Is BERT Looking at the Right Place? Attention Analysis on Adversarial QA Examples

Hannah Chen, James Ku, Lahiru Nuwan
Department of Computer Science
University of Virginia
yc4dx,jk2mf,lnw8px@virginia.edu

Abstract

In recent years, transfer learning methods and architectures for NLP has emerged and reached state-of-the-art result on a variety of NLP tasks. However, pre-trained language models like BERT [1] has shown to be vulnerable to adversarial examples [2] We would like to evaluate the over-sensitivity and over-stability of BERT towards adversarial examples by analyzing and visualizing the attention weights. We targeted on question answering task, and found that the attention weight does appear to be related to BERT's predicted answer.

1 Problem Statement

Jia and Liang (2017) [3] revealed that QA models fail to make correct predictions on adversarial examples. Their crafted "distracting" sentences target model's over-stability, which means that the model is too stable to identify semantic changes in syntactically similar sentences. On the other hand, models can also be over-sensitive to the adversarial inputs. In this case, the model is too sensitive to imperceptible noises added to the inputs even if the meaning of the input sentence does not change at all.

Vulnerability	Overly-sensitive	Overly-stable
Adversarial Input	Word or synonym substitutions / paraphrases	Word-swapping / Negations
Semantic Change	N	Y
Adversarial Example	Google bought YouTube. Google acquired YouTube. (Paraphrase) YouTube was sold to Google. (Non-paraphrase)	This movie is bad. (Sentiment: Negative) This movie is not bad. (Sentiment: Negative)
Model's Mistake (Original vs. adversarial)	Consider the two to be different	Consider the two to be the same

Figure 1: Vulnerabilities in NLP Models

Most of the prior works of generating adversarial examples for NLP models only evaluated on simple models. Therefore, we instead target on the state-of-the-art pre-trained language model, BERT. Besides analyzing its robustness against adversarial attacks, we would like to know why and how it make such decision towards adversarial examples. Since BERT is consisted of 12 attention layers, its output predictions depend on the attention weights for each word token. We can understand how BERT make decisions through visualizing the attention maps for each input example.

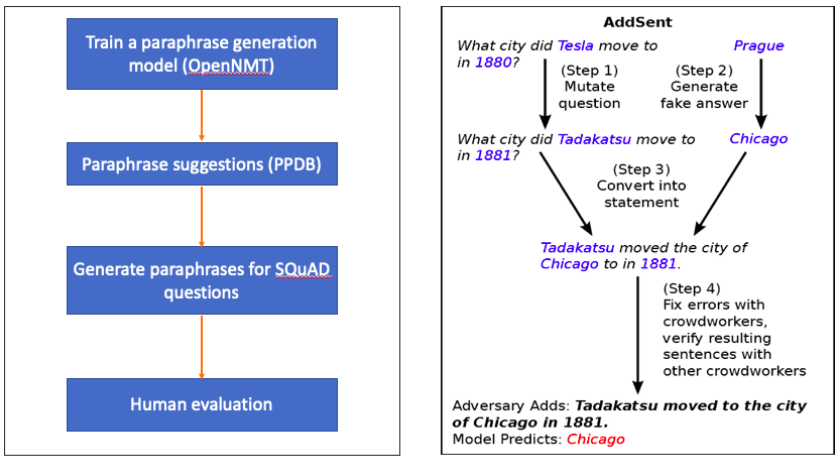
33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

In this project, we would like to understand how BERT react to these two types of adversarial examples for Question Answering (QA) task. Specifically, we would like to find answer to the following questions:

1. How vulnerable is BERT to adversarial examples?
2. How do attention weights affect BERT's prediction?
3. Would the attention map for the adversarial examples different from normal examples?

2 Proposed Method

We would first analyze the adversarial robustness of the BERT model by testing on adversarial examples in two different aspects. For over-stability, we use the adversarial SQuAD dataset proposed by (Jia and Liang, 2017). This dataset would be used for testing whether the model is too stable and not be able to notice the semantic changes with adversarially crafted examples that looks similar to the original sample. For over-sensitivity, we would generate paraphrase questions for each examples, and see if BERT is too sensitive to some lexical changes that actually do not affect the original meaning of the sentence. We chose to implement the adversarial paraphrase generation method proposed by Iyyer et al. (2018) [4] at first. However, the quality of the generated sentences for questions from SQuAD was quite low. (Please see Section 4.3 for detailed explanations.) We then decided to use the adversarial paraphrased SQuAD dataset released by Gan and Ng (2019). Different from the previous method we tried, they trained a Transformer model on a combination of the WikiAnswers parpahrase corpus and the Quora Question Pairs dataset.



