How Does Different Knowledge Type (Parametric vs. Non-Parametric) Affect Pre-trained Language Models?

Xiaochen Zhu
University of Cambridge
xz479@cam.ac.uk

Abstract

Using pre-trained language models has become a standard in recent NLP research, as it shows superiority in refreshing NLP task benchmarks and even indicates human-like reasoning skills (Vaswani et al., 2017; Qiu et al., 2020; Dasgupta et al., 2022). Fine-tuning is the primary technique to update such models' knowledge (Hao et al., 2020). This is inefficient, especially in a dynamic world with rapid knowledge changes. Recent attempts at improvements are instead of tuning the model's parameter, providing more sophisticated inputs to achieve better prediction results (Zhu et al., 2022; Min et al., 2022). Motivated by such initiative, we investigate to what extent non-parametric knowledge such as textual input affects language model outputs and overrides the model's parametric knowledge. We implement a natural language inference task on a novel data set, the UKP data set, (Habernal et al., 2017) with warrants of different polarities as extra non-parametric knowledge to test on. Our experimental results indicate general pre-trained language models are not sensitive to non-parametric knowledge by providing poor prediction accuracy and hardly changing prediction outcomes regardless of the warrants given. Such insensitivity holds regardless of prompts we choose to phrase warrants differently. However, other results demonstrate the potential of additional fine-tuning to impel language models to value non-parametric knowledge more. [1]

1 Introduction

Recent advancements in natural language processing (NLP) can be mainly characterized as the development of larger pre-trained language models (PLM) and their applications on various downstream tasks (Vaswani et al., 2017; Qiu et al., 2020). Such PLMs can be considered as a knowledge base, as in the training phase, an extensive amount of data is encoded as the parameters (Petroni et al., 2019) and given models the ability to capture the distribution of the language (parametric knowledge).

To use PLMs in downstream tasks, the mainstream solution is to fine-tune a desired PLM on a specific data set (Hao et al., 2020). Given the constantly evolving nature of knowledge and the data sparsity in natural language processing, it is important for models to stay up-to-date and adaptable. However, under the current fine-tuning schema, we have to fine-tune PLM constantly for new tasks, or the same task with unseen data, which is rather inefficient.

In addition, recent studies show that PLM can actively use implicit parametric knowledge in downstream task inference (Talmor et al., 2020; Habernal et al., 2017). However, it is possible for these models to incorporate factual errors into their parametric knowledge, which can lead to incorrect inferences. To address this issue, it may be necessary to edit the model's knowledge to ensure its accuracy. Such knowledge editing also requires additional training (De Cao et al., 2021), which is time-consuming as well.

The issues above advocate a careful rethink of the fine-tuning schema for PLM applications. A desired model should be robust to unseen input and flexible enough for online learning. Inspired by In-Context Learning (Min et al., 2022), it's possible to freeze the model parameters and only modify the input (non-parametric knowledge), as a sample of fine-tuning text, to achieve zero-shot learning.

To validate our hypothesis, we use natural language inference (NLI) as a downstream task to test the following two aspects:

1. To what extent does unseen non-parametric knowledge help PLMs in inference?
2. To what extent does non-parametric knowledge overrides the parametric knowledge of PLMs and change the inference outcome?

[1] Code available at https://github.com/SpaceHunterInf/parametric_knowledge

By verifying these aspects, instead of fine-tuning the model frequently, we could freeze the model and use an extra editable knowledge base to extract relevant knowledge as input for better reasoning in a zero-shot fashion.

2 Related Works

In this section, we first review some traditional methods, editing parametric knowledge of PLMs by extra training, to illustrate their limitations. Then, we mention approaches that utilize non-parametric knowledge to assist PLMs in inference to demonstrate the feasibility of our idea.

2.1 Traditional Fine-tuning

Traditional fine-tuning is updating weights for all parameters of a PLM based on a given task with a considerable amount of supervised labels and back-propagate the loss (Hao et al., 2020). Although this popular approach beats many NLP benchmarks given a sufficient amount of data, its limitations are not negligible, including being time-consuming, requiring a large amount of data, poor out-of-distribution generalization and the potential of exploiting spurious features of training data (Brown et al., 2020). In our context, questions remain unanswered to clearly and easily measure the sufficiency of fine-tuning to encode new knowledge or correct old knowledge into the model's parametric knowledge, and if such modification perturbs the model's performance involving unrelated knowledge.

