TransMorpher: A Phonologically Informed Transformer-based Morphological Analyzer

Karahan Şahin[1], Ümit Atlamaz[2]

[1]Cognitive Science Program, Boğaziçi University, İstanbul, Turkey
[2]Department of Linguistics, Boğaziçi University, İstanbul, Turkey

Abstract

We introduce TransMorpher, a phonologically informed Transformer-based morphological analyzer that returns disambiguated morphological parses. TransMorpher consists of a phonological normalization module (inspired by the modular architecture widely adopted in the field of Generative Linguistics), a character-based encoder with multi-head self-attention for encoding words, and a pre-trained language model for encoding context, and a decoder with multi-head self-attention. The language-specific phonological normalization module maps phonological variants of morphemes into unique abstract representations. Normalized words are fed to the word encoder, whose output representations are concatenated with the contextual representation of each word obtained from the pre-trained model. The concatenated representations are then fed to the decoder, which auto-regressively generates a single disambiguated morphological parse for each word. We evaluate TransMorpher on Turkish, an agglutinative language with rich morpho-phonological variation in a relatively low-resource setting, and obtain promising results with 85% accuracy. Our experiments show that phonological normalization contributes to a 5% gain in tag accuracy and 10% in lemma accuracy. We also tested our model without the phonological normalization module on Danish, Russian, and Finnish in low-resource contexts and achieved acceptable accuracy rates.

Keywords

morphological analysis, transformer, agglutinative morphology

1. Introduction

We introduce TransMorpher, a phonologically informed transformer-based morphological analyzer that takes in sentences as input and produces disambiguated lemmas and morphological features of each word as the output. TransMorpher is a two-level analyzer that consists of a rule-based phonological normalization module and a sequence to sequence character translation module using the original Transformer architecture [1].

TransMorpher was explicitly developed for Turkish, an agglutinative language with 132 distinct derivational and inflectional morphemes and rich allomorphy due to vowel harmony and other productive phonological processes. For example, the past tense suffix can be realized as one of [*dı, du, di, dü, tı, tu, ti, tü*] depending on the final vowel and consonant of the stem it attaches to. One immediate corollary of the phonological alternation is that it explodes the morpheme space by outputting multiple surface forms for the same abstract morpheme. This

The International Conference and Workshop on Agglutinative Language Technologies as a challenge of Natural Language Processing (ALTNLP), June 7-8, 2022, Koper, Slovenia

karahan.sahin@boun.edu.tr (K. Şahin); umit.atlamaz@boun.edu.tr (Ü. Atlamaz)

complicates the morphological analysis task by increasing the variation in the dataset as well as causing an imbalance in the distribution of surface forms of abstract morphemes, causing a sparsity problem for some of the morphemes.

To alleviate this problem, we first pass the surface forms through a phonological module which normalizes the allophonic variants into a single abstract representation. This is inspired by the inverted Y model adopted in the Generative Linguistics and Distributed Morphology in particular, where morphology is considered to be the module that maps the output of the syntactic component into the phonological component of the grammar [2]. Our experiments show that phonological normalization increases accuracy significantly in low-resource settings. While the phonological normalization module is specific to Turkish, the sequence to sequence translation module is language-agnostic and can be used independently for other languages.

The organization of the paper is as follows: Section 2 discusses related work, Section 3 discusses the details of the phonological normalization module and the sequence translation model, Section 4 introduces the details of the datasets we use, Section 5 presents the experiments and the results, Section 6 concludes the discussion.

2. Related Work

Morphological analysis of agglutinative languages has been dominated by Finite State Transducers. The two-level morphology developed by Koskenniemi [3] allows modeling both phonological and morphological grammars that enable morphological analysis via Finite State machines. Some of the early implementations of Finite State Technology involve the PC-KIMMO system developed by Antworth [4] based on Koskenniemi [3] and the XFST Xerox Finite State Tool developed by Beesley and Karttunen [5].

Finite State Transducers have been widely used in morphological analysis of Turkish as well. Oflazer [6] built a two-level morphological analyzer for Turkish based on the PC-KIMMO system and later with the Xerox tools. Çöltekin [7] implemented the first publicly available free two-level FST analyzer.

One of the major challenges in the morphological analysis is ambiguity. FSTs end up generating multiple candidate analyses, which need further disambiguation. These candidates are usually disambiguated with various machine learning techniques. Sak et al. [8] developed an analyzer with a disambiguation component with the perceptron algorithm, and Sak et al. [9] implemented a stochasticized FST to address the disambiguation problem in Turkish.

Recently, a purely deep learning-based line of morphological analyzers started emerging. Malaviya et al. [10] used a neural factor model for cross-lingual morphological analysis. Akyürek et al. [11] introduced Morse, a recurrent encoder-decoder model that can do cross-lingual morphological analysis and disambiguation jointly. Morse uses an LSTM to do left-to-right character encoding of each word. Then, it uses a bi-directional LSTM to encode the context for each word to get a unique contextual embedding for each word that allows the model to do disambiguation. The decoder is also a unidirectional encoder. The main advantage of using deep learning-based approaches is their ability to be used across many languages. Unlike FSTs, they are not rule-based and can be quickly trained with sufficient data, alleviating the need for tedious rule-writing by experts.

TransMorpher takes Morse as a starting point and replaces the LSTM component with a Transformer based encoder to encode tokens and a BERT-based model [12] to perform the context encoding. In addition, it has a phonological normalization module that normalizes each token before it is fed to the encoder. Like Morse, TransMorpher can do both morphological analysis and disambiguation simultaneously. We achieve 85% accuracy in a low-resource setting and show that the phonological normalization module significantly improves accuracy in this low-resource setting. Although the accuracy of TransMorpher does not reach the current state-of-the-art (98.59% as reported by Akyürek et al. [11]), it achieves promising results with 10 times less data.

3. TransMorpher Architecture

TransMorpher is a character-based sequence to sequence translation system inspired by Sutskever et al. [13] and Akyürek et al. [11]. It has a Transformer based encoder-decoder module, a rule-based phonological normalization module for Turkish, and a BERT [12] based contextual token encoder for encoding the context of a word in a given sentence. TransMorpher takes the input of a word and the sentence containing the word and returns a disambiguated lemma and morphological analysis. Figure 1 depicts the TransMorpher architecture. In the following subsections, we explain the details of each component.

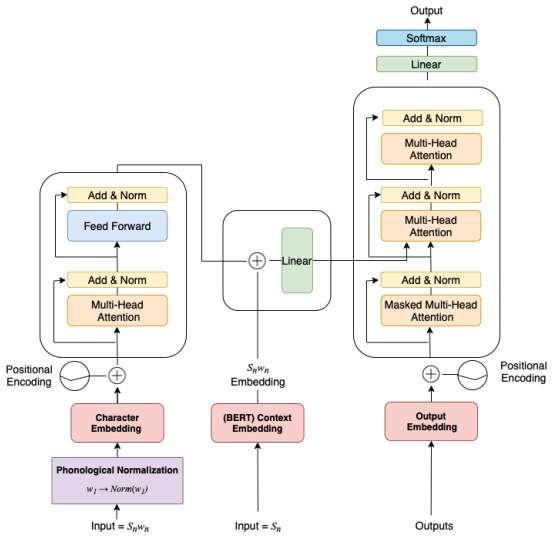

Figure 1: TransMorpher Architecture

3.1. Phonological Normalization Module

One of the major hypotheses maintained in the field of Generative Linguistics has been the modularity of the components of natural language. Ever since Chomsky and Lasnik [14], the field of generative linguistics assumed the inverted Y model that embeds natural language

between the Conceptual-Intentional system and the Articulatory-Phonetic system [15, 16]. In this modular view, the morphological component is considered to be embedded between the syntactic component and the phonological component. This has been clearly articulated in the modern generative theories of morphology like the Distributed Morphology by Halle and Marantz [2].

