Human Adversarial QA: Did the Model Understand the Paragraph?

Prachi Rahurkar, Matthew Olson, Prasad Tadepalli
School of EECS
Oregon State University
{rahurkap, olsomatt, prasad.tadepalli}@oregonstate.edu

Abstract

Recently, adversarial attacks have become an important means of gauging the robustness of natural language models as training and testing set methodology has proved inadequate. In this paper we explore an evaluation based on human-in-the-loop adversarial example generation. These adversarial examples aid us in finding the loopholes in the models and give insights into their working. In the published work on adversarial question-answering, perturbations are made on the questions without changing the background context on which the question is based. In the current work, we examine the complementary idea of perturbing the background context while keeping the question constant. We analyze the state-of-the-art language model BERT for the task of question-answering on SQuAD dataset using novel adversarial examples crafted by humans exposing the weaknesses of the model. We present the typology of the successful attacks here and encourage NLP researchers to use it as a baseline for stress-testing.

1 Introduction

Recently, deep transformer architectures [Vaswani et al., 2017] have been applied to a variety of language tasks with significant success. One of the top-performing systems is BERT, which achieved very high accuracies on many natural language tasks [Devlin et al., 2018]. Here, we focus on reading comprehension Question Answering with SQuAD [Rajpurkar et al., 2016], which is one of the most difficult tasks in Natural Language Understanding.

Research in natural language processing has been headed towards leveraging these state-of-the-art models for several domains which consist of language data, such as healthcare, robotics, databases, etc. BERT and its derivatives [HuggingFace, 2019] have also been used widely in tasks which incorporate both vision and language embeddings. While the performance of these systems has generally been high as measured in the traditional training-and-testing set paradigm, it is increasingly becoming clear that they are susceptible to a variety of problems including out-of-sample performance and adversarial attacks. Deploying such systems even in modestly important applications like call centers is currently unthinkable.

The goal of the current paper is to advance the state of understanding of question answering systems like BERT through human-in-the-loop evaluation. We created a HAMLET-like user interface [Nie et al., 2020]. On our interface, the users see some background context and a question which is correctly answered by BERT. The users can edit the background context and re-run BERT. The goal of the users is to change the context in a way that fools BERT while still containing the correct answer to the same question. We conducted a user study with 15 users, who are active deep learning practitioners, giving them examples from the SQuAD test dataset. Here is an illustrative example (from SQuAD) showing BERT's failure. The question here is: "How are chloroplasts similar to mitochondria?" (corresponding image in Figure 1). With the original paragraph from SQuAD dataset,

Presented at NeurIPS 2020 Workshop on Human And Model in the Loop Evaluation and Training Strategies.

the model correctly predicts the ground truth. When the sentence that contains the ground truth (henceforth called "key sentence") is split into two sentences that are placed apart, the prediction changes to an incorrect answer. If the same adversarial paragraph is given to a human, the human correctly predicts the ground truth. In Figure 3 is another example of an adversarial attack crafted by a user where all the sentences other than the key sentence are ablated.

This raises an important question: Can the model be trusted to answer correctly given the same question but with different paragraphs which still contain the answer? We dive deeper into this question with the study. We categorize the user crafted adversarial paragraphs that are successful in misdirecting model's predictions into different types of successful attacks.

Our contributions are as follows: 1) We conduct a "human-as-an-adversary" user study, consisting of ten different paragraphs having a mix of difficulty and complexity levels. 2) We show, via the quantitative analysis, that users can come up with multiple adversarial examples that can break the model misleading it to incorrect predictions, in a very short span of time. 3) We provide a detailed qualitative analysis of the collected adversarial samples by clustering the successful adversarial attacks into different types. A demo of the interface we used in our study is available at `http://165.227.25.235/0`.

