr_ufal	poly	uk_iu	ru_syntagrus	elmo
hr_set	sr_set	poly	ur_udtb	fa_seraji	elmo
hsb_ufal	cs_pdt	poly	vi_vtb	en_ewt	elmo
hu_szeged	et_edt	elmo	yo_ytb	es_gsd	poly
hy_armtdp	ru_pud	poly	yue_hk	zh_gsd	poly
id_gsd	es_gsd	elmo	zh_cfl	zh_gsd	elmo
it_isdt	it_partut	elmo	zh_gsd	ja_gsd	elmo
it_partut	it_isdt	elmo			

Table 8: This shows for each dataset, with which dataset it has the highest word overlap, and what their best common embeddings type is. Three datasets could not be paired, as they had 0% overlap with all other datasets (ignoring punctuation and numericals).

B External Embeddings per Dataset

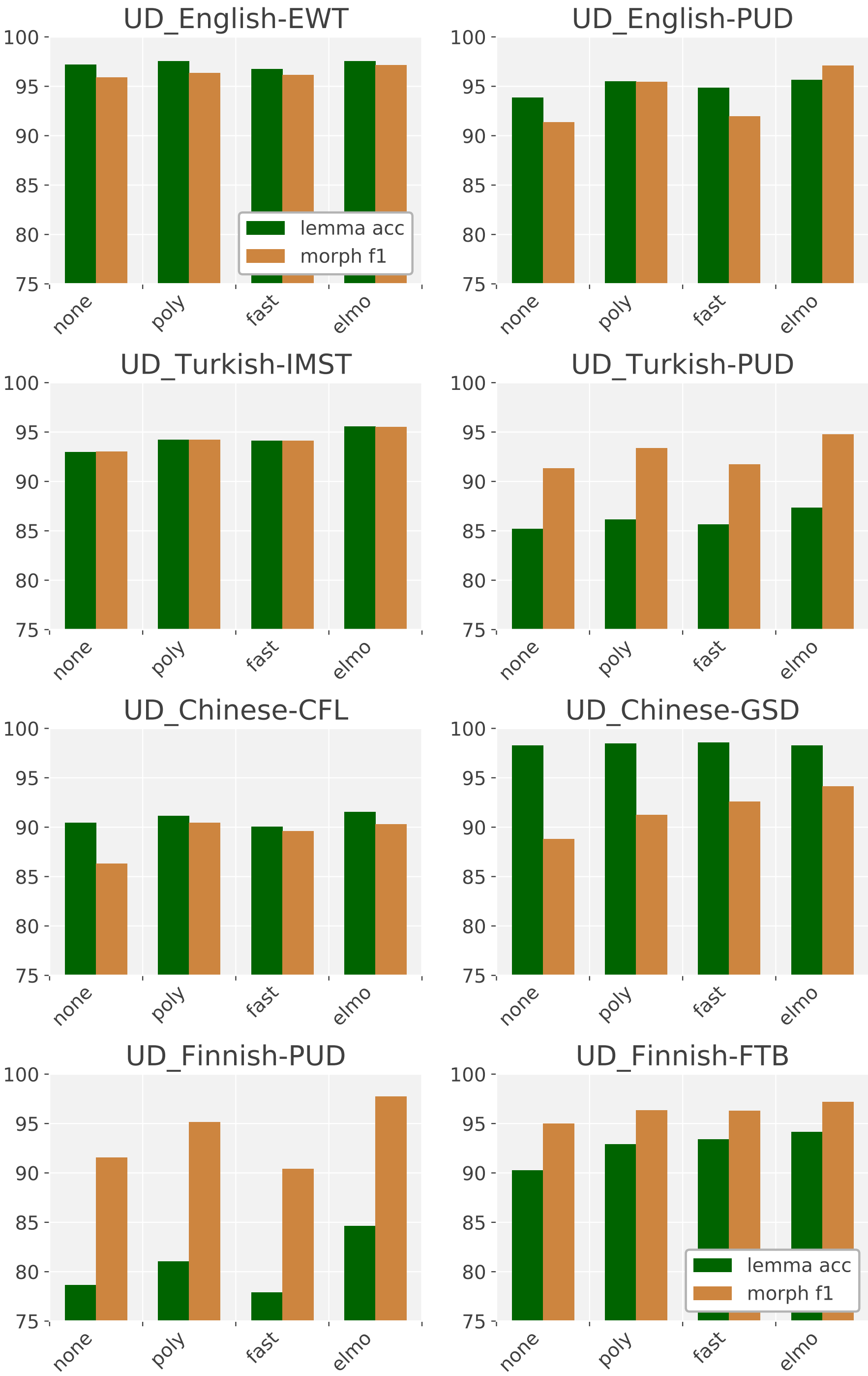

Figure 6: Results of different types of embeddings on the development splits of our tune datasets.

C Dataset Embeddings per Dataset

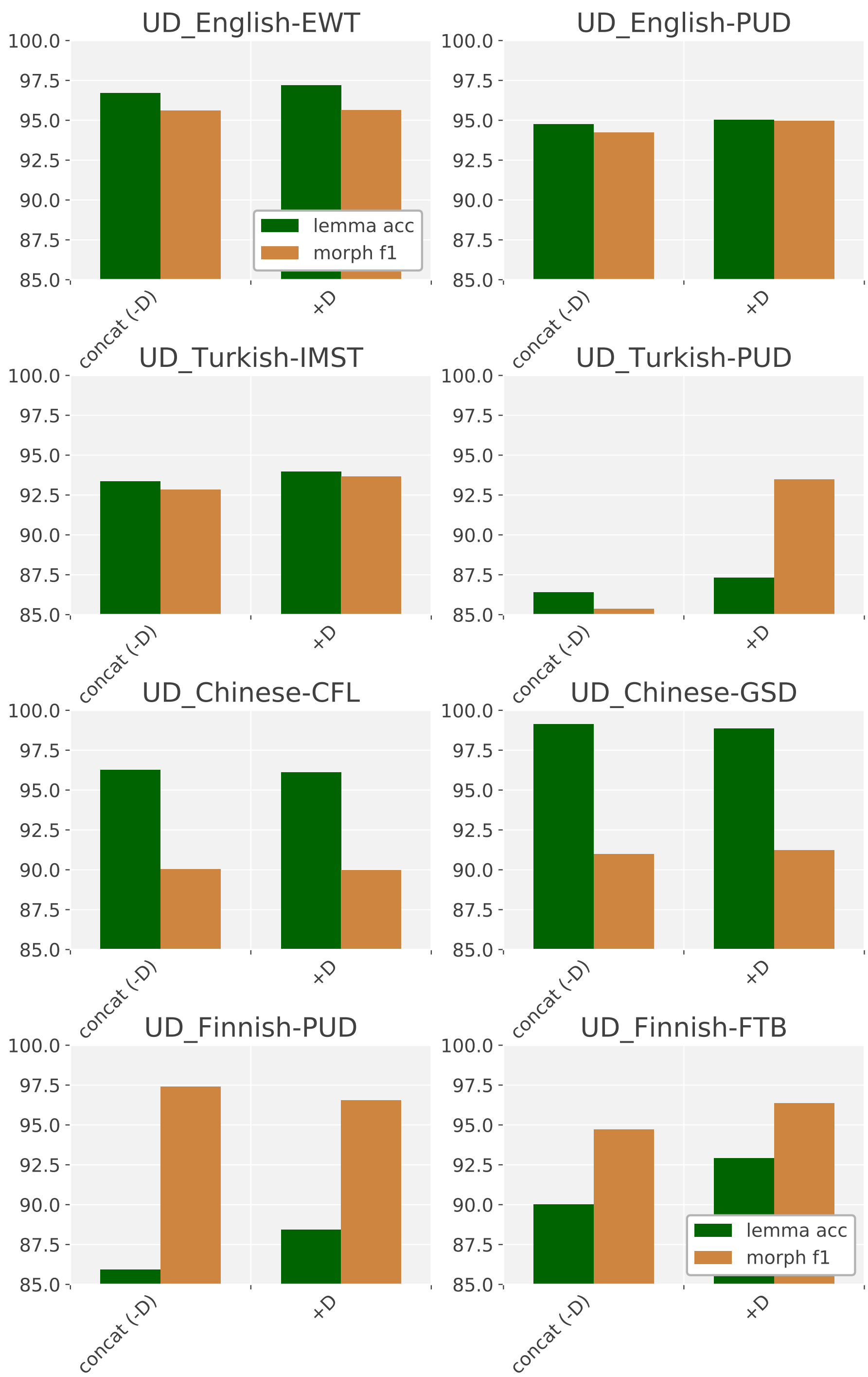

Figure 7: Results of dataset embeddings on the development splits of our tune datasets. We compare the dataset embeddings with a simple concatenation of the datasets.

