Pre-Training to Learn in Context

Yuxian Gu[1,2,*], **Li Dong**[2], **Furu Wei**[2], **Minlie Huang**[1,†]

[1]The CoAI Group, DCST, Institute for Artificial Intelligence, State Key Lab of Intelligent Technology and Systems,
Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
[2] Microsoft Research

guyx21@mails.tsinghua.edu.cn, {lidong1,fuwei}@microsoft.com
aihuang@tsinghua.edu.cn

Abstract

In-context learning, where pre-trained language models learn to perform tasks from task examples and instructions in their contexts, has attracted much attention in the NLP community. However, the ability of in-context learning is not fully exploited because language models are not explicitly trained to learn in context. To this end, we propose PICL (**Pre**-training for **In**-**C**ontext **L**earning), a framework to enhance the language models' in-context learning ability by pre-training the model on a large collection of "intrinsic tasks" in the general plain-text corpus using the simple language modeling objective. PICL encourages the model to infer and perform tasks by conditioning on the contexts while maintaining task generalization of pre-trained models. We evaluate the in-context learning performance of the model trained with PICL on seven widely-used text classification datasets and the SUPER-NATURALINSTRCTIONS benchmark, which contains 100+ NLP tasks formulated to text generation. Our experiments show that PICL is more effective and task-generalizable than a range of baselines, outperforming larger language models with nearly 4x parameters. The code is publicly available at `https://github.com/thu-coai/PICL`.

1 Introduction

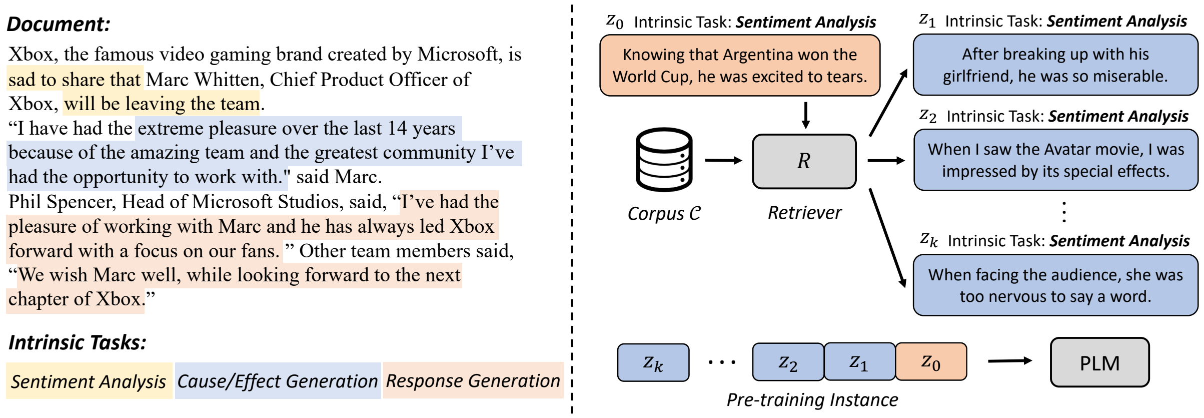
Pre-trained language models (PLMs; Han et al., 2021; Qiu et al., 2020) have shown strong abilities of learning and performing unseen tasks conditioning on several task examples or instructions in its context, which is called *in-context learning* (ICL; Brown et al., 2020). Compared to conventional fine-tuning methods, ICL adapts PLMs to downstream tasks only through inference, without parameter updates, which is computationally cheaper in practice and is closer to general AI.

However, PLMs trained on massive corpora to predict the next word given previous words are not explicitly taught to learn in the context. This makes ICL a surprising emergent ability but also indicates that the ICL ability of PLMs is not fully exploited. Garg et al. (2022) has shown that by directly training to do ICL in a meta-learning paradigm, models show strong performance on learning simple function classes in the context. In practical NLP scenarios, previous works (Min et al., 2022b; Chen et al., 2022b) also enhance the ICL performance by meta-fine-tuning PLMs on a large collection of downstream tasks and evaluating them on unseen tasks. However, the low diversity of human-annotated downstream tasks restricts the performance of the meta-tuned model. Direct training on downstream tasks also brings undesired bias on specific input formats, label spaces, or domains, which hurts the generalization of PLMs.

To enhance the ICL ability while maintaining generalization, we propose PICL (**Pre**-training for **In**-**C**ontext **L**earning), a framework that exploits the PLM's ICL ability by pre-training models on data automatically constructed from the general plain-text corpus. Our framework is based on a simple observation that many paragraphs in the text documents contain "intrinsic tasks". As shown in the left part of Figure 1, each paragraph in the document contains an intrinsic task. When doing language modeling on each paragraph, models implicitly perform the corresponding intrinsic tasks simultaneously. This shares a similar idea with the prompt-learning paradigm (Liu et al., 2021), where downstream data examples from NLP tasks are transformed into text sequences, and the model learns to perform the original tasks when trained on the text sequences with language modeling. Different from the downstream data, text paragraphs contain more diverse intrinsic tasks and have little bias on input formats, label spaces, or domains because they are free-form texts from the large-

[†] Corresponding author.
[*] Contribution during internship at Microsoft Research.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4849–4870
July 9-14, 2023 ©2023 Association for Computational Linguistics



Document:

Xbox, the famous video gaming brand created by Microsoft, is sad to share that Marc Whitten, Chief Product Officer of Xbox, will be leaving the team.
"I have had the extreme pleasure over the last 14 years because of the amazing team and the greatest community I've had the opportunity to work with." said Marc.
Phil Spencer, Head of Microsoft Studios, said, "I've had the pleasure of working with Marc and he has always led Xbox forward with a focus on our fans. " Other team members said, "We wish Marc well, while looking forward to the next chapter of Xbox."

Intrinsic Tasks:

Knowing that Argentina won the World Cup, he was excited to tears.

After breaking up with his girlfriend, he was so miserable.

When I saw the Avatar movie, I was impressed by its special effects.

When facing the audience, she was too nervous to say a word.

Figure 1: **Left:** An example of intrinsic tasks found in a document from the OpenWebText (Gokaslan et al., 2019) corpus. **Right:** The overall framework of PICL. For each paragraph z_0 in the corpus $\mathcal{C}$, we retrieve k paragraphs that share the same intrinsic task (Sentiment Analysis) as demonstrations and then concatenate them with z_0 to construct a pre-training instance. We compute the language modeling loss on the whole instance to train the model.

scale general corpus. By gathering and concatenating paragraphs with the same intrinsic tasks (right part of Figure 1), we can construct a meta-training dataset to pre-train the model to perform the intrinsic task conditioning on paragraphs in the context, and thereby improve the ICL ability.

We adopt a retrieval-based approach to gather paragraphs sharing the same intrinsic tasks from a general corpus. We first train an encoder to represent each paragraph in a vector space where paragraphs with the same intrinsic task have close embeddings. The encoder is trained with contrastive learning (Khosla et al., 2020) on a collection of downstream datasets by taking examples from the same tasks as positive pairs and those from different tasks as negative pairs. Then, treating any paragraph in the corpus as a query, we retrieve the paragraphs with close representations to the query, namely, sharing the same intrinsic task with the query. Finally, we concatenate the query and the retrieved paragraphs to get a pre-training instance. Note that although we use downstream datasets, the model is trained on instances constructed from the general corpus, which ensures its generalization.

We evaluate the ICL performance of the model pre-trained with PICL on seven widely-used text classification datasets and SUPER-NATURALINSTRUCTIONS (Wang et al., 2022), a benchmark whose test split contains more than 100 tasks formulated into text generation. Empirical results show the effectiveness of PICL, enabling the model to reach or even outperform larger models with nearly 4x parameters. Besides, we find that the PICL-trained model is more generalizable on various tasks than previous meta-fine-tuning methods. We also conduct extensive experiments to analyze several key factors of PICL.

2 Method

We first present an overview of PICL and then describe the details in the following sections. As shown in the right part of Figure 1, we construct the pre-training instances from a corpus $\mathcal{C}$ consisting of paragraphs split from full documents by "\n". For each paragraph z_0 in $\mathcal{C}$, we first use a retriever R to find k paragraphs $\{z_1, z_2, \cdots, z_k\}$ sharing the same intrinsic task (Sentiment Analysis) with z_0. Then the retrieved paragraphs are treated as demonstrations and concatenated with z_0 to form a pre-training instance: $z_k \oplus z_{k-1} \oplus \cdots \oplus z_1 \oplus z_0$. Finally, we adopt a language modeling objective to pre-train the model on the constructed instances.

In this way, the pre-training stage can be regarded as a meta-training process, where the model learns to solve the intrinsic task in z_0 conditioning on its context $z_k \oplus z_{k-1} \oplus \cdots \oplus z_1$. Since $\mathcal{C}$ is a large-scale general corpus, it contains a variety of intrinsic tasks and little domain bias, which ensures the generalization of the pre-trained model.

2.1 Retriever

The main component of the retriever R is a task-semantics encoder E that represents a text paragraph as a d-dimensional vector in a space V, where paragraphs with the same intrinsic tasks have similar representations. We define the similarity between two paragraphs z_0 and z using the dot product of their representations: $E(z_0) \cdot E(z)$.

