Language-Independent Approach for Morphological Disambiguation

Alymzhan Toleu and **Gulmira Tolegen** and **Rustam Mussabayev**
Institute of Information and Computational Technologies, 050010, 28 Shevchenko str., Almaty, Kazakhstan
alymzhan.toleu@gmail.com and gulmira.tolegen.cs@gmail.com
and rmusab@gmail.com

Abstract

This paper presents a language-independent approach for morphological disambiguation which has been regarded as extensions of POS tagging, jointly predicting complex morphological tags. In the proposed approach, all words, roots, POS and morpheme tags are embedded into vectors, and contexts representations from surface word and morphological contexts are calculated. Then the inner products between analyses and the context's representations are computed to perform the disambiguation. The underlying hypothesis is that the correct morphological analysis should be closer to the context in a vector space. Experimental results show that the proposed approach outperforms the existing models on seven different language datasets. Concretely, compared with the baselines of MarMot and a sophisticated neural model (Seq2Seq), the proposed approach achieves around 6% improvement in average accuracy for all languages while running about 6 and 33 times faster than MarMot and Seq2Seq, respectively.

1 Introduction

Morphological disambiguation (MD) is the task of jointly predicting lemma/root, part of speech (POS)(Toleu et al., 2020), and morpheme tags. For a Turkish word "yeni" (new), it can be analyzed as: 1)yen+Noun+[A3sg, Pnon, Acc]; 2)yen+Noun+[A3sg, P3sg, Nom]. If one counts analyses as tags, MD can be cast as a tagging problem with an extremely large tagset. This fact discourages direct application of the state of the art approaches designed for small fixed tagsets.

For instance, many approaches treat each analysis as a tag, and apply sequence labeling models to perform tagging (Mueller et al., 2013; Müller and Schütze, 2015; Malaviya et al., 2018). Treating each analysis as a tag leads to an oversized tagset and corresponding data-sparsity issues, which can be a concern for morphologically complex languages such as Turkish and Kazakh, where the number of morphological analyses is theoretically unlimited (Yuret and Türe, 2006). To address this problem, a sequence to sequence (Seq2Seq)(Tkachenko and Sirts, 2018) based approach was proposed, which treated each morphological analysis as a sequence of a composite tags and explicitly modeled their internal structure. This approach was inspired by the neural sequence-to-sequence models for machine translation (Cho et al., 2014). The LSTM networks were applied to model morphological analyses and context as a pair of sequences, which involves more sophisticated architectures, namely using double layers of biLSTM, one for characters and another for words. This approach was almost challenging to simplify the architecture if one wanted to keep the performance as the original Seq2Seq has. Because all types of architecture of recurrent neural networks are more fit to the nature of sequence to sequence problems (Sutskever et al., 2014).

This paper presents a language-independent MD approach that applies an uncomplicated neural architecture and obtains comparable results in accuracy and speed with the current best. A sequence of morphological analysis is an expansion sequence of its surface words with morphological information. The idea of the approach is to measure the distance between analyses and surface word context by embedding each morphological analysis and the surface word context into a single vector space. The underlying hypothesis is that the vector representation of the correct analysis should be closer to the context vector. Two types of contextual embedding for words are presented: i) surface word context; ii) morphological context, which improves the model's performance significantly. In the following, the proposed approach is referred to as language-independent morphological disambiguation (LIMD).

Our contribution amounts to the following: i) a

Proceedings of the 29th International Conference on Computational Linguistics, pages 5288–5297
October 12–17, 2022.

general language-independent approach for MD, its neural architecture is simple, decoding is fast and can be implemented easily in practice. It achieves comparable results with the current best. ii) two types of context representation are explored: word and morphological context representations.

2 Related Work

Morphological disambiguation/tagging has been studied extensively for decades, and here we review the work most relevant to this paper. We categorize the common approaches into three groups:

i) Modeling the structure of complex morphological labels with structured prediction models (Mueller et al., 2013; Müller and Schütze, 2015; Malaviya et al., 2018). The work (Mueller et al., 2013) presented a pruned CRF (PCRF) for tagging and proposed to use coarse-to-fine decoding and early updating to train the higher-order CRF. Experiments on six languages show that the PCRF gives significant improvements in accuracy. We evaluate this model on our data-sets as one of the baselines. (Müller and Schütze, 2015) compared the performance of the most important representations that can be used for across-domain MT. One of their findings is that the representations similar to Brown clusters perform best for POS tagging and that word representations based on linguistic morphological analyzers perform best for tagging. The study (Malaviya et al., 2018) combines neural networks and graphical models presented a framework for cross-lingual tagging. Instead of predicting full tag sets, the model predicts single tags separately and modeling the dependencies between tags over time steps. The model is able to generate tag sets unseen in training data, and share information between similar tag sets. This model is about cross-lingual tagging and we do not make comparisons with monolingual tagging models.