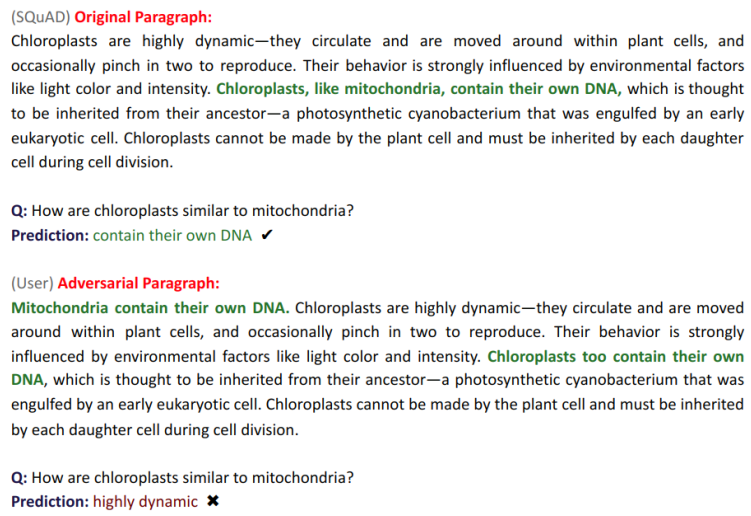

Figure 1: Adversarial attack example: Split-Reorder attack

2 Related Work

The use of adversarial examples to find weaknesses in deep learning models has been widely studied for a long time. Recently, there have been a number of techniques for crafting adversarial attacks on image-based deep learning models [Goodfellow et al., 2015]. While there is limited literature for such approaches in NLP models, there have been some studies that have exposed the vulnerabilities of neural networks in text-based tasks like machine translations and question answering.

We draw inspiration from the work by [Jia and Liang, 2017]. In this work, two kinds of adversarial attacks namely ADD-SENT and ADD-ANY are proposed. In ADD-SENT attack, a distractor sentence is concatenated to the reading comprehension paragraph. In the ADD-ANY attack, a sequence of random words regardless of grammar is concatenated to the paragraph. While our work posed here is along similar lines, the major difference is that we propose a novel categorization of different text attacks crafted by users in a limited span of time. We present interesting insights from the user study we conducted for the same.

There has also been question answering focused work [Mudrakarta et al., 2018] wherein an attribution technique called Integrated Gradients [Sundararajan et al., 2017] is employed to isolate question words that a deep learning system uses to produce an answer. In this work, integrated gradients are applied to attribute the model's predictions to words in the questions. The weaknesses identified in

the model's logic, as exposed by the attributions, are leveraged to craft adversarial questions. We are motivated by this work and while they perturb questions to craft targeted attacks, we present patterns in the adversarial paragraphs (instead of questions) which successfully change model's behaviour.

The most recent work by [Nie et al., 2020] describes a never-ending learning setting called HAMLET. In this work, the task of Natural Language Inference is studied. A new large-scale NLI benchmark dataset is introduced which is collected via an iterative, adversarial human-and-model-in-the-loop procedure. We are also motivated by this work and while this work focuses on the task of Natural Language Inference, our work is specifically focused only on the Question-answering task.

There has also been work on generating black-box adversarial examples for text classifiers using deep reinforced models [Vijayaraghavan and Roy, 2019]. Unlike text classifiers, we focus solely on question-answering with adversarial testing on context, keeping the question as it is. Another related work [Ribeiro et al., 2018] discusses rules to generate adversarial questions which are semantically equivalent to the original question for the tasks of sentiment analysis, machine comprehension and visual question-answering. The recent work CHECKLIST [Ribeiro et al., 2020] proposes a task-agnostic methodology for testing NLP models.

3 Methodology

User Study We employed fifteen expert users having a background in deep learning including a graduate level course and some research experience. The users were presented with a context paragraph along with the corresponding question and its correct answer. Below that, we provided an editable text-box where the user can input their perturbed custom paragraph. After they are done with editing the paragraph, they can click PREDICT. When they are successful in their attempt to fool the model, i.e. when the model predicts an incorrect answer, the user can click on SUBMIT.

Before the start of the study, users were briefly informed on BERT and question-answering. Each user tested ten different paragraphs and corresponding ten different questions. We set a timer of four minutes for every paragraph-question. So each user study lasted for forty minutes. We randomly collected ten different topics from the SQuAD test dataset. One paragraph from each of these topics was taken for the study. The paragraph too was randomly selected from the list of paragraphs under the particular topic.