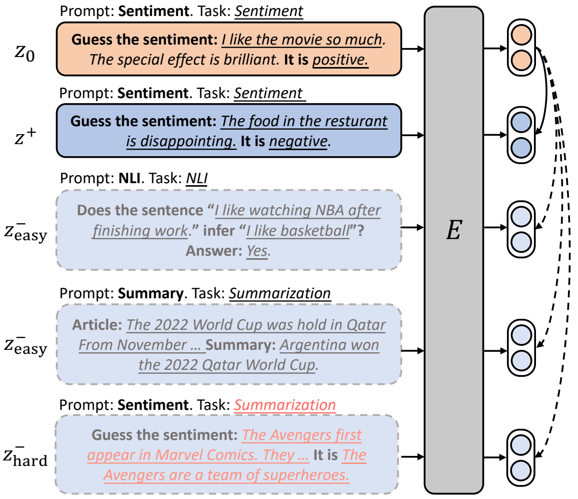

Prompt: Sentiment. Task: Sentiment
z_0
Guess the sentiment: I like the movie so much. The special effect is brilliant. It is positive.
Prompt: Sentiment. Task: Sentiment
z^+
Guess the sentiment: The food in the resturant is disappointing. It is negative.
Prompt: NLI. Task: NLI
z^-_{easy}
Does the sentence "I like watching NBA after finishing work." infer "I like basketball"? Answer: Yes.
Prompt: Summary. Task: Summarization
z^-_{easy}
Article: The 2022 World Cup was hold in Qatar From November ... Summary: Argentina won the 2022 Qatar World Cup.
Prompt: Sentiment. Task: Summarization
z^-_{hard}
Guess the sentiment: The Avengers first appear in Marvel Comics. They ... It is The Avengers are a team of superheroes.
E

Figure 2: An example of how we construct the positive and negative pairs to train the task-semantics encoder E. The solid line means positive pairs, and the dashed lines mean negative pairs.

Encoder We use RoBERTa$_{\text{BASE}}$ (Liu et al., 2019) as the base model of E. The output vector is computed by averaging the last-layer representation of each token in the input paragraph.

Retrieval We approximate that paragraphs whose representations are close to each other in V share the same intrinsic task. Therefore, for every paragraph z_0 in $\mathcal{C}$, R searches for k paragraphs with embeddings closest to $E(z_0)$:

$$R(z_0) = \{z_k, z_{k-1}, \cdots z_1\} = \text{top-k}_z(E(z_0) \cdot E(z)). \quad (1)$$

We employ the FAISS library (Johnson et al., 2019) for efficient searching.

Contrastive Training We adopt contrastive learning (Khosla et al., 2020; Karpukhin et al., 2020) to train the task-semantics encoder E. As shown in Figure 2, we take two paragraphs with the same intrinsic task as positive pairs and those from different tasks as negative pairs. However, the annotation of a paragraph's intrinsic task is usually unavailable. To this end, we use a collection of downstream NLP datasets from various tasks whose examples are converted into text sequences with human-written prompts to train E. In this way, treating each text sequence as a paragraph, we can regard the corresponding downstream task as the intrinsic task annotation. We assume that the instances from all downstream tasks form a dataset $\mathcal{D}$. For each $z_0 \in \mathcal{D}$, we have a positive instance z^+ sharing the same task with z_0 and a set $\mathcal{N}(z_0)$ consisting of negative instances with different tasks than z_0, the loss function takes the form:

$$\begin{aligned}&\mathcal{L}(z_0, z^+, \mathcal{N}(z_0)) \\ &= -\log \frac{e^{E(z_0)\cdot E(z^+)}}{e^{E(z_0)\cdot E(z^+)} + \sum\limits_{z^- \in \mathcal{N}(z_0)} e^{E(z_0)\cdot E(z^-)}}.\end{aligned} \quad (2)$$

Positive and Negative Instances For each $z_0 \in \mathcal{D}$, we randomly sample a positive instance z^+ belonging to the same task with z_0 from $\mathcal{D} \backslash \{z_0\}$. As shown in Figure 2, $\mathcal{N}(z_0)$ contains two kinds of negative instances: (1) Easy Negatives z^-_{easy} sampled from $\mathcal{D}$ and belonging to different tasks than z_0. (2) Hard Negatives z^+_{hard} sharing the same prompt with z_0 but containing mismatched tasks. For instance, in Figure 2, we apply the prompt from the sentiment task to the summarization task to create the hard negative instance z^-_{hard}. This prevents the model from hacking the contrastive objective using prompts like "Guess the sentiment" and learning a trivial pattern matching but forces the model to extract task semantics from the whole paragraph.