ii) Modeling complex morphological labels as sequences of morphological feature values through neural networks (NN) (Tkachenko and Sirts, 2018) and statistical approaches (Hakkani-Tur et al., 2000; Schmid and Laws, 2008). The work (Tkachenko and Sirts, 2018) presented a sequence to sequence model for tagging. The model learns the internal structure of morphological labels by treating them as sequences of morphological feature values and applies a similar strategy of neural sequence-to-sequence models commonly used for machine translation (Sutskever et al., 2014) to do tagging. The authors explored different neural architectures and compare their performance with PCRF (Mueller et al., 2013). Double layer of biLSTMs were applied in those neural architectures as Encoder (Ling et al., 2015; Labeau et al., 2015; Ma and Hovy, 2016). The encoder uses one biLSTM to compute character embedding and the second biLSTM combine the obtained character embedding along with pre-trained word embedding to generate word context embeddings. The output of those neural networks are different: one of the baselines is to use a single output layer to predict whole morphological labels. As the second baseline, the output layer can be changed to predict the different morphological value of tag with multi output layers. An improved version of the second one is to use a hierarchical multi output layers in order to capture dependencies between tags.

iii) Modeling the output of morphological analyzer as candidates then use the different classifiers to do disambiguation (Hakkani-Tur et al., 2000; Zalmout and Habash, 2017; Toleu et al., 2017). The work (Zalmout and Habash, 2017) presented an improved tagging system for Arabic by using the results of biLSTM output from words and characters and a character-aware MD model(Toleu et al., 2017) was proposed for Kazakh and Turkish. A voted-perceptron approach for Kazakh MD was proposed in the work (Tolegen et al., 2020), and explored many features impact on MD.

3 Approach

This section describes the proposed MD approach, which embeds a context and its morphological analysis into a vector space, then calculates similarity scores to rank them for performing disambiguation.

3.1 Notation

Given a sentence $(w_1, ...a_{1j}), ..., (w_n, ...a_{nj})$ consisting of n words with all possible morphological analysis a_{ij} of each word w_i, we want to predict the sequence $a_1*, ..., a_n*$ of morphological analysis which best fit to the context of the given sentence. $j \in N_i$ is the index of analyses for a word w_i. We treat a morphological analysis a_{ij} as a combination of three main constituents: root r_j, POS p_j and morpheme chain m_j. A morpheme chain m_j consists of several morphological tags, each of tags is denoted as t_{jk}, means the k-th tag in morpheme chain m_j. Vector representations of a context and a morphological analysis a_{ij} are denoted as $\mathbf{S_i}$ and

$\mathbf{M_{ij}}$, respectively. $[... \circ ... \circ ...]$ concatenation operation of inside vectors.

3.2 Morphological Embedding

For the j-th analysis a_{ij} of given word w_i, we embed its root r_j, POS p_j and morpheme tags m_j into dense vector representation. In order to handle the various length of morpheme tags, we define a value $maxT$ as the largest length of morpheme tags in the dataset. Then a vector representation for a analysis is calculated as follows[1]:

$$\mathbf{M_{ij}} = \sigma(\mathbf{W_a} * [\mathbf{r_j} \circ \mathbf{p_j} \circ \mathbf{m_j}]) \quad (1)$$

where $\mathbf{M_{ij}} \in R^{d_h \times 1}$ is a vector representation of i-th word's j-th morphological analysis. $[\mathbf{r_j} \circ \mathbf{p_j} \circ \mathbf{m_j}] \in R^{(d_r+d_p+maxT*d_m)\times 1}$ is the concatenation of corresponding vectors of root, POS and morpheme tags. $\mathbf{W_a} \in R^{d_h \times (d_r+d_p+maxT*d_m)}$ is the model parameter. d_r, d_p, d_m is the dimension of root, POS and each morpheme tag embeddings respectively. σ is a activation function. The bias term was left out for clarity. Representation for all N_i analyses of i-th word is denoted as $\mathbf{M_i} \in R^{d_h \times N_i}$

3.3 Contextual Embedding

A sentence is a sequence of surface words; its corresponding series of morphological analyses could be considered its expansion with morphological information. Two sequences are dissimilar in their formation but are similar in the language meaning. The former is made of a series of surface words, and the latter is composed of morphological analyses with certain ambiguities that depend on the context. This subsection introduce two context representations, and describe how to obtain vector representations for them:surface word context and averaged morphological context.