Our phonological normalization module is inspired by the modular division of labor standardly assumed in the field of Generative Linguistics. Turkish is an agglutinative language with rich morphology and phonology. The richness of phonological variation increases the number of surface forms to be analyzed significantly. In addition, phonological variants introduce an imbalance in the distribution of surface forms and raise sparsity problem for certain surface forms. We alleviate these challenges by using a rule-based shallow phonological normalization component that converts phonological variants of certain strings into a single abstract form. For example, the two surface variants of the plural morpheme [*ler*] and [*lar*] are turned into the single abstract form [*lAr*]. Similarly, the various surface forms of the past tense morpheme [*dı, du, di, dü, tı, tu, ti, tü*] are transduced into [*DI*]. An example of an input-output pair for the phonological normalization module is given in (1).

(1) Input: *başlamışlardı*
Output: *başlamışlArDI*
start.Nar.PastCop.V3pl
'They had started.'

The phonological normalization module consists of a set of phonological rules that skip the lemmas and apply to all the morphemes after the lemma. In its current implementation, it mainly focuses on vowel harmony and some of the common phonological alternations (e.g., voicing, insertion, elision). It is not built as a comprehensive phonological module as some of the phonological rules require morphological information and the task becomes counter-cyclic. The main goal is to alleviate the sparsity problem in low-resource settings rather than providing a precise phonological component.

3.2. Word Encoder

Word encoding is done via a character-level Transformer-based encoder. Each character w_{ij} is mapped to an n-dimensional vector $v_{ij} \in \mathbb{R}^n$. Input tokens are appended with [start] and [end] tokens in addition to padding for tokens below the maximum sequence length. We used the original Transformer encoder architecture introduced by Vaswani et al. [1]. The encoder is composed of a stack of N = 3 identical layers. Each layer consists of a multi-head attention sublayer with 8 attention units and a position-wise fully connected feed-forward network sublayer. Each sublayer is followed by a residual connection and normalization. The left part in Figure 1 depicts the word encoder.

3.3. Context Encoder

Context encoder is a BERT [12] based pre-trained language model that takes a sentence as input and returns a contextual embedding for the target word. We specifically use BERTürk

by Schweter [17]. For a word w_i, we define its corresponding context embedding $c_i \in \mathbb{R}^h$ as the output of the BERT embeddings. The output of the context encoder is concatenated with the output of the word encoder and passed through a linear transformation before it is fed into the decoder.

3.4. Decoder

The decoder is also composed of a stack of N = 3 identical layers with an additional sub-layer in each layer to perform multi-head attention over the embeddings created by the concatenation of the word and context encoders. The output of the decoder stacks is passed through a linear layer before they are passed through a softmax layer.

4. Datasets

We evaluated TransMorpher on Turkish in a relatively low-resource setting as well as with Danish, Rusian, and Finnish to test its multilingual capabilities.

4.1. Turkish Dataset

There are several Turkish datasets that are available for training morphological analyzers Hakkani-Tür et al. [18], Sak et al. [8], Sulubacak et al. [19], Akyürek et al. [11], Kayadelen et al. [20] to name a few. We evaluate our model on TWT by Kayadelen et al. [20] as it is the only gold standard dataset with sufficient data size.

TWT is a gold standard dependency treebank for Turkish developed by Kayadelen et al. [20]. TWT annotations were done manually by trained linguists with reported inter-annotator agreement scores above 90%, indicating a high degree of consistency. TWT consists of 4,851 sentences scraped from Wikipedia and various websites and annotated with the Universal Dependencies tags in the CoNLL-U format. Details of the data distribution are given in Table 1.

Source	Sentence	Token
Wikipedia	2310	39932
Web	2541	26508
Total	4851	66440

Table 1
Treebank Statistics of TWT [20]

Following the convention introduced by Hakkani-Tür et al. [18], we convert the dependency annotations for each token into morphological parses by concatenating the lemma, part-of-speech tag, and morphological features as in (2).

(2) $lemma + POS - Tag + Feature_1 + ... + Feature_n$

After transforming the data into morphological parses, we randomly split the data into training, validation, and test sets using a ratio of 70:15:15. Table 2 provides the details of the split.

Data	Count
Training	46512
Validation	9966
Test	9966

Table 2
Token Counts for Training, Validation, and Test Sets

4.2. Multilingual Datasets

We evaluated TransMorpher without the phonological normalization component on Danish[1], Russian[2], and Swedish[3], using the annotations from the Universal Dependencies Repository. As the context encoder, we used Multilingual BERT Devlin et al. [12]. Table 3 summarizes the token quantities in each dataset. We used a 80:10:10 ratio for training, validation, and test sets.