2.2 Data Construction

For each paragraph $z_0 \in \mathcal{C}$, we concatenate the retrieved paragraphs $\{z_1, z_2, \cdots, z_k\} = R(z_0)$ with z_0 to get a pre-training instance $z_k \oplus z_{k-1} \oplus \cdots \oplus z_1 \oplus z_0$. To improve the quality of the constructed data, we derive an approach to filter out instances that are less informative to ICL. We consider the following score to measure the informativeness of an instance based on the perplexity difference of the paragraphs in the instance before and after they are concatenated as a sequence:

$$s = \frac{-\sum_{i=0}^{k} \log P(z_i) + \log P(z_k \oplus z_{k-1} \oplus \cdots \oplus z_0)}{|z_k \oplus z_{k-1} \oplus \cdots \oplus z_0|}, \quad (3)$$

where $|\cdot|$ is the length of a sequence and $P(\cdot)$ is the language modeling probability based on any uni-direct PLMs. Given a manually set threshold δ, we retain the instances that satisfy $s > \delta$. This criterion leverages the original ICL ability of the PLM. If concatenating the paragraphs results in lower perplexity, they are more correlated and may be more informative for ICL. We finally construct a pre-training corpus containing N instances $\mathcal{C}_{\text{pre-train}} = \{z^i_k \oplus z^i_{k-1} \oplus, \cdots, \oplus z^i_1 \oplus z^i_0\}^N_{i=1}$.

2.3 Pre-Training

We pre-train the model with auto-regressive language modeling on $\mathcal{C}_{\text{pre-train}}$. Unlike previous works (Min et al., 2022b; Chen et al., 2022b), which only compute the language modeling loss on the label tokens, we compute the loss on the whole sequence. There are two reasons for this choice. First, the intrinsic tasks are already in the natural language format, and it is unnecessary to split the input and the label. Second, we argue that computing loss on the whole sequence ensures a large token number in a forward batch, which is critical to maintaining the basic in-weights ability (Chan et al., 2022). Therefore, the loss function is:

$$\mathcal{L}_{\text{ICL}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log P(z_k^i \oplus z_{k-1}^i \oplus \cdots \oplus z_0^i; \theta), \quad (4)$$

where θ is the parameters of the model. In addition, we find that adding a language modeling loss $\mathcal{L}_{\text{LM}}(\theta)$ on the original full documents before being split into paragraphs benefits the performance. Therefore, the final optimization objective is:

$$\min_{\theta} \alpha\mathcal{L}_{\text{ICL}}(\theta) + (1-\alpha)\mathcal{L}_{\text{LM}}(\theta). \quad (5)$$

where we set $\alpha = 0.5$ in our main experiments.

3 Experimental Setup

3.1 Pre-training Data

We merge OpenWebText (Gokaslan et al., 2019), WikiCorpus (Foundation, 2022), and BookCorpus (Zhu et al., 2015) to construct the pre-training data, where full documents are split into paragraphs by "\n". The corpus $\mathcal{C}$ consists of 80M paragraphs, totaling about 30GB. For each paragraph, we search for $k = 20$ demonstrations and concatenate them until 1024 tokens, the maximum input length constraint of the language model we used. This ensures that the model sees various demonstration numbers during pre-training. We use GPT2-Large (Radford et al., 2019) to compute $P(\cdot)$ in Equation 3 and set $\delta = 0.0$ for filtering. More details of data processing and statistics are shown in Appendix A.

3.2 Baselines

We consider four baselines in our experiments:

- **VanillaICL** directly prompts a PLM with the concatenation of training examples to do ICL.
- **ExtraLM** further pre-trains the PLM on the original full documents before being split into paragraphs with the language modeling objective.
- **Self-Sup** (Chen et al., 2022a) designs four self-supervised pre-training objectives, including Next Sentence Generation, Masked Word Prediction, Last Phrase Prediction, and Classification, to enhance the ICL performance. We conduct the self-supervised pre-training on our merged corpus for a fair comparison.
- **MetaICL** (Min et al., 2022b) meta-trains the model on a large collection of downstream human-annotated datasets for learning to learn in context. The meta-training instances are constructed by concatenating several training examples in each dataset to a single text sequence. We replicate the method on the training set of our task-semantics encoder for a fair comparison.

3.3 Evaluation

We evaluate the model trained with PICL on two kinds of downstream tasks.

