Explaining BERT for Paraphrase Identification

Shailesh Mani Pandey
shailesh.pandey@utexas.edu

VietDuy Nguyen
vietduy.nguyen@utexas.edu

Abstract

BERT is a relatively new pre-trained language representation model that achieves near state-of-the-art performance in many NLP benhcmarks. We specifically look at its performance on paraphrase identification task. We explain the predictions made by BERT on specific examples and also test its generality and robustness. We show that although BERT achieves competitive performance and is fairly robust, it suffers from lack of generality across datasets, results in asymmetric behaviour on the inherently symmetric task, and has hard time interpreting numbers present in the input text. We believe that an exhaustive explanation of a model's predictions can help us to understand its limitations, improve its training procedure and ultimately extract better results.

1 Introduction

Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) is a language representation model designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, including the General Language Understanding Evaluation (GLUE) (Wang et al., 2019) benchmark.

The unprecedented success of BERT in NLP tasks has motivated a growing area of research that investigates the features of language that BERT learns from the unlabeled pre-training data. In addition to helping us understand and reason about the model, explanation also enables us to improve the performance of our model. For example, understanding that key words are missed, or not so important words are repeated in the output of a decoder in seq2seq models enabled us to improve their performance by using attention mechanisms.(Vaswani et al., 2017) Similarly, realization that a model only looks at one-sided context motivated the origin of bidirectional models such as BERT. Moreover, at times, although the evaluation metrics may suggest a good performance of the model, the model may be just learning statistical information about the dataset rather than general characteristics of the task itself. In such cases, explanation of the model can help us to realize such undesired learning and correct course.

In this work, we specifically try to understand and explain the performance of BERT on the paraphrasing task. Two sentences are termed as *paraphrases* of each other if they express the same meaning using different words. Given a pair of sentences, the goal is to identify whether the sentences are paraphrases of each other. Paraphrase identification is an important problem that finds application in many other NLP tasks, such as question answering (QA), machine translation (MT), semantic text similarity, and plagiarism detection. Moreover, a good paraphrase identification system can help in generating paraphrases of better quality by filtering from a set of generated paraphrases. We find that the performance of $\text{BERT}_{\text{BASE}}$ fine-tuned on the task gives performance close to the current state-of-the-art (GLUE, 2019 (accessed Dec 10, 2019). Given this good performance of BERT on this task and its similarity to the Next Sentence Prediction (NSP) task used in pre-training BERT, we expect this simple task to give us useful insights into the functioning of BERT.

We try to understand the features of input that BERT uses to make its decision by using model-agnostic techniques, such as LIME (Ribeiro et al., 2016). This means our set of experiments to qualitatively analyze and explain the predictions of a classifier generalize to any other classifier just as well as to BERT. Specifically, we design and conduct experiments to test the following three hypotheses regarding BERT:

1. BERT generalizes well across different datasets of the same task.

2. BERT assigns reasonable relative importance to different words in the input text.

3. BERT is resilient to minor (but common) grammatical mistakes in the input text.

The rest of this report is structured as follows. Section 2 discusses previous work related to this project and also covers the background material in brief. In section 3, we describe the datasets that we use in this project. Section 4 describes the design and rationale of our experiments while section 5 discusses the results and inference that we draw from these experiments. We mention our ideas about possible future work in section 6 and conclude with section 7.

2 Related Work

2.1 BERT

In 2018, BERT was introduced by Google as a new pre-training language representation model (Devlin et al., 2018). There have been a significant number of such models. One of the most recent ones is ELMo (Peters et al., 2018). ELMo used a mechanism of extracting context-sensitive features in multidimensional model to generalize word-embedding representations. In other words, ElMo approach is feature based and not deep bidirectional (Devlin et al., 2018). BERT, on the other hand, pre-trains a multi-layer deep bidirectional representation model from unlabeled text by multidimensional conditioning. The work is based on the original implementation of (Vaswani et al., 2017). There are various BERT models such as $\text{BERT}_{\text{BASE}}$ or $\text{BERT}_{\text{LARGE}}$. In the scope of this project, we only investigate on $\text{BERT}_{\text{BASE}}$ (number of layers = 12, hidden size = 768, the number of self-attention heads = 12, and the total parameters = 110M)(Devlin et al., 2018)

2.2 Paraphrase Identification

The early exploration of this task mainly relied on conventional methods and small scale datasets. The availability of multiple large datasets with thousands of human annotated sentence pairs has enabled a wide application of neural networks to this task. (Choi et al., 2018) use cell-aware stacked LSTMs while (Yang et al., 2019) use a residual convolution network with rich alignment features. The current state of the art method for this task is achieved by a multi-task deep neural network (MT-DNN) that uses a multi layer bidirectional transformer encoder (Liu et al., 2019). This encoder in MT-DNN learns the representation using multi-task objectives, in addition to pre-training stage like that of the BERT model. Recent advances made by using pre-trained BERT in many NLP tasks motivates us to further explore and explain its performance.

