Enhancing Robustness of Pre-trained Language Model with Lexical Simplification

Anonymous ACL submission

Abstract

For both human readers and pre-trained language models (PrLMs), lexical diversity may lead to confusion and inaccuracy when understanding the underlying semantic meanings of given sentences. By substituting complex words with simple alternatives, lexical simplification (LS) is a recognized method to reduce such lexical diversity. In this paper, we leverage a novel improved LS approach which can enhance robustness of PrLMs, resulting in improved performances in downstream tasks. A rule-based simplification process is applied to a given sentence. PrLMs are encouraged to predict the real label of the given sentence with auxiliary inputs from the simplified version. Using strong PrLMs (BERT and ELECTRA) as baselines, our approach can still further improve the performance in various text classification tasks.

1 Introduction

Pre-trained language models (PrLMs) such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and ELECTRA (Clark et al., 2020) have led to strong performance gains in downstream natural language understanding (NLU) tasks. However, (Li et al., 2020; Jin et al., 2019) demonstrate that it only takes a few simple synonym replacements to mislead the prediction of PrLMs on various text classification tasks. Such result indicates that lexical diversity can pose a negative impact on the accuracy of semantic meaning understanding for PrLMs.

In order to reduce lexical diversity, previous works have proposed some approaches for lexical simplification (LS) (Gooding and Kochmar, 2019; Qiang et al., 2020). By substituting complex words with their simpler alternatives in original sentences, LS can generate a simplified sentence version, which is much easier to understand for human readers. Inspired by these studies, we leverage LS as a paraphrasing tool to enhance the robustness of PrLM, which is verified in text classification tasks.

It is crucial to let well-designed LS rules customized to neural network (eg. PrLM in our case). However, existing LS methods (Gooding and Kochmar, 2019; Qiang et al., 2020) do not well suit PrLMs as they were originally proposed for human readers to simplify reading process, but not for strengthening neural network. Especially, the current LS methods are very time-consuming. This is because they apply large pre-trained neural networks to detect and replace the complex words in a recursive way(Qiang et al., 2020). Therefore, we design a lexical simplification method based on lemmatization and rare word replacement (abbreviated as LRLS), which is more effective and serves better to our purpose, to generate simplified version of give sentence.

In order to better accommodate the LRLS lexical simplification method with PrLMs and improve the overall performance, an framework (LRLS-Aux) which leverage simplified sentence generated by LRLS as an auxiliary input in both training and inference phase of PrLMs is designed and executed. In this way, PrLMs are able to make the right decision based on both the original sentence and the simplified perspective. Thus, the challenge posed by lexical diversity can be significantly reduced.

A series experiments are conducted on various text classification tasks. Empirical results show that our approach can notably improve the performance PrLMs. Meanwhile, ablation studies further verify the effectiveness of our LRLS method. Furthermore, we also compare our LRLS method with other paraphrasing method used in data augmentation, such as randomly replacement of several words by synonyms (Wu et al., 2019; Wei and Zou, 2019), back-translation (Xie et al., 2019; Edunov et al., 2018), cutoff (Shen et al., 2020). Analysis result demonstrate that our LRLS method remains

Model	SST-2	MR	CR	SUBJ	AG	Avg
BERT$_{BASE}$	92.4	86.1	90.0	97.3	94.2	92.0
+LRLS-Aux	93.5(+1.1)	88.1(+2.0)	90.8(+0.8)	98.0(+0.7)	95.0(+0.8)	93.1(+1.1)
ELECTRA$_{LARGE}$	96.7	90.0	94.3	97.4	94.6	94.6
+LRLS-Aux	**97.5**(+0.8)	**91.4**(+1.4)	**94.5**(+0.2)	**98.1**(+0.7)	**95.3**(+0.7)	**95.3**(+0.7)

Table 1: performances (%) across five text classification tasks for models with and without LRLS-Aux.

the most effective.

2 Method

2.1 LRLS Lexical Simplification Process

Previous works (Li et al., 2020; Jin et al., 2019) show that the prediction of PrLMs would be easily misled by replacing only a few words with their synonyms in the given sentences. By carefully observing the adversarial examples, we find that changing the tense of verbs, changing the singular and plural form of nouns, and replacing words by its less frequent synonyms compose the majority of the adversarial examples. The observation is also confirmed by (Mozes et al., 2020).

Inspired by such observation, our LRLS method is developed with two major steps: (1) lemmataization by transforming verbs and nouns into corresponding lemmas, and (2) replacing rare words with theirs more common synonyms. Firstly, we employ a third-party part-of-speech tagger [1] to annotate verbs and nouns in given sentences, and transform every verb to its infinitive form and every noun to its singular form. Secondly, according to a word frequency list[2], we label every word whose frequency is less than a frequency threshold n_f as a rare word. We then use a word embedding from (Mrkšić et al., 2016), which is specially curated for locating synonyms, to find the top n_s synonyms of identified rare words with the highest cosine similarity. Each rare word is replaced by its synonym with the highest frequency. A part-of-speech (POS) check is also applied to ensure that all the synonymous candidates hold the same POS as the original words.

2.2 Simplified Sentence As Auxiliary Input

Following (Devlin et al., 2018), the original sentence and its simplified version are combined together in to a single sentence. In our approach, the original and simplified sentences are differentiated in two ways. First, a special separation token ([SEP]) is inserted between the two sentences. Second, a learned segmentation embedding is added to every token which indicates whether it belongs to the original sentence or the simplified sentence. In both training and inference phases, we feed PrLMs the original-simplified sequence as inputs. The rest of implementations remain the same as the original PrLMs.

[1]Natural Language Toolkit (NLTK) (Bird et al., 2009)
[2]https://github.com/hermitdave/FrequencyWords

3 Experimental Setup

3.1 Benchmark Datasets

To verify if our proposed method can indeed enhance the performance of PrLMs, we conduct our experiments on five benchmark text classification tasks: (1) SST-2: Stanford Sentiment Treebank (Socher et al., 2013), (2) CR: customer reviews (Hu and Liu, 2004; Liu et al., 2015), (3) SUBJ: subjectivity/objectivity dataset (**?**), (4) MR:Movie reviews (Pang and Lee, 2005), and (5) AG: AG's News (Zhang et al., 2015), classification task with regard to four news topics: World, Sports, Business, and Science.

3.2 Baseline Models

We use (1) BERT-base (Devlin et al., 2018) with 12 layers, 768 hidden units, 12 heads and 110M parameters, and (2) ELECTRA-large (Clark et al., 2020) with 24 layers, 1024 hidden units, 16 heads and 340M parameters as our baseline PrLMs.

4 Experiments

In this section, comprehensive experiments and analysis are conducted. For all the experiments, we average results from three different random seeds.

4.1 Our Approach Make Gains

As shown in Table 1, we run both BERT-base (Devlin et al., 2018) and ELECTRA-large (Clark et al., 2020), with and without LRLS-Aux, across all five datasets. The average gain is 1.1 for BERT-base and 0.7 for ELECTRA-large. As ELECTRA-large is a very strong baseline, the result prove the effectiveness of our approach. As show in Figure 1,

		Input sentence	Result	
1	**Original**	You may feel **compelled** to watch the film twice or pick up a book on the subject.	**Baseline**	Negative
	Simplified	You may feel **obliged** to watch the film twice or pick up a book on the subject.	**LRLS-Aux**	Positive
			Gold label	Positive
2	**Original**	No film could possibly be more **contemptuous** of the single female population.	**Baseline**	Positive
	Simplified	No film could possibly be more **demeaning** of the single female population.	**LRLS-Aux**	Negative
			Gold label	Negative
3	**Original**	Rarely has **leukemia** looked so **shimmering** and **benign**.	**Baseline**	Positive
	Simplified	Rarely have **cancer** look so **shining** and **gentle**.	**LRLS-Aux**	Negative
			Gold label	Negative

