Team 21a's Submission to the SIGTYP 2024 Shared Task on Word Embedding Evaluation for Ancient and Historical Languages

Anonymous ACL submission

Abstract

In this paper, we describe Team 21a's submission to the constrained track of the SIGTYP 2024 Shared Task. Using only the data provided by the organizers, we built transformer-based multilingual models finetuned on the Universal Dependencies (UD) annotations of a given language. We also explored the cross-lingual capability of our trained models. [Our systems achieved]$_{\text{LJ}}$

1 Introduction

This paper describes Team 21a's submission to the *constrained* track of the SIGTYP 2024 Shared Task on Word Embedding Evaluation for Ancient and Historical Languages. Our general approach involves pretraining a transformer-based multilingual model on the shared task dataset, and then finetuning the pretrained model using the Universal Dependencies (UD) annotations of each language. Throughout this paper, we will refer to the pretrained model as LiBERTus. We also explored data sampling and augmentation techniques during the pretraining step to ensure better generalization performance.

Our systems achieved...[stuff]$_{\text{LJ}}$. Table 1 shows our systems' performance on the shared task test set.

We detail our resource creation, model pretraining, and finetuning methodologies. In addition, we also show the results of our cross-lingual transfer learning setup.

2 Methodology

2.1 Resource creation

We constructed the pretraining corpora using the annotated tokens of the shared task dataset. Then, we explored several data augmentation techniques to ensure that each language is properly represented based on the number of unique tokens.

	POS tag.	Morph. annot.	Lemma.
CHU			
FRO			
GOT			
GRC			
HBO			
ISL			
LAT			
LATM			
LZH			
OHU			
ORV			
SAN			

Table 1: SIGTYP 2024 Shared Task final results as evaluated on the test set.

From our experiments, upsampling underrepresented languages helped reduce our pretraining validation loss. Figure **??** shows that LATM has the most number of unique tokens in the corpora. We upsampled each language by randomly duplicating a document in the training pool until the number of unique tokens is greater than or equal to that of LATM. The same figure also shows the token counts after the augmentation process.

2.2 Model Pretraining

Using the pretraining corpora, we trained two variants of LiBERTus, a base model with XXXM parameters and a large model with XXXM parameters, that will later on serve as bases for finetuning downstream tasks. LiBERTus follows RoBERTa's pretraining architecture (Liu et al., 2019) and takes inspiration from Conneau et al. (2020)'s work on scaling BERT models to multiple languages.

Our hyperparameter choices closely resemble that of the original RoBERTa implementation as seen in Table 2. We also trained the same BPE tok-

	Base	Large
Hidden size	768	1024
Intermediate size	3072	4096
Max position embed.	512	512
Num. attention heads	12	16
Hidden layers	12	24
Dropout	0.1	0.1

Table 2: Hyperparameter configuration for the base and large variants of LIBERTUS.

enizer (Sennrich et al., 2016) using the constructed corpora. During model pretraining, we used the AdamW optimizer with β_2=0.98 and a weight decay of 0.01. The base model underwent training for 100k steps with a learning rate of 2e-4 whereas the large variant trained for 300k steps. We used a learning rate scheduler that linearly warms up during the first quarter of the training process, then linearly decays for the rest. Figure **??** shows the training curve for both variants.

2.3 Model Finetuning

For each language, we finetuned a multitask model using spaCy (Honnibal et al., 2020). The final system consists of a parts-of-speech (POS) tagger, morphological analyzer, and lemmatizer.

Parts-of-speech (POS) tagger. We employed a standard classifier that predicts a vector of tag probabilities for each token. Each POS tag is a unique class that we assign exclusively to a token. We trained a model by taking the context-sensitive vectors from our pretrained embeddings, and passing it to a linear layer with a softmax activation. The network is then optimized using a categorical cross-entropy loss.

Morphological analyzer. Similar to the POS tagger, we treat morphological annotation as a token classification task. Instead of directly modeling each feature, we made every unique combination of morphological features as a class. The limitation of this approach is that it can only predict combinations that were present in the training corpora.

Lemmatizer. We trained a neural-based edit tree lemmatizer (Müller et al., 2015) by first extracting an edit tree for each token-lemma pair. Because this process can result to hundreds of edit trees, we treat the problem of picking the correct tree as a classification task. Here, each unique tree serves as a class and we compute a probability distribution over all trees for a given token. To obtain the most probable tree, we passed the context-sensitive embeddings from our pretrained model to a softmax layer and trained the network with a cross-entropy loss objective.

3 Results and Discussion

3.1 Benchmarking results

3.2 Cross-lingual transfer

4 Conclusion

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2015. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

A Appendix

A.1 Full results for benchmark experiments

Here, we show the full results table for the benchmarking experiments in Sections 3.1 and 3.2.

A.2 Effect of different sampling strategies on pretraining performance

We also explored different sampling strategies and their effect on the pretraining loss curve. In our final system, we used the upsampling strategy as it provides clear benefit compared to others.

We examined the following configurations:

- **None:** use the original dataset without any data sampling or augmentation.
- **Upsampling:** we upsampled each language to ensure that the number of their unique tokens is greater than or equal to the most dominant language.
- **Averaging:** we took the average number of unique tokens in the whole set and up-/downsampled each language based on this value.
- **PROPN augmentation:** similar to upsampling, but instead of duplicating sentences, we replace their PROPN with another.

Figure **??** shows that upsampling provides the clearest benefit for our pretrained model.

A.3 Finetuning a model per language vs. monolithic system

We also investigated if finetuning a model per language is more effective against a monolithic system, i.e., training on the full multilingual annotated corpora. We found out that although more expensive, finetuning a model per language still yields the best results. Table **??** illustrates these results.

A.4 Reproducibility

The pretrained models can be found in ... while the source code to replicate the experiments can be found on ...