

Composing Byte-Pair Encodings for Morphological Sequence Classification

Adam Ek,

Centre for Linguistic Theory and Studies in Probability,
Department of Philosophy, Linguistics and Theory of Science,
University of Gothenburg,
adam.ek@gu.se

1 Introduction

Things added to model:

- Layer attention
- Adaptive learning rate
- Label smoothing
- Fine-tuning
- Dropouts
- Trained character representations

Things to add:

- (results) More detailed accuracy for $\text{len}(\text{BPE}) > 2$
- (results) With fine-tuning/without fine-tuning
- (data) Data statistics: number of MSD-tags

In this project we explore the transformer model applied to sequence classification. In sequence classification a dataset is split into words (or tokens) and each word is assigned a label. Neural model can predict the label of a word by assigning it a representation and passing it through the network. However, the transformer model use byte-pair encoding (BPE) [7], or some variation of it [2], to tokenize text. BPE tokens are sub-strings that occur significantly often in the corpus, and does not correspond directly to words. For example, the word "scientifically" may be composed of the the BPE tokens: ["scient", "ifical", "ly"]. In sequence classification, we need to assign a label to the word "scientifically". But the word is now composed of three BPE embeddings, so to predict a label for the word we need to compose all of the BPE embeddings into one representation.

In this project is to explore six different methods of creating word representations from multiple BPE embeddings.

Task: We explore combination of BPE embeddings with the task of Morphological-Tagging-in-Context [5]. The task is simply to predict the morphological features (such as case, number person, ...) of each word (as defined in the dataset) given the sentence which the word occur in.

Data: For the task we will use the Universal Dependencies dataset [6] annotated with the UniMorph schema [4]. We look at 6 different languages: Arabic, Basque, Finnish, Czech, Spanish and Turkish. The language typology, ratio of BPE-tokens per word and training/development and test data can be found in Table 1.

Language	Typology	BPE-word ratio	Train	Validation	Test
Arabic-PADT	Fusional	1.39	225494	28089	28801
Basque-BDT	Agglutinative	1.79	97336	12206	11901
Czech-CAC	Fusional	1.77	395043	50087	49253
Finnish-TDT	Agglutinative	1.98	161791	19876	20541
Spanish-AnCora	Fusional	1.34	439925	55196	54449
Turkish-IMST	Agglutinative	1.73	46417	5708	5734

Table 1: Treebank statistics.

Base model For the task we will use the XLMR [1] model¹. XLMR is a masked language model based on the transformer (specifically, RoBERTa [3]) trained on data from 100 different languages, using a shared vocabulary. In this experiment we use XLMR_{LARGE}, the model have 12 encoder layers, 16 attention heads and use 1024 dimensions for its hidden size.

2 Method

In this section we present different methods of composing BPE embeddings into word representations and the structure of our neural network. We align BPE tokens to words by a simple string matching algorithm.

2.1 Byte-pair combination methods

Sum: For the sum method, we simply take the sum for each dimension of all BPE embeddings. Thus, for dimension i we add all the values in all BPE embeddings $x_i^0 + \dots + x_i^n$:

$$x'_i = \sum_{j=1}^n x_i^j \quad (1)$$

Mean: In the mean method we calculate the sum and divide by the number of BPE embeddings in the word. Thus, for dimension i we calculate $\frac{x_i^0 + \dots + x_i^n}{n}$:

$$x'_i = \frac{1}{n} \sum_{j=1}^n x_i^j \quad (2)$$

Max: In the max method we take the most activated value of dimension i over all BPE embeddings $x^0 \dots x^n$ in the word.

$$x'_i = \operatorname{argmax}(x_i^0, \dots, x_i^n) \quad (3)$$

RNN: In this method we employ a bidirectional RNN to learn how to compose the BPE embeddings. For each multi-BPE token, we pass the sequence of BPE embeddings through an RNN and use the final output as the word representation.

Att1: In the first attention method we use self-attention [8] with one linear transformation. Like in the transformer we use three copies of the BPE sequence as inputs and transform one of them using a linear layer. We then calculate pairwise scores and apply softmax. We then use the softmax to create a weighted sum for the word representation.

$$x = \operatorname{softmax}\left(\frac{q(kW_k)^\top}{\sqrt{d}}\right) \cdot V \quad (4)$$

¹We use the fairseq implementation: <https://github.com/pytorch/fairseq/tree/master/examples/xlmr>

Att3: Like in the first attention method we use self-attentions. However, in the second attention approach we use three linear transformations, one for each copy of the input.

$$x = \text{softmax}\left(\frac{qW_qkW_k^\top}{\sqrt{d}}\right) \cdot vW_v \quad (5)$$

2.2 Model

The model we employ use XLMR as a feature extractor, i.e. we only extract BPE embeddings from the model and do not fine-tune it. An outline of the model is presented in Figure 1, in the outline f represent the different methods we use to combine BPE embeddings.

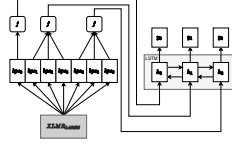


Figure 1: Procedure outline

The model we use to predict morphological features is simple. For each sentence we extract n BPE embeddings from $XLMR_{LARGE}$ then align them to words. We then run all words who consist of more than one BPE embedding through a function f which combine the BPE embeddings. This produce one embedding per word which we pass to a bidirectional LSTM to extract contextual features. We pass the LSTM outputs to a linear transformation layer that compute scores for each class in the output. We then apply softmax and compute the loss.

3 Results

We present the results of our experiments in Table 2. In general, our simple model using XLMR as a feature extractor yield good performance compared to the baseline. One exception is Basque, where the baseline is 0.1% better than our model. We see the best improvement for Czech, where our model score above the baseline by 16.8%. For the other languages, our best model increase the performance compared to the baseline by about 4%.

In general the results indicate that combining BPE embeddings into word representations using a RNN works well for all languages. It achieves the best score for tokens overall for all six languages. Looking at the accuracy of word representations containing two or more BPE embeddings, RNN also appear to work best. One exception is for Spanish-AnCora where Att3 performed marginally better (0.1%) than the RNN method.

Treebank	Ratio	All Tokens Accuracy						BPE ≥ 2 Accuracy					
		Sum	Mean	RNN	Att1	Att3	max	Sum	Mean	RNN	Att1	Att3	max
Finnish-TDT	.751	.818	.820	.833	.821	.822	.817	.738	.739	.773	.741	.747	.722
Basque-BDT	.676	.652	.659	.675	.652	.666	.656	.473	.482	.525	.474	.499	.478
Turkish-IMST	.620	.623	.626	.646	.623	.624	.610	.396	.396	.449	.396	.410	.368
Spanish-AnCora	.842	.881	.811	.881	.881	.880	.880	.829	.831	.835	.831	.836	.828
Arabic-PADT	.770	.822	.827	.829	.824	.825	.825	.709	.715	.730	.716	.718	.707
Czech-CAC	.771	.933	.935	.939	.933	.935	.914	.890	.891	.905	.890	.893	.851

Table 2: Accuracy for morphological tagging. Evaluated on all tokens and on all tokens that are composed of 2 or more BPE tokens.

4 Discussion

We can observe an interesting trend for the treebank samples. For the languages with agglutinative morphology² the average increase using RNN compared to Mean is 4.3%. For the fusional³ languages,

² 1-to-1 correspondence between morphemes and grammatical/syntactic features.

³ 1-to-many correspondence between morphemes and grammatical/syntactic features.

the difference between RNN and Mean is only 1.1%.

Interestingly, this observation seem task-dependant rather than BPE dependant. We can see in Table 1 that Czech is a fusional language, but with a BPE-to-word ratio (1.77) comparable to the agglutinative languages. But, Czech also exhibit the pattern of benefiting less from RNN than from Mean, indicating that fusional languages⁴ regardless of BPE-to-word ratio is less affected by the composition method than agglutinative languages.

References

- [1] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [4] Arya D McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. Marrying universal dependencies and universal morphology. *arXiv preprint arXiv:1810.06743*, 2018.
- [5] Arya D McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian J Mielke, Jeffrey Heinz, et al. The sigmorphon 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. *arXiv preprint arXiv:1910.11493*, 2019.
- [6] Joakim Nivre et al. Universal dependencies 2.2. 2018. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.
- [7] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.
- [8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

⁴In our language/treebank sample.