Few-Shot Text Classification We consider seven widely-used text classification datasets, including SST-2 (Socher et al., 2013), SST-5 (Socher et al., 2013), Subj (Pang and Lee, 2004), MR (Pang and Lee, 2005), RTE (Dagan et al., 2006), CB (De Marneffe et al., 2019), and AGNews (Zhang et al., 2015) to evaluate the few-shot ICL performance of the trained models (see Appendix B.1 for more details). Note that these tasks are not included in the training set of the task-semantics encoder. We randomly sample 4 or 8 demonstrations from the official training sets of each dataset. Effects of other demonstration numbers can be found in Section 4.3. We compute the average accuracy scores on at most 1000 samples from the validation split of each dataset across five random seeds for selecting demonstrations.

Instruction Following To test the generalization of PICL, we also evaluate the trained model on a larger range of tasks with more free-form inputs, including both human instructions and few-shot examples. We use the test split of Super-NaturalInstructions (Wang et al., 2022) as the benchmark and exclude the tasks that appear in the training set of the task-semantics encoder, resulting in 105 evaluation tasks (see Appendix B.2 for a full list of tasks). Each task is specified with a human-written instruction and two or three demonstrations. We follow Wang et al. (2022) to formulate all tasks to the text generation format and score the outputs with ROUGE-L (Lin, 2004).

3.4 Settings

Retriever We train the task-semantics encoder on 37 tasks (see Appendix C) using up to 10K examples per task. To enhance generalization, we

Shot	Method	Param.	SST2	SUBJ	MR	RTE	AgNews	CB	SST5	Average
4-shot	VanillaICL	770M	$67.5_{9.2}$	$57.7_{7.8}$	$50.3_{0.3}$	$50.8_{1.7}$	$67.5_{2.3}$	$68.1_{2.4}$	$24.4_{5.4}$	$55.2_{0.5}$
	VanillaICL	1.5B	$74.9_{9.7}$	$65.2_{10.0}$	$61.9_{6.5}$	$50.4_{0.4}$	$65.6_{4.8}$	$67.8_{5.6}$	$32.4_{4.6}$	$59.7_{2.5}$
	VanillaICL	2.7B	$75.0_{7.5}$	$65.4_{2.9}$	$71.4_{13.3}$	$49.8_{1.8}$	$65.6_{2.8}$	$60.0_{2.1}$	$32.1_{5.4}$	$59.9_{1.1}$
	ExtraLM	770M	$68.9_{11.3}$	$63.9_{6.4}$	$60.3_{6.4}$	$51.2_{1.7}$	$64.5_{1.5}$	$63.7_{5.3}$	$27.8_{5.1}$	$57.2_{2.1}$
	Self-Sup	770M	$55.0_{7.4}$	$50.3_{0.6}$	$59.7_{3.5}$	$52.2_{2.0}$	$50.3_{7.0}$	$63.4_{7.1}$	$28.8_{3.3}$	$51.4_{2.2}$
	MetaICL	770M	$69.8_{4.0}$	$63.5_{4.6}$	$65.6_{7.5}$	$\mathbf{57.6}_{2.3}$	$66.3_{2.4}$	$65.2_{3.0}$	$31.7_{2.1}$	$60.0_{1.5}$
	PICL	770M	$\mathbf{79.7}_{8.6}$	$\mathbf{66.8}_{7.4}$	$\mathbf{81.0}_{1.3}$	$54.5_{1.8}$	$\mathbf{67.7}_{3.4}$	$\mathbf{69.6}_{4.3}$	$\mathbf{34.8}_{4.0}$	$\mathbf{64.4}_{1.6}$
8-shot	VanillaICL	770M	$68.7_{6.0}$	$66.6_{9.8}$	$60.2_{5.5}$	$51.8_{1.6}$	$60.2_{5.6}$	$68.8_{3.2}$	$31.4_{3.8}$	$58.2_{2.9}$
	VanillaICL	1.5B	$72.1_{12.6}$	$63.4_{6.5}$	$63.3_{5.4}$	$52.7_{2.8}$	$54.2_{8.4}$	$\mathbf{70.4}_{5.7}$	$33.5_{3.3}$	$58.6_{2.5}$
	VanillaICL	2.7B	$71.0_{11.6}$	$65.2_{4.0}$	$70.4_{6.3}$	$51.3_{2.0}$	$63.1_{2.4}$	$69.6_{4.0}$	$\mathbf{34.1}_{2.8}$	$60.6_{3.2}$
	ExtraLM	770M	$69.7_{3.4}$	$65.2_{6.5}$	$63.6_{6.0}$	$52.6_{1.6}$	$58.9_{7.0}$	$69.6_{3.8}$	$32.2_{4.7}$	$58.8_{1.6}$
	Self-Sup	770M	$61.4_{6.5}$	$54.3_{4.5}$	$73.8_{8.1}$	$53.0_{2.4}$	$52.1_{3.8}$	$63.0_{6.9}$	$33.7_{1.8}$	$55.9_{2.1}$
	MetaICL	770M	$73.6_{6.2}$	$67.2_{8.8}$	$70.1_{5.6}$	$\mathbf{53.6}_{2.1}$	$56.1_{0.7}$	$65.8_{4.1}$	$33.7_{4.7}$	$60.0_{2.2}$
	PICL	770M	$\mathbf{78.0}_{10.6}$	$\mathbf{69.3}_{9.5}$	$\mathbf{77.5}_{5.0}$	$53.0_{1.6}$	$\mathbf{64.7}_{4.4}$	$\mathbf{70.4}_{2.1}$	$\mathbf{34.1}_{3.8}$	$\mathbf{63.9}_{1.3}$