Figure 1: Examples that show how auxiliary inputs from simplified sentences help the PrLMs to mlrake the right prediction. In the result column. **Baseline** demonstrates the original prediction made by BERT, and **LRLS-Aux** shows the prediction generated with the auxiliary inputs from simplified sentences.

we select several examples from MR and SST-2 to further illustrate how PrLMs can benefit from the auxiliary input of simplified sentences.

4.2 Impact of Lexical Simplification Process

Since our LRLS method is composed of two steps: transformation of verbs and nouns into their lemmas, and replacement of rare words. To investigate the impact of different LS methods, we firstly apply the two steps separately and compare with our LRLS method. We also include BERT-LS (Qiang et al., 2020), which leverages masking language model of BERT to generate synonym candidates of rare words, for further comparison.

As shown in Table 2, the lemma transformation and rare words replacement are both effective, but we can further improve the performance by combining these two methods together. The performance of our method also exceeds that of BERT-LS. Moreover, our method is more than a hundred faster than BERT-LS, since our method is entirely rule-based, while BERT-LS uses a large pre-trained neural network to detect and replace the complex words recursively.

Method	MR	SST-2
BERT$_{BASE}$	86.4	92.4
Lemma	87.6	93.1
RR	87.7	92.9
BERT LS	87.9	93.1
LRLS	**88.1**	**93.5**

Table 2: Performances (%) using different LS methods. **Lemma** represents the transformation of verbs and nouns into their lemma, **RR** represents the replacement of rare words.

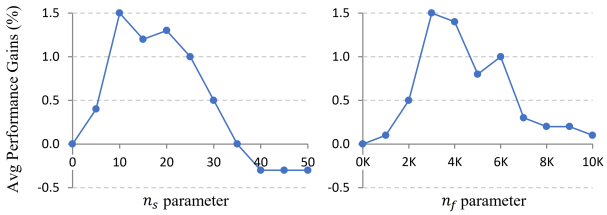

Figure 2: Average performance gain over MR and SST-2. n_s is the number of synonym candidates and n_f is the threshold of the frequency under which the word will be replaced.

4.3 Words Replacement Hyperparameters

The process of the rare word replacement is controlled by two hyperparameters: n_f and n_s. n_f is the frequency threshold under which the word will be labelled as rare word and replaced. The larger the n_f, the more words will be replaced. n_s is the number of synonym candidates. The larger the n_s, the larger possibility that the rare words will be replaced by more common but less similar candidates. In order to investigate the effect of these two hyper-parameters, we change these two hyperparameters separately and conduct experiments on MR and SST-2 to see the impact on the performance.

As shown in Figure 2, the best performance gain is obtained with middle-sized n_f and n_s, which is consistent with our expectation. Because if n_f and n_s are too small the simplified sentence will be almost the same as the original version, on the contrary if n_f and n_s is too large, it may change the underlying meaning of the sentence.

4.4 Alternative Frameworks

We use simplified sentences as auxiliary inputs to improve the robustness of PrLMs. However, there are other frameworks to incorporate lexical

simplification with PrLMs.

One alternative framework is to feed PrLM only the simplified sentences in both training and inference phases. In this case, prediction is made solely based on simplified versions.

Another framework is to leverage LS as a data augmentation technique. To illustrate, let $D = \{x_i, y_i\}_{i=1...N}$ denote the training dataset. For a given sample $\{x_i, y_i\}$ in the training dataset, we generate an augmented sample by simplifying the sentence x_i to x_i' and preserving the label y_i. In this way, we generate an augmented dataset $D' = \{x_i', y_i\}_{i=1...N}$. PrLMs can thus learn from both the training set D and the augmented set D'.

Experiments are conducted to compared our framework with the two alternative frameworks mentioned above on BERT-base.

Method	MR	SST-2
BERT$_{BASE}$	86.4	92.4
LRLS-Only	86.5	92.1
LRLS-Aug	87.9	92.6
LRLS-Aux	**88.1**	**93.5**

Table 3: Performances (%) using different frameworks to leverage simplified sentences. **LRLS-only** represents predictions made solely based on simplified sentences, **LRLS-Aug** represents the use of simplified sentences for training data augmentation, **LRLS-Aux** represents using simplified sentences as auxiliary inputs.

As show in Table 3, the framework using simplified sentences as the only input (**LRLS-Only**) would slightly hurt the performance of PrLM. This is because a part of semantic meanings carried by original sentences may be lost during the simplification process. Experiments also show that leveraging lexical simplification for data augmentation (**LRLS-Aug**) is also beneficial for the overall performance. However, this framework would double the training time and the performance is still worse than our framework (**LRLS-Aux**) .

4.5 Alternative Paraphrasing Methods

While we leverage LRLS method to paraphrase the original sentence and generate auxiliary inputs for PrLMs, we wonder if other commonly used paraphrasing techniques are effective.

These paraphrasing methods include (1) random replacement of several words by their synonyms (Wu et al., 2019; Wei and Zou, 2019), (2) translating an existing example x in language A into another language B, and then translating it back into A to obtain a paraphrased example x' (back-translation) (Xie et al., 2019; Edunov et al., 2018) , and (3) randomly delete several words in the sentence (cutoff) (Shen et al., 2020).

The upper mentioned paraphrasing methods are applied on original sentences respectively to generate auxiliary inputs, and then incorporated into PrLMs. Performance on MR and SST-2 from different paraphrasing methods are compared.

As show in Table 4, cutoff would slightly harm the overall performance. This is because it simply randomly deletes several words in the original sentence to generate a paraphrased version, which tends to twist the original semantic meaning and adds noise for predictions. Although back-translation and random replacement can slightly boost the performance of PrLMs, our LRLS method remains the most competitive.

Method	MR	SST-2
BERT$_{BASE}$	86.4	92.4
+back-translation	87.0	92.8
+cutoff	86.3	91.6
+random replacement	87.3	92.5
+LRLS	**88.1**	**93.5**

Table 4: Performances (%) using different paraphrasing techniques to generate auxiliary inputs.

5 Conclusion

This paper proposes a novel approach that leverages lexical simplification and to reduce lexical diversity and an enhance robustness of PrLMs, resulting in improved performances in downstream tasks. Experiments on various text classification tasks demonstrate that our approach consistently improves strong baselines.

Within the framework, we incorporate a specially designed lexical simplification process based on lemmatization and rare word replacement (LRLS) for better performance. Our comprehensive analysis also show that compared with other paraphrasing techniques used in previous works, LRLS is more effective. Furthermore, an effective framework (LRLS-Aux) leveraging LRLS as auxiliary information is designed. Unlike data augmentation which only leverages paraphrased information in training phase, LRLS-Aux incorporates the information in both training and inference phase and achieves better performance gains. Such framework may shed the light for more future studies.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

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*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale.

Sian Gooding and Ekaterina Kochmar. 2019. Recursive context-aware lexical simplification. 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 4855–4865, Hong Kong, China. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, 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. *arXiv preprint arXiv:1907.11932*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015. Automated rule selection for aspect extraction in opinion mining. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1291–1297. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Maximilian Mozes, Pontus Stenetorp, Bennett Kleinberg, and Lewis D. Griffin. 2020. Frequency-guided word substitutions for detecting textual adversarial examples.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Jipeng Qiang, Yun Li, Zhu Yi, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. *AAAI*.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6383–6389, Hong Kong, China. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *International Conference on Computational Science*, pages 84–95. Springer.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pages 649–657. Curran Associates, Inc.