CS391D Final Project: Does BERT Use Syntax in its decisions?

Michael Li, Shijing Zhong
eid:mal4565, sz6539
{michael.li, zsjgary}@utexas.edu

Abstract

Different word embedding has been proven to be essential in many tasks of the Natural Language Processing. Information such as world knowledge and syntax are encoded into the embedding, but the relationship between these information and performance of the model are unclear and not well-defined. There are plenty of papers noted on the existence of syntactic structure inside embeddings such as ELMO and BERT, but we don't know whether the model is using that knowledge. Given the structural probe for finding syntax from embedding, we wanted to quantify how much of that syntax knowledge the model uses to make its decisions of the output.

1 Introduction

Neural Networks have been very effective in making huge leaps in performance in machine learning. With the advent of GPUs and their usage with neural network training starting with AlexNet (Krizhevsky et al., 2012), their usage has exploded. It started in computer vision, but it also spread to natural language processing (NLP). With lots of work being done in the field, the most dominant approach has primarily dealt with Transformers.

Transformers (Vaswani et al., 2017) and BERT (Devlin et al., 2018) have really transformed the different approaches people take when trying to solve NLP tasks. We only know at a high level why these method work such as "attention" provides a way for the model to focus on particular parts of the input. However, despite these approaches' effectiveness, we do not know specifically why these approaches work very well. There has been prior work using probes to examine whether or not there exists forms of language within the model or embeddings. Our work will build off the structural probes work (Hewitt and Manning, 2019).

2 Related Work

(Hewitt and Manning, 2019) find evidence of syntax trees in BERT embeddings. Specifically, they are able to project the BERT embeddings of all the words in a sentence into a smaller dimensional space and extract a minimum spanning tree that connects all of the words in the new embedding space. They then use two evaluation metrics to determine how close this minimum spanning tree is to the true syntax tree. Their findings is that there is a good amount of the syntax tree accounted for in the minimum spanning tree. Our work sets out to ask the question: if we can find the syntax tree embedded in the word embeddings, does that necessarily mean BERT uses that knowledge with the classification token to determine its final predictions in classification problems?

3 Data

3.1 Tree Distance Evaluation Metrics

The data we use to train the structural probe is the Penn Tree bank which contains gold parse syntax tree as the ground truth of the probe. The Penn Tree bank data set is used for training and evaluating a baseline structural probe. From the original experiment of the structural probe (Hewitt and Manning, 2019), the metric for evaluating how well each embedding encodes the syntax structural is the percent of undirected edges placed correctly—against the gold tree. Therefore, the gold parse syntax tree derived from the Penn Tree bank will be our benchmark on other data set.

3.2 GLUE Benchmark

The test data includes several GLUE Tasks(Wang et al., 2018) such as The Corpus of Linguistic Acceptability(CoLA) and other language understanding tasks built on established existing datasets and selected to cover a diverse range of dataset sizes, text genres, and degrees of difficulty. GLUE is able to objectively evaluate the performance of the word embedding. After fixing a control model for the benchmark, we will compare the Tree Distance Evaluation Metrics of the embedding over the corpus with its correspondence GLUE score.

4 Approach

At a high level, we want to compare the performance of BERT on various data sets with the accuracy of the structural probe on those same sentences. This involves two main components. The first component is that we would either take the accuracy scores on the test set for the various data sets or run BERT on those data sets and evaluate ourselves. The second component has two main parts. The first part is that we would want to take the SOTA syntax parsers and construct the ground truth syntax trees. The second part is that we would want to take the fully trained structural probe and evaluate in a similar way to that paper with parse distance and parse depth.

These two components give us enough information to tell us how well BERT is doing on the specific task at hand and enough information on how well BERT is encoding the syntax trees. With these two pieces of information, we are able to determine a correlation between how well the task is performing and how much of the syntax tree is encoded into BERT. This would help us answer the question: does BERT use its knowledge of the syntax tree in its classification problems?

4.1 Syntax Trees

5 Current Progress

1. We are able to run the structural probes and calculate the parse distance and parse depth.

2. We are also able to load up BERT with Hugging Face and tokenize all of the corresponding inputs.

3. We are currently trying to recreate an experiment in (Hewitt and Manning, 2019).

4. We are able to train a structural probe with our own parameters.

5. We do have a list of data sets we want to run our evaluations on such as Corpus of Linguistic Acceptability (CoLA), Stanford Sentiment TreeBank, Microsoft Research Paraphrase Corpus, and Multi-Genre Natural Language Inference. We're keeping it somewhat limited so that we can do more involved analysis on the results we end up generating.

6 Results and Evaluation

We do not have enough results to make any conclusions yet.

However, what we do hope to evaluate with is to calculate a Pearson Correlation Coefficient between the performance of the data set for each data point and the performance of the structural probe on the extracted syntax trees.

In particular, the analysis would contain some form of the magnitude of correlation between the accuracy of the task and the accuracy of the syntax tree. Perhaps we would also run a significance test on how closely correlated the two accuracies are.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805. http://arxiv.org/abs/1810.04805.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., pages 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR* abs/1706.03762. http://arxiv.org/abs/1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding.