Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 3, No. 1, 2023

Using Bidirectional Prompt Learning in NLP Few Shot Tasks

Li Ding, Shiren Ye *

School of Computer Science and Artificial Intelligence Changzhou University, Changzhou 213164, China

* **Corresponding author**: Shiren Ye (Email: yes@cczu.edu.cn)

Abstract: Few-Shot Learning, a subfield of machine learning, aims to solve the problem of learning new tasks with only a small amount of annotated data. Compared to traditional supervised learning, Few-Shot Learning is more challenging because there are very few training samples available during the training process, which means that the model must learn quickly and generalize to new samples. Prompt learning is a recently emerged training paradigm in natural language processing, which can quickly leverage the language capabilities of large pre-trained language models to achieve fast start-up. Based on the prompt learning paradigm, this paper proposes to use its conjugate tasks to further enhance the model's ability in few-shot learning. The experimental results show that the proposed method can effectively improve the performance of the model on multiple datasets and can be combined quickly with other methods for joint optimization.

Keywords: Few-shot; Natural Language Processing; Prompt learning.

1. Introduction

Natural Language Processing (NLP) is an important research direction in the fields of computer science and artificial intelligence. Its aim is to enable computers to understand, process, and generate natural language in order to better interact with humans.

The development history of natural language processing can be traced back to the 1950s. During this period, computer scientists began using limited grammar rules and dictionaries to attempt to solve natural language processing problems. Linguist Noam Chomsky [1] proposed Chomsky's grammar in 1956, which includes four levels: Type 0 grammar (unrestricted grammar), Type 1 grammar (context-sensitive grammar), Type 2 grammar (context-free grammar), and Type 3 grammar (regular grammar). The difference between these grammar levels lies in the complexity of the language rules they describe. Chomsky's grammar is important for the field of natural language processing because it provides a formal grammar description method. However, due to the complexity and diversity of language, the method based on limited grammar rules quickly encountered bottlenecks.

In the 1980s, with the introduction of statistical and probabilistic models, natural language processing underwent a significant transformation. Machine learning language models based on statistics and probability used large-scale corpora to train models, allowing them to better understand and generate natural language. Naive Bayes is a machine learning algorithm based on Bayes' theorem and the assumption of feature independence. Assuming that the impact of each feature on the classification result is independent of each other simplifies the model, making training and prediction more efficient. Naive Bayes classification algorithm performs extremely well in spam email classification, effectively identifying and filtering a large amount of spam emails. Support Vector Machine (SVM)[2] is a commonly used machine learning algorithm that can be used for classification, regression, and anomaly detection tasks. In text classification, SVM was initially used for binary classification problems and can be extended to multiclass classification using the one-vs-all or one-vs-one approach. Hidden Markov Model (HMM)[3] is a probabilistic model based on state transitions that can consider contextual information and predict the next state based on the previous state, commonly used for the conversion between speech and text.

Deep learning originates from the neural network structure in machine learning, which enables the learning of more complex expressions and features from raw data by constructing multiple levels of abstract feature representations to improve the accuracy of prediction and classification. In recent years, deep learning has rapidly developed from a branch of traditional machine learning to a mainstream method. The growth of GPU computing power in recent years has greatly improved the training and inference speed of neural network models, allowing for the implementation and application of larger and more complex models.

The development of deep learning in natural language processing can also be divided into several stages. Early work (2000 to 2016) focused on unsupervised vectorization of words and structural engineering of networks. The bag-of-words model (BOW)[4] was originally used in text classification. This model treats each word in the text as an independent feature and represents the document as a feature vector. The basic idea is to assume that for a text, its word order and grammar syntax can be ignored, and it can be regarded as a collection of vocabulary. Each vocabulary in the text is an independent feature. N-Gram [5] is an algorithm based on statistical language model. Its basic idea is to slide a window of size N over the text, which is segmented into byte fragments, to form a sequence of byte fragments with a length of N. Each byte fragment is called a gram. By counting the frequency of occurrence of all grams and filtering them according to a pre-set threshold, a key gram list is formed, which is the vector feature space of the text. Each gram in the list is a feature vector dimension. From 2017 to 2019, there were significant changes in the learning of NLP models, and the completely supervised paradigm is now playing a decreasing role. Specifically, the standard has shifted towards

