BLCU-NLP at SemEval-2020 Task 5: Data Augmentation for Efficient Counterfactual Detecting

Chang Liu
Beijing Language and
Culture University
liuchang2014@gmail.com

Dong Yu
Beijing Language and
Culture University
yudong_blcu@126.com

Abstract

Counterfactuals describe events counter to facts and hence naturally involve common sense, knowledge, and reasoning. SemEval 2020 task 5 is focusing on this field. We participate in the subtask 1 and we use BERT as our system. Our Innovations are feature extraction and data augmentation. We extract and summarize features of counterfactual statements, augment counterfactual examples in training set with the help of these features, and two general methods of data augmentation is experimented in our work. We demonstrate the effectiveness of our approaches, which achieves 0.95 of subtask 1 in F1 while using only a subset of giving training set to fine-tune the BERT model, and our official submission achieves F1 0.802, which ranks us 16th in the competition.

1 Introduction

Counterfactual statements describe events that did not actually happen or cannot happen, as well as the possible consequence if the events have had happened. Detecting counterfactual statements involves common sense, knowledge, and reasoning, it is also the basis for all down-stream counterfactual related causal inference analysis in natural language.(Son et al., 2017)

The problem of counterfactual have been studied in many domains, like the perspective of literally logical relations between the antecedent and consequent of counterfactual forms and the outcomes(Goodman, 1947), and conducting counterfactual thought experiments for hypothetical tests on historical events, policies, or other aspects of a society and assess them by political scientists(Tetlock and Belkin, 1996). (Son et al., 2017) use a combination of a rule-based approach and a supervised classifier to capture counterfactual statements from Twitter, which is close to our problem.

SemEval 2020 task 5 focus on the problem of detecting counterfactual statements, as (Yang et al., 2020) described in their work. Subtask 1 of task 5 is a two-classification problem, in which we need to find out whether a statement is counterfactual or not. We choose to apply feature extraction and data augmentation to tackle this problem, for the biggest challenge exists when doing this task is unbalanced data: although the training set has 13,000 examples in total, seems to be enough to fine-tune a BERT model, the number of both classes is highly unbalanced: there are only 1454 counterfactual examples, and 11546 non-counterfactual examples seriously dilute the proportion of counterfactual examples in the training set.

By observing the training set, we conclude a set of feature words and rules to describe the occurrence of these words in counterfactual examples. In terms of data augmentation, we use two different methods, perform experiments on both augmented datasets to determine which method is more effective. We limit the number of non-counterfactual examples, which keeps the same as the number of counterfactual examples, in actual used training set to balance data.

Using our best model setting, we get F1 0.95 on our test set, and F1 0.802 on official test set, while we only use 4334 training examples to fine-tune our model.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 633–639
Barcelona, Spain (Online), December 12, 2020.

The rest of this paper is organized as follows. Section 2 contains a description of our system. Experiments and analysis of the results are introduced in Section 3. We describe the conclusions in Section 4.

2 System description

Our system consists of two components: data augmentation, which includes 2 different augmentation methods, and BERT model.

Since the data provided by the task organizers are highly unbalanced in numbers of examples in two classes, we believe that the key to solve the task is to expand the number of counterfactual examples in some way. We apply 2 data augmentation methods to solve this problem: Back Translation and a simple EDA (Easy Data Augmentation) system(Wei and Zou, 2019) combine with WordNet and feature extraction.

2.1 Data augmentation

This section briefly introduces the datasets of subtask 1, then we introduce our two data augmentation approaches.

Data Overview The training data provided by organizers has 13000 examples in total, each example consists of 3 parts: sentence ID, gold label and sentence. Labels are either 0, denotes a non-counterfactual example, or 1, denotes a counterfactual example. While sentence length of the examples is between 6 to 3273 words, the mean and median length of the examples are 193 words and 177 words respectively. When doing data augmentation, we use Baidu Machine Translate API as our Back Translation method, and WordNet to work with our EDA system.

During our experiments, we test our methods on a freeze test set, has 500 examples in total, sampled from the training set from task organizer, which contains 250 counterfactual and non-counterfactual examples each, to make sure we can evaluate our methods in a consistent standard. As a result, number of counterfactual examples we can use to train our model further reduced to 1204.

Label balancing Before we set off to build our system, we did some preliminary experiments, which proof that the huge imbalance between numbers of two classes will lead to poor performance of model. Therefore, when we build our data augmentation system, label balancing is taking into concern from the beginning. Basically, we use down sampling and up sampling to control numbers of both classes.

As described in (Provost, 2000), down sampling means for a relatively small set of 1 label examples, we randomly sample the same number of 0 label examples from the whole 0 label set to merge with 1 label examples as final augmented training data; and up sampling means for a relatively big set of 0 label examples, we augment 1 label examples several times or simply repeat 1 label examples to increase the size of 1 label example set to match the size of 0 label example set.

Both up and down sampling has its own merit: up sampling uses all available 0 label examples, and down sampling avoid introduce redundant information, which may be useless or even harmful, to the system. In experiment section, we performed experiments on both approaches to find out which leads to better performance.

Back Translation In this section, we describe our data augmentation approach using back translation. Back Translation is an effective method to improve neural machine translation with monolingual data, which is to augment the parallel training corpus with back-translations of target language sentences.(Edunov et al., 2018)

Baidu Machine Translate API is an open online machine translate platform developed by Baidu(`api.fanyi.baidu.com`). Based on Baidu's high-level NLP technology, it is able to provide good and reliable translation service.

Based on Baidu's official SDK, we build a back translation pipeline, as shows in Figure 1.

By observing the training data, we consider that the main problem needs to be solved is that the number of 1 label examples is too small. Therefore, the main target of our data augmentation is 1 label examples. In a typical run of back translating, we first take all 1 label examples we have to be translated into a certain language, then translate the output of previous step back to English. Baidu Machine Translate offers

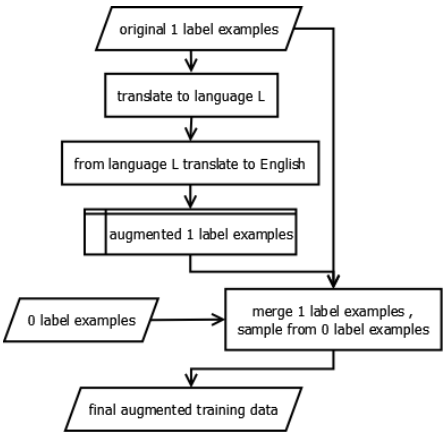

Figure 1: Back Translation

many languages we can use. We test some of them, choose languages capable of keep the sentence's semantic information while generate paraphrases vary from the original sentence. We use Chinese in down sampling scenario. French, Russian, German, Portuguese, Italian, Spanish and Greek are used in up sampling scenario.

EDA Combined with WordNet and Feature Extraction EDA, or Easy Data Augmentation techniques, consists of four simple but powerful operations: synonym replacement, random insertion, random swap, and random deletion. EDA is presented by (Wei and Zou, 2019), which claims to have strong results for small datasets. In our system, we apply only synonym replacement, which shows good performance in (Wang and Yang, 2015)'s work, and random deletion combined with WordNet and a feature words set.

WordNet(Miller, 1995) is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. We use WordNet to do synonym replacement.

Through observing the data, we obtain a set of feature words which appear frequently in counterfactual examples with relatively fixed structures, but seldom in non-counterfactual examples. Part of the set shows in table 1.

would have been	if
should ... have done	... should be ...
could have done	when ... were ...
were it not ...	that ... would have been
if ... had been ...	may/might have done
had (sb) done ...	... would have done ...
supposing ... were ...	... could ...
unless	might

Table 1: Feature Words Set

Based on the feature words set and observation of data, we propose a hypothesis as follows: Whether the feature words appear and are organized in specific forms carries key information to distinguish counterfactual statements from non-counterfactual statements. Therefore, when doing data augmentation with EDA, we need to pay attention to keep these information from being removed or altered during the augmenting process, while generate paraphrases differ to original sentences. And as such, we build our EDA system as shows in figure 2.