Users were given the freedom to introduce both word-level and sentence-level perturbations. They were also allowed to submit more than once if they were successful in multiple different attempts done in the four-minute timespan. The instructions given to the users were as follows:

1. There will be given ten paragraphs with corresponding ten different questions.
2. Each paragraph-question will be given four minutes.
3. Create adversarial custom paragraphs in such a way that the model cannot predict the ground truth while still keeping the ground truth (i.e. correct answer) inside the paragraph.
4. The button "Predict" should be used to find model's predictions on the perturbed paragraphs and it can be used multiple number of times.
5. When the attempt is successful, click on "Submit" and submit the successful adversarial paragraph. Submissions can be made multiple times within the four minutes time.

If there is still time left and the user wishes to go back to previous examples for the remaining time (of the current question), they are allowed to do so. So the total time of every user study remained a constant forty minutes.

To ensure the meaning of the passage was indeed preserved, all the adversarial paragraphs collected in the user study were examined by a human expert. After the adversarial paragraph was approved by the human expert, the users were allowed to submit it.

4 Results

All the adversarial paragraphs the users created were collected in the data file. The successful paragraph submissions were also collected separately. Duplicate successful paragraphs and multiple

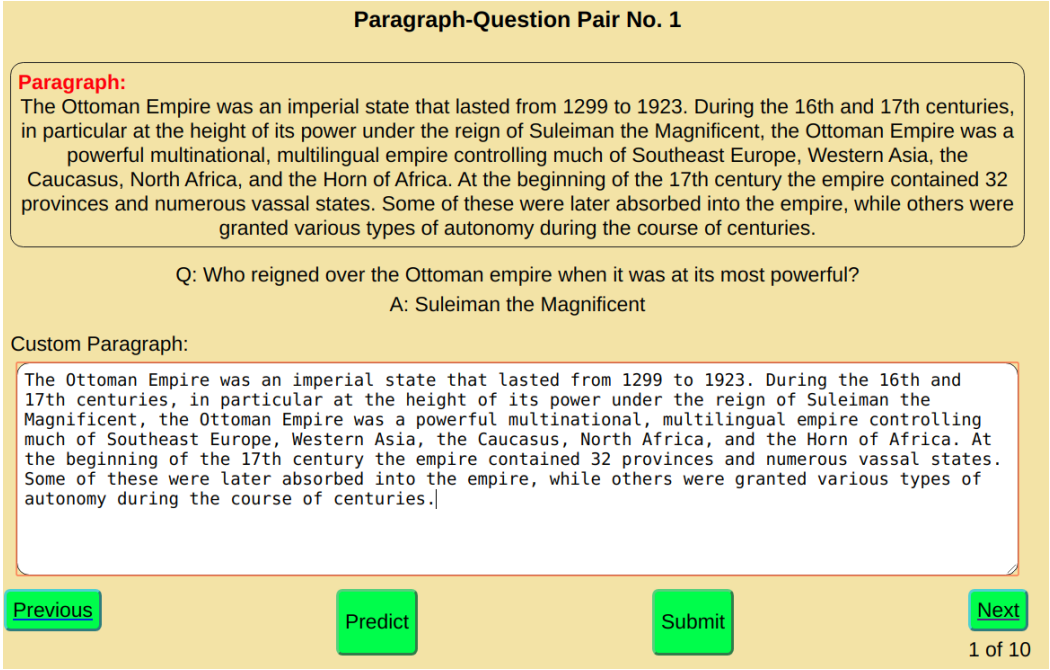

Figure 2: An example of the interface used to query the model where a user can make edits to the context in our question-answer setting.

Table 1: For every paragraph-question pair, % of successful attempts out of total attempts for all users

Task Index	1	2	3	4	5	6	7	8	9	10
Success Proportion	10%	51%	23%	41%	37%	32%	57%	17%	64%	43%

submissions of the same were handled by eliminating all the duplicates and keeping only one. We analysed both the data files and present some of the interesting insights here. The findings from this experiment are summarized into two sections as follows: quantitative analysis and qualitative analysis.