2.2 Knowledge Editing

Knowledge editing focuses on editing weights of neural networks with a specialized training routine to correct factual knowledge errors (Zhu et al., 2020; Sinitsin et al., 2020). A typical work uses a hyper-network and constrained optimization to ensure knowledge updates can be made easily and do not affect unrelated knowledge (De Cao et al., 2021). In such a schema, training is still required and we need to know the knowledge we want to change in advance. In large-scale experiments, if many facts need to be edited, it could be time-consuming as well.

2.3 In-Context Learning

In-context learning is an example of how non-parametric knowledge helps PLMs produce meaningful predictions in downstream tasks without training (Min et al., 2022). It only requires few-shot samples of input and output pairs formatted in natural language as a demonstration. With testing input appended to the end of the demonstration, PLMs output meaningful prediction competitively enough compared with traditional fine-tuned models (Xie et al., 2021). In our context, we can provide such additional information as non-parametric input.

2.4 Prompting

In addition, prompting is another non-parametric knowledge that greatly enhances the performance of PLM. Prompting is defined as a function that modifies the input text, this could be a text template, discrete prompts (certain tokens) or even continuous prompts with no corresponding text (Liu et al., 2021). In our context, we can use prompts to emphasize certain parts of the input and make better predictions.

The works above illustrate that training-related approaches still share limitations such as being inefficient. They also indicate the great potential of using non-parametric knowledge to improve PLMs' inference in test time without training. However, these studies did not reach conclusion about how non-parametric and parametric knowledge interacts with each other, and which one the model values more. These questions will be our main focus in the following sections.

3 Task

We use the NLI task to demonstrate the effect of non-parametric knowledge over PLMs. NLI is also known as recognizing textual entailment (RTE). The task is defined as determining the inference relation between two (short, ordered) texts, a premise P and a hypothesis H: entailment, contradiction, or neutral (Bowman et al., 2015). In NLI, it is generally accepted that strict logical deduction is not necessary. Instead, what is required is the ability to make inferences similar to those made by humans when there are logical gaps and additional information, such as common sense, is available to assist in the inference process (Manning, 2006; Williams et al., 2017).

We perform NLI on a specific data set of argument reasoning with implicit warrant reconstruction, the UKP data set (Habernal et al., 2017). In this setting, there are 4 components, a reason R, and claim C, a correct warrant W and an alternative warrant AW. The data set is designed so that by formulating the correct warrant and the reason as the premise, we could infer the claim as the hypothesis. By contrast, using the incorrect warrant together with the reason refutes the claim.

R:	Cameras should monitor police officers 24/7.
C:	Police officers should be required to wear cameras.
W:	Police officers have no right to privacy in many situations
AW:	Police officers also have the right to privacy in many situations.

Table 1: UKP data example.

Taking an example from the data set as above, we describe excepted human annotator behaviour and compare it in the context of using PLMs. A human annotator could be asked for a label given only the pair R and C. In this situation, human annotator will inevitably use their additional knowledge (i.e. commonsense) for prediction. This would be invoking additional parametric knowledge of PLMs. When associating R and W together as a premise, and C as a hypothesis, a human annotator would give entailment as the NLI label. By contrast, if R is associated with AW, the label would be a contradiction. Sometimes a warrant is not known in advance by a human annotator, in this case, it's the case of extra non-parametric knowledge assisting PLMs in inference. In addition, an alternative warrant is mostly inconsistent with the real-world knowledge, or the conception of the human annotator, however, a human annotator can be asked to take the alternative warrant for granted and derive a result. This is when non-parametric knowledge overrides the original parametric knowledge of PLMs.

The routine of this task consists of 3 aspects as the following:

1. Probe model's initial belief of pair R, C.

2. Give $R + W, C$ and examine if the model yields a different result.

3. Give $R + AW, C$ and examine if the model yields a different result.

4 Experiments

We experimented with a wide range of PLMs with different architectures in order to reach a general conclusion. We used encoder-only models, BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and decoder-only models, GPT2 (Radford et al., 2019), and also encoder-decoder model T5 (Raffel et al., 2020) for the following experiments.