Table 1: Main results of few-shot text classification. We report the average accuracy scores and the standard deviations across 5 random seeds for selecting demonstrations. We use GPT2-Large (770M), GPT2-xLarge (1.5B), and GPT-*Neo* (2.7B) for VanillaICL. The best scores on each dataset under 4 or 8 evaluation shots are in **boldface**.

apply multiple prompts from PromptSource (Bach et al., 2022) to one example and use 320 prompts in all. We use the in-batch negative trick (Chen et al., 2020) to compute the contrastive loss. We set the learning rate to 5×10^{-5}, the batch size to 64, and construct 4 hard negatives for each instance. The encoder is trained from RoBERTa$_{\text{Base}}$ for 1 epoch.

Language Model We test PICL based on the 770M GPT2-Large (Radford et al., 2019) unless otherwise specified. Results on larger models can be found in Appendix E.1. To save computational resources, we train the model from its pre-trained checkpoints. We also test the VanillaICL performance of larger models, including GPT2-xLarge (Radford et al., 2019) (1.5B) and GPT-*Neo* (Black et al., 2021) (2.7B) for reference.

Pre-Training We set the maximum learning rate to 1×10^{-6} and use the "inverse square root" scheduler (Vaswani et al., 2017) with 1000 steps warmup. The model sees 131K tokens in a step and is pre-trained for 100K steps. It takes less than a day to finish pre-training on 64 V100 32G GPUs.

4 Results

4.1 Few-Shot Text Classification

Table 1 shows the results of few-shot text classification, from which we have 3 observations.

First, among the baselines with 770M parameters, simply further training the model on our corpus with language modeling improves the performance (ExtraLM). This is likely due to the higher domain diversity of our corpus. MetaICL is helpful on most datasets, which verifies the effectiveness of meta-training for ICL. Self-Sup fails to bring benefits on most datasets against VanillaICL, probably because the constrained label space of the Classification training task (only contains "True" and "False") brings bias to the model's output. This emphasizes the importance of using training objectives with little bias.

Second, we observe that the PICL-trained model outperforms the baselines with the same model sizes by a large margin on most datasets across different shots, verifying the effectiveness of PICL. An exception is RTE, where MetaICL performs the best. We speculate the reason is that some training tasks of MetaICL share the same label space with RTE ("Yes"/"No"), such as paraphrase identification. Min et al. (2022c) has shown that the label space plays a vital role in ICL, which explains the good performance of MetaICL on RTE.

Thrid, comparing models across different sizes, we find that increasing the model parameters boosts the performance, but PICL enables the 770M model to beat a 2.7B counterpart. This indicates that the ICL ability can be enhanced not only through scaling up the parameters. Improving the structure of the pre-training data is also beneficial. In Appendix E.1, we can see that PICL is also effective when applied to a 1.5B model.

4.2 Instruction Following

The results on SUPER-NATURALINSTRUCTIONS are shown in Table 2. We can see that PICL

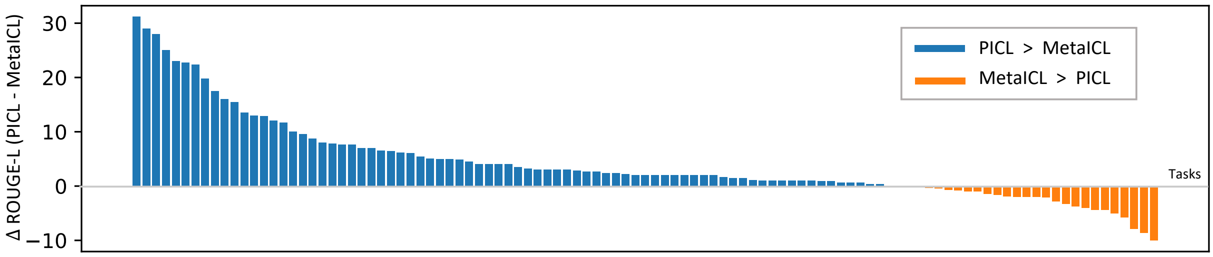


Figure 3: Comparison between PICL and MetaICL on SUPER-NATURALINSTRUCTIONS (Wang et al., 2022). Each bar represents an evaluation task. The y-axis means the ROUGE-L score difference between the two methods.

