Using BERT for Unsupervised Word Sense Disambiguation

First Author
Iker Garca Ferrero
igarcia945@ikasle.ehu.eus

Abstract

Word sense disambiguation consists of associating words in context with their most suitable entry in a pre-defined sense inventory. In this paper, we will explore the possibility of using contextualized word representations such as BERT to improve the results in the word sense disambiguation task. We want to achieve an unsupervised method that can successfully perform word sense disambiguation. To be able to implement such a system we will study two different approaches, directly assign BERT and WordNet and using BERT to improve the performance of the UKB toolkit. We hope that this work can serve to find possible future lines of research.

1 Introduction

Word sense disambiguation (WSD) attempts to identify which sense of a word is used in a sentence. We understand as senses the different meanings a word has in different contexts. The most used sense inventory for English in WSD is WordNet (Miller, 1995). For example, given the word "mouse" and the sentence: "the mouse was being chased by the cat" we would assign "mouse" with the small rodents sense (the 1st sente in WordNet). Understanding the word the correct word sense given a word in a sentence is a crucial task in natural language processing (NLP). A system being able to accomplish this task would probe that it can understand the meaning of a word in a given context, that is, understand a sentence. Unfortunately, while WSD is generally a very easy task for a human, it is a very challenging task for automatic systems and it remains an open task in NLP.

Current WSD systems assign a sense to a word by taking into account the other words in the sentence. Multiple systems have been proposed. (Ando, 2006; Ng et al., 2003; Zhong and Ng, 2010) made use of discrete word features, which involves training a classifier using surrounding words and collocations. This classifier can be improved making use of continuous word representations of the surrounding words (Taghipour and Ng, 2015; Yuan et al., 2016). Neural based systems have become the state of the art in most of the NLP task (Otter et al., 2018), multiple authors have also explored the possibility of feeding continuous word representations into a neural network that learns abstract representations of the whole sentence and the words in the sentence (Kågebäck and Salomonsson, 2016; Raganato et al., 2017; Luo et al., 2018). Another line of research makes use of knowledge-based systems, these systems use the information encoded in knowledge bases such as WordNet (Miller, 1995) or BabelNet (Navigli and Ponzetto, 2012) to perform WSD (Agirre et al., 2014; Moro et al., 2014).

However, as shown in (Ruder, 2019), WSD systems have reached an upper-bound performance. All the systems achieve similar performance and this performance has not improved recently. One of the causes for this is that all the systems make use of word representations that are independent of the context. Recently, contextualized word embeddings (Melamud et al., 2016; McCann et al., 2017; Peters et al., 2018; Devlin et al., 2018; Radford et al., 2019; Yang et al., 2019) have been shown to improve the performance of many NLP tasks. Pre-trained contextualized word representations are publicly available. These representations are obtained through neural sentence encoders trained on a huge amount of texts. These representations are yet to be tested in WSD.

In this work, we explore the possibility of using contextualized word representations (BERT (Devlin et al., 2018)) in the WSD task. Given a sentence where a word has been masked, for exam-

Sentence	BERT prediction
the [MASK] of my computer does not work, I can not write anything	keyboard
the [MASK] of my computer does not work, I can not see anything	screen
Ben wanted to eat so he went to a [MASK] near his house	restaurant
artificial intelligence should always [MASK] humans	help

Table 1: Bert predictions for masked words (model: bert base uncased)

ple, the sentence "Ben wanted to eat so he went to a [MASK] near his house" BERT will predict that the most probable word the masked position is "restaurant". This shows that BERT has a very deep understanding of the meaning of the sentences, and we think that knowledge can be very helpful for the WSD systems. The main objective of this work is to explore how can we take advantage of the knowledge encoded in BERT to study future research lines in this topic. The following sections are organized as follows: section 2 describe the methodology of the tests, in this section, we will describe the resources that we have used, the implemented systems and the evaluation framework. section 3 present and discuss the obtained results. Finally section 4 concludes and discusses future work.

2 Methodology

In this section, we will describe the resources, algorithms, models and datasets used.

2.1 Using Bert for WSD

The contextualized word representations used in this word is BERT (Devlin et al., 2018), which is a bidirectional transformer encoder model (Vaswani et al., 2017) that has been trained in a huge corpus. This model is trained in two tasks, masked words and next sentence prediction. In both tasks, prediction accuracy is determined by the ability of the model to understand the context.

In this work, we will exploit the ability of BERT to predict masked sentences. Given a sentence an a masked word, BERT can predict the most probable words in the masked position. As an example, Table 1 shows the predictions of the BERT uncased base model for some masked words. Bert can also be used without masked words, that is, given a sentence, it can predict the words that can better substitute the word in a given position. For example, given the sentence 'the [mouse] eats cheese" BERT predicts that the most suitable words to substitute mouse are mice, cat, crow, worm, child, bird, fox, rat, rabbit, minor. We will use BERT in this way, given a sentence, we will predict the 10 most suitable words to substitute the words that we want to disambiguate. We will use for all the experiments the BERT base uncased model, we decided to use the smaller version of BERT for this work because the bigger models can make the required computations very slow.

In the previous example, given the word mouse and the list of words mice, cat, crow, worm, child, bird, fox, rat, rabbit, minor, every human will be able to easily assign to the word mouse the small rodents sense of WordNet. However, we need to implement an algorithm to be able to do this on a computer. We have tried two different approaches, the first one makes direct use the WordNet graph, and the second one makes use of UKB (Agirre et al., 2014), a toolkit to perform graph-based word sense disambiguation using random walks over a knowledge base (we will use WordNet).