Surface word context. For a sentence $w_1, ..., w_n$, consider its contextual information, we want to compute surface word context representation to each word. With the purpose of simplifying the model architecture, we choose a window-based feed forward neural network as the encoder. The encoder takes a window of words and embed them to vector representation by one linear and non-linear layers:

$$\mathbf{C_i} = [\mathbf{w_{i-d_{win/2}}} \circ \mathbf{w_i} \circ \mathbf{w_{i+d_{win/2}}}] \quad (2)$$

where d_{win} is the window size and $[... \circ \mathbf{w_i} \circ ...] \in R^{(d_{win}*d_w)\times 1}$ is the concatenation of word embeddings. Here, to simplify the model architecture, we did not apply a non-linear layer to generate surface word context. It will be integrated with averaged morphological context embeddings to capture the interaction between pairs of sequences.

Averaged morphological context representation A sequence of morphological analyses is another ambiguous realization (each word has several analyses) of word series. Regardless of the ambiguities, we can compute averaged vector representations to the morphological context and apply them to handle better the dependencies issue among morpheme tags and the dependencies among analyses located in different positions of the sentence. Here, we expect the averaged morphological context to impact MD positively and will conduct corresponding experiments to find it out.

More formally, instead of only using surface word for a current word w_i, we can use the information from the previous $i \in (i-win, ..., i-1]$ words' morphological analyses as well as the next $i \in [i+1, ..., i+win)$ words' analyses. Because there are large dependencies in the morphological tags. Morphological context $\mathbf{C_{pre}} \in R^{(d_r+d_p+maxT*d_m)\times 1}$ and $\mathbf{C_{next}} \in R^{(d_r+d_p+maxT*d_m)\times 1}$ are defined by:

$$\mathbf{C_{pre}} = \sum_i \frac{1}{N_i} \sum_{j=1}^{N_i} [\mathbf{r_j} \circ \mathbf{p_j} \circ \mathbf{m_j}] \quad (3)$$

where N_i is the number of morphological analyses that i-th word has. win is the window size for a morphological context. Similar calculation goes for right side morphological context $\mathbf{C_{next}}$. The final morphological context is obtained by averaging the embedding of all analyses for the corresponding side. After obtaining all context vectors, the final vector representation for the context is calculated by concatenating three (surface, left side, and right side morphological context) and then going through a non-linear layer to extract interactive features between these contexts.

$$\mathbf{S_i} = \sigma(\mathbf{W_c} * [\mathbf{C_i} \circ \mathbf{C_{pre}} \circ \mathbf{C_{next}}]) \quad (4)$$

Where $\mathbf{W_c} \in R^{d_h \times (d_{win}*d_w+2*(d_r+d_p+maxT*d_m))}$ is the model parameter.

[1] only consider the the presence of each tag in a morpheme chain. If the number of tags in a chain less than $maxT$, after looking-up the existing tags, the remaining positions fill with zero vector.

3.4 Disambiguation

For disambiguation, we score each analysis by computing the inner product between analyses and the context's representations:

$$a_i^* = argmax(softmax(\mathbf{M_i^T} \odot \mathbf{S_i})) \quad (5)$$

where a_i^* denotes the most probable analyses for a word w_i in a context. The underlying hypothesis is that the embedding of the probable morphological analysis should be most similar to the context. The training procedure of the proposed method is given in algorithm 1.