D Results of External and Treebank Embeddings on Development Data

Dataset	Base	-E-D	-E+D	+E-D	+E+D	Lem	Mor	Dataset	Base	-E-D	-E+D	+E-D	+E+D	Lem	Mor
af_afribooms	95.48	95.14	95.95	96.63	**97.03**	96.49	97.57	it_postwita	92.19	95.12	0.00	**96.19**	0.00	95.24	97.14
akk_pisandub	74.76	63.65	**64.76**	59.64	63.07	46.67	82.85	it_pud	94.14	94.06	**97.02**	90.29	96.80	97.25	96.79
am_att	93.53	**95.51**	0.00	93.77	0.00	98.90	92.12	ja_gsd	93.98	96.96	97.27	98.08	**98.21**	98.91	97.52
ar_padt	93.40	93.74	90.05	93.93	**95.92**	94.90	96.94	ja_modern	94.26	94.61	**96.55**	94.19	96.47	96.45	96.65
ar_pud	86.58	84.28	76.56	88.02	**88.65**	82.70	94.60	ja_pud	92.91	95.16	98.34	97.02	**99.02**	99.46	98.59
be_hse	84.82	80.69	88.10	81.32	**89.13**	87.48	90.78	kmr_mg	89.17	87.36	0.00	**87.81**	0.00	88.57	87.06
bg_btb	95.57	97.43	0.00	**97.97**	0.00	97.15	98.79	ko_gsd	89.38	91.49	0.00	**92.82**	0.00	90.45	95.19
bm_crb	88.86	91.34	0.00	**91.50**	0.00	88.70	94.30	ko_kaist	91.99	94.89	0.00	**95.57**	0.00	94.69	96.45
br_keb	91.00	86.51	0.00	89.14	**91.39**	91.49	91.29	ko_pud	93.89	94.02	96.57	96.32	**97.55**	98.80	96.30
bxr_bdt	83.66	83.46	86.34	84.33	**86.36**	86.83	85.89	kpv_ikdp	67.44	61.19	**73.16**	60.82	72.83	71.08	75.24
ca_ancora	96.91	98.43	0.00	**99.08**	0.00	99.11	99.06	kpv_lattice	75.14	63.65	74.16	62.89	**76.29**	78.57	74.01
cop_scriptorium	94.53	**95.53**	0.00	94.54	0.00	95.12	95.94	la_ittb	95.85	97.89	0.00	**98.17**	0.00	98.39	97.94
cs_cac	95.86	97.31	0.00	**98.30**	0.00	98.34	98.26	la_perseus	83.11	82.97	89.34	87.26	**91.25**	90.80	91.70
cs_cltt	95.53	94.32	97.20	93.19	**97.67**	97.82	97.51	la_proiel	94.03	95.05	0.00	**96.74**	0.00	96.88	96.59
cs_fictree	94.10	96.09	0.00	**97.85**	0.00	97.99	97.72	lt_hse	76.09	70.49	74.93	76.93	**81.48**	80.69	82.27
cs_pdt	95.26	96.96	0.00	**98.00**	0.00	98.02	97.98	lv_lvtb	92.38	93.88	0.00	95.67	**95.70**	94.73	96.67
cs_pud	89.85	88.22	96.04	94.17	**96.94**	97.05	96.83	mr_ufal	76.22	74.79	76.64	74.80	**77.87**	76.71	79.02
cu_proiel	93.47	**94.94**	0.00	94.78	0.00	95.00	94.87	nl_alpino	94.25	95.77	0.00	**96.17**	0.00	96.35	95.99