2.3 Interpreting deep neural networks

Interpreting a deep model has attracted many researchers due to the growth in the number of work that employed deep neural network models, as well as policy and legal ramifications (Goodman and Flaxman, 2017). There are several ways to interpret a model.

Firstly, we can look at the weights of features in the trained neural network model. However, this can be complex for large models such as BERT where even the basic model has more than 100 million parameters to be learnt. Scrutinizing such a large number of parameters and identifying those germane to the output requires extensive effort and resources.

The second approach is employing counterfactual to learn the impact on the model's prediction at a smaller scale (sentence $\rightarrow$ words, or images $\rightarrow$ separable components). The LIME model, Locally-interpretable Model-Agnostic Explanations, proposed by (Ribeiro et al., 2016) analyzes those impacts by a sparse linear mechanism. To be more specific, LIME looks at one example at a time to interpret the input model and treats it as a black box (Ribeiro et al., 2016). In the first step, LIME breaks the example into a set of smaller components x' such that $x \in R^d \rightarrow x' \in \{0,1\}^{d'}$. Thereafter, LIME checks prediction of the input model on x'. Lastly, LIME trains a linear regression model to predict prediction based on x' and then looks at that model's weights to explain the decision.

As discussed later in section 4, we prefer this later approach for its simplicity and wider applicability. We use a modified version of LIME as one of our approaches to understand the prediction of BERT in paraphrase identification task.

3 Datasets

We use the two datasets for paraphrase identification task that are part of the GLUE benchmark: Quora Question Pairs and Microsoft Research Paraphrase Corpus

3.1 Quora Question Pairs

Quora Question Pairs (QQP) dataset consists of over $400,000$ lines of potential question duplicate pairs. Each line contains IDs for each question in the pair, the full text for each question, and a binary value that indicates whether the line truly contains a duplicate pair (Quora, 2017 (accessed Oct 14, 2019). However, the annotation of question pairs in Quora dataset is not entirely correct. Manually scrutinizing the dataset, we found some such examples. For instance, the following sentences are classified as paraphrases.

Sentence 1: Why don't I feel uncomfortable making eye contact?

Sentence 2: Why am I feeling uncomfortable to make eye contact?

Indeed, these sentences are negations of each other.

3.2 Microsfot Research Paraphrase Corpus

Microsoft Research Paraphrase Corpus (MRPC) dataset contains 5800 pairs of sentences which have been extracted from news sources on the web, along with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship (Microsoft, 2005 (accessed Oct 14, 2019).

MRPC is a short dataset with 408 examples in the development set which allows us agile experimentation with different models and input perturbations under constrained availability of resources. Hence, we perform most of our experiments on MRPC and use QQP only to test the generality of BERT.

4 Experiments

Before we describe the experiments to test the hypotheses stated in section 1, we describe our model and training in detail. We use the pre-trained $\text{BERT}_{\text{BASE}}$-cased (cased means we retain the case of each letter in the input) model that uses 12 hidden layers with 768 units each and 12 self-attention heads. This amounts to a total of around 110 million parameters. Although $\text{BERT}_{\text{LARGE}}$ is expected to attain better performance, it requires learning more than 340 million parameters. Hence, in the interest of time, we limit our experiments to $\text{BERT}_{\text{BASE}}$. As mentioned in (Devlin et al., 2018), the input to our model are the token indices of two sentences separated by the `[SEP]` token and prepended with a `[CLS]` token. The hidden state corresponding to the `[CLS]` token is fed to an output layer followed by a Softmax function for the classification task.