pre-training and fine-tuning paradigms. In this paradigm, models with fixed architectures are pre-trained as language models (LMs) to predict the probability of observed textual data. Since a large amount of raw text data is required to train LMs, these LMs can be trained on large datasets to learn robust and general language modeling features. Then, additional parameters are introduced, and task-specific objective functions are used to fine-tune the aforementioned pre-trained LMs to adapt to different downstream tasks. In this paradigm, the focus has mainly shifted to target engineering and designing training objectives for the pre-training and fine-tuning stages. BERT (Bidirectional Encoder Representations from Transformers) [6] is a pre-trained transformer-based model developed by Google. GPT (Generative Pre-trained Transformer) [7] is a pre-trained transformer-based model developed by OpenAI, which is trained on a large corpus of text to generate high-quality language outputs such as text completion, summarization, and translation. Nowadays, natural language processing is in the midst of a second major transformation, with the "pre-train and fine-tune" paradigm being replaced by what we call the "pre-train, prompt, and predict" paradigm. In this paradigm, instead of adapting pre-trained LMs to downstream tasks through target engineering, downstream tasks are reformulated with the help of textual prompts to resemble the tasks solved during the original LM training. For example, when identifying the sentiment of social media posts, for the sentence "I lost my wallet today.", we can prompt the LM with "I feel __" and ask it to fill in the blank with an emotional word. Alternatively, if we choose to prompt the LM with a sentence like "English: I missed the bus today. Chinese: __", we can evaluate its ability to perform machine translation.

Due to the fact that prompt learning is closer to the pre-training task of the language model, it has shown great competitiveness in zero-shot learning and few-shot learning domains. To be able to stimulate the language ability acquired by the model during the pre-training phase, the selection of prompt templates is particularly important. The prompt can be divided into two categories, hard prompts (discrete prompts) and soft prompts (continuous prompts). Hard prompts refer to prompts that are manually designed, that is, templates designed by humans. Manual design is generally based on human natural language knowledge, striving to obtain semantically fluent and efficient templates. In 2020, soft prompts were proposed. Soft prompts are exactly the opposite of hard prompts. They learn the generation of prompts as a task, which is equivalent to changing the generation of prompts from human (discrete) to machine learning (continuous).

Hard prompts and soft prompts each have their advantages and disadvantages. The advantage of hard prompts is that they have good interpretability and can be presented in a form that humans can read. The disadvantage, however, is that what humans consider to be good hard prompts may not necessarily be good hard prompts for a language model. This property is known as the sub-optimality of hard prompts, where the selection of hard prompts can have a significant impact on the performance and stability of pre-trained models. On the other hand, soft prompts are the opposite. They are learned by the model and are more stable than hard prompts, but lack interpretability, making them difficult to apply in tasks that require strong interpretability.

In our work, we propose a learning strategy that is conjugate to prompt learning for few shot tasks. In prompt learning, the input text is fixed and the model is required to fill in the target word. We reversed this task by fixing the target word and requiring the model to fill in the input text. This is based on the same assumption as prompt learning, that the language ability learned by the language model can not only fill in the target word, but also deduce the missing content in the original text from the target word. The results show that on multiple public datasets, whether our method is applied alone or trained jointly with the original prompt learning, it can effectively improve the model's ability in few-shot tasks.

2. Related Work

The work on prompt emerged around 2019. Many researchers began to focus on extracting knowledge from large pre-trained models and applying it to different downstream tasks. Lewis et al [8]. introduce RAG models where the parametric memory is a pre-trained seq2seq model and the non-parametric memory is a dense vector index of Wikipedia, accessed with a pre-trained neural retriever. The result shows that RAG models generate more specific, diverse and factual language than a state-of-the-art parametric-only seq2seq baseline. Jiang et al[9]. propose methods to improve the accuracy of estimating the knowledge contained in LMs by automatically generating better prompts for querying. Our methods include mining-based and paraphrasing-based approaches to generate high-quality and diverse prompts, and ensemble methods to combine answers from multiple prompts. They conducted extensive experiments on the LAMA benchmark, which extracts relational knowledge from LMs, and found that methods improved accuracy from 31.1% to 39.6%. Radford et al [10]. show that substantial improvements in performance on various tasks can be achieved by employing a two-stage approach: generative pre-training of a language model using a diverse corpus of unlabeled text, followed by discriminative fine-tuning on each specific task. By tailoring the architecture of the model to multiple tasks, their methods were able to significantly surpass the state of the art in 9 out of the 12 tasks.