Firstly we fine-tune our model to reach a similar performance with an NLI data set, SNLI (Bowman et al., 2015). The models are chosen to have a similar number of parameters for better comparison.

Model	Parameters	Accuracy
BERT	110M	89.35%
RoBERTa	125M	90.33%
GPT2	117M	88.31%
T5	60M	87.65%

Table 2: Accuracy on SNLI

4.1 Model Setup

For models with different architectures, we define their inference as the following.

- **Encoder Models** For BERT and RoBERTa, we add an additional classification layer to the model directly. Firstly we add special tokens to separate the premise and hypothesis, and then the last hidden state, the sentence embedding, is used as the input of the classification layer.

- **Decoder Models** For GPT and T5, we first formulate the premise and hypothesis into a natural language paragraph with the template "Inference the following sentence, premise: $\{P\}$; hypothesis:$\{ H \}$. L", where P is the premise, H is the hypothesis and L is the label token. We fine-tune the model with such a template and take the last token as the prediction label.

4.2 Knowledge Probing

Initially, we also want to investigate whether any warrant is learnt by PLM as a piece of factual knowledge, used in inference and affects the prediction result. We set up a small and simple experiment to verify this by trying to probe the truthfulness of the given knowledge. We used the technique of zero-shot classification pipeline (Yin et al., 2019). We give 3 classes $True$, $False$ or $Unkown$, for each warrant in the data set. We want to see if this classification task has any correlation with the result of the NLI prediction. If a correct warrant is believed as $True$ and NLI prediction is entailment by just giving R and C or the alternative warrant is believed as $True$ and NLI result is a contradiction, then there is one more evidence to support the finding that large PLMs have human-like reasoning competence (Dasgupta et al., 2022).

However, the results show no significance. In most of the scenarios, all PLM consider both correct and alternative warrants of a single data instance as $True$. Instead of giving the truthfulness of a statement, PLMs classify statements containing negation such as“no, not” as $False$ and others as $True$. Thus, under current settings, it's difficult to determine whether non-parametric knowledge, a warrant, is new to PLM or in contrast with the model's original parametric knowledge. Though the warrant cannot be probed explicitly, we can still combine the CM rate together with accuracy to observe belief change.

4.3 Evaluation Metrics

The first part of the evaluation follows the traditional NLI benchmark by testing the prediction accuracy against the gold labels, however, for each aspect motioned in section 3 we have different labels.

1. UKP data set is designed such that C should always be entailed if a correct warrant is chosen. When only R and C are given, we set the gold label as entailment, to test the model's initial belief.

2. If W is given, the gold label is entailment, since $R + W$ should entail C.

3. If AW is given, the gold label is contradiction, since $R + AW$ would refute C.

UKP data set has larger logical gaps for NLI task given only R and C since it's designed to infer with warrants. We expect to have a lower accuracy with only R and C given. However, the NLI accuracy should not be a stand-alone metric. We are more focused on the impact of non-parametric knowledge, the warrants, on NLI predictions. The accuracy is calculated only for reference.

The second part of our evaluation is based on the NLI result. We define a change-mind rate (CM) indicating after the percentage of instances that the model originally predicted a wrong label, but changed to the gold label providing a warrant. For a given tuple $(R, W, AW, C) = D$, a data set $\mathbf{D} = \{D_1, ..., D_n\}$, a model taking two arguments and return a label $NLI(,)$, and a function [] returns 1 for $True$ and 0 for $False$, CM rate of W is below:

$$CM_W = \frac{\sum_i^n [NLI(R_i + W_i, C_i) = entailment]}{\sum_i^n [NLI(R_i, C_i) \neq entailment]} \quad (1)$$

4.4 Raw Input Results

We run 3 aspects of the task directly using all PLMs we have on the UKP testing set and achieved the results below. We simply concatenated the additional warrant to the end of the original reason as the premise. If a warrant is given, higher accuracy and CM rate indicate a PLM is more sensitive to parametric knowledge. As mentioned above, for each test, there is a single gold label. For example, if AW is given, then all gold labels should be "contradiction". In this case, a random baseline for each test is 33%.

