International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 3, pp: 370-373 www.ijaem.net ISSN: 2395-5252

BERT Approach for Word Sense Disambiguation

Anu P C, Rameez Mohammed

Computer Science and Engineering GEC Palakkad Palakkad, India
Computer Science and Engineering GEC Palakkad Palakkad, India
Corresponding Author: Anu P C

Date of Submission: 25-07-2020 Date of Acceptance: 05-08-2020

ABSTRACT—Word Sense Disambiguation (WSD) in Natural Language Processing (NLP) may be defined as the ability to determine which meaning of the word is activated by the use of word in a particular context. Words have different meanings based on the context of its usage in the sentence, they are known as polysemy words. Examples for polysemy words are Tie, Book, Interest, Bank, etc. WSD is easy for humans, but it is challenging task for automatic systems and it remain an open problem in Natural Language Processing. Multiple WSD systems have been proposed and it uses discrete word features, which involves training a classifier using surrounding words and collocations. This classifier can be improved making use of continuous words of surrounding words. All have similar performance and not improved recently. One cause for this is that all the systems make use of word representation that are independent of context. Recently contextualized word embeddings have been shown to improve the performance of many NLP tasks. Pre-trained contextualized word representations are publically available, here BERT model is used. BERT stands for Bidirectional Encoder Representations from Transformers. Given a sentence where a word has been masked, BERT will predict the most probable word in the masked position.

Index Terms—WordNet, Bidirectional Encoder Representa- tions from Transformer (BERT), unsupervised learning, natural language processing.

I. INTRODUCTION

When a word is present in text, which has multiple meanings Word Sense Disambiguation is used. It is the task performed by a computer program where in a correct sense for a word in a part of a sentence or context is determined. Ambiguity is com- mon in natural language. Word Sense Disambiguation (WSD) deals with lexical ambiguity, they are known as polysemy in the sentences. Bank, Book, Interest, Mouse, etc are some examples of polysemy words. Understanding the correct sense of a word according to the context can be easily detected by humans, but WSD is a challenging task for automatic machines and it remains an open problem in NLP. Systems has to apply some algorithms and then find out the senses correctly. For example, consider the sentences She is a regular bass player and Bass is my favorite dish. According to both sentences bass has two different meanings, one is a musical instrument and another is something related to food, i.e some eatable thing.

The basic WSD task works on three procedures. First we have to rank the senses of a word, because there are lots of words in a context. So ranking is necessary to realize the importance of each word in the context. Second is the selection of the window to be considered, that means the length of the sentence is considered like Bag of Words methods. Finally the usage of knowledge base such as WordNet. The most used sense inventory in English is WordNet. WordNet is the lexical database specifically designed for NLP. A pretrained contextualized word representation such as BERT model is used for the WSD task. Given a sentence where a word has been masked, for example consider the sentence ”Artificial intelligence should always [MASK] humans”. BERT will predict the most probable word in the masked position is ”help”. This shows that BERT has very deep understanding of the many of sentences or context, and for sure this knowledge can be very useful for the WSD system. The input to the system are a sentence and an ambiguous word, and the output is target sense of the word.

Word Sense Disambiguation is an open problem in natural language processing and researches under this area is ongoing using various approaches. Figure 1 shows the basic idea of a Word Sense Disambiguation System. The input of the system are sentence and an ambiguous word, correct sense of the ambiguous word will be generated as the output.

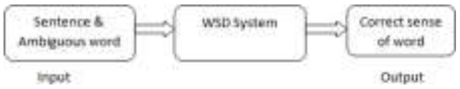

Fig. 1. Basic WSD Architecture

The WSD task is challenging because it is difficult to determine how much context to use to achieve the most accurate disambiguation. Determine if the word should be disambiguate for a more generic sense or for a finer sense in a given context is another challenge And also it is difficult define the senses of words and the level of detail represented by a particular sense with respect to sense usage.

In computational linguistics, Word Sense Disambiguation (WSD) is an open problem concerned with identifying which sense of a word is used in a sentence. The solution to this issue impacts other computer-related writing, such as discourse, enhancing relevance of search engines, anaphora resolution, coherence, and inference. The human brain is quite expert at word-sense disambiguation. That natural language is formed in a way that requires so much of it is a reflection of that neurological reality[5]. In other words, human language

succeeded in a way that reflects (and also has helped to shape) the innate potential provided by the brain's neural networks.