2.1.1 BERT+WordNet

In this approach, we will make use of the WordNet graph. Given all the senses of the word that we want to disambiguate (target), and a list (L) containing all the senses of the 10 words predicted by BERT, we want to calculate which of the senses of the target word is closer to the senses in L. Given a sense T1 of the target word and a sense S1 of the list L, we will calculate the distance between them using 4 different metrics.

- **path similarity** ($BertWN_{PS}$): Min path similarity calculated using NLTK in the WordNet graph between T1 and S1.

- **distance to lowest common hypernym** ($BertWN_{PS+LCH}$): Min sum of the path similarity calculated using NLTK in the WordNet graph between T1 and the lowest common hypernym between T1 and S1 and S1 and the lowest common hypernym between T1 and S1.

- **nearest lowest common hypernyms** ($BertWN_{LCH}$): Min sum of the number of nodes between T1 and the lowest common hypernym between T1 and S1 and S1 and the lowest common hypernym between T1 and S1 using WordNet graph.
- **vote nearest lowest common hypernyms** ($BertWN_{Vote.LCH}$): Similar to $BertWN_{LCH}$, but instead of calculating the distance between every sense of the target word and every sense in L and selecting the min. We will calculate the $BertWN_{LCH}$ between every sense of the target word and every sense each one of the 10 predicted words. That is, we will obtain a list containing 10 results, one for each predicted word, and we will select the sense that appears more times in the list.

2.1.2 BERT+UKB

In this approach, we will make use of the UKB toolkit. UKB receive as input a sentence and the word that we want to disambiguate and it will apply random walks over WordNet to disambiguate the word. We will try to improve the performance of UKB using BERT. To do that, given a sentence and a word to disambiguate, we will calculate the 10 most suitable words that can replace the word that we want to disambiguate using BERT. Then we will generate a new sentence containing the first 5 most probable words, the word to disambiguate and the next 5 most probable words. This new sentence is going to be the input for UKB. As an example, if we have the sentence "the [mouse] of my computer does not work" the 10 predicted words by BERT are screen, keyboard, rest, power, computer, monitor, display, battery, memory and back. So we will generate the sentence "screen keyboard rest power computer mouse monitor display battery memory back" that will be the input to UKB. We expect that UKB will perform better disambiguation using the generated sentences. For UKB we use the default parameters.

2.2 Dataset

We will use the Semeval 2007 WSD dataset to evaluate our models. We decided to use this dataset because it provides a sentence and the words to disambiguate. More recent datasets only provide a sentence and the system needs to identify the ambiguous words, which can be multiword expressions. To avoid dealing with this to be able to completely focus in the WSD task we have chosen this dataset. The Semeval 2007 dataset uses the sense inventory of WordNet.

System	precision	recall	f1
Unsupervised methods			
$BertWN_{PS}$	22.8	17.4	19.7
$BertWN_{PS+LCH}$	23.1	17.4	19.8
$BertWN_{LCH}$	47.0	40.0	43.2
$BertWN_{Vote.LCH}$	47.5	33.4	39.2
BERT+UKB	51.9	51.9	51.9
UKB (Agirre et al., 2014)	**53.2**	**53.2**	53.2
Babelfy (Moro et al., 2014)	-	-	51.6
Supervised methods			
context2vec (Melamud et al., 2016)	-	-	61.3
supWSDemb (Papandrea et al., 2017)	-	-	**63.1**
ELMo (Peters et al., 2018)	-	-	62.2

Table 2: Results of the systems in the SemEval 2007 dataset

3 Results

Table 2 Shows the results scored by the different systems in the SemEval 2007 WSD dataset. As we can see the models using BERT and WordNet do not achieve the same performance as UKB. $BertWN_{LCH}$ and $BertWN_{Vote.LCH}$ achieve a precision relatively high with means that the systems can perform WSD, more complex distance metrics could be able to achieve a performance similar to other unsupervised methods. However, the implemented algorithms fail to give an answer to every word for disambiguate in the dataset, this causes a lower recall and f1 score. I think that these algorithms can be improved to be able to give a response to every word that we want to disambiguate. On the other side using BERT to generate the input of UKB does not improve the performance of UKB.

4 Conclusions

We have implemented and evaluated different systems that make use of BERT to perform WSD. Contrary to the initial expectations we have not been able to achieve good results. Even so, contextualized word representations have great potential and I believe that they can be very useful in the WSD task. I think that implementing an unsupervised method that makes use of these representations that achieves state-of-the-art result is possible, however, the systems implemented in this word are not good enough to successfully use the knowledge encoded in BERT. Also, I think that the combination of BERT and UKB has great potential, but I also think that to be able to successfully

combine both of them some modifications to the UKB algorithm should be made to make it able to make a better usage of the knowledge encoded in BERT, more specifically to make a better use of the 10 most probable word predicted by BERT.

References

Eneko Agirre, Oier Lpez de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Rie Kubota Ando. 2006. Applying alternating structure optimization to word sense disambiguation. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 77–84. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional lstm. *COLING 2016*, page 51.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pages 51–61.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Hwee Tou Ng, Bin Wang, and Yee Seng Chan. 2003. Exploiting parallel texts for word sense disambiguation: An empirical study. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 455–462, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. 2018. A survey of the usages of deep learning in natural language processing. *CoRR*, abs/1807.10854.

Simone Papandrea, Alessandro Raganato, and Claudio Delli Bovi. 2017. Supwsd: A flexible toolkit for supervised word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 103–108, Copenhagen, Denmark. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Sebastian Ruder. 2019. Word sense disambiguation nlp-progress. `http://nlpprogress.com/english/word_sense_disambiguation.html`. Accessed: 2019-07-20.

Kaveh Taghipour and Hwee Tou Ng. 2015. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 314–323.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. *arXiv preprint arXiv:1603.07012*.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 system demonstrations*, pages 78–83.