Robustness to Modification with Shared Words in Paraphrase Identification

Zhouxing Shi
DCST, Tsinghua University,
Beijing 100084, China
zhouxingshichn@gmail.com

Minlie Huang*
DCST, THUAI, SKLits, BNRist
Tsinghua University,
Beijing 100084, China
aihuang@tsinghua.edu.cn

Abstract

Revealing the robustness issues of natural language processing models and improving their robustness is important to their performance under difficult situations. In this paper, we study the robustness of paraphrase identification models from a new perspective – via modification with shared words, and we show that the models have significant robustness issues when facing such modifications. To modify an example consisting of a sentence pair, we either replace some words shared by both sentences or introduce new shared words. We aim to construct a valid new example such that a target model makes a wrong prediction. To find a modification solution, we use beam search constrained by heuristic rules, and we leverage a BERT masked language model for generating substitution words compatible with the context. Experiments show that the performance of the target models has a dramatic drop on the modified examples, thereby revealing the robustness issue. We also show that adversarial training can mitigate this issue.

1 Introduction

Paraphrase identification is to determine whether a pair of sentences have the same meaning (Socher et al., 2011), with many applications such as duplicate question matching on social media (Iyer et al., 2017) and plagiarism detection (Clough, 2000). It can be viewed as a sentence matching problem, and many neural models have achieved great performance on benchmark datasets (Wang et al., 2017; Gong et al., 2017; Devlin et al., 2018).

Despite this progress, there is not much work on the *robustness* of paraphrase identification models, while natural language processing (NLP) models on other tasks have been shown to be vulnerable and lack of robustness. In previous works for the robustness of NLP models, constructing semantic-preserving perturbations to input sentences while making the model prediction significantly change appears to be a popular way, in tasks such as text classification and natural language inference (Alzantot et al., 2018; Jin et al., 2019). However, on specific tasks, it is possible to design modification that is *not* necessarily semantic-preserving, which can further reveal more robustness issues. For instance, on reading comprehension, Jia and Liang (2017) conducted modification by inserting distracting sentences to the input paragraphs. Such findings can be important for investigating and resolving the weakness of NLP models.

On paraphrase identification, to the best of our knowledge, the only previous work is PAWS (Zhang et al., 2019) with a cross-lingual version (Yang et al., 2019), which found that models often make false positive predictions when words in the two sentences only differ by word order. However, this approach is for negative examples only, and for positive examples, they used back-translation to still generate semantically similar sentences. Moreover, it was unknown whether models still easily make false positive predictions when the word overlap between the two sentences is much smaller than 100%.

In this paper, we propose an algorithm for studying the robustness of paraphrase identification models from a new perspective – via modifications with **shared words (words that are shared by both sentences)**. For positive examples, i.e., the two sentences are paraphrases, we aim to see whether models can still make correct predictions when some shared words are replaced. Each pair of selected shared words are replaced with a new word, and the new example tends to remain positive. As the first example in Figure 1 shows, by replacing "purpose" and "life" with "measure" and "value" respectively, the sen-

* Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 164–171

(P)	What is ultimate **purpose** of **life**?
(Q)	What is the **purpose** of **life** , if not money?
(P')	What is ultimate **measure** of **value**?
(Q')	What is the **measure** of **value** , if not money?
Label	*Positive*
Output	*Positive* (99.4%) → *Negative* (85.2%)
(P)	How can I get my **Gmail account** back ?
(Q)	What is the best **school management** software ?
(P')	How can I get my **credit score** back ?
(Q')	What is the best **credit score** software ?
Label	*Negative*
Output	*Negative* (100.0%) → *Positive* (68.3%)

Figure 1: Examples with labels *positive* and *negative* respectively, originally from Quora Question Pairs (QQP) (Iyer et al., 2017). "(P)" and "(Q)" are original sentences while "(P')" and "(Q')" are modified. Modified words are highlighted in bold. "Output" indicates the change of output labels by BERT (Devlin et al., 2018), where the percentage numbers are confidence scores.

