Soft and Hard Prompting for Document Classification with Only Label Names

Luyao WANG† Zhewei XU‡ and Mizuho IWAIHARA‡

Graduate School of Information, Production and Systems, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka 808-0135 Japan
E-mail: †luyao@fuji.waseda.jp, ‡xuzhewei@toki.waseda.jp, ‡iwaihara@waseda.jp

Abstract The existing prompt-based tuning methods can tune discrete or continuous prompts with a frozen language model, and achieve performance comparable to fine-tuning on many tasks. However, it is often difficult to achieve good results without any annotated data. Especially for continuous prompt optimization, it is usually necessary to provide a certain amount of labeled data to train a prompt vector. Therefore, we propose a method that combines soft and hard prompts for text classification tasks. First, we construct discrete prompts and a verbalizer to obtain initial prediction results by the masked language model. Then a part of the labelled data with high confidence are used as pseudo-labeled training samples to optimize the continuous prompt vector. Finally, a classification model can be obtained by self training. We find that the ideas of prompting can be applied to weakly supervised learning methods, and it is worth exploring in the future research.

Keyword Document classification, continuous prompt, weakly supervised learning, pretrained language model, self-optimizing

1. Introduction

Recent years pretrained language models (PLMs) have been widely applied to various natural language processing tasks due to their powerful language understanding ability. To explore the principles of such effectiveness of PLMs, researchers have conducted extensive studies and suggested that PLMs have obtained rich knowledge during pre-training [2][14]. As a result, more and more attention is being paid to how to better utilize and maximize the potential of language models. A common method to achieve this goal is fine-tuning [3], in which an additional classifier is added to the top of a PLM and the model is further trained under its classification objectives. Fine-tuning has yielded satisfactory results on supervised tasks. Nevertheless, applying fine-tuning in few-shot learning [1] and zero-shot learning [19] scenarios remains challenging because additional classifiers require sufficient training samples for fine-tuning.

With the advent of GPT-3 [1], researchers have also increasingly focused on prompt-based learning approaches. The basic idea is to link downstream tasks to pre-trained process. A series of research using prompts [7][15] demonstrates that such discrete or continuous prompts show better performance on zero-shot and few-shot tasks, even on small-scale models. After the Pattern-Verbalizer-Pair (PVP) component is proposed, a routine flow of using discrete prompts is formed. That is, we first put the input sentence into a natural language template, then let the PLM make the masked language prediction, and map the result to the class using the verbalizer. For example, to classify the topic of a sentence x: “Amazing film!” into the “GOOD” category, we wrap it into a template: “x This film is [MASK]” and the generated words in “[MASK]” token are mapped into categories via the verbalizer. However, it was found that even small differences in the hand-constructed templates could have a significant impact on the results. [17] have attempted to automatically search for discrete templates, but this approach tends to fall into local optima. To overcome this dilemma, a series of continuous prompts methods [6][8][9] have been proposed. The prefix-tuning [6] optimizes consecutive word embeddings as prefixes to indicate text generation tasks, with the parameters of the pre-trained language model frozen. It shows that only a small amount (about 0.1%) of optimization parameters are needed to achieve good results.

When a prompt-based approach applied to a class name-only text classification task, we have no labeled text that can be used to optimize continuous prompts, and the template and verbalizer construction may produce more bias in the results if we use discrete prompts. Therefore, we propose a new method combining hard prompts and soft prompt-tuning (HSPT). As shown in Figure 1, Our approach can be divided into three stages. Firstly, we obtain highly-credible pseudo-label data by using a hard prompt and the masked language model, and then it is used as training samples to optimize the continuous prompts in the second stage. Finally, we can obtain the final classification model after several iterations of optimization. We conduct experiments

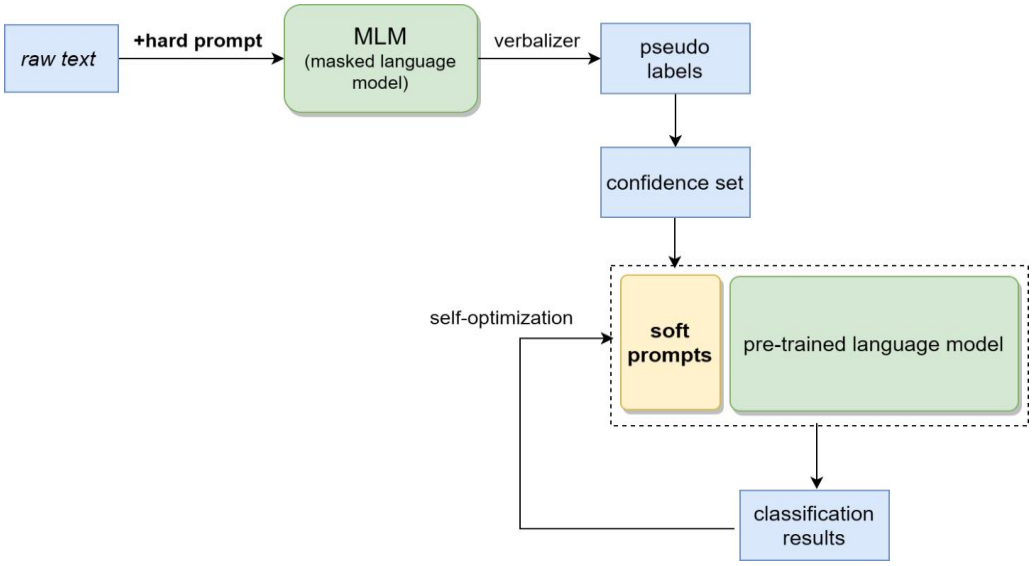

Figure 1. Overview of the prediction process using Pattern-Verbalizer-Pair and discrete prompt

on four datasets without any labeled data. Experimental results demonstrate that our design is feasible and effective, which achieves good results on most of the datasets, so this direction is worth exploring.

2. Related Work

2.1. Prompt-tuning

Prompt-tuning originated with the emergence of the GPT-3 [1] model, with arguments that large-scale language models can maximize their reasoning and understanding capabilities with appropriate templates. GPT-3 pioneers the concept of in-context learning, enabling few-shot or zero-shot learning without modifying the model. Later, Schick et al. [15] demonstrate that this approach is also feasible on small-scale language models like BERT [3]. They attempt to convert all classification tasks into cloze questions consistent with the masked language model (MLM) and designs an important component, Pattern-Verbalizer-Pair (PVP). Based on this framework, current research has focused on how to select or construct appropriate templates and verbalizers. A simple approach is to design templates manually based on the nature of the task and prior knowledge, but it is found that the choice of hard prompts has a very large impact on the pre-trained model. To solve this problem, soft prompt is proposed [6][8][9], introducing new parameters into the model to make prompt generation a task for the machine to learn. The essence of prompt-tuning is to reformat the task so as to cater to the performance of the large-scale model. Prompt-tuning has been applied to a large variety of tasks such as text classification, natural language understanding, relation extraction, and knowledge probing, etc. However, the hard prompt is not precise enough to be used as a basis for weakly supervised classification, and the soft prompt can not be applied directly to this task either, because we do not have any ground-truth labeled data in our setting, there is a lack of material for training soft prompt parameters. As a result, we consider the way of combining these two kinds of prompts to solve the problem, by first generating pseudo-labels through hard prompts, and then extracting the parts of them with high confidence to train soft prompts.

2.2. Class Name-Only Text Classification

Class name-only text classification means that only label surface names or limited word-level descriptions of each category can be used for classification. Recently, a number of researches [11][12][15][16] trained neural text classifiers in an weakly supervised way. They generate a set of documents with pseudo labels to train a supervised model over them. LOTClass [12] utilizes BERT to query replacements for label names to obtain candidate words for categories, and performs self-training on unlabeled data after fine-tuning the PLM. X-Class [14] leverages BERT to represent documents and labels, and generates document-class pairs by clustering to train a classifier. Compared with previous work, our proposed HSPT method obtain supervision data by MLM with discrete prompts, instead of utilizing a set of category vocabulary, or calculating the similarity between classes and texts. We further introduce continuous prompts to obtain classification results by prompt-tuning rather than fine-tuning, which requires less

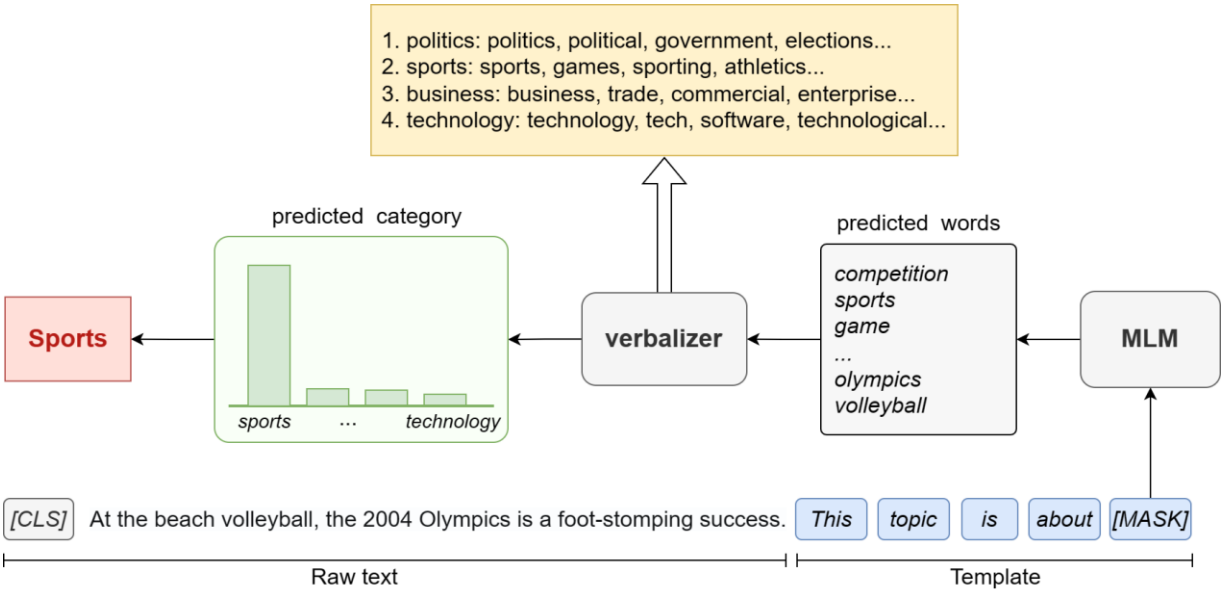

Figure 2. Overview of the prediction process using Pattern-Verbalizer-Pair and discrete prompt

training parameters.

3. Methodology

3.1. Hard Prompt

We denote M as a pre-trained language model. In text classification task, we suppose to classify the input text set $X = \{x_0, x_1, \ldots, x_n\}$ into class set $Y = \{y_0, y_1, \ldots, y_m\}$. Discrete prompting aims to formalize the classification task into a masked language modeling problem in cloze style. For example, assuming that we need to classify the text $x =$ "At the beach volleyball, the 2004 Olympics is a foot-stomping success." into a label from set {politics, sports, business, technology}. We can add a template and change the raw text into:

$$x_t = [\text{CLS}]\ x\ This\ topic\ is\ about\ [\text{MASK}]$$

Taking it as input, MLM can generate the probability of each word v in the vocabulary appearing at the [MASK] position. Then we design a verbalizer, which consists of a set of related words from the vocabulary of each class. We define a label word set V as the words that are used by the verbalizer. As Figure 2 shows, the verbalizer can be considered as a mapping $f: V \to Y$ to map the probabilities of predicted words into the probabilities of categories. Then the probability of label y for sentence x_t can be calculated as

$$P(y|x_t) = P_M([\text{MASK}] = v|x_t) \mid v \in V_y, \quad (1)$$

Finally, we assign the category with the highest probability to text x as a pseudo label.

Verbalizer Construction The verbalizer is an important part of the PVP component, mapping a set of predicted words into the label space. A common idea is to look for category related words, so we utilize the category vocabulary built in LOTClass [12], which is obtained by using MLM to predict the position of category words and rank them by frequency of occurrence. We take the top 10 words of each class in the category vocabulary and add them to the verbalizer. The procedure of predicting masked words based on the context is not a single-choice process, that is, there is no standard correct answer, but abundant words may fit this context. Therefore, we also introduce WordNet [13] as an external structured knowledge base to expand the verbalizer with comprehensive label words. We generate synset for each class, and choose another top 10 words as verbalizer words, ranked by word similarities.

Generating Highly Confident Data After obtaining the classification result for each document, we sort all texts by their maximum class probabilities. In order to optimize the soft prompt, we tend to choose samples with high quality in the first stage, where a large amount is not so necessary. From this perspective, we select the top α% of sorted results and add them to the high confidence set. In the experiments, we set α as 20.

3.2. Soft Prompt-tuning

It is suggested that manually constructed templates and

verbalizer may create uncertainty into the results. However, continuous prompt learning does not suffer from this problem. Therefore, we use high confidence data generated in the first stage to optimize the continuous prompts to improve the stability and generalization ability of the model.

Liu et al. [7] and Lester et al. [5] introduce trainable continuous prompts as a substitution to natural language prompts for natural language understanding (NLU) with the parameters of pretrained language models frozen. Prompt tuning has been proved to be comparable to fine-tuning on 10-billion-parameter models on simple classification tasks [5][7]. As Figure 2 shows, continuous prompts are added as prefixes to the input of each layer. We keep the parameters of the pre-trained model unchanged and only optimize the prefix vector with pseudo-labeled text data. The work [8] demonstrates that this kind of deep prompt-tuning optimizes more smoothly and more effective than just adding prompts to the embedding layer like [9] for natural language understanding (NLU) tasks. Then we utilize the traditional method for classification through [CLS] tokens, with randomly initialized linear heads, rather than the verbalizer. We use soft prompts to solve the text classification task without any labeled data. Therefore, compared with the previous work, our training of soft prompts is not based on ground-truth labeled data, but on pseudo-labeled data, which can exploit the potential of prompt-tuning to a greater extent.

3.3. Self-Optimization

Generally, after optimizing the soft prompt using selected pseudo-labelled data with high confidence, the pretrained language model can generate more accurate classification results. As a result, we propose to self-optimize the prefix vector on the entire unlabeled data. The process of self-optimization is to iteratively use the model's current predictions with high confidence to compute a target distribution which optimize the continuous prompt simultaneously. We define r_{ij} as the logits of the LM for document i belongs to class j, and we obtain a probability distribution $P = [p_{i1}, p_{i2}, \dots, p_{im}]$ over labels using the softmax function:

$$[p_{i1}, p_{i2}, \dots, p_{im}] = Softmax([r_{i1}, r_{i2}, \dots, r_{im}]), \quad (2)$$

Then the confidence score of x_i is obtained by:

$$CS_i = max\ p_{ij}, j = 1,2, \dots, m, \quad (3)$$

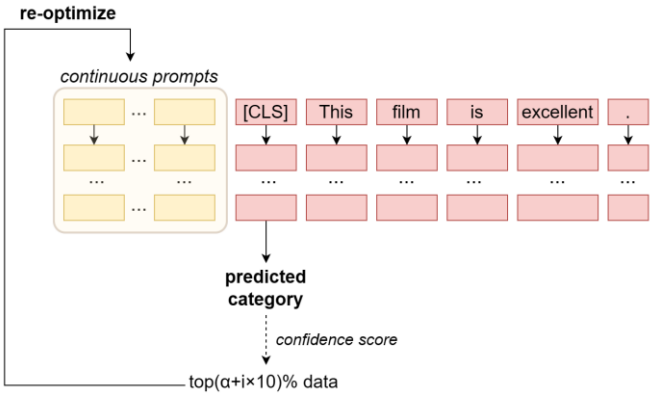

Figure 3. Overview of the soft prompt-tuning and self-optimizing stage

Then, as illustrated in Figure 3, we sort all documents by the confidence score and use $top(\alpha + 10 * i)\%$ of them as high confidence predictions for prompt optimization in the i-th self-optimizing loop.

4. Experiments

4.1. Datasets

We use three benchmark datasets for document classification, including three topic datasets AGNews [20], DBpedia [4] and 20News and a sentiment dataset IMDB [10]. The dataset statistics are shown in Table 1.

4.2. Compared Methods

We compare our method with a wide range of weakly-supervised methods and also state-of-the-art supervised methods. Fully supervised methods use the entire training set for model training. Weakly-supervised methods use the training set as unlabeled data.

WeSTClass [11]: This method first generates pseudo labels for texts that include user-provided candidate words. Also, pseudo-labelled documents are used as training data to train a neural network. Then it performs a self-training process.

LOTClass [12]: It builds a category vocabulary for each category, by utilizing a pre-trained masked language model to find replaceable words. Then the LM is finetuned through word-level class prediction task, and finally generalized classifier is obtained after self-training on unlabeled data.

X-Class [14]: It leverages BERT representations to generate class-oriented document presentations, then generates document-class pairs by clustering methods, and

Name	Type	#Class	#Train set
20News	Topic	5	17,871
AGNews	Topic	4	120,000
DBpedia	Topic	14	560,000
IMDB	Sentiment	2	25,000

Table 1 Dataset statistics

Model	20News	AGNews	IMDB	DBpedia
WeSTClass	71.28/69.90	82.3/82.1	77.4/-	81.42/81.19
LOTClass	73.78/72.53	86.89/86.82	86.5/-	86.66/85.98
X-Class	78.62/77.76	85.74/85.66	82.20/82.18	91.32/91.17
HSPT (proposed)	82.51/81.33	88.66/88.65	89.78/89.73	78.45/78.32
Supervised	93.99/93.99	93.99/93.99	94.55/94.54	98.96/98.96

Table 2 The Micro and Macro-F1 scores of our HSPT method compared with the baselines and supervised method

then fed pairs to a supervised model to train a text classifier.

BERT [3]: We use the pretrained BERT-base-uncased model and fine-tune it with the labeled training data for text classification. It shows an upper bound of weakly supervised methods.

4.3. Experimental Settings

We use the pre-trained BERT-base-uncased model as the base neural language model. For the three datasets AGNews, IMDB and DBpedia, the learning rates are set as 5e-3, 5e-3, 6e-4, and the length of soft prompt is set to 5 for all three datasets. With α set to 20, we repeat the self-optimization process for five times, each containing 5-10 epochs. The training batch size is 12, and we test all results on given training dataset. The model is run on one NVIDIA GeForce GTX 3090 GPU.

4.4. Result Comparison and Analysis

Table 2 presents the performance of our method compared with other weakly supervised text classification baselines and the supervised SOTA method over four different datasets. Our proposed hard and soft prompt-tuning (HSPT) method outperforms previous weakly supervised classification methods by about 3 percent points of the Micro-F1 and Macro-F1 scores on 20News, AGNews and IMDB, but there is still a gap between our result and the supervised ones. However, our method does not perform as well on the DBpedia dataset, which may be related to the imperfect construction of the verbalizer and the imbalance of the data in each category.

The results show that our idea and design are reasonable, and our model can effectively do text classification under the condition of weak supervision, and the performance is comparable to the results of supervised classification method.

5. Conclusion

In this paper, we proposed a method using both hard and soft prompts to solve the text classification problem under the situation that only unlabeled documents and label names are available. Since the optimization of continuous prompts requires certain amount of labeled data, while manual discrete prompts suffer from instability, we choose to combine these two methods to improve the performance. Firstly, we construct templates and verbalizer, and obtain the class distribution of each document by masked language model. Then a set of pseudo label prediction results with high confidence are selected as the subsequent training data. Then the plausible samples are used to optimize the continuous prompt vector added before the input of every layer in the model. After that, the optimized

model is used to make predictions on the whole unlabeled data, and part of the prediction results are used to re-optimize the prefix vector. Finally, we can obtain the final classification model after repeating this process several times.

In the experiments, we find that our HSPT method exhibits competitive performance on four different datasets. The results on three datasets are outperforming the baseline models, while the results on the DBPedia dataset is falling behind of X-Class, which may be related to the nature of the dataset and the construction of the verbalizer.

For future work, we should first continue to improve the hyperparameters of the model and expand the selection of verbalizers by using other knowledge base. In addition, we will construct and combine more templates for each datasets. Moreover, the soft trainable prompt is added at the very beginning of the input as a prefix currently, perhaps investigating the influence of different locations of prompts on the effect of the model can be taken as the next research direction, such as adding the continuous prompt to the end of the input.

References

[1] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., "Language models are few-shot learners. Advances in neural information processing systems", 33, pp.1877-1901, 2020.

[2] Davison, Joe, Joshua Feldman, and Alexander M. Rush. "Commonsense knowledge mining from pretrained models." Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp. 1173-1178, 2019.

[3] Devlin, Jacob., et al. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." Proceedings of the 2019 Conf erence of the North, 2019.

[4] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S. and Bizer, C. "Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia." Semantic web, 6(2), pp.167-195, 2015.

[5] Lester, Brian, Rami Al-Rfou, and Noah Constant. "The power of scale for parameter-efficient prompt tuning." arXiv preprint arXiv:2104.08691, 2021.

[6] Li, Xiang Lisa, and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation." arXiv preprint arXiv:2101.00190, 2021.

[7] Liu, Xiao, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. "GPT understands, too." arXiv preprint arXiv:2103.10385, 2021.

[8] Liu, Xiao, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. "P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks." arXiv preprint arXiv:2110.07602, 2021.

[9] Liu, Xiao, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. "P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks." In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 61-68, 2022.

[10] Maas, Andrew, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning word vectors for sentiment analysis." In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pp. 142-150, 2011.

[11] Meng, Yu, Jiaming Shen, Chao Zhang, and Jiawei Han. "Weakly-supervised neural text classification." In proceedings of the 27th ACM International Conference on information and knowledge management, pp. 983-992, 2018.

[12] Meng, Yu, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. "Text classification using label names only: A language model self-training approach." arXiv preprint arXiv:2010.07245, 2020.

[13] Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. "WordNet:: Similarity-Measuring the Relatedness of Concepts." In AAAI, vol. 4, pp. 25-29, 2004.

[14] Petroni, Fabio, et al. "Language models as knowledge bases?." arXiv preprint arXiv:1909.01066, 2019.

[15] Schick, Timo, and Hinrich Schütze. "Exploiting cloze questions for few shot text classification and natural language inference." arXiv preprint arXiv:2001.07676, 2020.

[16] Schick, Timo, and Hinrich Schütze. "It's not just size that matters: Small language models are also few-shot learners." arXiv preprint arXiv:2009.07118, 2020.

[17] Shin, Taylor, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. "Autoprompt: IV, Eric Wallace, and Sameer Singh. "Autoprompt: Eliciting knowledge from language models with automatically generated prompts." arXiv preprint arXiv:2010.15980, 2020.

[18] Wang, Zihan, Dheeraj Mekala, and Jingbo Shang. "X-class: Text classification with extremely weak supervision." arXiv preprint arXiv:2010.12794, 2020.

[19] Yin, Wenpeng, Jamaal Hay, and Dan Roth. "Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach." arXiv preprint arXiv:1909.00161, 2019.

[20] Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." Advances in neural information processing systems 28, 2015.