Learning Sense Embeddings for Word Sense Disambiguation Using BERT

Sara Sultan, 968430
Department of Linguistics
University of Kontanz

Abstract

Contextualized word embeddings (CWEs) from latest language models result in different embeddings for the same word based on its context, thus encoding kind of a word sense. Supervised word sense disambiguation methods encode each sense as a separate class, which does not take advantage of sense meanings the same way as CWEs. In this work, we propose fine tuning BERT with sense classes encoded as word embeddings and included into the training corpus alongside their corresponding words. For prediction, sense positions are masked in the sentence and the prediction scores from BERT at the masked positions are used to select the correct sense. While results show that sense disambiguation can be achieved with this method, further work is needed to get the most out of it.

1 Introduction

Word Sense Disambiguation (WSD) is a task which aims to clarify a text by assigning to each of its words the most suitable sense labels, given a predefined sense inventory (Navigli, 2009). Supervised WSD depends on semantically annotated data for training classifiers. Recently, language models like BERT are being used to vectorize the sentences (Raganato et al., 2017) and resulting CWEs are fed into a sense classifier. Even using a simple near neighbor classifier on CWEs achieves state of the art results (Wiedemann et al., 2019; Loureiro and Jorge, 2019). More complex classifiers improve the performance further (Scarlini et al., 2020).

Sense vocabulary compression method (Vial et al., 2019) reduces the number of unique sense tags needed and helps with training classifiers on limited data. Reduced number of sense tags makes it possible to add them into BERT vocabulary and make sense tags a part of language modeling itself. To the best of our knowledge, no one has tried to include sense tags as part of the language. By leveraging the advances in language modeling, not only can sense tags be robustly classified, but relationships between sense tags can also be quantified.

2 Methods and Material

Previous supervised approaches to word sense disambiguation either cluster the CWEs from the output of a language model or use SoftMax classifiers to annotate the senses. We propose making sense tags part of the language itself and then fine-tuning the pre-trained language models on this modified language construct. Sense tags are added next to each corresponding word in the sentence and these sense tags are also introduced into the vocabulary of the language model.

Since a pretrained language model like BERT has 30000 words in its vocabulary, it is important that the number of sense tags added to the vocabulary is not too large. To this end, sense vocabulary compression methods based on semantic relations between senses are used (Vial et al., 2019). This results in 9,500-13,000 unique sense tags, down from 206,941 unique senses. Whether to add the sense tag before or after the corresponding word in the sentence and if that makes a difference still needs to be tested. As a default, taking inspiration from adjectives and adverbs, the sense tag is added before the word.

Sense annotated corpuses, SemCor and WordNet GlossTag (WNGT), are used for training. Language model is trained by randomly masking some words in the sentence and calculating loss on how well the model predicts the correct word. At prediction time, sense tags are masked in the sentence if the corresponding word has the possibility of having an alternative sense tag. If the word has only one possible sense tag, that sense tag is not masked. This results in multiple masked places in

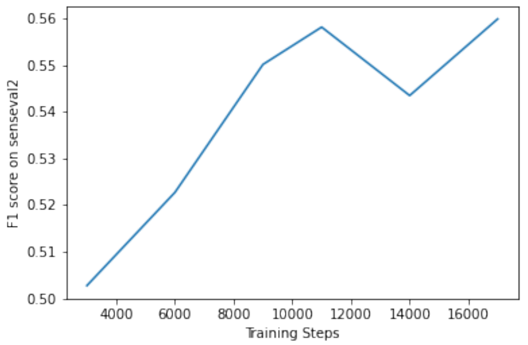

Figure 1: F1 score of sense prediction for Senseval-2 dataset at different stages in training. Each training step consists of a batch of 32 and one epoch is completed in 600 training steps

Method	Senseval-2
First Sense Baseline	65.6
SVC (Vial et al., 2019)	79.7
Our Method	56

Table 1: Comparison of F1-scores from our method against the current state of the art.

the sentence and language model is used to predict scores for all its vocabulary at each masked position. Since at each masked position, only a subset of sense tags is possible, scores for this subset of sense tags is recorded and the sense tag with highest score is chosen.

3 Results

A pre-trained BERT model (bert-base-cased) with 12 layers and 110M parameters was fine-tuned on training data of 150,000 sentences with embedded sense tags. Model was trained for 28 epochs on a TPU with a batch size of 32 and while further improvement was still possible, training had to be stopped over a lack of compute resources and time. Performance of the model was evaluated on Senseval-2 corpus over the period of training. Metric chosen for evaluation is F1-score averaged over all sense tags. Improvement of F1 score over the training steps can be seen in Figure 1. As seen in the figure, performance of the model continuously increases over the period of the training with 1 exception. Table 1 shows the results of our trained model as well as the state of the art results.

To understand if the model is really learning the sense embeddings and that sense embeddings different sense tags are distinguishable from each other, t-SNE is used to plot the embeddings of most frequently occurring sense tags in Senseeval-2 dataset. After reducing the embeddings to two-dimensions with t-SNE, a scatterplot is shown in Figure 2 with sense tags represented with different colors. As seen in the figure, embeddings for each sense tags are grouped well together and do not overlap with any other sense tag.

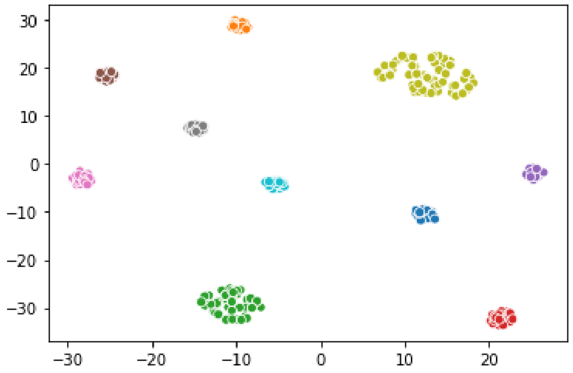

Figure 2: t-SNE plot of BERT-based embeddings of several sense tags present in Senseval-2. Each color represents a unique sense tag.

4 Discussion

While initial findings prove this to be a workable approach, the results are much below the state of the art and even the baseline results. Amount of data available for training is low and that possibly affects the convergence of a big language model like BERT. Adding OMSTI corpus for training can be interesting. Another big factor is the training setup itself and the loss used by masked language modelling. It is expected that the best results may be achieved by only masking sense tags during training of the language model and modifying loss to account for the limited number of possible senses at any masked position. Such an implementation is out of the scope of this work and will be looked into in the future.

Despite low F1 score, it can be seen that the model does start to learn the embeddings for different sense tags. Even if directly including the sense tags into the language does not prove to be a valid approach, sense embeddings learned by the model can replace the typical softmax classification method where each sense tag is a separate class and the classifier has to predict almost 11000 unique classes. Using sense embeddings, this problem can instead be converted into regression of 1000 numbers (embedding size).

References

Daniel Loureiro and Alipio Jorge. 2019. Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation. *arXiv preprint arXiv:1906.10007*.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *AAAI*, pages 8758–8765.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. *arXiv preprint arXiv:1905.05677*.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint arXiv:1909.10430*.