tences change from asking about "purpose of life" to "measure of value" and remain paraphrases, but the target model makes a wrong prediction. This indicates that the target model has a weakness in generalizing from "purpose of life" to "measure of value". On the other hand, for negative examples, we replace some words and introduce new shared words to the two sentences while trying to keep the new example negative. As the second example in Figure 1 shows, with new shared words "credit" and "score" introduced, the new example remains negative but the target model makes a false positive prediction. This reveals that the target model can be distracted by the shared words while ignoring the difference in the unmodified parts. The unmodified parts of the two sentences have a low word overlap to reveal such a weakness. In contrast, examples in PAWS had exactly the same bag of words and are not capable for this investigation.

In our word replacement, to preserve the label and language quality, we impose heuristic constraints on replaceable positions. Furthermore, we apply a BERT masked language model (Devlin et al., 2018) to generate substitution words compatible with the context. We use beam search to find a word replacement solution that approximately maximizes the loss of the target model and thereby tends to make the model fail.

We summarize our contributions below:

- We study the robustness of paraphrase identification models via modification with shared words. Experiments show that models have a severe performance drop on our modified examples, which reveals a robustness issue.
- We propose a novel and concise method that leverages the BERT masked language model for generating substitution words compatible with the context.
- We show that adversarial training with our generated examples can mitigate the robustness issue.
- Compared to previous works, our perspective is new: 1) Our modification is not limited to be semantic-preserving; and 2) Our negative examples have much lower word overlap between two sentences, compared to PAWS.

2 Related Work

2.1 Paraphrase Identification Models

There exist many neural models for sentence matching and paraphrase identification. Some works applied a classifier on independently-encoded embeddings of two sentences (Bowman et al., 2015; Yang et al., 2015; Conneau et al., 2017), and some others made strong interactions between the two sentences by jointly encoding and matching them (Wang et al., 2017; Duan et al., 2018; Kim et al., 2018) or hierarchically extracting features from their interaction space (Hu et al., 2014; Pang et al., 2016; Gong et al., 2017). Notably, BERT pre-trained on large-scale corpora achieved even better results (Devlin et al., 2018).

2.2 Robustness of NLP Models

On the robustness of NLP models, many previous works constructed semantic-preserving perturbations to input sentences (Alzantot et al., 2018; Iyyer et al., 2018; Ribeiro et al., 2018; Hsieh et al., 2019; Jin et al., 2019; Ren et al., 2019). However, NLP models for some tasks have robustness issues not only when facing semantic-preserving perturbations. In reading comprehension, Jia and Liang (2017) studied the robustness issue when a distractor sentence is added to the paragraph. In natural language inference, Minervini and Riedel (2018) considered logical rules of sentence relations, and Glockner et al. (2018) used single word replacement with lexical knowledge. Thus methods for general NLP tasks alone are insufficient for studying the robustness of specific tasks. In particular, for paraphrase identification, the only prior work

is PAWS (Zhang et al., 2019; Yang et al., 2019) which used word swapping, but this method is for negative examples only and each constructed pair of sentences have exactly the same words.

3 Methodology

3.1 Algorithm Framework

Paraphrase identification can be formulated as follows: given two sentences $P = p_1 p_2 \cdots p_n$ and $Q = q_1 q_2 \cdots q_m$, the goal is to predict whether P and Q are paraphrases. The model outputs a score $[Z(P, Q)]_{\hat{y}}$ for each class $\hat{y} \in \mathcal{Y} = \{positive, negative\}$, where *positive* means P and Q are paraphrases and vice versa.

We first sample an original example from the dataset and then conduct modification. We take multiple steps for modification until the model fails or the step number limit is reached. In each step, we replace a word pair with a shared word, and we evaluate different options according to the model loss they induce. We use beam search to find approximately optimal options. The modified example evaluated as the best option is finally returned.

In the remainder of this section, we introduce what modification options are considered available to our algorithm in Sec. 3.2 and how to find optimal modification solutions in Sec. 3.3.

