Typo BERT-ATTACK

Seungil Lee
KAIST School of Computing
silly5921@kaist.ac.kr

Jongchan Park
KAIST School of Computing
jongchan.park@kaist.ac.kr

Eugene Seo
KAIST School of Computing
eugene.s21@kaist.ac.kr

Dohyung Kim
KAIST School of Computing
sidsince0428@kaist.ac.kr

Abstract

We suggest **Typo BERT-ATTACK**, an efficient and effective model generating adversarial examples using both typo and BERT. We addressed the original paper, BERT-ATTACK's problem, and added typo generation to provide various attack methods. As a result, **Typo BERT-ATTACK** generates human unrecognizable attack examples, while keeping the strengths of BERT-ATTACK. Through our experiments, the power of using two attack methods is verified. In addition, the optimal ratio between perturbation generated by typo and BERT are found in each dataset.

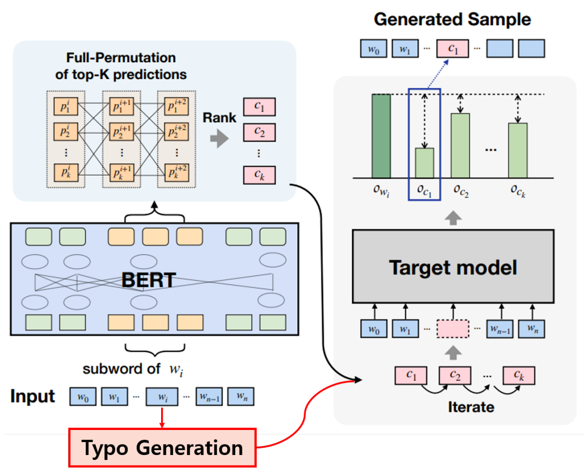

Figure 1: One step of the replacement strategy

1 Introduction

Recently, deep learning has achieved state-of-the-art advances in various topics. Nonetheless, recent adversarial attack works have verified the vulnerability of neural networks. Adversarial attack studies attempt to attack the model to make a mistake via adversarial examples. However, it is not easy to generate fluent adversarial examples on NLP models since the texts are discrete.

Previous approaches achieved successful attacks by adopting heuristic rules. HotFlip (Ebrahimi et al., 2017) introduced character-level error in a white-box setup, yet it can attack in a limited boundary. TextFooler (Jin et al., 2019) suggested changing the most vulnerable word into its synonym. It used sentence similarity to rank the semantic similarity, but it is still a context unaware.

BERT-ATTACK (Li et al., 2020) solved the previous papers' limitations via BERT (Devlin et al., 2018). Through BERT, BERT-ATTACK succeeded in generating fluent and semantic-consistent substitutions. Also, it reduced time costs by eliminating perturbation scoring steps.

We propose **Typo BERT-ATTACK**[1], which generates perturbed sentences using typos and BERT. Typos do not preserve the semantic of the original input, but it is difficult for humans to distinguish because it still looks almost identical to the original sentences. To generate appropriate typos, we referred to five methods suggested by Hagen et al. (2017). In addition, we found various severe problems during replication experiments and modified the code to generate more natural attack examples while keeping the original strengths.

Experimental results show that **Typo BERT-ATTACK** has a higher attack success rate than only using BERT-ATTACK. The optimal ratio between perturbations produced using typo and BERT is observed.

2 Dataset

On our development and experiments, we selected four datasets with a different tasks to test the overall performances of attack model.

Text Classification

[1]Codes are available here.
https://github.com/ChoiIseungil/BERT-Attack

Dataset	Model
IMDB	fabriceyhc/bert-base-uncased-imdb
AG's News	fabriceyhc/bert-base-uncased-ag_news
SNLI (p/h)	boychaboy/SNLI_bert-large-uncased
HateXplain (3 labels)	Hate-speech-CNERG/bert-base-uncased-hatexplain
HateXplain (2 labels)	Hate-speech-CNERG/bert-base-uncased-hatexplain-rationale-two

Table 1: Models used in the experiments.

- **IMDB**[2] Document-level binary sentiment classification dataset of movie reviews.
- **AG's News** Sentence-level news topic classification task by Zhang et al. (2015), which can classify world, sports, business, and science.
- **HateXplain** (Mathew et al., 2021) Hate speech detection dataset composed of posts on Twitter and Gab. Each post is annotated from three different perspectives: hate, offensive or normal.