4.1 Quantitative Analysis

We summarize our findings via two aspects. In Table 1, the first row denotes an ID number of the paragraph-question pair. The second row represents the % of successful attacks out of the total attacks crafted by all the fifteen users, for each corresponding paragraph-question pair.

We calculated the proportion of successful attempts of total attempts for all participants over all paragraph-question pairs: the mean success rate was 36.2 % with a standard deviation of 13.32 (and a variance of 177.36). Table 1 shows the exact success rates for each question answering task. Some paragraphs were significantly harder to generate an adversarial (such as task 1 and 8), whereas other paragraphs were successfully attacked by the majority of participants (task 7 and 9). This points to diversity being an important aspect in the selection process of examples where evaluating models for adversarials.

4.2 Qualitative Analysis

We collected all the custom paragraphs i.e. the adversarial paragraphs crafted by the users to find model predictions. If the model prediction does not match the original ground truth, then the adversarial attack is a successful attempt. If it does, then the attack is an unsuccessful attempt. We also evaluated the successful adversarial attacks by users based on human judgements, and confirmed that the semantics in the paragraph and the ground truth to the presented question is indeed

preserved inside the paragraph. We found that the perturbations produced by the users affect the model predictions significantly and are indeed novel and innovative.

We categorize and report all the successful attacks into distinct categories here. These count of these attacks denotes the amount of times it was successful, for entire user population and all ten questions combined.

Sentence ablation attacks In this attack, the paragraph is reduced at a sentence-level keeping the ground truth intact. For example, in Figure 3 (below), when the original paragraph is reduced to just the most important key sentence, the correct prediction changes to an incorrect prediction. But not for all the examples. This makes the model's behaviour very unclear as to how much context is actually required by the model for it to answer questions correctly. These attacks were used a total of 13 times successfully in the entire study of ten questions with all the users.

Reordering attacks In this attack, the sentences in the paragraph are re-arranged in such a way that the meaning of the paragraph is still preserved. However, the model is not able to predict the answer correctly at some times, and at other times, it is. This attack was successful 15 times. So is the order not important? This means that the model picks the answer tokens from the sentence which contains the maximum words of the question. For instance, this attack was used in combination with the splitting attack in Figure 1.

Splitting key sentence attacks The key sentence (for SQuAD dataset) is the one that contains the ground truth. If this key sentence is split into two or more sentences, the model predictions take a hit. This attack were used for a striking 24 times successfully, that is the highest number of times. For example, this attack was used in combination with the re-ordering attack as shown in Figure 1.

Sentence merging attacks In this attack, the key sentence is merged along with another sentence within the para forming a long sentence inside the paragraph. This causes a change in model predictions as well. This attack was used for a total of 4 times successfully. For example, in Figure 2, if the first two sentences are joined with the conjunction ".. because ..", the prediction changes.

Distractor sentence attacks A distractor sentence is a sentence which has almost all the words from the question but does not contain the answer. When a distractor sentence is introduced in the paragraph, the model tends to attend to the trigger words which causes inaccurate predictions. This attack was used for a total of 19 times successfully. For example in Figure 2, concatenating the sentence "XYZ didn't reign over the Ottoman empire when it was at its most powerful." at the start gives a wrong prediction.

Misspelling attacks This is one of the easiest attacks introduced by far. In this word-level or sentence-level misspellings can be introduced which help in fooling the model. A special case of

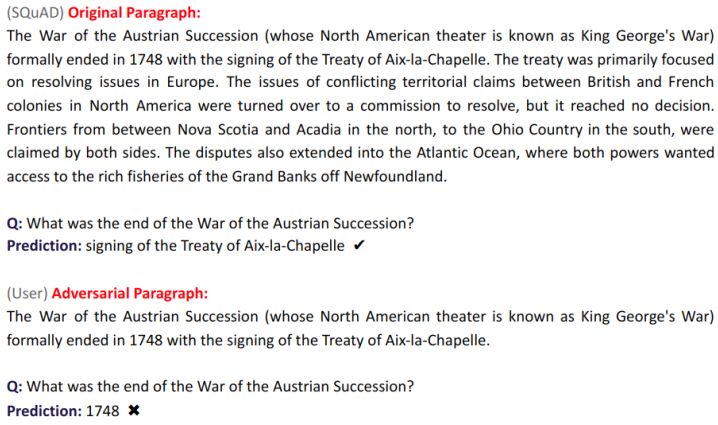

Figure 3: Adversarial attack example: Ablation attack

Table 2: Attacks and their counts

Sentence ablation	13
Reordering	15
Splitting key sentence	24
Sentence merging	4
Distractor sentence	19
Misspelling	6
Garbage concatenation	2
Paraphrasing	15
Key sentence elongation	6
Synonym replacement	10
Coreference ambiguation	4