3.2 Modification Options

Original Example Sampling To sample an original example from the dataset, for a positive example, we directly sample a *positive* example from the original data, namely, $(P, Q, positive)$; and for a negative example, we sample two different sentence pairs (P_1, Q_1) and (P_2, Q_2), and we then form a negative example $(P_1, Q_2, negative)$.

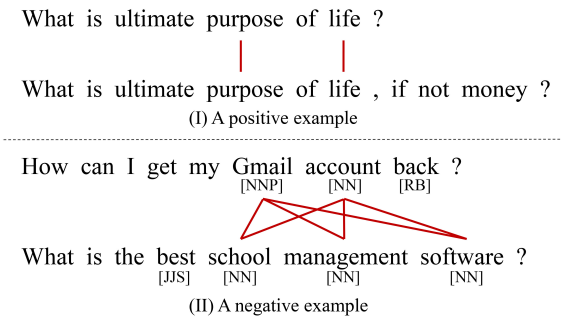

Figure 2: Examples of identifying replaceable position pairs that are linked with red lines. In the negative example, POS tags of non-stopwords are also shown.

Replaceable Position Pairs For a sentence pair under modification, we impose heuristic rules on replaceable position pairs. First, we do not replace stopwords. Besides, for a positive example, we require each replaceable word pair to be shared words, while for a negative example, we only require them to be both nouns, both verbs, or both adjectives, according to Part-of-Speech (POS) tags obtained using Natural Language Toolkit (NLTK) (Bird et al., 2009). Two examples are shown in Figure 2. For the first example (positive), only shared words "purpose" and "life" can be replaced, and the two modified sentences are likely to talk about another *same* thing, e.g. changing from "purpose of life" to "measure of value", and thereby the new example tends to remain *positive*. As for the second example (negative), nouns "Gmail", "account", "school", "management" and "software" can be replaced. Consequently, the modified sentences are based on templates "How can I get $\cdots$ back ? " and "What is the best $\cdots$?", and the pair tends to remain negative even if the template is filled by shared words. In this way, the labels can usually be preserved.

Substitution Words We use a pre-trained BERT masked language model (Devlin et al., 2018) to generate substitution words compatible with the context, for each replaceable position pair. Specifically, to replace word p_i and q_j from the two sentences respectively with some shared word w, we compute a joint probability distribution

$$\begin{aligned}&\mathcal{P}(w|p_{1:i-1}, p_{i+1:n}, q_{1:j-1}, q_{j+1:m})\\ =&\mathcal{P}(w|p_{1:i-1}, p_{i+1:n}) \cdot \mathcal{P}(w|q_{1:j-1}, q_{j+1:m}),\end{aligned}$$

where $s_{i:j}$ denotes the subsequence starting from i to j. $\mathcal{P}(w|p_{1:i-1}, p_{i+1:n})$ and $\mathcal{P}(w|q_{1:j-1}, q_{j+1:m})$ are obtained from the language model by masking p_i and q_j respectively. We rank all the words in the vocabulary of the model and choose top K words with largest probabilities, as the candidate substitution words for the position pair.

This method of generating substitution words enables us to find out possible substitution words and also verify their compatibility with the context simultaneously, compared to previous methods that have these two separated (Alzantot et al., 2018; Jin et al., 2019) – they first constructed a candidate substitution word list from synonyms, and using each substitution word respectively, they then checked the language quality or semantic

similarity constraints of the new sentence. Moreover, some recent works (Li et al., 2020; Garg and Ramakrishnan, 2020) that appeared later than our preprint have shown that using a masked language model for substituting words can outperform state-of-the-art methods in generating adversarial examples on text classification and natural language inference tasks.

3.3 Finding Modification Solutions

We then use beam search with beam size B to find a modification solution in multiple steps. At step t, we have two stages to determine the replaced positions and the substitution words respectively, based on a two-stage framework (Yang et al., 2018).