In computer science and the information technology that it enables, it has been a long-term provocation to develop the ability in computers to do natural language processing and machine learning. A rich variety of techniques have been investigated, from dictionary-based methods that use the knowledge encoded in lexical resources, to supervised machine learning methods in which a classifier is trained for each distinct word on a corpus of manually sense-annotated examples, to completely unsupervised methods that cluster occurrences of words, thereby inducing word senses. Among these, supervised learning approaches have been the most fortunate algorithms to date.

In the reminder of this paper Section 2 includes various neural methods to Word Sense Disambiguation; Section 3 includes detailed methodology of Word Sense Disambiguation using BERT

II. METHODS FOR WORD SENSE DISAMBIGUATION

This section deals with various methods for Word Sense Disambiguation.

A. Methodology-1 Sreelakshmi Gopal et al. 2016

The work[1] is done based on Malayalam Word Sense Disambiguation using Naive Bayes Classifier. Supervised ma- chine learning technique is used here. As compared to other methods of WSD, it is easy to identify the sense for each ambiguous word in a sentence using Naive Bayes classifier.A corpus based approach has been adopted here in this work. The proposed system can give exactness in Malayalam word sense disambiguation system using machine learning approach than the previously used knowledge base learning approach. This system provide us 95% reliability using a corpora of 1 lakh words.

B. Methodology-2 Alexander Popov et. al. 2017

This methodology[2] is referred to as RNN (Recurrent Neural Networks) for word sense disambiguation. The model is inspired by the very successful recent applications of LSTM cells to NLP problems. This paper also explores the utility of combining word embeddings learned from a large corpus of text with lemma embeddings learned from an artificially generated corpus based on a knowledge resource (WordNet). A comparison with the bestscoring systems on a popular evaluation dataset shows that the neural network is well- positioned with respect to them. The fact that the addition of the lemma embeddings from the pseudo-corpus improves significantly the results, signals that they could be further boosted by exploring different feature spaces and combinations of them.

C. Sawan Kumar et. al. 2018

This method[3] is an Extended WSD Incorporating Sense Embeddings (EWISE), a supervised model to perform WSD by predicting over a continuous sense embedding space as opposed to a discrete label space. This allows EWISE to generalize over both seen and unseen senses, thus achieving generalized zeroshot learning. To obtain target sense embed- dings, EWISE uses sense definitions. EWISE learns a novel sentence encoder for sense definitions by using WordNet relations and also ConvE, a newly suggested knowledge graph embedding method. BiLSTM is used to encodes the context of a word to be disambiguated.

D. Methodology-4 Loc Vial et al. 2019

Sense vocabulary compression through the semantic knowl- edge of WordNet for Neural Word Sense Disambiguation[4] gear the issue of the limited quantity of manually sense annotated corpora for the task of word sense disambiguation, by exploiting the semantic relationships between senses such as synonymy, hypernymy and hyponymy, in order to compress the sense vocabulary of Princeton WordNet, and thus reduce the number of different sense tags that must be observed to disambiguate all words of the lexical database. BERT large model is used. For training SemCor and The SemCor, and the concatenation of the SemCor and the Princeton WordNet Gloss Corpus (WNGC) are used.

III. WORD SENSE DISAMBIGUATION USING BERT

For finding the correct sense of a word in a particular context or sentences, many WSD approaches have been ap- plied like knowledge based, supervised, semi-supervised and unsupervised methods also. Apart from all these techniques BERT is a new and efficient procedure for this task.

The system architecture is shown in the Figure 2. The proposed system mainly have two stages, word prediction using BERT and make use of WordNet for finding the correct sense. $\text{BERT}_{\text{BASE}}$ model is used for word prediction. The sentence is fed into the BERT model after the completion of tokenization and preparation of the data, then the predicted word is passed to the WordNet to get the senses. Input to the system is a sentence. Output from this system is correct sense of the ambiguous word. Each stage is described in detail in the following sections.