The process runs in lexical level. For every word in an example statement, words in feature words set will not be altered. We replace other words with their synonym from WordNet, and drop some of them in a certain percentage.

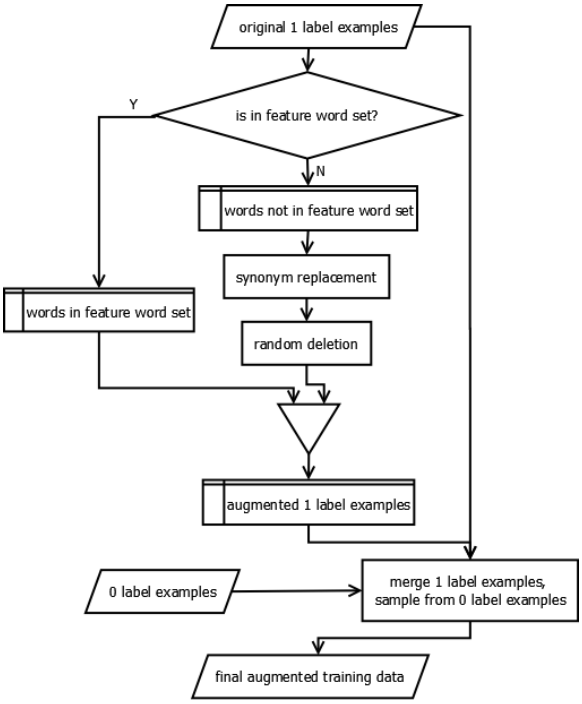

Figure 2: EDA System

As mentioned in Data Overview, the training data provided by task organizer contains 1454 counterfactual examples, and we can only use 1204 of them to train our model. After the EDA process, we doubled the number of available counterfactual examples to 2408. Then we merged same number of non-counterfactual examples with them to compose the final training set, which has 4816 examples in total. Meanwhile, when fine-tuning the model, 10% of training data is used as dev set, so only 4334 examples are actually used in model fine-tune.

2.2 Model

We use BERT as our model. BERT, which stands for Bidirectional Encoder Representations from Transformers. The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks without substantial task-specific architecture modifications(Devlin et al., 2019). As revealed by (Jawahar et al., 2019), BERT is able to capture languages' structure properties from phrase-level information to semantic features, which is critical for our task: we found that even humans try to identify counterfactual statements from non-counterfactual statements requires understanding these statements in the first place. We use BERT large uncased as our pre-trained model.

3 Experiments

In this section, we present experiments we conduct on our system and make detailed analysis. We compare the performance among different data augmentation and label balancing strategies. The model is evaluated on the freeze test set we keep untouched from the beginning.

3.1 Experiment Settings

We perform experiments on our BERT model with max sequence length at 64 and batch size at 4, models are fine-tuned on 2 GTX 1080 Ti GPUs. We fine-tune our model on all available training data provided by task organizers (except those used as freeze test set) as a general baseline (marked as No-aug in result tables), and on each augmented dataset generated by different approaches. Among EDA experiments, we perform multiple runs on different settings to find which operation contributes more to overall performance boost.

3.2 Results

We use F1 score, accuracy and recall as evaluation criteria, No-aug as general baseline, and task organizer's baseline model's evaluation result as submission baseline. Table 2 shows our performances on back translation augmented datasets, compares down sampling and up sampling in detail. We get 0.90 of F1 using back translation approach on our freeze test set.

Back Translation Experiment Result			
	Precision	Recall	F1
No-aug	0.91	0.90	0.90
Down sampling	0.86	0.86	0.86
Up sampling	0.90	0.90	0.90

Table 2: Back Translation Experiment Results

Table 3 shows our performance on EDA augmented datasets. WN represents doing synonym replacement using WordNet, FW represents Feature Words set, drop represents random deletion process. We get 0.95 of F1 using EDA approach on our freeze test set.