First, for replaced positions, we enumerate all replaceable position pairs and replace words on each pair of positions with a special token *[PAD]* respectively. We then query the model with these new examples and take top B examples that minimizes the output score of the gold label. Next, we enumerate all words in the candidate substitution word set of positions with *[PAD]* and replace *[PAD]* with each candidate substitution word respectively. We again query the model with the examples after each possible replacement, and we take top B examples similarly as in the first stage. For the topmost example, if the label predicted by the model is already incorrect, we finish the modification process. Otherwise, we take more steps until the model fails or the step number limit S is reached.

4 Experiments

4.1 Datasets and Target Models

We adopt two datasets. The Quora Question Pairs, **QQP** (Iyer et al., 2017), consists of 384,348/10,000/10,000 question pairs in the training/development/test set as we follow the partition in Wang et al. (2017). And the Microsoft Research Paraphrase Corpus, **MRPC** (Dolan and Brockett, 2005), consists of sentence pairs from news with 4,076/1,725 pairs in the training/test set. Each sentence pair is annotated with a label indicating whether the two sentences are paraphrases or not (*positive* or *negative*).

We study three typical models for paraphrase identification. **BiMPM** (Wang et al., 2017) matches two sentences from multiple perspectives using BiLSTM layers. **DIIN** (Gong et al., 2017) adopts DenseNet (Huang et al., 2017) to extract interaction features. **BERT** (Devlin et al., 2018) is a pre-trained encoder fine-tuned on this task with a classifier applied on encoded representations. These models are representative in terms of backbone neural architectures: BiMPM is based on recurrent neural networks, DIIN on convolutional neural networks, and BERT on Transformers.

4.2 Performance on Modified Examples

We train each model on the original training set and then try to construct modification that makes the models fail. For each dataset, we sample 1,000 original examples with balanced labels from the test set, and we modify them for each model. We evaluate the accuracies of the models on our modified examples. Table 1 shows the results. We focus on rows with "normal" for column "training" in this section. The models have high overall accuracies on the original data, but their performance drops dramatically on our modified examples (e.g., the overall accuracy of BERT on QQP drops from 94.3% to 24.1%). This demonstrates that the models indeed have the robustness issue we aim to reveal. Some examples are provided in Appendix B.

4.3 Adversarial Training

To improve the model robustness, we further fine-tune the models using adversarial training. A training batch consists of original examples and modified examples from the training data, where modified examples account for around 10% in a batch. The proportion of modified examples is directly chosen to demonstrate the effectiveness of adversarial training while preventing the model from overfitting on modified examples. During training, we modify examples with the current model as the target and update the model parameters iteratively. The beam size for generation is set to 1 to reduce the computational cost. We evaluate the adversarially trained models as shown in Table 1 (rows with "adversarial" for column "training"). The performance on modified examples of all the models raises significantly (e.g. the overall accuracy of BERT on modified examples raises from 24.1% to 66.0% for QQP and from 23.8% to 87.0% for MRPC). This demonstrates that adversarial training with our modified examples can significantly improve the robustness, yet without remarkably hurting the performance on original

Table 1: Accuracies (%) of target models: "Original full" indicates the full original test set, "original sampled" indicates original examples sampled from the test set (see Sec. 3.2), and "modified" indicates examples modified by our algorithm. "Pos" and "neg" indicate results on positive examples and negative examples respectively. The "training" column indicates whether the models are normally trained or adversarial trained (see Sec. 4.3).