4 Experiments

4.1 Datasets

We run experiments on Arabic-PADT (ar) (Hajič et al., 2009), Czech-PDT (cs)(Bejček et al., 2013), Spanish-AnCora (es) (Taulé et al., 2008), German-GSD (de)(McDonald et al., 2013), Russian-SynTagRus (ru) (Droganova et al., 2018), Turkish-IMST (tr)(Sulubacak et al., 2016) and Kazakh-KTB (kk)(Tyers and Washington, 2015) from Universal Dependencies version 2.3[2]. We use default data splits except for Kazakh because the default training set is significantly less than test set, we put the larger set as the training set and the less one for the test set. We tested the proposed language-independent approach on various types of language: Arabic is a Semitic language with nonconcatenative morphology. We used default Arabic script without any pre-processing. Czech and Russian are highly inflecting Slavic languages. Spanish and German belong to Romance and Germanic language groups, respectively. Kazakh and Turkish are agglutinative languages. Table 1 shows statistics of the corpora. As given, German has large ambiguous data in terms of analyses per word, it has 6.06 analyses per word on average and the maximum number of analyses reach to 51 for some certain words. It should be noted that average analyses per word are calculated based on all tokens (total number of analyses of all tokens divided by the total number of all tokens) not based on all unique tokens.

Figure 1 shows the percentage information about the number of analyses in the corpora. It can be seen that for Arabic and Russian, $20\% \sim 30\%$ tokens have two analyses and the remaining portions of tokens have analyses in the range of [3,11]. Czech and German have long-tailed distributions

[2]https://universaldependencies.org

Algorithm 1: The training and prediction process of the proposed method.

Input: $(w_1, ...a_{1j}), ..., (w_n, ...a_{nj})$, a sentence with its all possible morphological analyses of each word.
Output: $a_1^*, ..., a_n^*$, a sequence of correct morphological analysis.
Parameter : θ, the set of the model parameters.
for $epoch \leftarrow 1$ **to** $totalEpoch$ **do**
  for $i \leftarrow 1$ **to** n **do**
    if $N_i > 1$ **then**
      $\mathbf{C_{pre}}$ and $\mathbf{C_{next}} \leftarrow$ use equation (3) to calculate morphological context embedding.
      $\mathbf{S_i} \leftarrow$ use equation (4) to compute contextual embedding.
      Define a matrix $\mathbf{M_i} \in R^{d_h \times N_i}$.
      for $j \leftarrow 1$ **to** N_i **do**
        $\mathbf{M_{ij}} \leftarrow$ use equation (1) to calculate j-th morphological embedding for i-th word.
      end
      $a_i^* = argmax(softmax(\mathbf{M_i^T} \odot \mathbf{S_i}))$
      if $a_i^* \neq$ *the correct analysis* **then**
        $\theta^* \leftarrow$ use back-propagation to compute the gradient of the corresponding object function with respect to the model parameters.
        $\theta \leftarrow \theta + \eta\theta^*$ update parameters.
      end
    end
    else
      if i-word has only one analysis, then treat it as the correct analysis.
    end
  end
  if $epoch > totalEpoch$ *or reach the expected accuracy* **then**
    stop training;
  end
  epoch ++;
end

Table 1: Corpora statistics. *avg.* denotes the average number of analyses per word. *max.* is the maximum number. *ambig. rate* denotes the percentage of the ambiguous tokens (the words have more than one analysis).

Lang.	Training Set				Test Set			
	tok.	label per word		ambig. rate (%)	tok.	label per word		ambig. rate (%)
		avg.	max.			avg.	max.	
ar	254340	2.69	12	64.88	32128	2.71	12	66.19
cs	1175374	2.65	25	48.18	174252	2.67	25	49.16
es	446145	2.80	11	53.69	52801	2.81	11	65.58
de	268414	6.06	51	62.54	16772	5.89	51	70.66
ru	871521	1.88	15	41.34	117523	1.86	15	40.60
tr	38871	1.24	5	17.40	10193	1.24	5	16.79
kk	10063	1.27	5	19.06	547	1.32	5	21.38

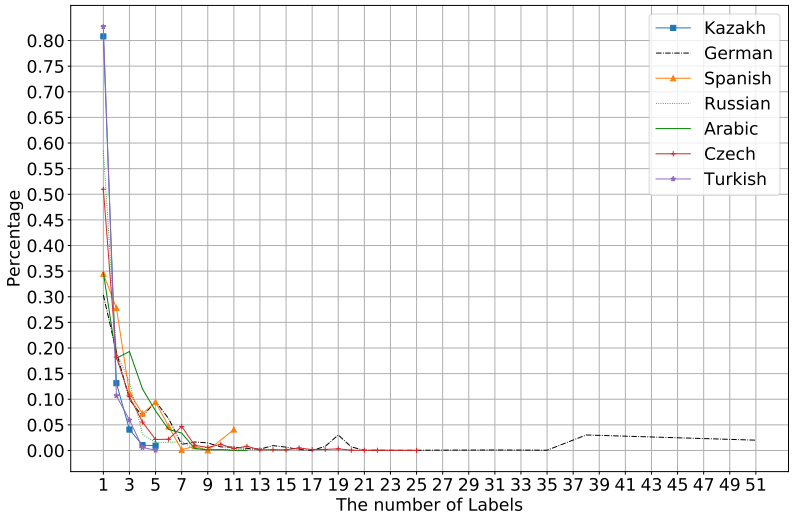

Figure 1: Distribution about the number of per word analyses. *x-axis* is the number of analyses and *y-axis* is the percentage of those analyses number in the corpus.

in the number of analyses. Kazakh and Turkish have similar distributions, and the large portion $50\% \sim 80\%$ of their analyses number are in the range [1,3).

4.2 Baselines

We use two models as baselines, the CRF-based MarMoT (Mueller et al., 2013) and Seq2Seq-based model (Tkachenko and Sirts, 2018): i) MarMoT[3] is the pruned CRF (PCRF)-based morphological tagger which has been shown to achieve competitive performance across several languages. The model is based on coarse-to-fine decoding, which means that the model first predicts POS and based on that, constrains the morphological tags. We train the second-order of MarMot following the result of (Mueller et al., 2013). ii) Seq2Seq[4] is a recent new sophisticated neural model, which is inspired from neural seq2seq models commonly used for machine translation. Encoder models the context of each word and decoder predicts morphological tags in a analysis as a sequence of its category value. Seq2Seq was trained with same hyper-parameters reported in (Tkachenko and Sirts, 2018).

4.3 Model Setup

It can be seen from Table 1 and Figure 1 that German has the most ambiguous test set, we optimize the hyper-parameters of LIMD on the German development set and apply the resulting values to other languages. We set the embedding of d_r, d_p, d_m to 35; the hidden layer size d_h is 100; the word context size is set to $d_{win} = 7$ and morphological context uses leftmost and rightmost word analyses. To compare the decoding times we run all experiments on the same test environment: In-

[3]http://cistern.cis.lmu.de/marmot/
[4]https://github.com/AleksTk/seq-morph-tagger

tel Core i7-8700 CPU with 6 cores and 16 GB of memory.

5 Results

Table 2 presents the experimental results. We report accuracy of Part-of-speech (POS), Morpheme and POS+Morpheme for all tokens. POS+Morpheme indicates that both POS and all morphological tags are correctly predicted. It can be seen from Table 2, LIMD performs comparable with MarMot (Mueller et al., 2013) and the seq2seq-based model (Seq2Seq) (Tkachenko and Sirts, 2018) in most cases for all three types of tagging. As a state-of-the-art, Seq2Seq outperforms MarMot that is a CRF-based strong baseline.

For POS, Seq2Seq and MarMot yield similar results (76.78% and 77.14%) for Kazakh such small dataset (Table 1), in contrast, the proposed approach significantly outperforms MarMot and Seq2Seq by $\approx 18\%$. Also, similar results can be observed for Turkish, the second smallest dataset in this work. LIMD outperforms baselines by $\approx 4\%$. For German and Arabic, LIMD gives above 1% improvement over baselines, and its results for Czech, Russian and Spanish datasets are slightly better than baselines.

For morpheme, LIMD gives comparable accuracy with MarMot and Seq for the most of the languages. Again, it shows promising results for the smallest (Kazakh, the improvement is $\approx 25\%$) and the second smallest (Turkish, the improv. is $\approx 7.5\%$) datasets. For German morpheme prediction, Seq2Seq (88.44%) gives 1.88% improvement over MarMot (86.56%), and LIMD yields 92.23% accuracy in this case. Compared to other languages, LIMD achieves a larger improvement over the baselines on the German data that is highest ambiguous among all datasets.

For POS+morpheme joint prediction, LIMD performs much higher than Marmot and Seq2Seq for the German, Turkish, Kazakh data, and for other languages, they give very competitive accuracies. Cross-task comparisons (morpheme vs. POS+morpheme and POS vs. POS+morpheme) reveal that the morpheme tagging is the most challenging part for all models, as it can be observed that morpheme's accuracies are much lower than POS one. It worth noting that Seq2Seq applies double-layer of biLSTM network as encoder to model the character and word embeddings for context. This architecture has been applied recently to context representation learning for MD and achieved the notable results (Heigold et al., 2017; Tkachenko and Sirts, 2018; Yu et al., 2017).

6 Analysis And Discussion

Analysis of surface word context. To explore the influence of different window-sized surface contexts, we fixed the morphological context with the leftmost and rightmost ones and tuned the window size only for the surface context. We choose the German dataset for the exploration because it is the most complex data in its ambiguous analysis in this work. Table 3 shows the results for POS, Morpheme and POS+Morpheme prediction. It can be seen that the model's accuracy grows gradually in window size (1-9), then it starts to drop slightly at window size 11, which indicates words outside of window 7 become "noise" when performing joint tagging. At window size (7,9), the model has minor differences for POS+Morpheme. Thus, we choose window size for word context to 7.

Analysis of averaged morphological context. Figure 2 shows the error rate of the training and test process for German data when incorporating two types of context embeddings: leftmost and rightmost analyses as morphological context. First, make it clear that all training curves are without markers in the figure. With markers are testing curves. In which, we present the models' performance when applying the different contexts independently: word context, left and right analyses as morphological context.

It can be seen that compared to *word* context, the *left* morphological context improves model's performance both in terms of the process for training and test. The error rate of training and test curve has a fast decrease when the model utilizes *left+right* morphological contexts compared to other settings. The model yields 84.38% (word), 85.17% (left) and 87.50% (left+right) accuracy at 25 epochs. It indicates that the morphological context plays an important role in MD. In other words, it could improve the model's performance and also reduces the training time.

Error analysis. Figure 3 shows the largest error rates of the distinct morphological categories for MarMot, Seq2Seq and LIMD models averaged over all languages. It can be seen that all models tend to have large errors for predicting the features of *Case*, *Number* and *Gender*. Among all the mod-

Table 2: Test accuracy results for POS, Morpheme and POS+Morpheme.

Lang.	POS			Morpheme			POS+Morpheme		
	Marmot	Seq2Seq	LIMD	Marmot	Seq2Seq	LIMD	Marmot	Seq2Seq	LIMD
ar	96.28	96.38	97.48	91.87	92.81	93.26	91.57	92.50	92.96
cs	98.56	98.67	98.95	93.24	94.57	94.82	92.97	94.40	94.45
es	98.25	98.17	98.40	97.79	97.56	98.00	97.11	96.83	97.30
de	92.96	93.34	94.76	86.56	88.44	92.23	81.75	83.67	88.11
ru	98.36	98.56	98.74	94.72	95.34	96.33	94.33	95.05	95.96
tr	92.99	93.66	97.66	88.42	90.47	97.04	86.20	88.15	96.03
kk	77.14	76.78	95.46	71.66	69.65	96.97	65.99	65.63	94.70

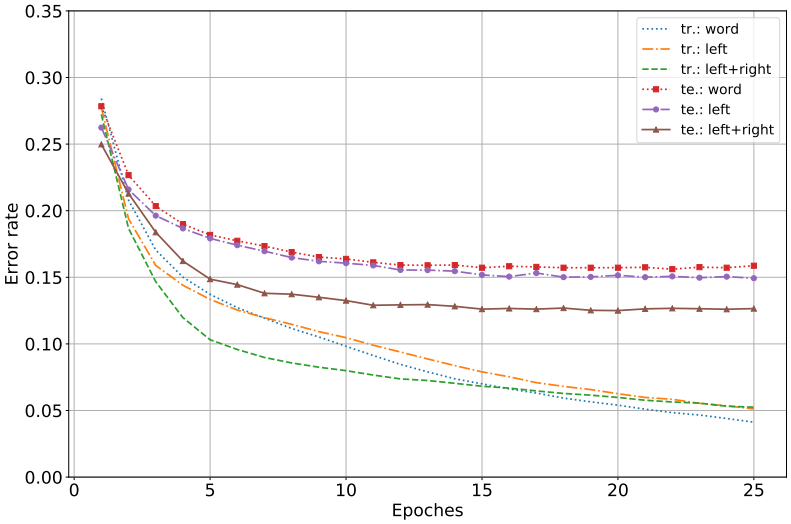

Figure 2: Example of training and test run of LIMD with two types of contexts for German data. *tr.* and *te* denote train and test. *word* - word context. *left, right* denote left and right morphological context.

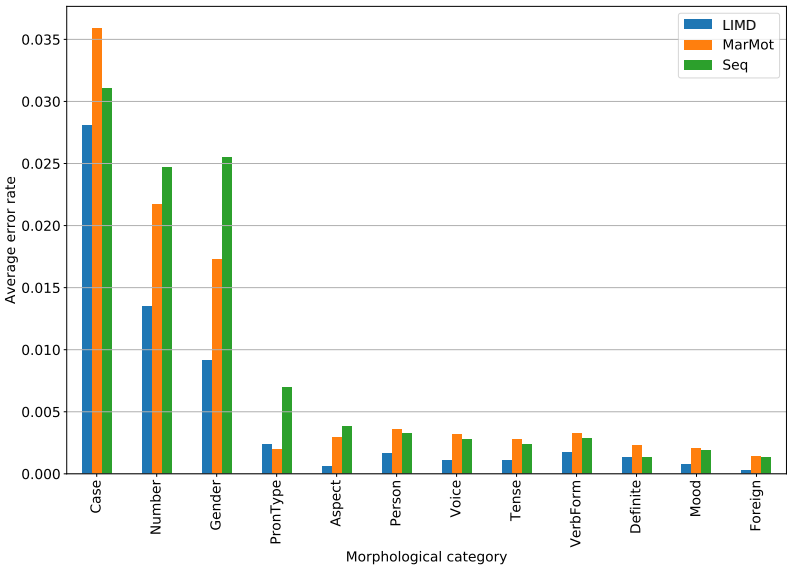

Figure 3: Average error rates of distinct morphological categories for LIMD, MarMot and Seq2Seq models.

els, it seems Seq2Seq performs worse on modeling *Number* and *Gender* features than others. It can be seen that LIMD's error rates are considerably lower in these two categories. For *Case* features,

Table 3: Test accuracy results for the German data using different window-sized surface contexts.

win	POS	Morpheme	POS+Morpheme
1	93.98	89.98	85.23
3	94.12	90.58	86.01
5	94.58	91.64	87.41
7	**94.76**	92.23	88.11
9	94.69	**92.30**	**88.15**
11	94.56	92.16	87.85

Table 4: Comparisons with previous work: Seq2Seq (Tkachenko and Sirts, 2018), Heigold (Heigold et al., 2017), Dozat(Dozat et al., 2017)

Lang.	Seq2Seq	Heigold	Dozat	LIMD
ar	**93.84**	93.78	92.85	92.96
cs	95.39	**96.32**	95.22	94.45
ru	**96.67**	96.45	96.20	95.96
tr	90.70	89.12	90.22	**96.03**
average	94.15	93.91	93.63	**94.85**

MarMot shows the largest error rates.

Comparison with previous work. It is difficult to make a direct comparison of our results to previously published results since UD data sets have various versions with differences. Here, we try to provide a very rough comparison in Table 4 only for reference. The original results were taken from (Tkachenko and Sirts, 2018) (Seq2Seq), which is obtained on UD2.1 version using a large pre-trained word embeddings[5] with sophisticated neural architecture and large well-tuned hyper-parameters. In contrast, LIMD starts by random initialization of parameters, then is tuned in the training process. Another previous tagger was presented in the work (Dozat et al., 2017), which used a more sophisticated encoder than Seq2Seq. In addition, we compare the results taken from (Heigold et al., 2017) obtained on UDv1.3. As we can see, the results are very competitive in most cases. For Turkish, LIMD shows a significant improvement.

Decoding time and accuracy. In Table 5, we report the final comparison to the baselines both in terms of accuracy and decoding time. Comparing with the baselines of MarMot and Seq2Seq, LIMD achieves around 6% gains in average accuracy for all languages, and running about 6 and 33 times faster than MarMot and Seq2Seq respectively. It can be seen from Table 5 that LIMD gains significant improvements on Kazakh, Turkish, and German datasets. The former two are the small datasets compared with other in this work, and the German is the complex one (it has around 6 analyses per word in average, see in Table 1). It may indicate that LIMD works well on morphologically complex languages with many analyses per token and the approach suffers less from the issue of lack of data.

[5]https://github.com/facebookresearch/fastText

Table 5: Comparison with the state-of-the-arts.

Lang.	POS+Morpheme		
	MarMot	Seq2Seq	LIMD
ar	91.57	92.50	92.96
cs	92.97	94.40	94.45
es	97.11	96.83	97.30
de	81.75	83.67	**88.11**
ru	94.33	95.05	95.96
tr	86.20	88.15	**96.03**
kk	65.99	65.63	**94.70**
avg.	87.13	88.03	**94.21**
tokens/s	1372 tok/s	257 tok/s	**8712 tok/s**

7 Conclusion

This paper presents a language-independent morphological disambiguation approach, LIMD. It embeds surface word and morphological context into vector representations, then calculates cosine similarly scores of two to perform disambiguation. Experimental evaluations show that LIMD outperforms other sophisticated models in both accuracy and speed. Results indicate that LIMD works well on morphologically complex languages with many analyses per token and the approach suffers less from the issue of lack of data.

Possible future work in this direction is to apply different methods to the model's output instead of a computing dot product for disambiguation. Also, there is still room for the improvement in the model's architecture, such as better capturing surface word context or modeling morphological analyses with more advanced architectures.

Acknowledgements

The work was funded by the Committee of Science of Ministry of Education and Science of the Republic of Kazakhstan under the grant AP09058174 and the grant AP08856034.

References

Eduard Bejček, Eva Hajičová, Jan Hajič, Pavlína Jínová, Václava Kettnerová, Veronika Kolářová, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Jarmila Panevová, Lucie Poláková, Magda Ševčíková, Jan Štěpánek, and Šárka Zikánová. 2013. Prague dependency treebank 3.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada. Association for Computational Linguistics.

Kira Droganova, O. Lyashevskaya, and Daniel Zeman. 2018. Data conversion and consistency of monolingual corpora: Russian ud treebanks.

Jan Hajič, Otakar Smrž, Petr Zemánek, Petr Pajas, Jan Šnaidauf, Emanuel Beška, Jakub Kracmar, and Kamila Hassanová. 2009. Prague arabic dependency treebank 1.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Diiek Z. Hakkani-Tur, Kemal Oflazer, and Gokhan Tur. 2000. Statistical morphological disambiguation for agglutinative languages. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*.

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, Valencia, Spain. Association for Computational Linguistics.

Matthieu Labeau, Kevin Löser, and Alexandre Allauzen. 2015. Non-lexical neural architecture for fine-grained POS tagging. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 232–237, Lisbon, Portugal. Association for Computational Linguistics.

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Chaitanya Malaviya, Matthew R. Gormley, and Graham Neubig. 2018. Neural factor graph models for cross-lingual morphological tagging. In *ACL*.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Thomas Mueller, Helmut Schmid, and Hinrich Schütze. 2013. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

Thomas Müller and Hinrich Schütze. 2015. Robust morphological tagging with word representations. In *HLT-NAACL*.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained pos tagging. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 777–784, Stroudsburg, PA, USA. Association for Computational Linguistics.

Umut Sulubacak, Memduh Gokirmak, Francis Tyers, Çağrı Çöltekin, Joakim Nivre, and Gülşen Eryiğit. 2016. Universal Dependencies for Turkish. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3444–3454, Osaka, Japan. The COLING 2016 Organizing Committee.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

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).

Alexander Tkachenko and Kairit Sirts. 2018. Modeling composite labels for neural morphological tagging. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 368–379, Brussels, Belgium. Association for Computational Linguistics.

Gulmira Tolegen, Alymzhan Toleu, and Rustam Mussabayev. 2020. Voted-perceptron approach for Kazakh morphological disambiguation. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 258–264, Marseille, France. European Language Resources association.

Alymzhan Toleu, Gulmira Tolegen, and Aibek Makazhanov. 2017. Character-aware neural morphological disambiguation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 666–671, Vancouver, Canada. Association for Computational Linguistics.

Alymzhan Toleu, Gulmira Tolegen, and Rustam Mussabayev. 2020. Deep learning for multilingual pos tagging. In *Advances in Computational Collective Intelligence*, pages 15–24, Cham. Springer International Publishing.

Francis M. Tyers and Jonathan N. Washington. 2015. Towards a free/open-source universal-dependency treebank for kazakh. In *3rd International Conference on Turkic Languages Processing, (TurkLang 2015)*, pages 276–289.

Xiang Yu, Agnieszka Falenska, and Ngoc Thang Vu. 2017. A general-purpose tagger with convolutional neural networks. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 124–129, Copenhagen, Denmark. Association for Computational Linguistics.

Deniz Yuret and Ferhan Türe. 2006. Learning morphological disambiguation rules for turkish. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 328–334, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nasser Zalmout and Nizar Habash. 2017. Don't throw those morphological analyzers away just yet: Neural morphological disambiguation for Arabic. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 704–713, Copenhagen, Denmark. Association for Computational Linguistics.