Once proposed, prompt learning became popular in the zero-shot and few-shot learning fields. Compared to the paradigm of "pre-train and fine-tune" which requires the model to learn the specific features needed to output specific tasks on specific labels, prompt learning directly utilizes the language ability inherent in the pretrained model. Its usage is closer to the paradigm during model pretraining, and therefore, it can quickly start downstream tasks. Li et al [11]. proposed prefix-tuning, a lightweight alternative to fine-tuning for natural language generation tasks, which keeps language model parameters frozen, but optimizes a small continuous task-specific vector (called the prefix). Findings indicate that prefix-tuning achieves comparable performance to fine-tuning on the full dataset by learning only 0.1% of the parameters. In settings with limited data, prefix-tuning outperforms fine-tuning and exhibits superior extrapolation capabilities to examples featuring topics not seen during training. Izacard et al [12]. present Atlas, a carefully designed and pre-trained retrieval augmented language model able to learn knowledge intensive tasks with very few training examples. Kojima et al [13]. used chain of thought (CoT) prompting, a recent technique for eliciting complex multi-step reasoning through step-by-step answer examples, achieved the state-of-the-art performances in arithmetics and symbolic reasoning.

Due to prompt learning's direct utilization of the inherent capabilities of upstream networks, many research hotspots in the "pre-train, fine-tune" paradigm have gradually faded in prompt learning research, such as methods that use downstream networks to concatenate upstream networks, train heterogeneous networks in parallel, and adjust loss functions. The research focus in prompt learning has shifted towards how to leverage the capabilities of large-scale language models using prompts. Schick et al [14]. propose a cloze-style prompt-based fine-tuning method called Pattern Exploiting Training (PET). PET is a semi-supervised training procedure that redefines input examples as cloze-style phrases to help language models understand the given task. These phrases are used to assign soft labels to a large number of unlabeled examples. Finally, standard supervised training is performed on the resulting training set. For several tasks and languages, PET significantly outperforms supervised training and strong semi-supervised methods in low-resource settings. Gao et al [15]. utilized prompt-based fine-tuning, developed a novel pipeline for automating prompt generation. Moreover, they have devised a refined approach for selectively and dynamically integrating demonstrations into each context. Results show that it significantly surpasses standard fine-tuning procedures in low-resource scenarios, achieving up to a 30% absolute improvement and an average of 11% across all tasks. AutoPrompt was developed by Shin et al[16]. as an automated approach to generating prompts for a wide range of tasks, utilizing a gradient-guided search. With the help of AutoPrompt, masked language models (MLMs) possess an innate ability to carry out sentiment analysis and natural language inference, without any additional parameters or fine-tuning. Liu et al[17]. proposed P-Tuning.it trains several prompt vectors using regular gradient descent, and use an LSTM to generate a final continuous template from these prompt vectors. Additionally, for certain keywords in the input questions, preserve their tokens in the template. This approach achieves better results than both fine-tuning and manually designed prompt-based methods on superGLUE and some NLU tasks.

In addition to prompt engineering, another important component of prompt learning is answer engineering. Unlike traditional label-based learning, the learning objective of prompt learning is no longer a specific vector, but a answer space Z that is mapped from the label Y. Some work manually design the space of potential answers Z and its mapping to Y. Yin et al[18]. manually design lists of words relating to relevant topics (“health”, “finance”, “politics”, “sports”, etc.), emotions (“anger”, “joy”, “sadness”, “fear”, etc.), or other aspects of the input text to be classified. Obviously, there are limitations to manually constructing mappings. Some work focuses on automatic answer search, albeit less than that on searching for ideal prompts. These work on both discrete answer spaces and continuous answer spaces. Chen et al [19]. introduce a novel approach called KnowPrompt, which leverages this knowledge through the use of learnable virtual type words and answer words during prompt construction, resulting in synergistic optimization. The goal of KnowPrompt is to improve the ability of our system to identify relevant relations by incorporating latent knowledge present in the relation labels. Hambardzumyan et al [20]. explore possibility of using soft answer tokens which can be optimized through gradient descent. They present an alternative approach based on adversarial reprogramming, which extends earlier work on automatic prompt generation.

3. Approach

In traditional supervised learning, the model takes input x and outputs y. This process can be seen as computing $P(y|x;\theta)$.Prompt-based learning methods for NLP attempt to circumvent this issue by instead learning an LM that models the probability $P(x;\theta)$ of text x itself and using this probability to predict y. This approach is more akin to communicating with a human rather than debugging a machine. Table 1 shows the typical pipeline of prompt-based learning.