Dataset	Target Model	Training	Original full			Original sampled			Modified		
			Pos	Neg	All	Pos	Neg	All	Pos	Neg	All
QQP	BiMPM	Normal	88.5	87.8	88.1	88.0	99.4	93.7	14.4	7.8	11.1
	DIIN		91.5	85.9	88.7	89.6	99.6	94.6	31.0	8.2	19.6
	BERT		90.7	91.3	91.0	89.0	99.6	94.3	33.4	14.8	24.1
	BiMPM	Adversarial	89.6	88.0	88.9	89.4	99.8	94.6	15.0	27.8	21.4
	DIIN		82.1	91.7	86.9	81.2	99.8	90.5	35.0	72.2	53.6
	BERT		87.6	92.5	90.1	86.8	99.8	93.3	53.0	79.0	66.0
MRPC	BiMPM	Normal	90.2	40.0	73.4	87.2	97.4	92.3	3.2	0.2	1.7
	DIIN		89.9	49.5	76.3	90.4	100.0	95.2	48.2	0.4	24.3
	BERT		93.2	66.4	84.2	94.0	100.0	97.0	45.6	2.0	23.8
	BiMPM	Adversarial	96.8	26.3	73.2	95.6	100.0	97.8	73.2	0.6	36.9
	DIIN		85.8	58.0	76.5	82.8	100.0	91.4	59.8	67.6	63.7
	BERT		95.3	55.2	81.9	95.0	100.0	97.5	81.0	93.0	87.0

data. An improvement on the *original* data is not expected since they cannot reflect robustness and it is even common to see a small drop in previous works (Jia and Liang, 2017; Iyyer et al., 2018; Ribeiro et al., 2018).

4.4 Manual Evaluation

Table 2: Manual annotation results on original examples and modified examples respectively, including accuracies and grammaticality ratings.

Dataset	Metric	Original	Modified
QQP	Accuracy - Pos	86%	70%
	Accuracy - Neg	98%	88%
	Accuracy - All	92%	79%
	Grammaticality	2.48	2.15
MRPC	Accuracy - Pos	90%	94%
	Accuracy - Neg	100%	82%
	Accuracy - All	95%	88%
	Grammaticality	2.40	2.19

We also manually verify the quality of the modified examples in terms of the label correctness and grammaticality. For each dataset, using BERT as the target, we randomly sample 100 modified examples with balanced labels such that the model makes wrong predictions, and we present each of them to three workers on Amazon Mechanical Turk. We ask the workers to label the examples and also rate the grammaticality of the sentences with a scale of 1/2/3. We integrate annotations from different workers with majority voting for labels and averaging for grammaticality. Results are shown in Table 2. We observe that the workers achieve acceptable accuracies on our modified examples (79% on QQP and 88% on MRPC), while their performance on original examples is not perfect either (92% on QQP and 95% on MRPC). The grammaticality drop between original examples and modified examples is also satisfactory (from 2.48 to 2.15 on QQP and from 2.40 to 2.19 on MRPC). These results suggest that the labels and grammaticality of the modified examples can be preserved with an acceptable quality.

5 Conclusion

In this paper, we present a novel algorithm to study the robustness of paraphrase identification models. We show that the target models have a robustness issue when facing modification with shared words. Such modification is substantially different from those in previous works – the modification is not semantic-preserving and each pair of modified sentences generally have a much lower word overlap, and thereby it reveals a new robustness issue. We also show that model robustness can be improved using adversarial training with our modified examples.

Acknowledgments

This work was jointly supported by the NSFC projects (Key project with No. 61936010 and regular project with No. 61876096), and the Guoqiang Institute of Tsinghua University, with Grant No. 2019GQG1. We thank THUNUS NExT Joint-Lab for the support.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Paul Clough. 2000. Plagiarism in natural and programming languages: an overview of current tools and technologies.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

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

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Chaoqun Duan, Lei Cui, Xinchi Chen, Furu Wei, Conghui Zhu, and Tiejun Zhao. 2018. Attention-fused deep matching network for natural language inference. In *IJCAI*, pages 4033–4040.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655.

Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. 2019. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1520–1529.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2019. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*.