EDA Experiment Results			
	Precision	Recall	F1
No-aug	0.91	0.90	0.90
WN	0.94	0.94	0.94
WN + FW	0.94	0.94	0.94
WN + FW + drop	**0.95**	**0.95**	**0.95**
WN + drop - FW	0.58	0.50	0.34

Table 3: EDA Experiment Results

Using our best system, we evaluate the official test set. As table 4 shows, the F1 score of our official submission is 0.802 in subtask 1, which ranks us 16th in all participants. The baseline of the test set is 0.217, which is lower than our result.

Submission Results	
	Subtask 1
Baseline	0.217
Our System	**0.802**

Table 4: Submission Results

3.3 Analysis

Back Translation shows poor performance in our experiments, No-aug outperforms down sampling by 0.04 in F1, and achieves same F1 score with up sampling. The way we balance the 2 classes does affect the result evidently. As our result shows, up sampling outperforms down sampling by 0.04 in F1. We guess the reason might be that though counterfactual examples' critical information may lose or misinterpret during several times of back translation with different languages, up sampling keeps as many as possible non-counterfactual examples, which is useful for our model to predict whether a given example is counterfactual or not.

However, after a deep observing inside the back translation augmented contents, we discover that the variety of generated paraphrases is worse than we expected. As a result, back translation fails to introduce much new information to our model, while adding more noise to the training data, not to mention down sampling drops a lot of non-counterfactual examples.

EDA combined with WordNet and Feature Extraction shows better performance than Back Translation in our experiments, which increase F1 by 0.05 and 0.05 than Back Translation and No-aug respectively. The augmentation setting that achieved best result is to use WordNet to do synonym replacement for all words not in feature words set, while randomly drop 30% of them. We suppose that this setting works because it highlights critical information which differs counterfactual statements from non-counterfactuals by increasing density of these features and reduce other not-so-important parts.

To prove our supposing, we perform experiments on 3 other different settings: use WordNet to do synonym replacement for all words (WN), use WordNet and Feature Words set without random dropping (WN + FW), and use WordNet and random dropping while remove words in Feature Words set (WN + dropping - FW). As shows in table 3, all 3 settings yield worse results than best setting. Especially the third setting (WN + dropping - FW) get a result of 0.34 in F1, which is lower than any other experiments we performed. It proves the hypothesis we put forward in the beginning that whether the feature words appear and are organized in specific forms carries key information to distinguish counterfactual statements from non-counterfactual statements.

4 Conclusion

We use BERT model and data augmentation approach to participate in SemEval 2020 task 5 subtask 1. The goal of this task is classification between counterfactual and non-counterfactual statements. We demonstrate that using EDA combined with WordNet and Feature Extraction can output relatively good result. Our official submission achieves F1 0.802, which ranks us 16th in the competition.

As for future work, we think data extension from external data sources based on feature words set and its pattern might be useful, since it might enlarge feature information of counterfactual statements, which is critical for increase our model's robustness.

Acknowledgements

This work is funded by the Humanity and Social Science Youth foundation of Ministry of Education (19YJCZH230) and the Fundamental Research Funds for the Central Universities in BLCU (No.17PT05).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Nelson Goodman. 1947. The problem of counterfactual conditionals. *The Journal of Philosophy*, 44(5):113–128.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language?

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Foster Provost. 2000. Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI'2000 workshop on imbalanced data sets*, volume 68, pages 1–3. AAAI Press.

Youngseo Son, Anneke Buffone, Joe Raso, Allegra Larche, Anthony Janocko, Kevin Zembroski, H Andrew Schwartz, and Lyle Ungar. 2017. Recognizing counterfactual thinking in social media texts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 654–658.

Philip E Tetlock and Aaron Belkin. 1996. *Counterfactual thought experiments in world politics: Logical, methodological, and psychological perspectives*. Princeton University Press.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP/IJCNLP*.

Xiaoyu Yang, Stephen Obadinma, Huasha Zhao, Qiong Zhang, Stan Matwin, and Xiaodan Zhu. 2020. SemEval-2020 task 5: Counterfactual recognition. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain.