HIT-SCIR at SemEval-2020 Task 5: Training Pre-trained Language Model with Pseudo-labeling Data for Counterfactuals Detection

Xiao Ding, Dingkui Hao, Yuewei Zhang, Kuo Liao, Zhongyang Li, Bing Qin, Ting Liu*
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{xding, dkhao, ywzhang, kliao, zyli, qinb, tliu}@ir.hit.edu.cn

Abstract

We describe our system for Task 5 of SemEval 2020: Modelling Causal Reasoning in Language: Detecting Counterfactuals. Despite deep learning has achieved significant success in many fields, it still hardly drives today's AI to strong AI, as it lacks of causation, which is a fundamental concept in human thinking and reasoning. In this task, we dedicate to detecting causation, especially counterfactuals from texts. We explore multiple pre-trained models to learn basic features and then fine-tune models with counterfactual data and pseudo-labeling data. Our team HIT-SCIR wins the first place (1st) in Sub-task 1 — Detecting Counterfactual Statements and is ranked 4th in Sub-task 2 — Detecting Antecedent and Consequence. In this paper we provide a detailed description of the approach, as well as the results obtained in this task [1].

1 Introduction

Deep learning technologies have been truly remarkable and have caught many attentions of researchers. However, deep learning systems' lack of understanding of causal relations is perhaps the biggest roadblock to giving them human-level intelligence (Pearl, 2019). Causation is a fundamental concept in human thinking and reasoning, which indicates a special semantic relation between the cause with the effect (Stukker et al., 2008). Pearl and Mackenzie (2018) propose that there are three levels of causation, and the top level is the counterfactual analysis. Counterfactual statements describe events that did not actually happen or cannot happen, as well as the possible consequence if the events have had happened (Yang et al., 2020). SemEval 2020 Task 5 consists of two subtasks which aim to detect counterfactual descriptions in sentences, this paper presents solutions to both of two subtasks.

Subtask1 — Detecting counterfactual statements refers to determining whether a given statement is counterfactual or not in several domains. For example, given the sentence: "*Her post-traumatic stress could have been avoided if a combination of paroxetine and exposure therapy had been prescribed two months earlier.*", the system is required to determine whether it is a counterfactual statement or not. This categorization task is evaluated by three evaluation indexes: Precision, Recall and F1. The challenges of this subtask include: a) positive and negative samples are unevenly distributed in the training data and test data. b) the patterns of counterfactual statements can be hardly defined by rules or learned by traditional word embedding based models.

Subtask2 — Detecting antecedent and consequence aims to locate antecedent and consequent in counterfactuals. A counterfactual statement can be converted to a contrapositive with a true antecedent and consequent (Goodman and Nelson, 1947). Consider the "*post-traumatic stress*" example discussed above; it can be transposed into "*because her post-traumatic stress was not avoided, (we know) a combination of paroxetine and exposure therapy was not prescribed*". Such knowledge can be not only used for analyzing the specific statement but also be accumulated across corpora to develop domain causal knowledge (e.g., a combination of paroxetine and exposure may help cure post-traumatic stress). The results are evaluated by four indexes: Precision, Recall, F1 and Exact Match. Unlike normal sequence labeling tasks, in some

*Corresponding author

[1] Our codes are available on https://github.com/haodingkui/semeval2020-task5-subtask1 (subtask 1) and https://github.com/AutismNLP/Semeval2020-Task5-Subtask2 (subtask 2).

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 354–360
Barcelona, Spain (Online), December 12, 2020.

cases there is only an antecedent part while without a consequent part in a counterfactual statement. For instance, "*Thanks for the article on this new term that fits me so well, wish all your articles were worthy of praise*.", there is only the antecedent part "*wish all your articles were worthy of praise*" in this sentence.

Recently, pre-trained language models have achieved great success in various NLP tasks (Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019). Hence, we first explore multiple pre-trained language models (i.e., BERT, XLNet and RoBERTa) stacking methods with fine-tuned in counterfactual data. These models are basically trained in a supervised fashion with labeled data. However, There is not very sufficient labeled data in Task 5. Hence, we use a simple way of training neural networks in a semi-supervised fashion. Basically, each pre-trained language model is trained in a supervised fashion with labeled data. For unlabeled data, *Pseudo-Labels*, created by just picking up the class which has more votes from pre-trained language models, are used as if they were true labels.

2 Related Work

Subtask 1 is a text classification task. Traditional machine learning systems, such as Naive Bayes (Domingos and Pazzani, 1997) and Support Vector Machines (Cortes and Vapnik, 1995) perform well in this task. In recent years, text classification has made breakthroughs in deep learning. A few of studies (Kim, 2014; Liu et al., 2016; Lai et al., 2015) made a series of improvements to the structure of deep neural networks. BERT (Devlin et al., 2018), a dynamic word vector based on the language model, has achieved very competitive performance in multiple domains of text classification. In addition, pre-trained language models such as XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019) have also achieved high performances in various NLP tasks.

Subtask 2 is a sequence labeling task. Traditional work uses statistical approaches such as Hidden Markov Models (HMM) and Conditional Random Fields (CRF) (Ratinov and Roth, 2009; Passos et al., 2014; Luo et al., 2015). Deep learning sequence labeling models utilize word-level Long Short-Term Memory (LSTM) structures to represent global sequence information and a CRF layer to capture dependencies between neighboring labels (Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016; Peters et al., 2017). In similar manner, fine-tuning the pre-trained model to complete the task of sequence labeling such as BERT has achieved excellent performances.

3 Methodology

For two subtasks (i.e., text classification task and sequence labeling task), we start by fine-tuning pre-trained language models, including BERT, RoBERTa and XLNet. Then, we enlarge the training set with pseudo-labeled sentences, which are predicted on the test set by voting ensemble of several single pre-trained models. Finally, we use the ensemble model to classify counterfactual statements for subtask 1 and utilize a CRF model to perform sequence labeling for extracting antecedent and consequence in subtask 2. We will introduce each module in details in the following sections.

3.1 Fine-tuning Pre-Trained Models

BERT (Devlin et al., 2018) is a revolutionary self-supervised pre-training technique that learns to predict intentionally hidden (masked) sections of text. Crucially, the representations learned by BERT have been shown to generalize well to downstream tasks, and when BERT was first released in 2018 it achieved state-of-the-art results on many NLP benchmark datasets.

XLNet (Yang et al., 2019) is a new unsupervised language representation learning method based on a novel generalized permutation language modeling objective. Additionally, XLNet employs Transformer-XL as the backbone model, exhibiting excellent performance for language tasks involving long context. Overall, XLNet also achieves state-of-the-art results on various downstream language tasks.

RoBERTa (Liu et al., 2019) builds on BERT's language masking strategy and modifies key hyper-parameters in BERT, including removing BERT's next-sentence pre-training objective, and training with much larger mini-batches and learning rates. RoBERTa is trained on an order of magnitude more data

than BERT, for a longer amount of time. This allows RoBERTa representations to generalize even better to downstream tasks compared to BERT.

For subtask 1, we use the default models for sentence classification implemented in the huggingface library. For subtask 2, which can be formulated as a sequence labeling task, sentences are fed into RoBERTa and then into a classification layer, producing a sequence of tag scores for each sentence. The sub-token entries are removed from the sentences and the remaining tokens are passed to the CRF layer. The maximum context tokens are selected and concatenated to form the final predicted tags. Generally, we use Viterbi algorithm to simplify the calculation of CRF and adopt the standard CRF loss function

3.2 Pseudo-Labeling

Pseudo-labeling is a simple but effective semi-supervised learning method that can improve the performance of machine learning models by utilizing unlabeled data (Lee, 2013). In subtask 1, firstly we fine-tune different pre-trained language models on the training set as pseudo-label predictors, then we use these classifiers to provide pseudo-labels on sentences in the test set. For each pseudo-labeled sentence in the test set, we add it into the training set if and only if all classifiers agree on the labeling of this sentence. In subtask 2, we use k roberta-large models trained by k-fold cross-validation as pseudo label predictors to make predictions on the test set, and we add pseudo labels into the training set when at least n ($n \leq k$) models agree with them.

3.3 Ensemble

Ensemble has shown its power on effectively improving the robustness and accuracy of each individual prediction models (Opitz and Maclin, 1999; Rokach, 2010). By ensembling predictions from models with different hyper-parameters or architectures, we can get better results than each individual model. In our system, XLNet and RoBERTa have different training objectives, BERT and RoBERTa have different hyper-parameters. Thus, we propose to combine their probability predictions via weighted average ensemble, and then optimize the classification threshold for optimal F1 score on the combined prediction.

3.4 Sequence Labeling with CRF

In subtask 2, we use CRF model to perform sequence labeling task and extract antecedents and consequences in the counterfactual statements. With an additional CRF output layer, our model can calculate not only the emission score but also the transition score. The transition score from CRF is considered to guarantee the integrity and correctness of the output label. We label antecedents and consequences in the counterfactual statements with BIO tags.

4 Experimental Settings

4.1 Data

SemEval 2020 Task 5 released the training and test dataset, and the detailed statistics are shown in Table 1.

	Training	Test
#Positive instances	1,454	-
#Negative instances	11,546	-
#Total	13,000	7,000

(a) Data distribution of subtask 1.

	Training	Test
#Positive instances	3,031	-
#Negative instances	520	-
#Total	3,551	1,950

(b) Data distribution of subtask 2.

Table 1: Data distribution of SemEval2020-Task5

4.2 Evaluation Metrics

- Subtask1: Precision, Recall, and F1. The evaluation will verify whether the predicted binary "label" is the same as the desired "label" which is annotated by human workers, and then calculate its precision, recall, and F1 scores.

- Subtask2: Exact Match, Precision, Recall, and F1. Exact Match will represent what percentage of both predicted antecedents and consequences are exactly matched with the desired outcome that is annotated by human workers. F1 score is a token level metric and will be calculated according to the submitted antecedent_startid, antecedent_endid, consequent_startid, consequent_endid.

4.3 Experimental Details

Experimental Environment. Our experimental codes relied heavily on BERT, RoBERTa and XLNet models, which are implemented in the Huggingface Transformers package(2.5.1) with Pytorch(1.2.0). Computations were performed on Tesla P100-PCIE-16GB GPUs.

Hyper-Parameters of Pseudo-Labeling. For pseudo-labeling in subtask 1, we train a bert-large-cased model, an xlnet-large-cased model and a roberta-large model on the whole training set as pseudo label predictors, respectively. We use an AdamW optimizer to tune the parameters with learning rate = 2e-5, epsilon = 1e-8, max_seq_length = 128, batch_size = 16, seed = 42 and we train each model for 4 epochs. Then we use these pseudo-label predictors to predict on the test set and get 6,867 pseudo-labels which all predictors agree with them. For subtask 2, we use five different roberta-large models trained in 5-fold cross-validation as pseudo-label predictors to make predictions on the test set, and we add the pseudo-label to the training set if and only if at least N models agree with it. We also use an AdamW optimizer to tune the parameters with learning rate = 1e-5, epsilon = 1e-8, max_seq_length = 250, batch_size = 8, seed = 19,970,514 and we train each model for 5 epochs. Then we can obtain 822 pseudo-labels when $N = 4$ and 524 pseudo-labels when $N = 5$.

Hyper-Parameters of Our System. We firstly add all the pseudo-labeled sentences to the training set, then we train a bert-large-cased model, an xlnet-large-cased model and a roberta-large model on the training set, respectively. After that, we use weighted average ensemble to combine the predicted probabilities of BERT (weight=1), XLNet (weight=1) and RoBERTa (weight=2). Due to the imbalanced distribution of the training set for subtask 1, models trained on it tend to predict more negative labels with a threshold of 0.5. Hence, we used a lower threshold (0.3437), which is the best threshold for the average F1 of 10-fold cross-validation on the training set, for the final submission. The hyper-parameters of our system used in both subtasks are shown in Table 2.

	BERT(st1)	XLNet(st1)	RoBERTa(st1)	RoBERTa(st2)
batch size	12	12	8	8
learning rate	2e-5	2e-5	2e-5	5e-6
max epochs	3	3	3	3
max sequence length	128	128	128	250
random seed	42	42	42	19970514

Table 2: Hyper-Parameters of our system for subtask 1 and subtask 2

5 Results

As there is no validation set in Task 5, we use cross-validation of the training set as the validation set. We tune our system on the validation set and report the experimental results in Table 3 and Table 4. Then we choose the system that achieved the best performance on the validation set to be evaluated on the test set, and report the experimental results in Table 5 and Table 6.

Table 3 shows results of 10-fold cross-validation on the training set of subtask 1. We can find that models trained with extra pseudo-labels get higher F1 score than models trained only with the original training data. The weighted average ensemble model of single models trained with extra pseudo labels outperforms all single models and achieves the highest F1 score.

Table 4 shows results of 5-fold cross-validation on the training set of subtask 2. We can find that the model trained with extra pseudo-labels from single model gets higher Recall, Precision and Exact Match than the model trained with the original training data.

Table 5 shows results on the test set of subtask 1. We can find that the results on the test data are similar to the results on the validation set. We can also see that models trained with pseudo-labels gain better

#	Model	Comments	F1 (%)	Recall (%)	Precision (%)
1	BERT	bert-large-cased	87.50	85.10	90.05
2	XLNet	xlnet-large-cased	86.53	84.55	88.61
3	RoBERTa	roberta-large	89.49	88.27	90.76
5	BERT+PL	bert-large-cased	88.39	86.09	90.81
6	XLNet+PL	xlnet-large-cased	88.29	86.04	90.65
7	RoBERTa+PL	roberta-large	89.91	88.50	**91.37**
8	RoBERTa+BERT+XLNet ensemble+PL	5+6+7	**90.72**	**91.13**	90.32

Table 3: Results of 10-fold cross-validation on subtask 1. PL is the abbreviation for Pseudo-Labeling

#	Model	Comments	F1 (%)	Recall (%)	Precision (%)	Exact Match (%)
1	RoBERTa+CRF	roberta-large	**86.30**	87.50	85.00	46.60
2	RoBERTa+CRF+1commonPL	roberta-large	85.20	**88.30**	**86.20**	**49.10**
3	RoBERTa+CRF+4commonPL	roberta-large	82.70	86.70	83.40	45.30
4	RoBERTa+CRF+5commonPL	roberta-large	84.00	87.60	84.80	47.10

Table 4: Results of 5-fold cross-validation on subtask 2. 4commonPL means that we add a pseudo-labeled sample of the test set into the training set when 4 models agree with it.

F1 score than models trained with only the original training data. The ensemble model (#8) trained with pseudo-labels achieves the best result.

Table 6 shows that the models trained with only the original training data obtained better performance than models trained with pseudo-labels. Pseudo-labels on the test set do not provide obvious help. A possible explanation is that the pseudo-labels cannot contain enough high-confidence labeling data to provide any benefit. In addition, in order to ensure the correctness of the label format, we used some rules on the optimal model. For example, if the start of the antecedent is labeled as -1 by the best model, then we find reasonable result from output of other models.

6 Conclusion

Counterfactuals inference is a challenging problem in causation, which is crucial for today's AI system. In this paper, we designed and implemented an ensemble model for detecting counterfactuals. It achieved the top F1 score of 90.90% on the Evaluation-Subtask 1 leaderboard and 4th F1 score of 84.10% on the Evaluation-Subtask 2 leaderboard. Task 5 mainly explores how to detect counterfactuals, which is the basis of counterfactuals inference. We believe that causation, especially counterfactuals will attract more attention from researchers driven by this task.

References

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

#	Model	Comments	F1	Recall	Precision
1	BERT	bert-large-cased	88.10	86.60	89.60
2	XLNet	xlnet-large-cased	88.90	87.70	90.10
3	RoBERTa	roberta-large	89.20	88.60	89.70
4	RoBERTa+BERT+XLNet ensemble	1+2+3	89.70	90.80	88.60
5	BERT+PL	bert-large-cased	89.00	88.50	89.60
6	XLNet+PL	xlnet-large-cased	89.80	89.80	89.70
7	RoBERTa+PL	roberta-large	89.50	88.60	**90.50**
8	RoBERTa+BERT+XLNet ensemble+PL	5+6+7	**90.90**	**91.90**	90.00

Table 5: Results on the test set of subtask 1.

#	Model	Comments	F1	Recall	Precision	Exact Match
1	RoBERTa+CRF	roberta-large	0.818	0.826	0.836	0.470
2	RoBERTa+CRF+PL	roberta-large	0.808	0.812	0.828	0.461
3	RoBERTa+CRF+4commonPL	roberta-large	0.791	0.791	0.816	0.457
4	RoBERTa+CRF+5commonPL	roberta-large	0.785	0.798	0.796	0.430
5	RoBERTa+CRF+Rule	roberta-large	**0.841**	**0.846**	**0.868**	**0.471**

Table 6: Results on the test set of subtask 2.

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*.

Pedro Domingos and Michael J Pazzani. 1997. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2):103–130.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv: Computation and Language*.

Dong-hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

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*.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

David W Opitz and Richard Maclin. 1999. Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*, 11(1):169–198.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *arXiv preprint arXiv:1404.5367*.

Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*. Basic Books.

Judea Pearl. 2019. The seven tools of causal inference, with reflections on machine learning. *Communications of The ACM*, 62(3):54–60.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.

Lior Rokach. 2010. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39.

Ninke Stukker, Ted Sanders, and Arie Verhagen. 2008. Causality in verbs and in discourse connectives: Converging evidence of cross-level parallels in dutch linguistic categorization. *Journal of Pragmatics*, 40(7):1296–1322.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

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.