Gan & Ng (2019) Jia & Liang (2017)

Figure 2: Adversarial examples we used for this project.

After gathering the adversarial examples, we obtain and plot the attention weights for each examples. Since BERT contains 12 attention layers with each having 12 attention heads, it would be time-consuming to look through each layer. Previous studies have also shown that the first couple of layers tend to attend broadly over many word tokens. (Clark et al., 2019) [8] Therefore, we skipped the first few layers and focused on finding relationships between the tokens in the questions and contexts in the last couple of layers.

3 Expected Outcome

Self-attention allows each word in the sequence to look at other words to learn which word contribute to the current word. It help BERT to better understand each word based on its context. For Question Answering task in the normal setting, we expect the tokens from the question sentence to assign higher attention weights on the correct answer. A visualization of attention map for a normal QA sequence is shown in Figure 3. On the other hand, we expect the model would have higher attention weights to the fake answer when evaluating on the adversarial example.

In addition, we can also find out if the model have a more focus attention weights on the target answer or distribute attention weights on multiple tokens across the entire paragraph. And in which layer the model would assign attention weights to tokens regarding the target answer.

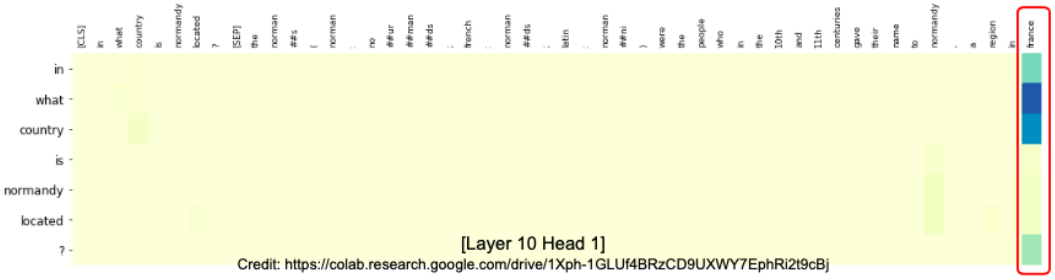


Figure 3: Attention visualization for a normal input sequence.

4 Project Progress

So far, we have finished training the BERT model on the SQuAD dataset, and also tested the model on the adversarial SQuAD dataset to evaluate the over-stability in the BERT model.

4.1 Model Training

We fine-tuned the BERT base uncased model on the SQuAD 1.1 dataset. We trained our model using huggingface's transformer library, which is developed in Pytorch environment. The base model contains 12 hidden layers and each layer has a hidden size of 768. We trained the model with a batch size of 12, a learning rate of 3e-5, and a max sequence length of 128.

After fine-tuning the model for 2 epochs, we evaluate the model's performance on the SQuAD development set and obtain an accuracy of 81.05% and 88.58% for F1 score. The performance we got is about the same as the original paper for BERT. The evaluation results are shown in Table 1.

Metric	Original Paper	Ours
Accuracy	80.8%	81.05%
F1 Score	88.5%	88.58%

Table 1: Model performance on SQuAD dev set

4.2 Over-stability Evaluation on Adversarial SQuAD

We evaluate our fine-tuned BERT model on ADDSENT and ADDONESENT from the adversarial SQuAD dataset. The adversarial sentences are generated by replacing nouns and adjectives from the target question with antonyms from WordNet. And a fake answer to this question is then generated with the same POS tag as the original true answer. Finally, the generated "distracting" sentence is appended to the end of the paragraph.

We compare the results with two baseline models that used in . One is the Match-LSTM[5], which essentially maps the attention-weighted first sequence to each token from the second sequence and make prediction based on the aggregated matching result. The other is the Bi-Directional Attention Flow (BiDAF) network [6], which uses a hierarchical multi-stage architecture including character-level, word-level, and contextual embeddings. Both baseline models also utilize attention mechanism to make predictions.

It is no surprise that BERT has better performance than other baseline models. From Table 2, we can see that the F1 scores of BERT decrease around 12% and 21% on ADDONESENT and ADDONESENT respectively. The decrease rate is significantly lower than other models. From

these results, it clearly shows that the BERT model is not overly stable to semantic changes and is more robust to this type of adversarial examples than other baseline models.

Model	Original	ADDSENT	ADDONESENT
Match-LSTM	71.4	27.3	39.0
BiDAF	75.5	34.3	45.7
BERT	**88.6**	**67.5**	**75.5**

Table 2: Model evaluation on adversarial SQuAD dataset

4.3 Adversarial Paraphrase Generation

For adversarial examples that evaluate model's over-sensitivity, we decide to generate paraphrase adversaries by using the Syntactically Controlled Paraphrase Networks (SCPNs) proposed by Iyyer et al. (2018) [4]. The model is trained to produce a paraphrase of the target sentence with the corresponding given syntax template. They first generate the training data for SCPNs with back-translation and use a parser to label the syntactic transformations. For adversarial example generation, the adversary would not only generate a semantically similar paraphrase for the target sentence but also make sure that the generated paraphrase would make the model output the incorrect prediction.

We tried using the pre-trained model from the paper's Github repository, and generated paraphrase questions for each example. However, the generated questions had really low quality and sometimes did not maintain the same meaning as the original question. One of the reason why it is not working could be that the model is trained on a more general dataset instead of a variety of questions. And also, the model requires user defined syntactic templates. The default templates they provided only has one type of parse template for questions. We tried adding several other types of syntactic form, but still with limited variations. Fortunately, we found an adversarial paraphrase dataset specifically generated for SQuAD (Gan & Ng, 2019). We then do further attention analysis based on this dataset.

4.4 Attention Visualization

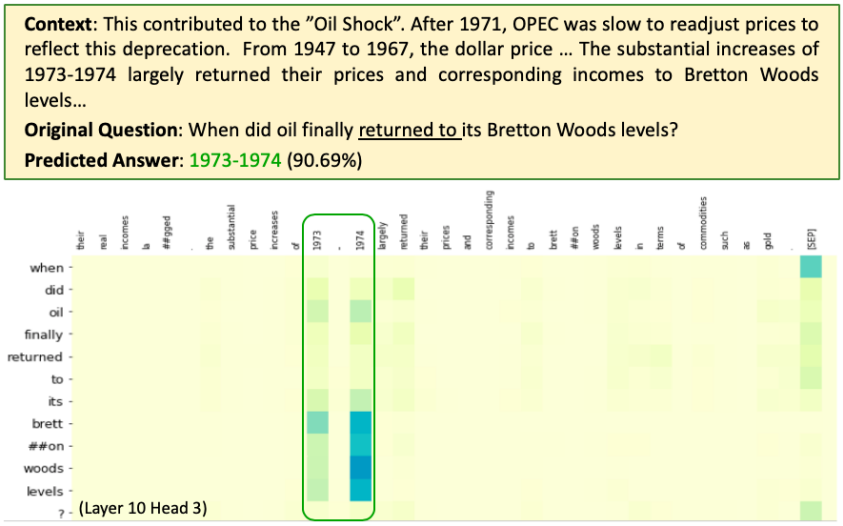


Figure 4: Attention map of the original example with correct predicted answer in Layer 10 Head 3.

We compared the attention maps for each example to their corresponding adversarial examples, and found that the tokens from the questions do attend to their predicted answers. More specifically, this pattern only and almost always happens in the third attention head of layer 10. The adversarial examples also have the same pattern. We found several examples that the model predicted correctly on the original question and also attend to the correct answer. (Figure 4) When we ask the model to

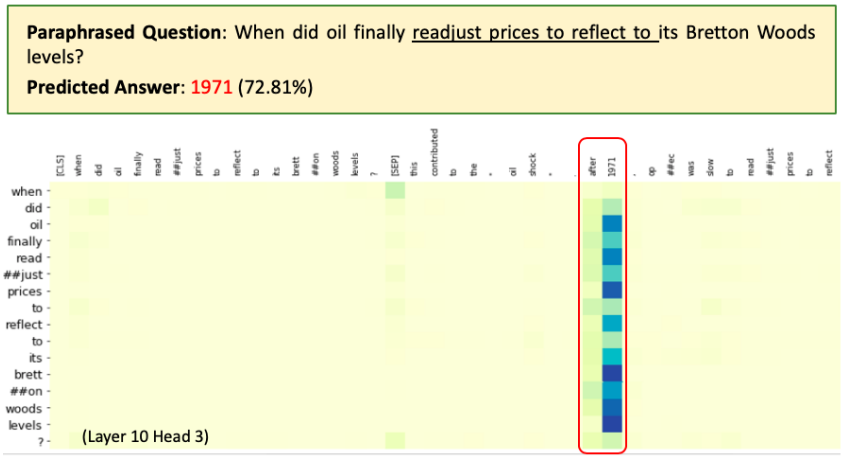


Figure 5: Attention map of the adversarial paraphrase example in Layer 10 Head 3.

predict the answer again but on the paraphrased question, the model output with another answer and the question tokens also attend to this newly predicted answer. (Figure 5)

Besides attending to the answer tokens, we found that a lot of the time it focuses on the deliminator token [SEP] in almost every attention layer, and some of the time it attends to itself. These patterns align to the findings from (Clark et al., 2019), so it is not a surprising results.

We also analyze the attention weight patterns in the adversarial examples that BERT predicts answers with lower probability. We showed one of our example in Figure 6. The adversarial question is generated with ADDSENT, which alters the question to have a different semantic meaning but similar structure as the original question. The BERT model still predicts the original answer "Tuesday" with the highest probability even though the answer to the new question should be "Monday". The correct answer "Monday" is actually one of the top 5 answers besides "Tuesday". And the attention weight for "Monday" is also slightly higher than other tokens, but still lower than the predicted answer, "Tuesday".

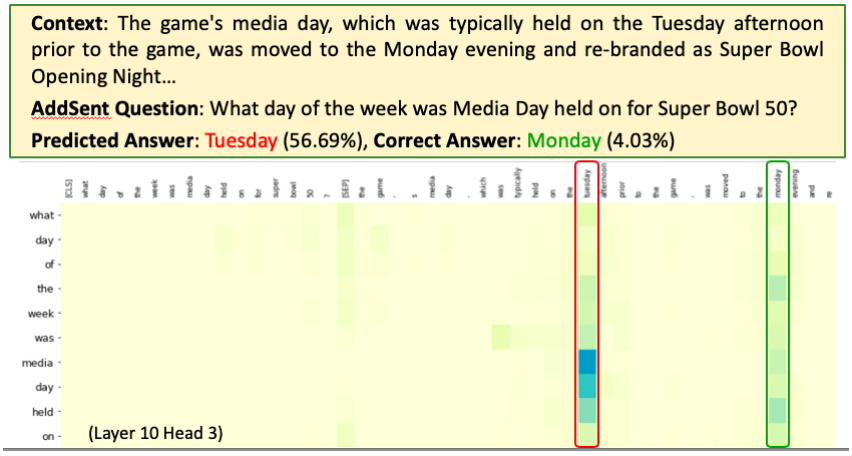


Figure 6: Attention map of the adversarial SQuAD example generated with ADDSENT method (Jia & Liang, 2017). The original question is *"As a norm, what day of the week is the traditional Media Day held prior to a Super Bowl?"* and the the original answer should be "Tuesday".

5 Conclusion

From our experiment results, we can see that the model's decision largely depends on the attention weight in layer 10 head 3. This pattern appears in both normal and adversarial examples. It would be interesting to see how well the prediction accuracy would be if we only use the information from layer 10. However, we still cannot avoid the existence of adversarial examples since the decisions for both normal and adversarial examples happen in the same attention head and layer. It would be hard to disentangle the attention weights. Yet, we can utilize this information to find possible solution to defend the BERT model for QA tasks against adversarial attacks.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, & Kristina Toutanova. (2018) BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*.

[2] Di Jin, Zhijing Jin, Joey Tianyi Zhou, & Peter Szolovits. (2019) Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment

[3] Robin Jia & Percy Liang. (2017) Adversarial examples for evaluating reading comprehension systems. *Empirical Methods in Natural Language Processing (EMNLP)*

[4] Mohit Iyyer, John Wieting, Kevin Gimpel, & Luke Zettlemoyer. (2018) Adversarial Example Generation with Syntactically Controlled Paraphrase Networks. *ACL*

[5] Shuohang Wang & Jing Jiang. (2016) Machine Comprehension Using Match-LSTM and Answer Pointer.

[6] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, & Hannaneh Hajishirzi. (2016) Bidirectional attention flow for machine comprehension. *CoRR*

[7] Wee Chung Gan & Hwee Tou Ng. (2019) Improving the Robustness of Question Answering Systems to Question Paraphrasing. *ACL*

[8] Kevin Clark, Urvashi Khandelwal, Omer Levy, Christopher D. Manning. (2019) What Does BERT Look At? An Analysis of BERT's Attention. *ACL*