Table 1. Pipeline of prompt-based learning

Name	Notation	Example
Input	x	I love this move
Output	y	++ (very positive)
Prompting Function	$f_{prompt}(x)$	$[X]$ Overall, it was a $[Z]$ movie.
Prompt	x'	I love this movie. Overall, it was a $[Z]$ movie.
Filled Prompt	$f_{fill}(x', z)$	I love this movie. Overall, it was a bad movie.
Answered Prompt	$f_{fill}(x', z^*)$	I love this movie. Overall, it was a good movie.
Answer Space	z	“good”, “fantastic”, “boring”

Different templates can adapt to different downstream tasks, as shown in Table 2.

Table 2. Different template

Type	Input	Template
Text CLS	I love this movie	$[X]$ The movie is $[Z]$.
Text-span CLS	Poor service but good food.	$[X]$ What about service? $[Z]$.
Text-pair CLS	$[X1]$: An old man with ... $[X2]$: A man walks ...	$[X1]$? $[Z]$, $[X2]$
Tagging	$[X1]$: Mike went to Paris $[X2]$: Paris	$[X1][X2]$ is a $[Z]$ entity
Text Generation	你好	Chinese: $[X]$ English: $[Z]$

However, direct application of manual templates for learning results in significant performance fluctuations in the model. Table 3 from the work of Liu et al [21]., shows that a change in a single word can have a great impact on the model.

Table 3. Case study on LAMA-TREx P17 with bert-base-cased

Prompt	P@1
$[X]$ is located in $[Y]$. (original)	31.29
$[X]$ is located in which country or state? $[Y]$.	19.78
$[X]$ is located in which country? $[Y]$.	31.40
$[X]$ is located in which country? In $[Y]$.	51.08

The same situation also occurs in the answer space, where different words can lead to huge differences in model performance. Table 4 taken from the work of Gao et al., illustrates this point.

It can be seen that relying solely on manual creation for both templates and answer spaces is inefficient and unstable. Compared to soft prompts that lack interpretability, most tasks are more inclined to select or generate templates and

answer spaces through algorithms. AutoPrompt defines the model's prediction for label y as the sum of the probability outputs of the model on all candidate words V_y:

Table 4. The impact of templates and label words on SST-2 dataset

Template	Label words	Accuracy(std)
[*X*] **It was** [*Y*].	great/terrible	92.7 (0.9)
[*X*] **It was** [*Y*].	good/bad	92.5 (1.0)
[*X*] **It was** [*Y*].	cat/dog	91.5 (1.4)
[*X*] **It was** [*Y*].	dog/cat	86.2 (5.4)
[*X*] **It was** [*Y*].	terrible/great	83.2 (6.9)

$$p(y|\boldsymbol{x}_{\text{prompt}}) = \sum_{w\in\mathcal{V}_y} p\left([y] = w|\boldsymbol{x}_{\text{prompt}}\right) \quad (1)$$

When replacing any word in V_y, the gradient of the model's output p changes significantly, indicating a strong correlation between the replaced word and label y, thus it can be included in the answer space.

$$\mathcal{V}_{\text{cand}} = \underset{w\in\mathcal{V}}{\text{top}-k}\left[\boldsymbol{w}_{\text{in}}^{T}\nabla \log p(y|\boldsymbol{x}_{\text{prompt}})\right] \quad (2)$$

Gao et al. proposed a method for automatically searching templates, which involves adding placeholder tokens around the target words and filling these tokens using the T5 model[22] to generate a set of template candidates. Then, the performance of these templates on the training set is calculated by selecting the templates with the highest sum of output probabilities across all samples.

$$\sum_{j=1}^{|T|}\sum_{(x_{\text{in}},y)\in\mathcal{D}_{\text{rain}}} P_{\text{T5}}\,\text{big}(t_j \mid t_1,\ldots,t_{j-1},\mathcal{T}_{\text{g}}(x_{\text{in}},y)) \quad (3)$$

Another way to enhance the modeling capability proposed by Gao et al. is to use demonstrations to indicate to the model what it should learn. The method is to randomly sample examples and concatenate them during the training process. Figure 1 shows this process.