da_ddt	93.74	91.53	95.66	97.19	**97.35**	97.01	97.68	nl_lassysmall	92.62	**95.05**	0.00	94.19	0.00	95.16	94.95
de_gsd	0.00	93.59	94.35	**94.44**	0.00	95.23	93.64	no_bokmaal	95.53	97.45	0.00	**97.62**	0.00	97.91	97.32
el_gdt	95.23	95.83	95.36	95.76	**96.00**	94.50	97.50	no_nynorsk	0.00	**97.23**	0.00	96.26	0.00	97.34	97.13
en_ewt	93.99	94.49	96.41	97.66	**97.78**	98.26	97.29	no_nynorsklia	91.80	95.52	95.25	95.61	**96.99**	97.79	96.18
en_gum	93.74	92.04	96.04	93.90	**97.53**	97.80	97.26	pcm_nsc	89.24	95.87	96.10	95.86	**96.17**	100.00	92.35
en_lines	94.61	96.24	0.00	**97.60**	0.00	98.01	97.20	pl_lfg	92.07	95.56	95.66	96.82	**97.43**	97.03	97.84
en_partut	93.33	95.15	96.05	96.11	**97.24**	98.24	96.25	pl_sz	91.10	93.78	95.00	96.15	**97.09**	97.37	96.80
en_pud	91.62	92.70	95.00	96.23	**97.07**	97.20	96.94	pt_bosque	94.88	97.08	0.00	**97.91**	0.00	98.32	97.50
es_ancora	96.87	98.27	98.38	**98.89**	0.00	98.96	98.83	pt_gsd	0.00	97.44	0.00	**98.14**	0.00	97.94	98.35
es_gsd	0.00	97.62	**98.14**	98.13	0.00	98.68	97.59	ro_nonstandard	93.62	95.73	**96.20**	96.11	0.00	96.32	96.09
et_edt	93.31	94.50	0.00	**96.20**	0.00	94.99	97.42	ro_rrt	95.56	97.39	97.46	**98.20**	0.00	98.23	98.16
eu_bdt	91.94	94.76	0.00	**95.80**	0.00	96.03	95.57	ru_gsd	55.99	94.85	0.00	**96.93**	0.00	97.10	96.75
fa_seraji	0.00	95.90	0.00	**96.61**	0.00	95.27	97.96	ru_pud	89.25	88.21	94.70	94.07	**95.94**	95.06	96.82
fi_ftb	92.27	94.23	94.65	**96.13**	0.00	94.81	97.45	ru_syntagrus	94.37	96.66	0.00	**97.25**	0.00	96.75	97.75
fi_pud	88.69	86.82	92.50	90.42	**93.12**	87.97	98.27	ru_taiga	85.09	85.38	92.94	91.17	**94.70**	94.28	95.13
fi_tdt	89.32	94.30	94.31	**95.56**	0.00	93.43	97.70	sa_ufal	68.45	61.87	64.32	62.96	**67.69**	63.92	71.46
fo_oft	88.93	88.18	**89.65**	87.98	89.28	88.11	91.19	sk_snk	92.44	93.12	0.00	**96.52**	0.00	96.20	96.85
fr_gsd	96.43	97.93	97.73	**98.30**	0.00	98.13	98.47	sl_ssj	92.97	95.59	0.00	**97.35**	0.00	97.60	97.09
fr_partut	94.09	94.62	96.89	96.54	**97.47**	97.10	97.83	sl_sst	89.78	89.19	94.26	93.20	**95.98**	96.89	95.06
fr_sequoia	95.59	96.64	97.98	**98.49**	0.00	98.43	98.54	sme_giella	89.69	89.60	87.56	88.27	**90.44**	88.32	92.56
fr_spoken	96.34	96.71	97.92	97.94	**98.63**	99.41	97.84	sr_set	93.82	95.64	0.00	**95.91**	0.00	95.84	95.98
ga_idt	86.92	84.13	0.00	**90.14**	0.00	89.15	91.13	sv_lines	93.54	93.97	95.07	**96.98**	0.00	96.90	97.07
gl_ctg	95.09	97.48	97.97	**98.15**	0.00	98.38	97.92	sv_pud	91.08	89.84	93.61	95.03	**95.68**	94.48	96.88
gl_treegal	92.77	93.38	95.64	**96.29**	94.91	96.03	96.55	sv_talbanken	0.00	96.13	95.95	**97.61**	0.00	96.85	98.38
got_proiel	94.19	**95.40**	0.00	95.00	0.00	95.63	95.17	ta_ttb	93.31	89.73	0.00	**91.46**	0.00	91.15	91.76
grc_perseus	0.00	93.48	0.00	**94.02**	0.00	92.46	95.57	tl_trg	68.66	73.36	70.36	69.13	**78.62**	76.00	81.25
grc_proiel	0.00	95.73	0.00	**97.06**	0.00	96.72	97.41	tr_imst	90.73	93.46	93.81	95.40	**95.43**	95.50	95.35
he_htb	94.31	95.82	96.55	**96.87**	0.00	96.47	97.27	tr_pud	87.86	88.63	90.39	90.83	**91.74**	87.96	95.52
hi_hdtb	96.36	97.43	97.55	**97.93**	97.71	98.53	97.32	uk_iu	91.39	92.41	94.06	**95.75**	0.00	95.24	96.25
hr_set	93.21	93.27	0.00	**95.25**	0.00	94.20	96.30	ur_udtb	92.12	92.85	0.00	**93.59**	0.00	95.61	91.57
hsb_ufal	84.64	82.83	82.98	84.67	**84.79**	86.28	83.31	vi_vtb	89.39	93.37	**94.48**	94.13	0.00	99.40	89.56
hu_szeged	91.06	91.65	91.03	90.94	**94.70**	93.42	95.98	yo_ytb	88.72	**92.41**	91.38	89.07	0.00	94.40	90.42
hy_armtdp	0.00	92.02	0.00	92.69	**93.08**	93.16	93.01	yue_hk	85.19	90.97	**94.10**	89.32	93.94	98.97	89.22
id_gsd	92.75	**96.05**	0.00	95.89	0.00	99.08	93.03	zh_cfl	85.31	89.34	93.05	91.33	**93.86**	96.26	91.46
it_isdt	95.73	**97.83**	97.62	97.12	97.80	97.63	98.02	zh_gsd	91.34	94.15	95.04	96.57	**96.79**	99.06	94.52
it_partut	95.36	95.59	**98.11**	97.84	97.73	97.85	98.37								

Table 9: Results on all development datasets. The average of lemma accuracy and morphological F1 score is used as main metric. base: baseline. E: external embeddings. D: dataset embeddings. Bold indicates which model is used on the test data. Lem: lemma accuracy of the bold model. Mor: morphologic tagging F1 score of bold model. A score of 0.00 means that we did not have time to run the model for this setting.