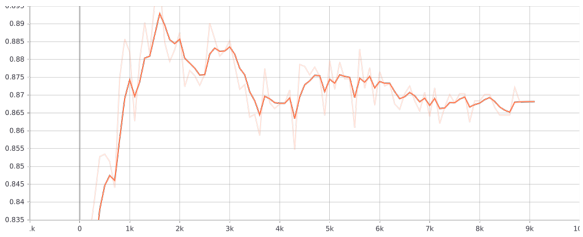

Figure 1: Combined accuracy and F1 score on the MRPC development set vs. number of training steps.

We fine-tune this pre-trained $\text{BERT}_{\text{BASE}}$-cased model on the MRPC dataset using Huggingface Transformers library (Wolf et al., 2019) with a decaying learning rate starting at $2e-5$ and the Adam optimizer (Kingma and Ba, 2014) for 20 epochs. We use 4 GeForce GTX 1080 Ti GPUs with a batch size of 2 examples per GPU. As shown by the graph in figure 1, we achieve the best performance of 91.88 F1 score and 88.48% accuracy on the development set after 1600 steps. Unless stated otherwise, we use this best performing model for the rest of our experiments. Also, we run all our experiments on the MRPC development set.

All our experiments are model-agnostic, that is to say we do not probe the internals of the model, such as weights or gradients. Instead, we perturb the input to our model and observe the change in the output of the model. This makes our experiment design applicable to any classifier. We favor this model-agnostic approach because investigating the model weights and gradients can get very complex for large models, such as BERT, as explained in section 2.

4.1 Sanity testing

We start the analysis by running two basic sanity testing experiments on our best performing trained model. Our rationale behind designing these experiments is that a human performing this task should not be affected at all by the changes that we make to our input.

In the first experiment, we replace the second sentence of all the examples in our development set by an exact copy of the first sentence. Since the two sentences in each example are now identical, we expect the model to always classify them as

paraphrases of each other.

For the second experiment, we interchange the positions of the sentences in each example in the development set. This means the previously first sentence in each example now becomes the second sentence of that example and vice versa. Since the paraphrase relation is symmetric, we expect the model to give same results as earlier for each example of the development set. Although this seems like an easy task to do for a human, this requires the model to learn that the parts of the input separated by the `[SEP]` token are interchangeable without any such specific indication while training.

4.2 Generalization across datasets

To test our first hypothesis that BERT generalizes well across datasets of the same task, we use the Quora Question Pairs (QQP) dataset, the only other paraphrase corpus listed in the GLUE benchmark. In our first experiment, we first fine-tune a pre-trained $\text{BERT}_{\text{BASE}}$-cased model on the QQP dataset. We compare the performance of this fine-tuned model on MRPC to the performance of the pretrained $\text{BERT}_{\text{BASE}}$-cased model without any fine-tuning. Further, we fine tune the best-performing QQP model on the MRPC dataset and compare its performance with our model that was fine-tuned just on MRPC. We expect the model to learn the common features from these datasets and be able to generalize those features to MRPC.

To test if BERT learns some useful features from the QQP dataset and is able to apply those to MRPC, we augment the original MRPC training set of 3668 examples with an equal number of training examples from the QQP dataset. It must be noted that we did not augment the development set. That is, we continued to judge the model based on its performance on the MRPC development set. Since both the datasets are for the paraphrase identification task and are from different sources, we expect this data augmentation to effectively work as a regularization scheme and help the model to learn more general rather than dataset specific features.

4.3 Importance of words

For our second hypothesis, we test if BERT assigns reasonable relevance to different words in the input. For these experiments, we keep our first sentence constant and perturb the second sentence. First, for each word in the dataset, we remove all its occurrences from the second sentence of all examples. We do this process one word at a time. We compare the predictions of the model with one word removed to the predictions of the model on the unaltered dataset. We expect the predictions of the model to be altered significantly on removing words that convey the core meaning of the sentence, such as verbs, and to be robust to removal of a few common stop tokens, such as 'a', 'the', ',' and '.'.

While our first experiment focuses on the weight of each individual word across the entire dataset, in the second experiment we focus on understanding the relative importance that the model places on words of a single instance. For this experiment, we modify LIME (Ribeiro et al., 2016) to take two sentences as input and only perturb the second sentence. We also modify it to only look at the second sentence in order to learn a sparse local linear model for the importance of tokens in model's prediction. For each instance, we use 5000 samples to learn the linear model and plot the top 6 most important words. We analyze the examples on which the model predicts the wrong label and compare those to the examples which the model predicts correctly.

4.4 Importance of grammar

In addition to the grammatical inconsistencies caused by our previous experiments, we wanted to look at the effect of manually introducing minor but common grammatical errors. By minor grammatical errors, we mean errors that do not alter the meaning of the sentence significantly. For this, we randomly select 20 examples from the MRPC development set and manually perturb the second sentence in each example to introduce minor grammatical errors, such as omission of articles, subject-verb disagreement, and inappropriate use of 'its' and 'it's'. We introduce $2-4$ such errors in each instance but are careful to not impact the meaning of the sentence with these changes. Following is one example of the sentence before and after we modified it:

Before: It has also said it would review all of its domestic employees more than 1 million to ensure they have legal status.

After: It has also said it **will** review all of **it's** domestic employees more than 1 million to ensure they **has** legal status.

As the reader would have noticed from this exam-

ple, some sentences are already ungrammatical. In this case we introduce more common grammatical errors. A well trained model should ignore these common grammatical errors and not alter its prediction for these perturbed examples.

5 Results and Discussion

In this section, we discuss the results of the experiments described in the previous section and draw out our inferences. We compare the results of all our experiments to the performance of our model on the unmodified MRPC development set. The model makes errors on 47 out of the 408 examples in the unmodified development set.

5.1 Sanity Testing

We find that our model classifies all the identical sentence pairs as paraphrases of each other, as expected. However, interchanging the positions of the sentences in the examples significantly alters the predictions of the model. 31 examples that were previously classified correctly are now misclassified while 21 examples on which the model previously made errors are now corrected. In total, the model now makes 57 errors on the development set which consists of 408 examples. This signifies that although BERT attains near state-of-the-art performance on this task, it learns some undesirable features that are based on the relative position of the sentence.

5.2 Generalization across datasets

Table 1 compares the performance of the $\text{BERT}_{\text{BASE}}$-cased model on the MRPC development set before and after fine-tuning on the QQP dataset. We expected the model to learn a few general features about the task and improve slightly after training on another dataset for the same task. However, as seen in table 1, the performance of the model actually deteriorates after fine-tuning on the QQP datset. This suggests that the features learnt (or adjusted) by BERT in the fine-tuning step are dataset specific.

To assert this further, we further fine-tuned this model on MRPC after fine-tuning on QQP. As seen in figure 2, BERT does not benefit from fine-tuning on QQP before MRPC or by mixing the datasets proportionally in the training dataset. This reinforces our inference that BERT does not learn any common features that are general to the paraphrasing task and hold across the datasets.

	Pre-trained	QQP-fine-tuned
Accuracy	68.38	69.61
F1	81.22	77.53
Acc & F1	**74.8**	73.57

Table 1: Performance of the pre-trained $\text{BERT}_{\text{BASE}}$-cased model on MRPC development set before and after fine-tuning on the QQP dataset.

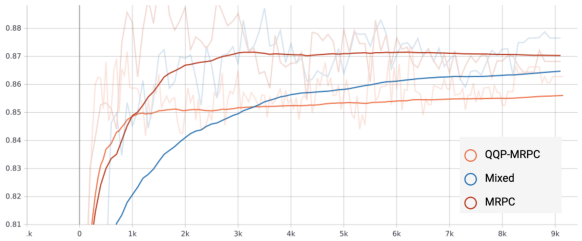

Figure 2: Combined accuracy and F1 score on the MRPC development set on: 1) fine-tuning directly on MRPC 2) fine-tuning on QQP followed by fine-tuning on QQP, and 3) fine-tuning on the mixed QQP-MRPC dataset.

Rather, the fine-tuning step modifies BERT to learn features specific to the dataset.

5.3 Importance of words

Table 2 shows selective results of our word-removal experiment for words that occur in at least 20 examples. The words are listed in increasing order of their importance. We calculate the importance of a word as the fraction of relevant examples for which our model changed its output on removal of that word. For this analysis, we ignore the case of the word. It can be seen that the model learns to ignore words such as 'is', 'are', and 'or' despite them occuring frequently in the dataset. It also learns to place a high importance on 'not', which is often a deciding token. Most of the words, such as 'talking', 'settle', 'california', 'only', occur in a small number of examples. As expected, removal of these words that convey core meaning of the sentence almost always affects the prediction of the model. In addition, we get similar scores for words with closely related semantics such as ['is', 'are'], and ['no', 'not'].