Model	Param.	ROUGE-L
VanillaICL	770M	34.3
VanillaICL	1.5B	34.9
VanillaICL	2.7B	37.3
ExtraLM	770M	34.6
Self-Sup	770M	30.5
MetaICL	770M	35.3
PICL	770M	**37.6**

Table 2: Results of instruction following on SUPER-NATURALINSTRUCTIONS. We report the average ROUGE-L score across all 105 evaluation tasks.

achieves higher overall instruction following performance than the baselines, outperforming a larger model with about 4x parameters.

In Figure 3, we compare the per-task performance of PICL and MetaICL because they share the most similar setting where human-annotated downstream datasets are used. We observe that PICL outperforms MetaICL on about 3/4 of evaluation tasks, indicating that compared to fine-tuning directly on downstream tasks, pre-training on intrinsic tasks constructed from the general plain-text corpus brings better ICL ability and ensures higher generalization performance across a broad range of tasks (see Appendix E.2 for more details).

Most tasks where MetaICL beats PICL belong to text classification whose output spaces are “Yes/No” or “True/False”. This matches the second observation in Section 4.1, where MetaICL predicts “Yes/No” well because of training on tasks that share the same label spaces. On the other hand, PICL performs much better on text generation, or tasks whose output spaces share the same semantics with “Yes/No” but use label words not in the training tasks of MetaICL (e.g., “Correct/Wrong”). This indicates that direct training on downstream datasets causes overfitting to specific labels. There are also tasks where PICL performs similarly to MetaICL, such as reasoning and word analogy. We notice that the improvements of PICL and MetaICL on these tasks are also marginal against VanillaICL probably because these tasks rely more on the “in-weights learning” ability (Chan et al., 2022), rather than in-context learning.

4.3 Analysis

Effect of Retriever We compare different approaches to retrieve paragraphs and test the final model performance. We try randomly selecting paragraphs (Random), retrieving using the non-parametric approach (BM25), encoding each paragraph with the original pre-trained encoder as it is (RoBERTa), or using the encoder for sentence similarity (Reimers and Gurevych, 2019) (SRoBERTa). We also study different numbers of hard negatives (0, 1, 4) and downstream tasks (7, 24, 37) to train the task-semantics encoder in PICL. From the results in Table 3, we can see that all retrieval methods except Random bring improvements against VanillaICL on both text classification and instruction following settings, indicating that improving the coherence of the paragraphs in the pre-training data benefits ICL. Using the task-semantics encoder in PICL achieves the best performance, showing the importance of retrieving paragraphs based on task semantics rather than word overlap or sentence meanings. Comparing different settings to train the task-semantics encoder, we observe that increasing the number of hard negatives and training tasks improves the final performance. This is in line with previous works (Karpukhin et al., 2020; Chen et al., 2020; He et al., 2020) that more challenging hard negatives benefit contrastive learning.

Effect of Demonstration Numbers Training with PICL brings two benefits: (1) PLMs learn a format where demonstrations from the same task are concatenated as the prefix, which is beneficial when the model is evaluated under the same number of demonstrations. (2) PLMs learn a better

Retriever	$n_{\text{HardNeg.}}$	n_{Tasks}	CLS Accuracy	SUP-NI ROUGE-L
VanillaICL	-	-	55.2	34.3
Random	-	-	56.7	29.3
BM25	-	-	59.2	34.5
RoBERTa	-	-	58.7	34.6
SRoBERTa	-	-	59.0	35.0
PICL	0	37	62.2	36.4
	1	37	63.1	36.5
	4	7	61.6	35.4
	4	24	63.4	36.6
	4	37	**64.4**	**37.6**

Table 3: Comparison of different retrievers. $n_{\text{HardNeg.}}$ and n_{Tasks} means the number of hard negatives and downstream tasks to train the task-semantics encoder in PICL. “CLS Accuracy” means the average accuracy scores on text classification tasks. “SUP-NI ROUGE-L” means the average ROUGE-L scores across the tasks in SUPER-NATURALINSTRUCTIONS.