Model	**R, C**	**R+W, C**	**R+AW, C**
BERT	43.91%	49.32%	29.95%
RoBERTa	34.23%	35.36%	31.30%
GPT2	16.44%	18.47%	51.8%
T5	33.11%	33.78%	54.95%

Table 3: Accuracy using simple concatenation.

Model	CM_W	CM_{AW}
BERT	19.28%	13.34%
RoBERTa	8.56%	10.53%
GPT2	7.28%	16.44%
T5	12.46%	36.05%

Table 4: CM rate using simple concatenation.

From the tables above, we can see without additional training, every PLM tends to stick with their original result. Indeed, slight improvements can be observed after providing additional warrants, but the general CM rate is still below 50% percent indicating PLMs value more about their parametric knowledge and are reluctant to use non-parametric knowledge as an override.

4.5 Prompted Input Results

In this section, we choose hard prompts to emphasize the warrants. Hard prompts are chosen as phrases that make the human evaluators take the warrant regardless of the warrant's consistency with one's belief or real-world knowledge. We also use the famous prompt "Let's think step by step.", though it's not a chain-of-thought type problem (Kojima et al., 2022). In addition, we added a random gibberish prompt just for comparison.

1. Given that {X}.

2. If we only consider the fact that {X}.

3. Given ground truth X. Let's think step by step.

4. asdfghjkl {X}.

The prompt templates chosen are above. X is the placeholder for warrants. After templating the warrant, it is concatenated to the end of the corresponding reason as the premise.

The accuracy and CM rate on the UKP testing set are shown in the tables below. Generally speaking, there is no substantial increase or decrease in both accuracy and CM rate using the prompt we chose for all models. One outlier is using the BERT model with random prompts, resulting in a CM rate close to 90% given the correct warrant. However, it also results in a dramatic decrease in the accuracy and CM rate gave the alternative warrant.

During test time, the polarity of a given prompt is hidden from the model. Therefore, a good prompt should increase the average accuracy and CM rate of both W and AW. In this case, the random prompt cannot show its superiority over other prompts.

Another observation is models that included decoder architecture tends to believe in alternative warrant-based premises. As the related change mind rate and accuracy are always higher than correct warrant-based premises. This is more obvious for GPT2 as it's accuracy for "entailment" as the gold label is below the random baseline.

Prompt	Avg Accuracy	Avg CM
prompt 1	37.08%	13.86%
prompt 2	35.98%	12.45%
prompt 3	40.09%	19.56%
prompt 4	40.93%	22.44%

Table 7: Average performance of each prompt.

Model	Avg Accuracy	Avg CM
BERT	42.12%	23.92%
RoBERTa	32.60%	12.81%
GPT2	34.94%	10.35%
T5	44.26%	21.40%

Table 8: Average performance of each model.

We take the average accuracy and CM of both W and AW to check the general performance regarding to each prompt and model. In general, the use of different prompts affects the model's performance, and surprisingly, prompt 4 (i.e the random prompt) is the most effective one as it has the highest accuracy and CM. BERT and T5 have better accuracy and are also more sensitive to non-parametric knowledge compared to other models. GPT2 is the most reluctant to be affected by additional warrants.

4.6 Fine-tuned Model Results

The previous experiments demonstrate the difficulty of using additional non-parametric knowledge to assist PLMs in inference. In this section, we investigate whether it's possible to train language models to value non-parametric knowledge when triggered by prompts.

We use the UKP training set as our data and use the prompt template "Given that {X}" to concatenate the warrant and the reason as the premise, the corresponding claim as the hypothesis and the correctness of the warrant as one of two NLI labels, entailment or contradiction. We fine-tune all models with 5 epochs and evaluate them again on the UKP testing set.

Model	R+W,C	R+AW,C
BERT	47.07%	70.95%
RoBERTa	70.50%	57.66%
GPT2	57.66%	63.96%
T5	45.72%	71.85%

Table 9: Accuracy on fine-tuned models with prompts.

Model	CM_W	CM_{AW}
BERT	50.20%	72.82%
RoBERTa	65.41%	50.66%
GPT2	55.80%	49.32%
T5	37.37%	65.31%

Table 10: CM rate on fine-tuned models with prompts.

From the tables above, an obvious increase in both accuracy and CM rate after fine-tuning can be observed across all language models. This proves that we can train the model to rely on non-parametric knowledge over its parametric ones.

