Joint Learning of Morphology and Context with Transformer Language Models

Karahan Şahin
Boğaziçi University
karahan.sahin@boun.edu.tr

Abstract

In this study, it is proposed a neural morphological analysis model using joint learning with language modeling. the approach utilizes the contextual information in both character-level and word-level representations. The main differences of this model from the relevant literature is the model is trained on sentence-level rather than word-level and the contextual representations are learned according to the morphological representations.

1 Introduction

Morphological Analysis is a NLP task that extraction of lemma, part-of-speech and other morpho-syntactic information. The task is crucial for agglutinative languages such as Turkish to reduce the sparsity in the representations. The task is also important for other tasks such as dependency parsing or sentiment analysis. In the initial research on morphological analysis, finite-state transducers were used for providing the all possible parsing of a given token. Although this method is provide a fine-grained morphological representations, the ambiguity is not resolved in this process. Therefore, disambiguation must be applied to the provided parses. In the previous research morphological disambiguation, models classify the correct parse with architectures perceptron disambiguation [**hakkani2018morphological**] model or LSTM model based sentence decoding [**shen2016role**]. The latter approaches than implements the seq2seq translation using character-level encoding of tokens in studies [**10.1162/tacl˙a˙00286**], Malaya. The morphological analysis is also utilized while providing pipelines such as UD or to provide dependency parsing, or named-entity recognition. However, the literature approaches the task as the linear process where the components required for morphological analysis learned separately as the contextual information is learned by a different language model where we propose method that learns each component jointly.

Looking at the previous literature, we can propose an novel architecture where rather than applying transfer learning approaches such as the concatenation of pre-trained contextual embeddings to the character-level encoding, our architecture proposes a joint learning of the morphology and the contextual information jointly. Figure 1 explains the architecture in more detail. The model also has the implication of the linguistic theory on the learning of morpho-syntactic information. While previous architectures proposes a learning where morphology and syntax are learning and joint decoding, while our architecture proposes that the learning of morphology and syntax simultaneously.

2 Architecture

The initial architecture is influenced by three studies as joint framework of [**can2022joint**], character-level contextual representations of [**shen2016role**] and sequence-decoder approach of [**10.1162/tacl˙a˙00286**]. In the morphological analysis component of the architecture, the latter two studies are in effect. The overall disambiguation process is similar to the Yüret's architecture where the morphological disambiguation carried with seq2seq approach where the character-level representations are learned via Bidirectional LSTM encoder and morphological features are generated via Bi-LSTM decoder. The language model component of this architecture is influenced by the joint framework of [**can2022joint**] where the multiple linguistic layers are learned at the same time. The loss calculation is carried as the same process as this framework.

2.1 Morphological Analysis Component

The architecture starts with the shared LSTM encoder for input token. The encoder works on the character-level one-hot encodings of the tokens. The tokens are initial fed into input embedding layer. The layer is important since the studies show the learned input embeddings on the character-level reduces the noise provided in the misspelled tokens.

The output embedding of each token in a given in sentence then feed through a Transformer encoder where the contextual relationship are learned between character-level token representations. The transformer encoder includes multi-head self-attention units where the contextual relations are learned between tokens. Although Yüret implements contextual information separately in their architecture, [**shen2016role**] shows that the contextual information can be learned via the surface representations since Turkish provides lesser ambiguity in the disambiguation process unlike Arabic. Therefore, we can hypothesize that the contextual information can be learned at the via surface-level encodings.

The outputted contextual representation is concatenated with the initial character-level encoding of token via the residual connections. This is implemented since there can be a information loss at the transformer encoder level. Therefore, outputted vector provides the information content of the context and the surface representation in one embedding.

The embedding fed through the LSTM decoder to generate the morphological output via softmax layer. The decoder provides two outputs. Initial output in the morphological disambiguation process each prediction is generated via LSTM node while the final state of the LSTM decoder is then feed through the language model to predict the next-word in the sentence. The morphological analysis output is calculated by the average of cross-entropy for each morphological analysis predictions as in the Formula 1. One caveat of this method is that the loss calculation is carried on the sentence basis rather than token basis. A further calculation of loss should be investigated.

$$L_{Morph} = \sum_{j=1}^{|sentence|} \sum_{i=1}^{|character_j|} Loss(\hat{y}_i^j, y_i^j) \tag{1}$$

The model is also provided with the Transformer-based encoder-decoder architecture where the character-level contextual representations are learned with transformer encoder component as well. The difference of this architecture is the character-level encoding of a token is formed as a matrix rather than a 1-d vector in LSTM based models. It is due to Transformer encoder outputs self-attention mechanism where the attention outputs of each token is provided. To implement the character-level contextual learning, the attention outputs of each character is flattened to extract the word-level representations. For better learning, the output is feed through dense layer to extract lower dimensional representations. The learned contextual representations than feed to same dimensional vectors with the feed-forward network to be reshaped to the character-level encoding.

2.2 Language Model Component

The language model compent is a carries a next word prediction task where for a word$_{\mathrm{i}}$ in a sentence, the model predicts the word$_{\mathrm{i+1}}$ in the decoder output. The model is a transformer encode architecture with one layer is selected since the dataset size is small for a pre-training objective. The number of transformer encoder in one layer of language model is equal to lenght of the sentence length. The language model is composed of transformer blocks connected to the softmax layer with next word prediction.

The loss in this output calculated as the summation of the cross-entropy loss in each prediction as the Formula 2.

$$L_{LM} = \sum_{t}^{i=1} L(\hat{y}_i, y_i) \tag{2}$$

Total loss will be calculated as the average of two losses which are loss of morphological parse of the token $\mathrm{L}_{\mathrm{Morph}}$ and the loss of the next-word prediction L_{LM} provided in the Formula 3.

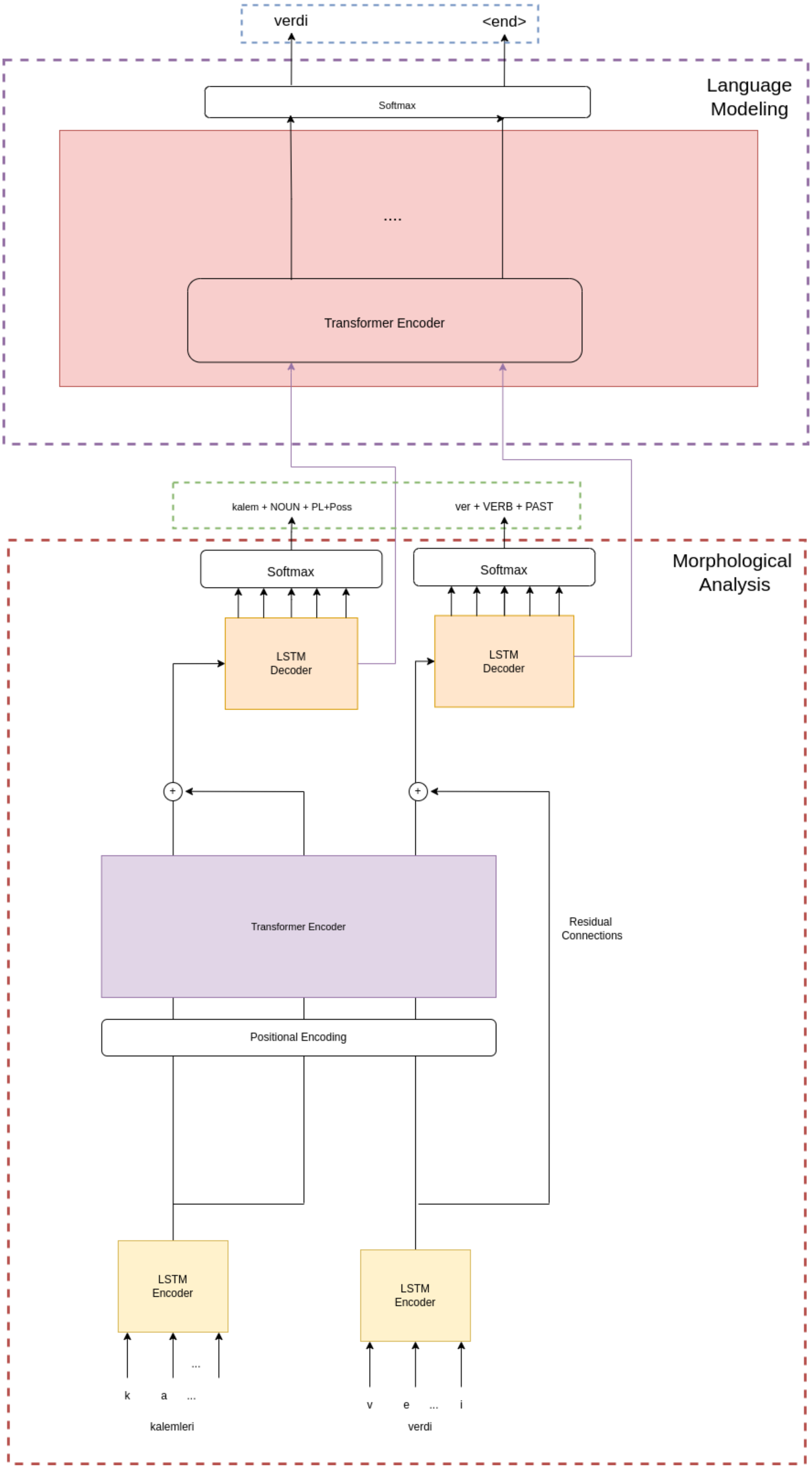

Figure 1: Example sentence *kalemlerini verdi*. The red area denoted with dashed lines points the **morphological analysis** component of the architecture, while purple area points the **language model** component. In the morphological component, there are 4 main components: 1. **LSTM Encoder** where character-level encoded, 2. **Transformer Encoder** where contextual information between character encodings, 3. **Residual Connections** where the character-level and contextual information is concatenated, 4. **LSTM Decoder** where the morphological disambiguation is carried via softmax layers.

Table 1: TrMor2006 Dataset Stats

Components	**train**	**test**
Documents	2383	3
Sentences	50674	42
Tokens	837524	862
Unambiguous	436406	482
Ambiguous	399216	379
Unknown	1902	1

$$L_{model} = \frac{(L_{Morph} + L_{LM})}{2} \tag{3}$$

2.3 Preprocessing

The data is cleaned from the character-case difference, non-Turkish encodings and the sentence-level/character-level special token addition such as the ‘¡bos¿‘, ‘¡eos¿‘, ¡unk¿, ‘¡pad¿‘. The max sentence length is decided to be 20 words where the sentences are divided according to achieve maximum sequence lengths. The model is trained on batch-gradient where the batch size is 20 as well.

3 Methodology

In this section, the dataset and the final hyperparameters that have been decided as a result of the experiments is explained.

3.1 Data

The data is from TrMor2006 dataset from the KUIS lab Github repository[1] which is the benchmark morphological analysis corpus. The corpus consists of 50674 sentences and approximetly 1 million tokens. The dataset consists of semi-automatically annotated morphological analysis of tokens. The overall statistics of the dataset are given in Table 1

One thing should be noted that ambiguous tokens have the multiple morphological parse outputs. In the training of the model, highest probability token is selected. In the evaluation process, the accuracy is calculated by the exact match between one of the possible parse is counted.

Hyperparameters: In the training of the model, there has 20 epoch in training where a results does not provide any exact match with validation set. The batch size is 20, and the learning rate is 5.0. For Adam optimizer, the epsilon is 1e-08 and the decay value is set to 0.01. The maximum sentence is determined as 20 while maximum token character length is 25.

4 Results and Discussion

4.1 Test Results

There are four statistical parameters, namely precision, recall, F1-score and accuracy to evaluate our proposed approaches however our model outputs 0.0 results in the test set. The reason of that can be argued as the number of parameters that the model is trained is too complex to provide any results in the limited number of epochs

4.2 Discussion

The main purpose of this study is to provide a new perspective to the morphological analysis where the morpho-syntactic information is learned with the character-level representations of context words are learned for both to predict next token and to decode morpho-syntactic content. However, the amount of training that the model requires more powerful setup than the current setup the model is trained.

[1] Dataset is freely available at: https://github.com/ai-ku/TrMor2018

5 Conclusion

Although the study provides a novel approach to the morphological analysis with language modeling, the complexity of the model outputs a drastically decreasing baseline performing model. However, the further research along with higher number of epochs can improve the model results.