Interestingly, we noticed that removing some of the tokens actually helped the model and it made less errors. For example, removing ',' led to the model correcting its predictions in 5 instances. One such pair of setntences is mentioned below:

Word	Occurrences	Error rate	Importance
is	41	0.073171	0.000000
are	20	0.050000	0.000000
or	29	**0.206897**	0.000000
that	53	0.132075	0.000000
said	114	0.105263	0.008772
the	308	0.107143	0.012987
of	145	0.117241	0.013793
.	**407**	0.115479	0.017199
,	240	0.120833	0.033333
a	136	0.147059	0.036765
and	130	0.123077	0.038462
has	39	0.051282	0.051282
his	31	0.193548	0.064516
it	35	0.142857	0.085714
not	29	0.137931	**0.103448**

Table 2: Effect of removing all occurrences of a word from the second sentence of all examples of development set. 'Occurences' is the number of examples in which the word occurs in second sentence. 'Error rate' is the fraction of relevant examples for which our model made a mistake. 'Importance' is the fraction of relevant examples for which our model changed its prediction on removing that word.

Sentence 1: In the United States, heart attacks kill about 460,000 year, in Canada about 80,000.
Sentence 2: In the United States, heart attacks kill about 460,000 yearly, in according to the National Institutes of Health.
These sentences are not paraphrases becaues the first sentence contains additional information about heart attacks in Canada. Our best model predicts these unmodified sentences to be paraphrases but removing ',' from the second sentence helps and the model does not identify these as paraphrases. We believe that the aligned positions of different occurences of ',' in both the sentences tricks BERT to believe that these are paraphrases but removing ',' helps it focus on other more important features of the input.

Although this approach gives an idea of global importance that BERT associates with a word, this importance may be a result of the specific context in each sentence. For example, we see a high error rate of around 20% in sentences that contain 'or' but we find its importance to be 0. This is because the sentences containing 'or' are inherently more complex. To capture such sentence specifc

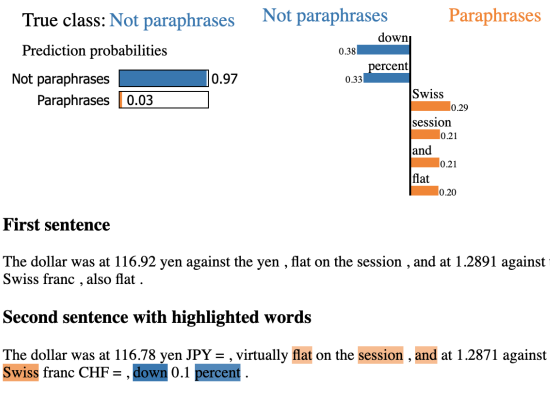

Figure 3: LIME based explanation of the model's correct prediction. The model correctly identifies words that differentiate the second sentence from first.

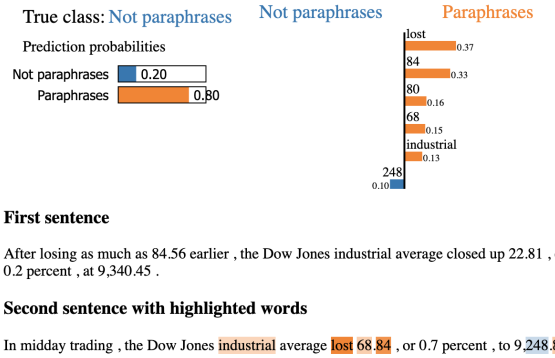

Figure 4: LIME based explanation of our model's wrong prediction. The model is unable to realize that the two sentences report different numbers.

importance, we modify and apply LIME(Ribeiro et al., 2016) to the paraphrase identification task as described in previous section.

Figure 3 shows the 6 most important words, as interpreted by LIME, in an example for which our model makes the correct prediction. The sentences are not paraphrases because the second sentence mentions that Swiss franc was down by 0.1 percent while the first sentence mentions that it was 'flat'. Although both the sentences have many similar words, the model assigns most importance to the two critical words, 'down' and 'percent' which convey information that is different from the first sentence. However, the model does not place much importance on the exact numbers quoted in the sentence which may be crucial for a human making this decision.