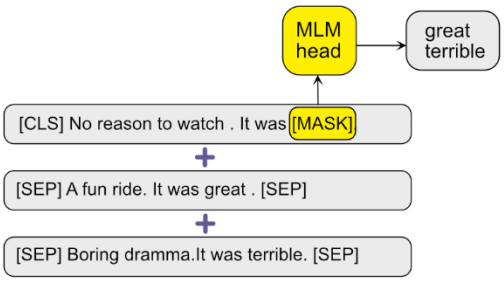

Figure 1. Prompt-based fine-tuning with demonstrations

In our work, we use a method opposite to prompt learning to enhance the model's ability. That is, we keep the target words in the template and randomly mask the tokens of the original corpus, and let the model predict the masked tokens. From the perspective of feature engineering, the model's prediction ability comes from extracting features from the samples, that is, the model establishes a mapping relationship between the samples and the features. If samples can be mapped to features, then features can also be mapped to samples. For example, Variational Encoder (VAE)[23] also known as a generative model, can map high-dimensional data such as images and audio into a low-dimensional latent space and generate new data from it. A more intuitive example is that humans can infer from the template "I love this movie. This movie is _" that the missing word should be "great", and also infer from the template "I _ this movie. This movie is great." that the missing word should be "love". Therefore, these two tasks mutually reinforce the model's ability in sentiment analysis. In fact, the training methods for modern natural language processing models are becoming more and more similar to teaching a child knowledge. The "mask" approach used in large-scale pre-training models actually originated from the common practice of fill-in-the-blank questions in exams. Another term for the demonstrations approach proposed by Gao et al is "example questions" and "real questions". Therefore, the bidirectional prompt learning method we propose is scientifically effective and in line with human intuition.

4. Experiment

We validate our method on four natural language classification datasets.

4.1. datasets

AG's News [24] is a widely used text classification dataset that includes news articles from four different topics - sports, technology, business, and health. The dataset contains 120,000 news articles, with 30,000 articles for each topic. Each article includes a title and a description, as well as a numeric representation of the topic category.

Yahoo is a widely used dataset for natural language processing and text classification tasks. Provided by Yahoo Research, it contains all the news articles collected from Yahoo News, covering various topics such as sports, politics, entertainment, and more. Each article in the dataset includes a news headline, description, and full text, as well as a corresponding topic label. With around 1,000,000 articles, this dataset is very large and suitable for training and testing various natural language processing algorithms and models.

IMDb (Internet Movie Database) is an online database of movies and television shows that includes the world's widest range of film and television information, including movies, TV shows, documentaries, short films, made-for-TV movies, and video games. The IMDB [25] dataset contains 50,000 highly polarized reviews from the database.

Amazon [26] is a large-scale dataset provided by Amazon, which contains millions of customer reviews and star ratings. These reviews cover various categories of products, ranging from books, electronic products, household items to food, health care, clothing and more.

4.2. Experiment Settings

The machine used in this test consists of AMD Ryzen 3600 processor, NVIDIA RTX 3090 graphics card, 32GB memory and win10 operating system. Python version is 3.8.5 and Pytorch version is 1.9.0+cu111. We use RoBERTa[27], an improved model based on the BERT model, as the framework of evaluation sig-loss. We employ Hugging Face library [28] to load and instantiate the pre-trained RoBERTa model.

4.3. Baselines

In this section, we provide a brief introduction to the compared baselines.

Fine-tuning (FT) is a typical way of combining pre-trained models with downstream tasks, which involves connecting the [CLS] token of the pre-trained model to a fully connected layer, and outputting the classification results of the model.

Prompt-tuning (PT) is the most basic way of prompt learning, which uses only a single template and a single answer space.

Better few-shot fine-tuning of language models(LM-BFF) is the method proposed by Gao et al.Includes prompt-

based fine-tuning together with a novel pipeline for automating prompt generation; and a refined strategy for dynamically and selectively incorporating demonstrations into each context.

AutoPrompt(AP) is was developed by Shin et al. It selects prompt tokens from candidate set by gradient.

Bidirectional prompt learning (BPL) is our method. It is worth mentioning that we have used various methods in natural language processing to expand the training samples, including synonym replacement, back-translation, etc. In addition, we have used part-of-speech tagging to select masked tokens, disregarding tokens such as articles and auxiliary verbs that do not have much significance for bidirectional learning.