misspellings is adding special characters to the key sentence within the paragraph. Appending or prepending special characters to the ground truth inside the para also causes the model to not predict anything at all. Misspelling attack was used for 6 times while the addition of special characters attack was used for two times successfully. For instance in Figure 1, if the para is changed to "....**Mitochondira**, contain their own DNA...", the prediction changes.

Garbage concatenation attacks Garbage, here, means a series of characters that do not make sense. For instance, when "xxxxx" (garbage value) is added inside the key sentence, the model predictions take a hit. This attack was used twice successfully. For example, in second paragraph-question, if the para is changed to "...In macroscopic closed systems, xxxxxxx non-conservative forces act to change the internal energies...", the prediction changes.

Paraphrase (or rephrase) attacks This is the most tried attack amongst all the users. It is not the most successful attack though. When the key sentence inside the paragraph is rephrased in a different way, model predictions change. This attack was used 15 times successfully in the user study. For example, in Figure 1, if the original sentence is changed to "... Chloroplasts contain their own DNA, as do mitochondria, which ...", the model gives an incorrect answer.

Key sentence elongation attacks In this attack, context-free phrases are added as suffix or prefix or in-between the key sentence in the paragraph. Few example phrases are "In reality", "Contrary to the popular incorrect belief", etc. This attack was used for 6 times successfully.

Synonym replacement attacks In this type of attack, the words in the key sentence which match with the words in the question are replaced with their synonyms. They can also be replaced with the negated antonym which will lead to the same meaning, thus preserving the semantics. This attack was successful for 10 times. For example, replacing the sentence "his father had originally wanted.." to "his dad had actually envisioned.." changes the model's prediction.

Coreference ambiguation attacks In this attack, common nouns which are related to the key nouns in the question are added in the paragraph. The key nouns and the remaining entities in the paragraph are referenced with respect to the introduced misleading noun. This means that entity resolution in language tasks still needs more work. This attack was successful for 4 times. For example, changing "...Water on the eastern side flowed toward the Atlantic, while to the west, water flowed toward the Pacific..." to "...Water flowed towards the eastern and western sides. The former flowed toward the Atlantic, while the latter flowed..." gives a wrong prediction.

Finally, based on our observations, these attacks work best when they are used in combination with other attacks described here.

5 Conclusion

To summarize, we put forward a novel categorization of the kinds of adversarial attacks made on context paragraphs in question answering tasks based on a study of expert users. The study shows that the human adversaries are fairly successful in generating a variety of adversarial examples. Our

research provides an alternative method to evaluate and stress test language models in the question answering task. In our future work, we will conduct the user study with non-expert users. Future work will also investigate how to address the weaknesses of the language models through robust training methods and alternative architectures. Our demo is publicly available on this webpage: http://165.227.25.235/0.

Broader Impact

Our findings from the experiment will help researchers looking to strengthen their language models. We encourage all researchers and software developers looking to deploy their question-answering models in production to evaluate their models on these attacks. Our research helps robustly evaluate QA systems and build appropriate trust before deploying them in the field.

References

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

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

HuggingFace. Pytorch-transformers. https://github.com/huggingface/pytorch-transformers, 2019.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems, 2017.

Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. Did the model understand the question?, 2018.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging NLP models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1079. URL https://www.aclweb.org/anthology/P18-1079.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://www.aclweb.org/anthology/2020.acl-main.442.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

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

Prashanth Vijayaraghavan and Deb Roy. Generating black-box adversarial examples for text classifiers using a deep reinforced model, 2019.