We further tested the performance of our fine-tuned language models again on the SNLI data. This is to verify when no additional non-parametric knowledge is given, can model still perform inference normally.

	Prompt 1		Prompt 2		Prompt 3		Prompt 4	
Model	W	AW	W	AW	W	AW	W	AW
BERT	51.13%	26.35%	54.73%	22.75%	52.70%	33.11%	**89.41%**	6.76%
RoBERTa	32.43%	29.05%	25.67%	32.08%	17.34%	56.98%	33.78%	34.46%
GPT2	17.57%	51.80%	20.05%	47.97%	18.47%	52.03%	16.44%	55.41%
T5	37.84%	50.68%	40.09%	45.50%	37.16%	52.92%	41.89%	49.32%

Table 5: Accuracy using prompted inputs.

	Prompt 1		Prompt 2		Prompt 3		Prompt 4	
Model	CM_W	CM_{AW}	CM_W	CM_{AW}	CM_W	CM_{AW}	CM_W	CM_{AW}
BERT	22.08%	10.26%	26.91%	7.18%	22.49%	11.79%	**85.54%**	5.13%
RoBERTa	6.85%	10.53%	37.67%	13.82%	2.39%	47.37%	7.88%	9.87%
GPT2	6.47%	12.33%	7.55%	8.22%	7.82%	19.18%	6.20%	15.97%
T5	13.13%	29.25%	13.80%	19.72%	14.14%	31.30%	18.51%	31.29%

Table 6: CM rate using prompted inputs.

Model	Accuracy
BERT	73.39%
RoBERTa	78.02%
GPT2	78.26%
T5	66.50%

Table 11: Accuracy on SNLI after fine-tuning

Though accuracy generally dropped, all language models maintain a high level of performance on the SNLI data set after fine-tuning. Considering UKP and SNLI has different data distribution, UKP data are generally longer and contains data inconsistent with a real-world scenario, the dropped can be explained by catastrophic forgetting. With better fine-tuning techniques, such as review learning or using adapter architecture, this can be easily improved.

5 Conclusion and Future Work

To investigate to what extent non-parametric knowledge affects PLMs, we set up an NLI task using UKP data set, by comparing the model's prediction with different warrants as non-parametric knowledge. Our experimental results demonstrate that the 4 PLMs we choose (Bert, RoBERTa, GPT2, T5) are not very sensitive to non-parametric knowledge. It's difficult to manipulate prediction outcomes by giving the model additional textual input. Prompting can also not help alter the model's prediction in a clear and stable fashion.

However, this does not eliminate the possibility of manipulating the model's output without further fine-tuning. In our results, using random prompts can greatly increase the accuracy and change the prediction outcome of BRET. With further prompt engineering, such as continuous prompt tuning, we can make models to take non-parametric knowledge more effectively (Zhu et al., 2022).

Furthermore, we managed to fine-tune PLMs to take non-parametric knowledge as an oracle. We fine-tuned PLMs on the UKP training set and evaluate them on both UKP and SNLI testing sets. By combining non-parametric knowledge with certain prompts as triggers, PLM values non-parametric knowledge more, while keeping the independence to use parametric knowledge for unperturbed predictions if no additional knowledge is available. It's also important to investigate the impact of confounding concepts of using non-parametric knowledge as an oracle. When parametric factual knowledge is overridden by non-parametric knowledge, can model adapt and change related concepts as well. As such, this mini-project constitutes an attempt towards parametric-efficient knowledge updates for PLMs.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ishita Dasgupta, Andrew K Lampinen, Stephanie CY Chan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. 2022. Language models show human-like content effects on reasoning. *arXiv preprint arXiv:2207.07051*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *arXiv preprint arXiv:2104.08164*.

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*.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2017. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. *arXiv preprint arXiv:1708.01425*.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2020. Investigating learning dynamics of bert fine-tuning. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 87–92.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

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*.

Christopher D Manning. 2006. Local textual inference: It's hard to circumscribe, but you know it when you see it–and nlp needs it.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *arXiv preprint arXiv:2004.00345*.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. *Advances in Neural Information Processing Systems*, 33:20227–20237.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. 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)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual prompt tuning for dialog state tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland. Association for Computational Linguistics.