Natural Language Inference

- **SNLI** (Bowman et al., 2015) Stanford natural language inference task dataset. The pair of one premise and one hypothesis sentences are given, and the goal is to predict the hypothesis is whether entailment, neutral, or contradiction.

3 BERT-ATTACK Improvement

Before implementing **Typo BERT-ATTACK**, we addressed the existing problems in BERT-ATTACK.

3.1 Approach

BERT-ATTACK attacks a word composed of sub-words as the following.

1. Pick top-K predictions for each sub-words
2. Make K^N permutations
3. Rank them with the perplexity
4. Use the top-K candidates

However, the official code calculates perplexity for only 24 candidates. We solved this problem by fixing the code, yet the attack time had soared. The reason was the existence of particular words composed of extremely many sub-words.

[2]https://datasets.imdbws.com/

In addition to the time cost, the original code generated a poor typo on words composed of sub-words. The perturbations on each sub-words resulted in a worse candidate word than a simple typo. Therefore, we limited the number of sub-words that can change. It prevented unacceptable attacks and solved the time-consuming problem.

Third, we filtered out the words containing punctuation marks from the target and the candidate list. The original BERT-ATTACK generated abnormal attack cases that generate a random text from a punctuation mark and vice versa. Those are not valid attacks since humans cannot infer the original test. Through our approach, the attack became natural.

Lastly, we changed the method of perplexity calculation. The original code calculated the perplexity by putting a single word into BERT. We concluded this is the usage out of the domain since BERT was not trained with single-word sentences. Therefore, we created a phrase containing words of the existing sequence before and after the candidate word, then the phrases are used as the input of BERT. In this way, we maximized BERT's functionality.

3.2 Experiment

We modify original model into 5 different types. Table 2 describes our setting for each models. For **Baseline** models, we only change the parameter of the original model. **All candidate** models are to resolve subword perturbation problem. They don't limit any candidates from subword perturbation. For punctuation problems, we introduce **Punc** models. They exclude any punctuation related words. To solve the execution time problem, **Subword** models limit the maximum subword that can be perturbed during subword perturbation. Finally, to improve the prediction from BERT, we change the policy of perplexity scoring. Instead of using a single word as the input, **Phrase** models use phrase.

Model name	Baseline		All candidates	Punc	Phrase	Subword
Details	Original Code		Consider all candidates	Punctuation excluded	Perplexity improvement	Limit subwords
Parameter	# of candidates	Subword perturbation	# of candidates	Subword perturbation	Length of phrase	# of subword

Table 2: Types of modified models

3.3 Result

With modified models, we experiment its performance with AG's news dataset. Table 3 shows our final experiment results.

Baseline We change parameter of original model to generate Baseline models. Baseline-1 model is equivalent to original model, Baseline-2 model reduces the number of candidates during subword perturbation, and Baseline-3 model doesn't attack the subwords. Among them, Baseline-1 and Baseline-2 show the best performance. Despite of their performances, we decide not to use those models because they are just changed their parameter so that still contains mentioned problems.

All candidates All candidates-1 generates 4 candidate per subword, but All candidates-2 makes 2 candidate per subword. Two All candidates models doesn't show the performance gap, yet All candidates-1 spends too much execution time.

Punc Punc-1 model excludes subword perturbation while Punc-2 does, while both of them solves problem related to punctuation marks. Between two models, Punc-2 shows better performance than Punc-1.

Subword To resolve the execution time problem of All candidates-1, we introduce subword model to limit subwords changes. Subword-1 allows only two subword to be changed during subword perturbation, and Subword-2 allows three of them. Subword-2 attacks successfully than Subword-1, and both models reduces execution time properly.

Phrase Phrase models improves the performance when model gets perplexity to generate candidates by using phrase rather than word itself. Phrase-1 and Phrase-2 uses 2 and 3 length phrase, respectively. Both improves the performance, while Phrase-2 model attacks better than Phrase-1.

Considering the performance of each model, we decided to make 2 different final models. Both final models use All candidates-1, Punc-2, and Subword-2. However, **Final-1** model uses Phrase-1 but **Final-2** model uses Phrase-2. We concluded to use **Final-2** as our new baseline since it shows the better performance.

Model	Accuracy	Time
Baseline-1	0.20	1290.892
Baseline-2	0.20	630.5436
Baseline-3	0.23	732.7911
All candidates-1	0.20	7205.729
All candidates-2	0.20	648.0994
Punc-1	0.30	743.0371
Punc-2	0.26	705.5875
Phrase-1	0.24	644.0462
Phrase-2	0.25	655.0946
Subword-1	0.26	698.7883
Subword-2	0.23	661.3538
Final-1	0.25	697.9477
Final-2	0.24	664.2827

Table 3: Performance of each modified models

Table 4 shows the performance of all other datasets. The overall accuracy after the attack inevitably became worse because of filtering all punctuation from the attack. Nonetheless, the number of unnatural attacks were significantly decreased.

3.4 Example

In Table 5, there is an example of how models attack the sentence. In this example, all changes of original model is related to bracket or punctuation. However, our new model (i.e. **Final-2**) doesn't show any punctuation marks during attack. Therefore, it gives more natural outputs, which are more suitable for adversarial attack.

4 Typo BERT-ATTACK

In this section, we designed the **Typo BERT-ATTACK** and evaluated its performance.

4.1 Approach

We set the result of Section 2 to our new baseline and added the typo attack method to increase the potential of the attack. We generated attack candidates with five different methods, and the details are the following.

Dataset	**Original Accuracy**	**Attacked Accuracy (original model)**	**Attacked Accuracy (new model)**
AG's News	92.6	18.4	26.0
IMDB	93.2	4.5	6.4
SNLI (hypothesis)	89.5	33.2	32.8
SNLI (premise)	89.5	32.8	33.1
HateXplain (Toxic/Non-toxic)	79.5	-	16.2
HateXplain (Hate/Normal/Offensive)	66.8	-	14.9

Table 4: Performances of the all datasets

Input sentence	teacher asks class where is pakistan little johnny replies outside with paki steve
Original model	(asks , where is pakistan little johnny (outside with paki steve
New model (Final-2)	teachers ask classes where is karachi kid jimmy reacts outside with katu steve

Table 5: Example of the adversarial attack of the original and new model

- Random Character Insertion
  (e.g. "search typo" → "search tyapo")
- Random Character Deletion
  (e.g. "search typo" → "search tpo")
- Random Character Substitution
  (e.g. "search typo" → "search type")
- Swap Neighbor Character
  (e.g. "search typo" → "search tyop")
- Swap Adjacent Keyboard Character
  (e.g. "search typo" → "search typi")

Algorithm 1 Typo Generation

```
procedure TYPO GENERATION(word, K)
    if length(word) < 5 then
        return []
    C ← []
    while length(C) < K do
        F ← Randomly chosen typo generation method
        w ← F (word)
        if w ∉ NLTK scope then
            C = C + w
    return C
```

With these typo generations, there might have some ambiguity like 'good' → 'gold'. It is because applying the above methods can generate the semantically different *complete* words, rather than *typo-like* words. Therefore, we filtered out these candidates with the NLTK WordNet dataset (Princeton, 2010). NLTK WordNet is the vocabulary-focused English dictionary dataset from Princeton University. It contains about 160k words and 120k synonym sets.

Also, we used *textattack* library (Morris et al., 2020) to implement the above typo generation methods.

Algorithm 1 describes the Typo Generation strategy.

Algorithm 2 is the pseudocode of the entire process of the **Typo BERT-ATTACK**. In the original algorithm, the process works in the following order,

1. Searching vulnerable words
2. Word replacement with BERT
3. Attacking the given model

Therefore, we add the previous typo generation algorithm, **Algorithm 1** before Step 3.

4.2 Experiment

Before conducting the experiments on **Typo BERT-ATTACK**, we designed our custom parameter, α. It is the proportion of the typo candidates among total candidates. Equation 1 is the expression of the parameter α.

$$\alpha = \frac{|typo\ candidate|}{|typo\ candidate| + |BERT\ candidate|} \quad (1)$$

We set up the attack experiment using the five different alpha values for each fine-tuned model. We purposed to find the ideal ratio of the typo candidates through this experiment.

Algorithm 2 Typo BERT-ATTACK

procedure IMPORTANCE RANKING(S)
    $S = [w_0, w_1, ...]$
    sort S using I_{w_i} in descending order
        where $I_{w_i} = o_y(S) - o_y(S_{\setminus w_i})$
        $o_y(S) =$ logit output for correct label y
    return S
procedure REPLACEMENT
    $S = [w_0, w_1, ...]$
    $H = [h_0, ..., h_n]$
    $L \leftarrow$ `Importance Ranking`(S)
    $P^{\in n \times K_{bert}} \leftarrow top - K_{bert}$ candidates for all subwords using BERT
    $S^{adv} = S$
    for $w_j \in L$ **do**
        if punctuation mark $\in w_j$ **then**
            Continue
        if w_j is a whole word **then**
            $C_{bert} \leftarrow Filter\ (P^j)$
        else
            $C_{bert} \leftarrow Filter\ (PPL(P^j))$
        $C_{typo} \leftarrow$ `Typo Generation` (w_j, K_{typo})
        $C_{total} = C_{bert} + C_{typo}$
        for $c_k \in C_{total}$ **do**
            if punctuation mark $\in c_k$ **then**
                Continue
            $S' = S^{adv}$
            $S'[k] = c_k$
            $Y \leftarrow$ gold-label
            if argmax$(o_j(S')) \neq Y$ **then**
                return $S^{adv} = S'$
            else
                if $(o_j(S')) < (o_j(S^{adv}))$ **then**
                    $S^{adv} = S'$
    return None

4.3 Result

Through experiments, we obtained an overall better attack success rate than BERT-ATTACK. Also, we observed the existence of an optimal α value for each dataset.

4.3.1 Attack Examples

There are three different cases of attack results.

- Case 1: Original BERT-ATTACK succeeded, 100% Typo-ATTACK failed.
- Case 2: Original BERT-ATTACK failed, 100% Typo-ATTACK succeeded.
- Case 3: Original BERT-ATTACK, 100% Typo-ATTACK failed, only Mixed ATTACK succeeded.

The detailed examples are in the Appendix A. Also, we noticed that the results of Mixed ATTACK is more fluent than the results of 100% Typo-ATTACK.

4.3.2 Attacked Accuracy

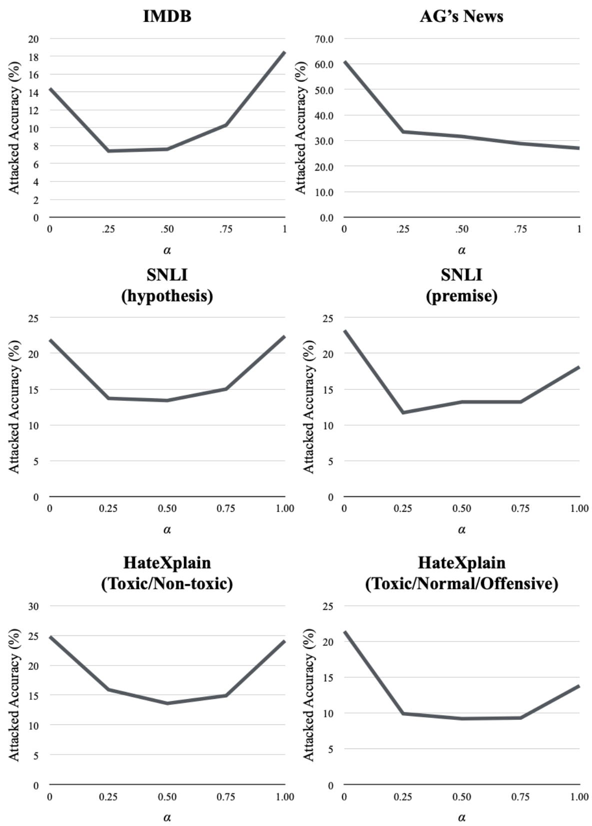

Figure 2: Attacked accuracy(%) of 6 datasets with resppect to α

Table 6 quantitatively shows attacked accuracies of 30 experiments. The last column is the original accuracies of the new baseline discussed in the Section 3.3. Remaining five datasets except for AG's News, the optimal α value was lied between 0 and 1. 100% typos was the best for the AG's News unlike other datasets.

	0	.25	.50	.75	1	Original
IMDB	14.4	**7.4**	7.6	10.3	18.5	93.2
AG's News	61.0	33.4	31.6	28.8	**27.0**	92.6
SNLI (hypothesis)	21.9	13.7	**13.4**	15.0	22.4	89.5
SNLI (premise)	23.2	**11.7**	13.2	13.2	18.1	89.5
HateXPlain (Toxic/Non-toxic)	21.4	15.9	**13.6**	14.9	24.1	79.5
HateXPlain (Hate/Normal/Offensive)	21.4	9.9	**9.2**	9.3	13.8	66.8

Table 6: Attacked accuracy(%). The bold typfaces are indicating the best attack

4.4 Limitation and Future Work

Although we tried to make a thorough analysis while implementing **Typo BERT-ATTACK**, there were two major limitations involved.

Input sentence	A man in a black tank top wearing a red plaid hat.
Attacked sentence	A man in a bmack tank top wearnig a red plyid hat.

Table 7: Examples of the unnatural typos

Typo unnaturalness The main reason we adopt typos to make an attack is that, we supposed the typos are difficult to visually detected by human, while it still decieves the neural models. Therefore the most important premise of typos is that it is visually inconspicuous to human. However, contrary to our goal, most of the generated typos were very unnatrual. Let's make an example in Table 7. While the word *wearing* is changed to the word *wearnig*, which is not easy to be detected, but word *black* and *plaid* are so easy to be caught by human. These examples are the out of our purpose. The future work will try an approach that restricts typo generation so that humans cannot recognize it well through making reliable human research.

Explainability It is remarkable that we performed better than the existing **Typo BERT-ATTACK**, the big limitation is the explainability. We could not explain the reason why mixing typos and original method is good at attack. Also we could not figure out while optimal α varies among the dataset. We expect this is due to the unique characteristics of each dataset. Further analysis have to be studied to figure out this reasons.

5 Conclusion

Through a thorough analysis in Section 3 and Section 4, we came to the following conclusions.

First, better than no typos at all. Table 6 shows that typos always enhances the attack performance. We suppose this is because typos are able to break semantics of the target words very easily, so it helps fooling the neural model.

Second, typos were not always the best. 100% typos were not the optimal except the AG's News. There were some optimal value for α, and this was quite surprising because we first expected typos are much more superior at attacking the model. Therefore figure 2 shows a U-shaped graph in which the attack performance decreases as the difference from the optimal α increases. This result means that the effect of the typos are varying with the characteristics of each dataset, and we suppose this is because the typos are not able to change short and important words, because we limited to create typos only for word with length of 5 or more.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *CoRR*, abs/1508.05326.

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.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for NLP. *CoRR*, abs/1712.06751.

Matthias Hagen, Martin Potthast, Marcel Gohsen, Anja Rathgeber, and Benno Stein. 2017. A large-scale query spelling correction corpus. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 1261–1264, New York, NY, USA. Association for Computing Machinery.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is BERT really robust? natural language attack on text classification and entailment. *CoRR*, abs/1907.11932.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: adversarial attack against BERT using BERT. *CoRR*, abs/2004.09984.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Princeton. 2010. About wordnet.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.

A Appendix: Attack Examples

Type	Sentence	Label
Original	Profiling Shaukat Aziz : economic reformist - / - PM . Shaukat Aziz , taking over as Pakistan # 39;s 23rd prime minister on Saturday , is a former private banker credited with infusing new life into an almost bankrupt economy .	World News
BERT	profiling shaukat aziz : monetary reformist - / - pm . shaukat aziz , taking over as pakistan # 39;s 23rd prime minister on saturday , is a former private banker credited with infusing new life into an almost bankrupt economics .	Business News
Typo	proflling shaukamt aziz : eqconomic reformitt - / - pm . shauat aziz , takibg over as apkistan # 39;s 23rd peime miniter on asturday , is a former privaye bankre eredited with infusind new life into an almost badnkrupt econony .	World News

Table 8: Example of Case 1. Original BERT-ATTACK succeeded / 100% Typo ATTACK failed

Type	Sentence	Label
Original	Giddy Phelps Touches Gold for First Time . Michael Phelps won the gold medal in the 400 individual medley and set a world record in a time of 4 minutes 8.26 seconds .	Sports News
BERT	iddy phelps touch silver for first times . mike phelps won the golden award in the butterfly 400 aquatics and sets a world records in a hours of 7 min 8.26 min .	Sports News
Typo	kgiddy phleps tbuches gold for first time . mkchael phelp won the gold mtdal in the 400 individkal medlye and set a world ecord in a time of 4 mknutes 8.26 seconds .	Business News

Table 9: Example of Case 2. Original BERT-ATTACK failed / 100% Typo ATTACK succeeded

Type	Sentence (Premises: a woman holding a scruffy cat.)	Label
Original	Hypothesis: An elderly lady holding a scruffy dog and smiling contently .	Contradiction
BERT	Hypothesis: An aged female holding a scruffy hound and smiling contently .”	Contradiction
Typo	Hypothesis: an eldejly lady hloding a csruffy dog and smilinjg contentyl .	Contradiction
Mixed	Hypothesis: an eplderly female holdihg a scruffy dogs and smilimg contently .	Entail

Table 10: Example of Case 3. Original BERT-ATTACK & Typo ATTACK failed / Mixed ATTACK succeeded