Seonhoon Kim, Jin-Hyuk Hong, Inho Kang, and Nojun Kwak. 2018. Semantic sentence matching with densely-connected recurrent and co-attentive information. *arXiv preprint arXiv:1805.11360*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Pasquale Minervini and Sebastian Riedel. 2018. Adversarially regularising neural nli models to integrate logical background knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 65–74.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. 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.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems*, pages 801–809.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4144–4150.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I Jordan. 2018. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *arXiv preprint arXiv:1805.12316*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3678–3683.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

A Implementation Details

We adopt open source codes for BiMPM[1], DIIN[2] and BERT ($\text{BERT}_{\text{base}}$ is used)[3], and the datasets are downloaded from the internet for both QQP[4] and MRPC[5]. There are 1.4, 42.8, and 109.5 million parameters in BiMPM, DIIN and BERT respectively.

For QQP, the step number limit of modification, S, is set to 5; the number of candidate substitution words suggested by the language model, K, and the beam size B are both set to 25. S, K and B are doubled for MRPC where sentences are generally longer.

We conduct the experiments on an NVIDIA TITAN X GPU. On QQP, the average time cost per example is around 4.7s for positive examples and 7.5s for negative examples. On MRPC, it is around 44.9s for positive examples and 61.6s for negative examples.

B Examples of Our Modifications

Table 3: Modified examples for BERT as the target model on QQP. "(P)" and "(Q)" indicate original sentences, and "(P')" and "(Q')" indicate modified sentences. Modified words are highlighted in bold.

(P)	How can I **lose weight** at age 55 ?
(Q)	What are some ways to **lose weight** fast ?
(P')	How can I **buy anything** at age 55 ?
(Q')	What are some ways to **buy anything** fast ?
Label	Positive
Output	Positive → Negative
(P)	If **infinite dark/vacuum/gravitational energy** can be created as universe expands , does it mean that their **potentiality** or potential **energy** is infinite ?
(Q)	What are **good gifts** for a **foreign** visitor to bring when they 're invited to someone 's **home** in **Vietnam** for the first time ?
(P')	If **local global interactions** can be created as universe expands , does it mean that their **existence** or potential **plane** is infinite ?
(Q')	What are **global interactions** for a **local** visitor to bring when they 're invited to someone 's **plane** in **existence** for the first time ?
Label	Negative
Output	Negative → Positive

Table 4: Typical modified examples for BERT as the target model on MRPC.

(P)	The **spacecraft** is scheduled to **blast** off as early as tomorrow or as late as Friday from the Jiuquan **launching site** in the **Gobi Desert** .
(Q)	The **spacecraft** is scheduled to **blast** off between next Wednesday and Friday from a **launching site** in the **Gobi Desert** .
(P')	The **match** is scheduled to **kick** off as early as tomorrow or as late as Friday from the Jiuquan **long day** in the **hot summer** .
(Q')	The **match** is scheduled to **kick** off between next Wednesday and Friday from a **long day** in the **hot summer** .
Label	Positive
Output	Positive → Negative
(P)	The **resolution** was approved with no **debate** by delegates at the bar association 's annual meeting here .
(Q)	Morales , who pleaded guilty in July , expressed " sincere regret and **remorse** " for his **crimes** .
(P')	The **loss** was approved with no **surprise** by delegates at the bar association 's annual meeting here .
(Q')	Morales , who pleaded guilty in July , expressed " sincere regret and **surprise** " for his **loss** .
Label	Negative
Output	Negative → Positive

We show some examples that our modification with shared words can make the target model fail, to further illustrate the robustness issue we reveal. Table 3 presents two examples using BERT as the target model on QQP. For the first example (positive), changing from asking about "lose weight" to "buy anything" fools the target model to alter the predicted label, though the modified sentences are still asking about the same thing and are paraphrases. For the second example (negative), introducing new shared words "local", "global", "interactions", "existence" and "plane" fools the target model to predict that the modified sentences are paraphrases, although the new sentences are still asking about different things. Similarly, Table 4 presents two examples on MRPC.

[1] https://github.com/zhiguowang/BiMPM

[2] https://github.com/YichenGong/Densely-Interactive-Inference-Network

[3] https://github.com/huggingface/transformers

[4] https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[5] https://www.microsoft.com/en-us/download/details.aspx?id=52398