This problem of not being able to understand the numbers is further depicted in figure 4. This is

an example where the model makes the wrong decision. These sentences clearly state different information about 'Dow Jones'. The model focuses on similar semantics of the sentence and uses common words such as 'lost', and 'industrial' to identify this pair as paraphrases. BERT is able to correlate different forms of a verb and understands that 'lost' and 'after losing' convey the same meaning. Unfortunately, it is not able to correlate numbers as expected. BERT correlates 84 in 68.84 to 84 in 84.56 and considers that as an indication of these sentences being paraphrases. This relatively poor understanding of numbers shows up in the fact that BERT makes mistakes in many such stock market related sentences which deal with numbers.

From these set of experiments, we infer that although BERT learns to place more importance on key words that convey the core meaning of a sentence, it suffers from a few drawbacks such as inability of interpret numbers, and considering positional correlation of unimportant tokens as an indication of paraphasing.

5.4 Importance of grammar

We have already seen from our previous experiment of removing tokens from the second sentence that BERT often does not alter its prediction on removal of stop words, articles etc. This suggests that BERT is robust to some of the common grammatical errors that are introduced by omission of these tokens. To further investigate this, we randomly altered 20 examples in the devlopment set to introduce common grammatical errors, as explained in the previous section. We find that BERT changed its prediction for only 2 of these 20 modified examples. This further strengthens our inference that BERT is resilient to minor and frequent grammatical errors.

All these results indicate the following about our hypotheses stated in section 1:

1. Although the pre-training of BERT helps in learning general language features, the fine-tuning stage only learns datset specific features. Thus BERT fine-tuned for one task on a given dataset does not generalize to another dataset for the same task.

2. BERT is able to recognize relatively important words that convey core meaning of a sentence from the common filler words. However, it has a hard time in understanding and correlating numbers with the overall meaning of the text.

3. BERT is also prone to minor grammatical mistakes in the input text and its prediction does not change on introduction of multiple such minor and common grammatical mistakes.

In addition to these hypotheses, we must emphasize that the prediction made by BERT depends on the order of input sentences even when the task is inherently symmetric.

6 Future Work

All our experiments in this project fix one sentence while perturbing the other sentence. It will be interesting to look at the behaviour of BERT when both the first and second sentences are modified. Also, in this work, we look at BERT's performance in a model-agnostic manner. The next step in explaining BERT better can be to observe and understand the changes in attention weights with variations in the input.

Furthermore, we believe that these insights into BERT can help us in improving the training procedure and obtaining better results, especially on paraphrase identification task. We believe annotating the datasets to specifically mark the core words in a sentence and training BERT to mark the important parts of a sentence in addition to predicting the class label can improve BERT's ability to understand natural language. Moreover, modifying BERT's pre-training to include 'Previous Sentence Prediction' in addition to 'Next Sentence Prediction' can help to alleviate the problem of unsymmetric behaviour. Similarly, including other tasks such as correlating numbers to their textual representations can help to improve the ability of BERT to deal with numbers.

7 Conclusion

In this work, we observe and explain the behaviour of BERT on the task of paraphrase identification in a model-agnostic manner. We test the generality of finetuned BERT model across different datasets of same task, BERT's ability to identify important words in the input, and its robustness to minor grammatical mistakes. We find that although BERT achieves close to state of the art performance, is good at identifying important words, and is resilient to minor grammatical mistakes, it

does not generalize across datsets for the same task. Moreover, it suffers from some basic and severe limitations such as asymmetric behaviour and inability to understand numbers. We believe such exhaustive explainatory efforts can help us in improving the training procedures and extracting better performance out of our models.

References

Jihun Choi, Taeuk Kim, and Sang goo Lee. 2018. Cell-aware stacked lstms for modeling sentences.

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

GLUE. 2019 (accessed Dec 10, 2019). *GLUE Leaderboard*. https://gluebenchmark.com/leaderboard.

Bryce Goodman and Seth Flaxman. 2017. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine* 38(3):50–57. https://doi.org/10.1609/aimag.v38i3.2741.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 4487–4496. https://www.aclweb.org/anthology/P19-1441.

Microsoft. 2005 (accessed Oct 14, 2019). *Microsoft Research Paraphrase Corpus*. https://microsoft.com/en-us/download/details.aspx?id=52398.

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

Quora. 2017 (accessed Oct 14, 2019). *First Quora Dataset Release: Question Pairs*. https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. pages 1135–1144.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv* abs/1910.03771.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 4699–4709. https://doi.org/10.18653/v1/P19-1465.