4.4. Results

Table 5. Results of text classification

Shot	Method	AG	Yahoo	Amazon	IMDB
5	FT	33.4	23.3	53.2	49.4
	PT	78.5	60.4	89.9	88.7
	LM-BFF	79.8	63.6	**90.5**	89.3
	AP	77.2	64.1	88.5	87.2
	BPL	80.3	65.5	90.3	**89.5**
	BPL+ AP	**81.2**	**66.3**	90.1	88.4
10	FT	68.9	38.7	78.6	73.2
	PT	82.3	62.1	90.8	90.5
	LM-BFF	83.9	63.6	91.1	91.4
	AP	81.2	64.6	90.5	89.6
	BPL	84.4	64.3	**91.2**	91.2
	BPL+ AP	**85.8**	**65.2**	90.5	**92.1**
20	FT	79.8	53.6	80.2	76.8
	PT	84.5	66.0	91.3	91.8
	LM-BFF	85.6	**67.7**	93.2	**93.3**
	AP	84.8	66.5	92.8	92.9
	BPL	85.3	67.2	**93.3**	92.6
	BPL+ AP	**86.1**	67.3	93.1	93.0

In Table 5, we compared the performance of various models with sample sizes of 5, 10, and 20, respectively. The results show that compared to the two baseline models, FT and PT, our method can effectively improve the model's ability in few-shot tasks and enable it to quickly boot. Our model can also be combined with existing methods, such as the BPL+AP model in the table, which uses both bidirectional prompt learning and Auto Prompt strategies. It is worth mentioning that compared to the baseline models, the variance of the results of several expansion models has been effectively reduced, demonstrating that these models can more effectively extract the language abilities of large models.

5. Conclusion

Prompt learning is a new paradigm in natural language processing that has emerged in recent years. It can effectively utilize the language abilities learned by large-scale models and unify the training methods for downstream tasks. In practical scenarios, the sample set size is often insufficient or unannotated, and research on few-shot learning requires models to quickly obtain good performance on small training and validation sets. Prompt learning has shown strong competitiveness in the field of few-shot learning, and this article proposes bidirectional prompt learning based on the prompt learning paradigm. By introducing a conjugate task, the model's ability to perform few-shot tasks can be further improved. It can also be easily integrated with most work.

References

[1] Chomsky, N. (2002). Syntactic structures. Mouton de Gruyter.

[2] Platt, J. (1998). Making large-scale support vector machine learning practical. Advances in Kernel Methods: Support Vector Machines.

[3] Eddy, S. R. (1996). Hidden markov models. Current opinion in structural biology, 6(3), 361-365.

[4] El-Din, D. M. (2016). Enhancement bag-of-words model for solving the challenges of sentiment analysis. International Journal of Advanced Computer Science and Applications, 7(1).

[5] Bengio, Y., Ducharme, R., & Vincent, P. (2000). A neural probabilistic language model. Advances in neural information processing systems, 13.

[6] Kenton, J. D. M. W. C., & Toutanova, L. K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT (pp. 4171-4186).

[7] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[8] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

[9] Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How can we know what language models know?. Transactions of the Association for Computational Linguistics, 8, 423-438.

[10] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training..

[11] Li, X. L., & Liang, P. (2021, August). Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 4582-4597).

[12] Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., ... & Grave, E. (2022). Atlas: Few-shot learning with retrieval augmented language models. arXiv preprint arXiv, 2208.

[13] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. Large Language Models are Zero-Shot Reasoners. In ICML 2022 Workshop on Knowledge Retrieval and Language Models.

[14] Schick, T., & Schütze, H. (2021, April). Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (pp. 255-269).

[15] Gao, T., Fisch, A., & Chen, D. (2021, August). Making Pre-trained Language Models Better Few-shot Learners. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 3816-3830).

[16] Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020, November). AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 4222-4235).

[17] Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., & Tang, J. (2022, May). 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).

[18] Yin, W., Hay, J., & Roth, D. (2019, November). Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. 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) (pp. 3914-3923)..

[19] Chen, X., Zhang, N., Xie, X., Deng, S., Yao, Y., Tan, C., ... & Chen, H. (2022, April). Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In Proceedings of the ACM Web Conference 2022 (pp. 2778-2788).

[20] Hambardzumyan, K., Khachatrian, H., & May, J. (2021, August). WARP: Word-level Adversarial ReProgramming. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 4921-4933).

[21] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., & Tang, J. (2021). GPT understands, too. arXiv preprint arXiv:2103.10385.

[22] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), 5485-5551.

[23] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[24] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. Advances in neural information processing systems, 28.

[25] Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). 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).

[26] McAuley, J., & Leskovec, J. (2013, October). Hidden factors and hidden topics: understanding rating dimensions with review text. In Proceedings of the 7th ACM conference on Recommender systems (pp. 165-172).

[27] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[28] McMillan-Major, A., Osei, S., Rodriguez, J. D., Ammanamanchi, P. S., Gehrmann, S., & Jernite, Y. (2021). Reusable templates and guides for documenting datasets and models for natural language processing and generation: A case study of the HuggingFace and GEM data and model cards. arXiv preprint arXiv:2108.07374.