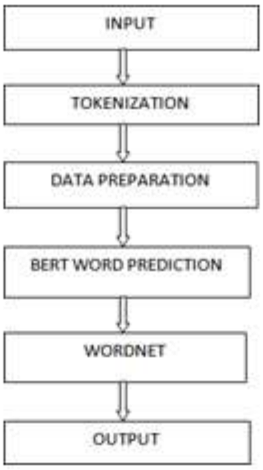

Fig. 2. System Overview

A. BERT for WSD

BERT model is pre-trained on a immense corpus and two novel unsupervised prediction tasks, that means, masked language model and next sentence prediction tasks are used in pre-training. In both tasks, prediction accuracy is determined by the capability of the model to understand the context. When incorporating BERT into downstream tasks, the fine-tuning procedure is recommended[6]. We fine-tune the pretrained BERT model on WSD task.

In this work, 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.Given a sentence where a word has been masked, for example consider the sentence ”Artificial intelligence should always [MASK] humans”. BERT will predict the most probable word in the masked position is ”help”. This shows that BERT has very deep understanding of the many of sentences or context, and for sure this knowledge can be very useful for the WSD system. Bert can also be used without masked words, that is, given a sentence and the position of ambiguous word, BERT will predict the suitable word in that position.

For example, given the sentence ”the [MASK] eats cheese” BERT predicts that the most suitable words to substitute MASK are boy, man, dog, child, girl, bird, animal, creature, one, bear and women. Utilizing BERT in this way, given a sentence, BERT will predict the 10 most suitable words to substitute the words that we want to disambiguate. BERT base uncased model is used in this experiments, that means the smaller version of BERT for this work because the bigger models can make the required computations very slow.

B. BERT and 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) holding all the senses of the 10 words predicted by BERT, we want to calculate which of the senses of the target word is adjacent to the senses in L. Given a sense T1 of the target word and a sense S1 of the list L, we will compute the distance between them using 4 different metrics.

- **Path Similarity**: Minimum path similarity calculated using NLTK in the WordNet graph between T1 and S1.
- **Distance to lowest common hypernym** : Minimum 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**: Minimum 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**: Similar to nearest lowest common hypernyms, but instead of calcu- lating the distance

between every sense of the target word and every sense in L and selecting the minimum.Calculated the nearest lowest common hypernyms between every sense of the target word and every sense each one of the 10 predicted words. That is, there will be a list containing 10 results, one for each predicted word, and will select the sense that appears more times in the list.

IV. CONCLUSION AND FUTURE SCOPE

Word Sense Disambiguation is a significant and challenging NLP issue. NLP applications include Machine Translation, Question answering, Information retrieval, Information extrac- tion etc. System to find which sense of a word is activated in a sentence is the difficulty. We proposed the semantic properties of contextualized word embeddings (CWEs) to address word sense disambiguation. System outputs the correct sense of the ambiguous word according to the given context. Integrating WordNet with BERT has proven to be an effective framework for encoding linguistic knowledge and improved performance in word sense disambiguation. The objective of this work is to improve output by using BERT method.

For further investigation, planing to use the relations be- tween senses, like hypernym and hyponym, to provide more accurate sense representations. Also we will investigate if more powerful classification algorithms for WSD based on contextualized embeddings and BERT are able to solve the issue (fnding the target sense) even in cases of extremely sparse training data.

REFERENCES

[1]. S. Gopal and R. P. Haroon, ”Malayalam word sense disambiguation us- ing naive bayes classifer,” in 2016 International Conference on Advances in Human Machine Interaction (HMI), pp. 14, IEEE, 2016.

[2]. A. Popov, ”Word sense disambiguation with recurrent neural networks,” in Proceedings of the Student Research Workshop associated with RANLP, pp. 2534,2017.

[3]. S. Kumar, S. Jat, K. Saxena, and P. Talukdar, ”Zero-shot word sense disambiguation using sense defnition embeddings,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 56705681, 2019.

[4]. L. Vial, B. Lecouteux, and D. Schwab, ”Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation,” arXiv preprint arXiv:1905.05677, 2019.

[5]. D. Yuan, J. Richardson, R. Doherty, C. Evans, and E. Altendorf, ”Semisupervised word sense disambiguation with neural models,” arXiv preprint arXiv:1603.07012, 2016.

[6]. L. Huang, C. Sun, X. Qiu, and X. Huang, ”Glossbert: Bert for word sense disambiguation with gloss knowledge,” arXiv preprint arXiv:1908.07245, 2019.

International Journal of Advances in Engineering and Management
ISSN: 2395-5252

IJAEM

Volume: 02 Issue: 01 DOI: 10.35629/5252

www.ijaem.net

Email id: ijaem.paper@gmail.com