Language	Token Count
Danish	100,733
Swedish	96,820
Russian	98,000

Table 3
Token Counts for Danish, Swedish, & Russian

5. Experiments

In this section, we present our training procedure and experiment results. We first discuss the results from Turkish experiments with and without phonological normalization. Then, we present our results for Danish, Russian, and Finnish.

5.1. Training

All the character embeddings have $n = 512$ dimensions, and all the hidden units have $H = 2048$ dimensions. The context embeddings from the BERTürk model have $h = 512$ dimensions. We used Xavier initialization for initializing the model parameters [21].

We trained the models using back-propagation through time with batch-gradient descent with a batch size of 128. We used Cross-Entropy Loss as our loss function. We used Adam optimizer with a learning rate of $lr = 0.0001$, betas between 0.9 and 0.98, and an epsilon value of 1e-9. Table 4 provides the details of our parameters. Vocabulary Size and Sequence Length were determined by the Turkish data and the table only reflects the values for TWT. These values were adjusted for each language.

[1]https://github.com/UniversalDependencies/UD_Danish-DDT
[2]https://github.com/UniversalDependencies/UD_Russian-GSD
[3]https://github.com/UniversalDependencies/UD_Swedish-Talbanken

Parameter	Value
Vocabulary Size	322
Sequence Length	20
Embedding Dimensions	512
Latent Dimensions	2048
Attention Heads	8
Batch Size	128
Epoch	9

Table 4
Encoder Parameters

5.2. Turkish Results

Table 5 presents the lemma, tag, and POS tag accuracies for the TWT dataset, as well as precision and recall scores for tags. Accuracy is calculated via exact-match for the whole-tag, excluding the lemma. POS tag and lemma accuracies are also calculated based on exact matches. Precision and Recall scores are calculated on a tag-by-tag basis, assigning partial credit to partially correct tag sets. The baseline numbers report the success metrics of the TransMorpher model without the Phonological Normalization module or the context encoding for disambiguation.

	Accuracy	Precision	Recall	POS Acc	Lemma Acc
Baseline	0.724	0.783	0.784	0.855	0.840
Phonological Norm.	0.777	0.839	0.838	0.886	0.940
Contextual Embedding	0.7933	0.906	0.899	**0.947**	0.9405
Phonological Norm.+Context. Emb.	**0.8501**	**0.9337**	**0.9323**	0.9287	**0.9703**

Table 5
TWT Results

Our results show that phonological normalization alone contributes to a 5% increase in tag accuracy and 10% gain in lemma accuracy over the baseline. We also observe that the contextual BERT embeddings yield a 7% gain on tag accuracy, whereas the impact on lemma accuracy is equivalent to the impact of phonological normalization. Our best results come from the combination of phonological normalization with contextual word embeddings. We make gains across all the metrics. We observe around 13% accuracy gains (compared to the baseline) in both tag accuracy and lemma accuracy, with tag accuracy reaching 85% and lemma accuracy reaching 97%. The only metric where the combined model does not outperform the other alternatives is on POS Tag accuracy, where the best accuracy is achieved with just contextual embedding without any phonological normalization.

5.3. Multilingual Results

To test the multilingual capabilities of TransMorpher, we evaluated it on Danish, Swedish, and Russian without the phonological normalization module but with the contextual encoder. Table 6 presents the results. The success rates on the multilingual data are significantly lower than the

Turkish scores. There are two main factors behind this difference. First, there is no phonological normalization module for these languages. Second, the multilingual BERT model used to encode context in these languages is not performing at the same level as the language-specific BERT model we used for Turkish. We believe that the accuracy will improve significantly once we use language-specific language models for each language. We leave this for future research.

	Accuracy	Precision	Recall	Lemma
Danish	0.749	0.836	0.834	0.826
Swedish	0.649	0.719	0.719	0.703
Russian	0.615	0.525	0.527	0.840

Table 6
Experiment Results

6. Conclusion

We presented TransMorpher, a phonologically informed Transformer-based morphological analyzer. TransMorpher consists of a linguistically motivated phonological normalization module and a Transformer-based encoder-decoder architecture with a BERT-based context encoder. We evaluated TransMorpher on Turkish, an agglutinative language with rich phonological variation, in a low-resource setting. TransMorpher takes an input of a target word and the sentence containing it and returns a disambiguated morphological analysis for the target word in that sentence. We achieve 85% accuracy on the TWT dataset [20]. Our experiments show that the phonological normalization component contributes to a significant gain in accuracy. Although TransMorpher does not reach state-of-the-art results reported in Akyürek et al. [11], it achieves promising results with a dataset whose size is an order of magnitude less than the TrMor2018 [11]. The logical next step is to evaluate TransMorpher on TrMor2018 and other Turkish datasets to evaluate its success in high-resource settings and compare it to the current state-of-the-art models. We leave this for future work. We also evaluated our model on multilingual data without the rule-based phonological normalization module and achieved acceptable results. We believe that multilingual results can be improved with better context embedding models, which we leave for future investigation.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762.

[2] M. Halle, A. Marantz, Distributed morphology and the pieces of inflection, in: K. Hale, S. J. Keyser (Eds.), The view from Building 20: Essays in Linguistics in honour of Sylvain Bromberger, 1993, pp. 111–176.

[3] K. Koskenniemi, Two-level model for morphological analysis., in: IJCAI, volume 83, 1983, pp. 683–685.

[4] E. L. Antworth, Pc-kimmo: a two-level processor for morphological analysis, Summer Institute of Linguistics (1990).
[5] K. R. Beesley, L. Karttunen, Finite-state morphology: Xerox tools and techniques, CSLI, Stanford (2003).
[6] K. Oflazer, Two-level description of turkish morphology, Literary and linguistic computing 9 (1994) 137–148.
[7] Ç. Çöltekin, A freely available morphological analyzer for Turkish, in: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), European Language Resources Association (ELRA), Valletta, Malta, 2010. URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/109_Paper.pdf.
[8] H. Sak, T. Güngör, M. Saraçlar, Morphological disambiguation of turkish text with perceptron algorithm, in: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, 2007, pp. 107–118.
[9] H. Sak, T. Güngör, M. Saraçlar, A stochastic finite-state morphological parser for turkish, in: Proceedings of the ACL-IJCNLP 2009 Conference short papers, 2009, pp. 273–276.
[10] C. Malaviya, M. R. Gormley, G. Neubig, Neural factor graph models for cross-lingual morphological tagging, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, volume 1, 2018, p. 2653–2663.
[11] E. Akyürek, E. Dayanık, D. Yüret, Morphological analysis using a sequence decoder, Transactions of the Association for Computational Linguistics 7 (2019) 567–579. URL: https://aclanthology.org/Q19-1036. doi:10.1162/tacl_a_00286.
[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[13] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, Advances in neural information processing systems 27 (2014).
[14] N. Chomsky, H. Lasnik, Filters and control, Linguistic Inquiry 8 (1977) 425–504. URL: http://www.jstor.org/stable/4177996.
[15] N. Chomsky, Lectures on government and binding: The Pisa lectures, 9, Walter de Gruyter, 1993.
[16] N. Chomsky, The minimalist program, MIT press, 1995.
[17] S. Schweter, Berturk - bert models for turkish, 2020. URL: https://doi.org/10.5281/zenodo.3770924. doi:10.5281/zenodo.3770924.
[18] D. Z. Hakkani-Tür, K. Oflazer, G. Tür, Statistical morphological disambiguation for agglutinative languages, Computers and the Humanities 36 (2002) 381–410.
[19] U. Sulubacak, M. Gokirmak, F. Tyers, Ç. Çöltekin, J. Nivre, G. Eryiğit, Universal Dependencies for Turkish, in: "Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers", "The COLING 2016 Organizing Committee", Osaka, Japan, 2016, pp. 3444–3454. URL: https://aclanthology.org/C16-1325.
[20] T. Kayadelen, A. Ozturel, B. Bohnet, A gold standard dependency treebank for Turkish, in: Proceedings of the 12th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 5156–5163. URL: https://aclanthology.org/2020.lrec-1.634.
[21] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y. W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International

Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.