Are BERT-based fact-checking models robust against adversarial attack?

Eliott Afriat[1]

Supervisor(s): Avishek Anand[1]**, Lijun Lyu**[1]**, Lorenzo Corti**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
February 17, 2023

Name of the student: Eliott Afriat
Final project course: CSE3000 Research Project
Thesis committee: Avishek Anand, Lijun Lyu, Lorenzo Corti, Marco Loog

Abstract

We seek to examine the vulnerability of BERT-based fact-checking. We implement a gradient based, adversarial attack strategy, based on Hotflip swapping individual tokens from the input. We use this on a pre-trained ExPred model for fact-checking. We find that gradient based adversarial attacks are ineffective against ExPred. Uncertainties about the similitude of the examples generated by our adversarial attack implementation cast doubts on the results.

1 Introduction

Natural language processing (NLP), is used to for translations, fact-checking. And in general, for human-computer interactions, NLP is being used more and more. BERT is a language representation model [1] that has been used in many applications of NLP such as sentiment analysis, fact-checking, text prediction and so on [2][3]. Testing these systems for vulnerability and robustness would help develop more usable and trustworthy systems.

Adverserial attack strategies, that aim to modify the input in a indetectable, or irrelevant from the human perspective, while changing the models original output. This technique has been widly used in the domain of image recognition. Adverserial attacks on NLP poses additional challanges as an indetectable change is harder to define, however these techniques have been gaining prominance in this domain [4].

Multiple Adverserial attack strategies have been applied to BERT models in previouse research. Black-box strategies, such as in [7], aim to create adverserial examples without knowing the innerworkings of the model. This represents a more realistic scenario from the perspective of a nefarious actor. As opposed to this white-box strategies, such as gradient based techniques based on HotFlip [5][6], have accesse to the innerworkings of the model, and are therefore considerd to be more powerfull then their black-box counterparts.

Both of these strategies have shown effective against BERT systems in other contexts [7][6].

Expred is a BERT-Based explain-then-predict NLP system [2]. First a model aims to generate an explanation by only showing the relevant parts of the text, and from that, a second model arrives to the prediction, as shown in Fig.1. This interpretability by design insures that the user has some understanding about how the conclusion was reached. However this model does not explain how the mask was generated, or how the prediction was reached from the masked inputs. Leaving room for possible vulnerabilities.

A model of expred trained on the ERASER FEVER dataset was used [2][8]. This dataset is designed to train fact-checking models with rationals (masked input).

An implemenation of HotFlip is presented, and applied on an ExPred model. The HotFlip strategy was chosen due to it being successful in previous situations against BERT [7][6], and was used to produce conterfactuals for ExPred for fact-checking [9]. As a white-box strategy, we can also expect better results than black-box strategies from similar resoutrces.

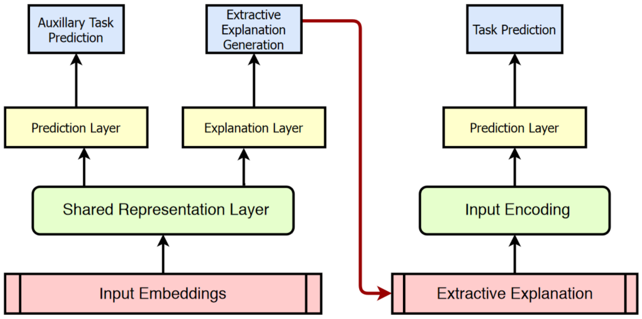

Figure 1: Diagram of expred from [2]

To test for effects of the explainer on the robustness of the system, the adverserial attacks where applied in two ways.

- By running the adverserial example through the entire model.
- By keeping the original mask, and only running the adverserial example through the predictor.

We determine this expred model, and potentially by extention other BERT models in this context, are robust against HotFlip-like gradient based strategies. And that the explainer step does not effect the vuneralbility of the system in any apparent way.

The paper continues as follows. After describing the problem in a formal maner in section 2, we follow by describing the hotflip implementations, and research methodology in section 3, followed by the experimental setup in section 4. The results are described in section 5. Section 6 discuses possible implication of these results, as well as highlighting possible shortcomings, and the steps taken to insure proper research are outlined in section 7. Lastly, the conclusion, and possible future works are outlined.

2 Problem Description

ExPred [2] is a pipeline of 2 models, the explanation generation network G, and the prediction network F.

$$G(c,x) = \{0,1\}^{|x|}$$
$$F(c, x \otimes G(c,x)) = y$$

$x \otimes G(c,x)$ is an elementwise operation replacing tokens in x with the wildcard token '.' where $G(c,x)$ is 0. This represents masking all token not in the explanation.

The explanation generation network takes a querry q and the text from the documents x, and returns a mask that only shows tokens it deems relevant to the claim m. The prediction network, takes the claim and the masked evidence, and returns its prediction y.

A valid adverserial example x' is defined as following.

$$F(c, x' \otimes G(c,x')) \neq F(c, x \otimes G(c,x))$$
$$F(c, x' \otimes G(c,x)) \neq F(c, x \otimes G(c,x))$$

The above passes the input through the whole expred system, and the second keeps the original mask.

No more than 20 tokens, and less than half of the tokens in x are permited to form the adverserial example x_{adv}. This is done to insure similarity.

3 Methodology

Our solution is based of HotFlip [5]. This method was chosen as it has shown effective against BERT models in past reasearch [6], since it's a white-box method, as well as being relativlly easy to implement. The tokens with the highest embedding gradients with respect to the opposite prediction are selected.

$$max_i \frac{\delta}{\delta x_i} L(\bar{y}, \bar{y} \neq y) \cdot \mathbf{e_i}$$

For each of these tokens, possible replacements are selected based on it's gradient.

$$max_{v \in V} \frac{\delta}{\delta x} L(\bar{y}, \bar{y} \neq y) \cdot \mathbf{e^v}$$

4 Experimental Setup

The code has been made available on a public repository [1].

4.1 Inputs

The HotFlip implantation is run on the first 1000 examples from the Eraser/Fever test dataset.

Each line from this dataset contains a query with its classification with multiple evidence set. Each evidence set contains documents, and is independently sufficient to determine the classification of the query. This dataset only retains examples labled as supports or rejects from the fever dataset, leaving examples that do not contain enough information out.

The inputs are generated by taking the query, and all the documents appearing in the evidence sets.

4.2 Parameters

To avoid unwanted tokens, only tokens with index $i = [1996 : 10000]$ are permited can be used as replacements.

This is to prevent unwanted tokens such as the separator token, that appear below that range, and uncommon tokens that appear above that range from appearing. Specific checks and masks are implemented to garanty that numbers can only be replaced by numbers, and suffixes by suffixes.

4.3 Model

An Expred model pre-trained on the ERASER/FEVER dataset is used.

5 Results

Adverserial examples are generated using hotflip and ran on the whole expred model, or only on the predictor model while keeping the mask the same. For all 1000 original inputs, we swap one token, and check if it changes the models predictions. We continue to add swaps to the input until the prediction changes or the maximum number of swaps is reached. Fig. 2 shows the success rate of the adverserial attacks by number of tokens swaped, and by proportion of tokens swaped for both models. This method only succeeds about 30% of the time after all the permitted swaps. This is far below our expectation.

There is no apparent difference between the behaviours of the whole ExPred model, and the predictor model by itself. A substantial difference in the results could have indicated that the explainer was providing extra robustness by shifting attention away from suspect tokens, if the whole expred model was more robust then the predictor by itself, or if the predictor was more robust then the expred model, this might suggest that the mask shape returned by the explainer, was used by the predictor as a factor in its decision making, this would reduce the likelyhood of the prediction flips when the meaning of the adverserial example was inadvertenly changed.

Some inputs are too small to have 20 swaps, and most are to large to have 50% of all their tokens swaped. This fact explains the log-shape of the curve to a large degree, especially for the second graph.

For each input, an adverserial attack containing the maximum amount of flips is created. Fig. 3 shows the number of times a specific token was selected as a replacement, and it's frequency.

The results correlate poorly with the distribution of most common words in the English language. The words ”the”, ”and”, and ”is” are the only ones in the top. It seems to make little sense to use tokens like ”shouldn” as a replacement in it's own right.

Other than ”archived” which is used 6% of the time, well above the rest, the usage of rest of the tokens are closer together in frequency. Non standard tokens such as ”...”, ”isbn”, tokens that do not make sense to change by themselfs such as ”shouldn” or foreign words such as ”buenos” or ”lanka”, this suggests that many adverserial examples are nonsensically.

6 Responsible Research

I have attempted to make this research as reproducable as possible, by uploading the code to a public repository, and by being as clear as possible about what dataset I have used, and how I have used it, and explain my implementation of my code, describing all relevant parameters. Further efforts to make my code more readable, and easier to run on other machines would have been made, but for time constraints.

Due to time constraints, the research questions, and purpose of the research where heavely modified, after some of the experiments where already carried out. This as a major flaw in the research process. Keeping this in mind, I have attempted to remain critical of my results, highlighting possible errors in the research process.

7 Discussion

After brief experimentation, a mask insuring numbers are swapped with numbers and suffixes with suffixes was implemented as it seemed to perform better. A more thorough look into this to explore other type of masks and there effects would be needed to guaranty better and sound results. We tried increasing the amount of candidate tokens examined for each swap, and using beam search to select the best combination, it was more effective at achieving prediction swaps. We chose not to continue with this aproach due to long executing time. But we do not believe these techniques would have a dramatic effect of success rates.

[1] https://github.com/somePersone/HotFlip-for-Expred

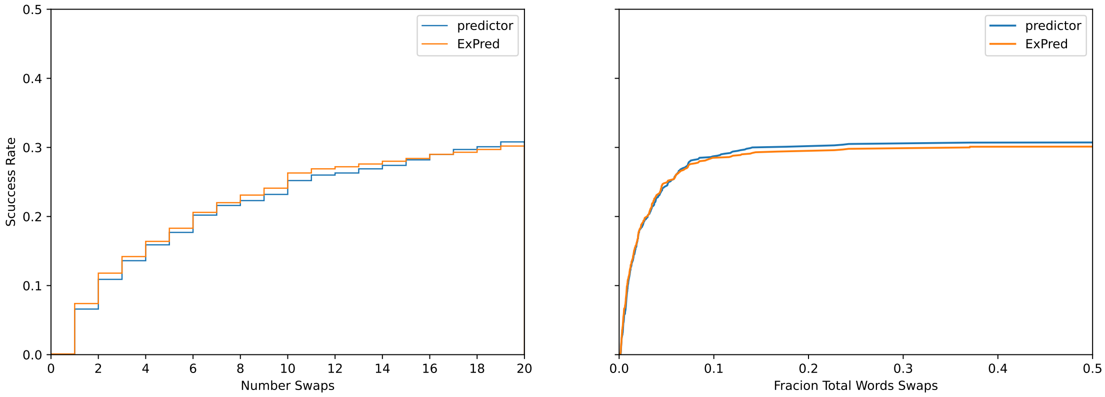

Figure 2: Success rate per number of tokens flipped, and fraction of tokens flipped

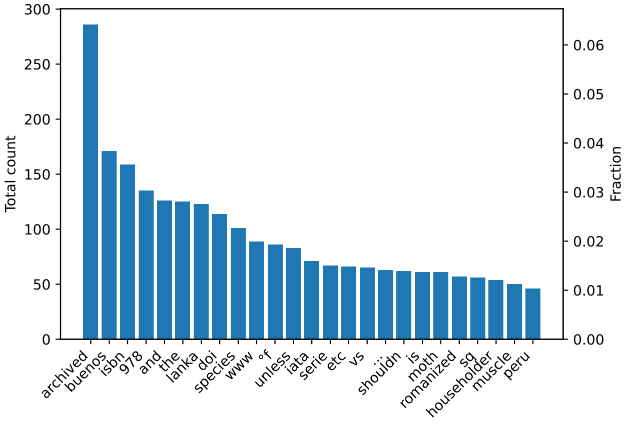

Figure 3: Frequency of words used to flip

There is no guaranty that the generated adverserial examples do not change the ”true” lable of the input, the examples could be grammatically incorrect, syntaxly nonsensical. They may also not have enough information to confirm or deny the query, or be a valid counterfactual example. Such cases go against the concept of asdverserial attack, that aims to keep the input similar from a human perspective. Since some of the examples might be counterfactuals, witch leads to a true flip in lable, the number of successfull adverserial examples might be lower then indicated. Conversaly, the successfull rate of ”true” adverserial attacks would be larger then Fig. 2 indicates if, for instance, constructed examples that do not contain enough information have a lower success rate at prediction swaps than ”true” adverserial predictions.

The dataset used for this research, and for the training of the BERT model did not contain a not enough information class. A model trained on such classes is likely to be a lot more susceptible to the same attack. [explain]

For future research, we recommand keeping either the number of swaps or fraction of token swaped consistent; doing it by both was a mistake, making both graphs in Fig. 2 hard to decipher.

Analysis on characteristics of examples with successful adversarial attacks: does it correlate with size of input, infrequent tokens in input?

Similar success rates of both models might hide significant differences, such as erratic changes in confidence of prediction after swaps. The ExPred model might not change it's prediction at all after a swap as the masked input returned by explainer hasn't changed, or conversely, additional swap causes a dramatic change in mask leading a massive change in confidence. More analysis on the results could have been performed, including number of successful swaps that happen on one model, but not the other.

8 Conclusions and Future Work

Our gradient based adverserial attack strategy is ineffective against the two step, explain-then-predict BERT model, ExPred, when applied to fact-cheking. These results should not be considered generalisable due to confounding factors, such as examples not containing enough information missing from the ERASER/FEVER dataset, that was used to train the model.

Arriving at a more reliable similarity metric, aided by manualy sorting the adverserial examples through crowed sourcing.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 10 2018.

[2] Z. Zhang, K. Rudra, and A. Anand, “Explain and predict, and then predict again,” pp. 418–426, ACM, 3 2021.

[3] Y. Qu, P. Liu, W. Song, L. Liu, and M. Cheng, “A text generation and prediction system: Pre-training on new corpora using bert and gpt-2,” 2020.

[4] A. Huq and M. T. Pervin, “Adversarial attacks and defense on texts: A survey,” 5 2020.

[5] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, “Hotflip: White-box adversarial examples for text classification,” vol. 2, pp. 31–36, Association for Computational Linguistics, 2018.

[6] Y. Wang, L. Lyu, and A. Anand, “Bert rankers are brittle: a study using adversarial document perturbations,” 6 2022.

[7] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, “Is bert really robust? a strong baseline for natural language attack on text classification and entailment,” 7 2019.

[8] J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace, “Eraser: A benchmark to evaluate rationalized nlp models,” pp. 4443–4458, Association for Computational Linguistics, 2020.

[9] Z. Zhang, V. Setty, and A. Anand, “Sparcassist: A model risk assessment assistant based on sparsegenerated counterfactuals,” *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3219–3223, 7 2022.