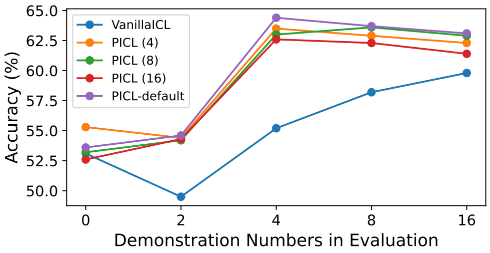
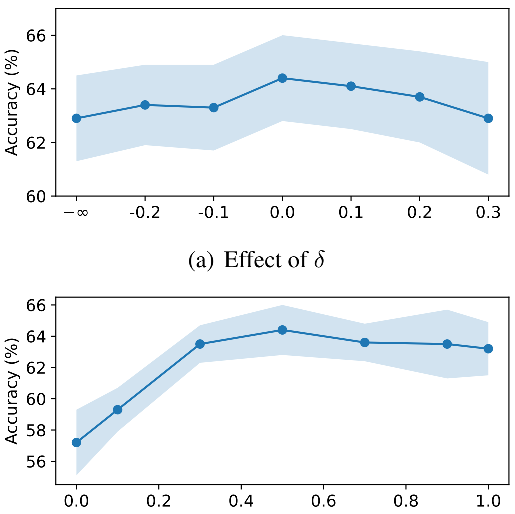


Figure 4: Average text classification accuracy when the pre-training instances contain different demonstration numbers in PICL (the number in the brackets). “PICL-default” means using a mixture of demonstration numbers as in previous experiments.

ability to infer and perform tasks from the context, even when the demonstration numbers in evaluation and pre-training do not match. To differentiate these effects, we conduct pre-training on instances containing only 4, 8, or 16 demonstrations and test the trained models under different text classification shots. Results in Figure 4 show that when pre-trained with different demonstration numbers, the models generalize well to unseen demonstration numbers in evaluation, achieving similar performance with the default setup where the model sees various demonstration numbers in pre-training (PICL-default). This indicates that the models learn more than the input formats in PICL.

Effect of Filtering We try different threshold values δ for filtering and report the scores on text classification tasks in Figure 5(a), while controlling the sizes of the constructed pre-training data the same. We find that $\delta = 0$ yields the best performance,



(a) Effect of δ

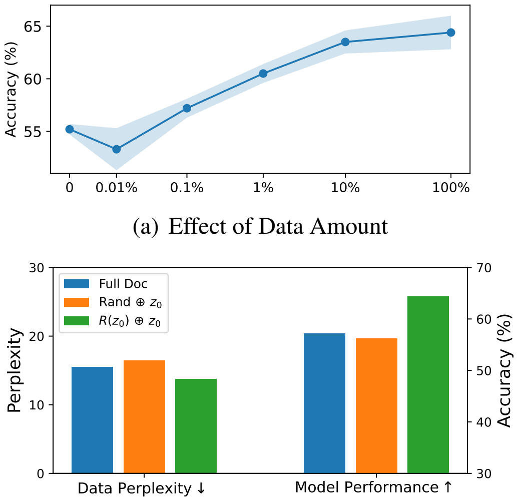
(b) Effect of α

Figure 5: Hyper-parameter analysis. **(a)**: average 4-shot text classification accuracy as a function of δ for filtering. $-\infty$ means we do not conduct filtering. **(b)**: average 4-shot text classification accuracy as a function of α to control the proportion of the full-documents.

which means we retain an instance if and only if the perplexity of individual paragraphs is higher than that of the concatenated sequence (Equation 3). For lower δ, the pre-training data contain too many uninformative instances for ICL. For larger δ, we speculate that the filtering process relies on the original GPT2-Large too much. Since we also pre-train based on GPT2-Large, the filtering process reduces the signals in the constructed data beyond the base model’s ability.

Effect of Full Documents In Figure 5(b), we report the model performance on text classification tasks when using different choices of α, which controls the proportion of the full-document data. We find that balancing the constructed and full-document data performs the best ($\alpha = 0.5$). When α is too large, the model is trained mostly on our constructed data and overfits its bias inevitably introduced by the task-semantics encoder in the data construction process. When α is too small, our method degenerates into ExtraLM.

Effect of Data Amount We study the size effect of the corpus used to construct the pre-training data in PICL and report the performance on text classification tasks in Figure 6(a). We conduct the data construction on 0.01%, 0.1%, 1%, and 10% of the original 80M paragraphs (100%) and pre-train models for at most 100K steps until the validation



(a) Effect of Data Amount

(b) Data Comparison

Figure 6: Data analysis. **(a)**: the average 4-shot text classification accuracies when constructing data using different proportions of the original corpus. **(b)**: perplexity of full-document data (Full Doc), random retrieved data (Rand $\oplus$ z_0) and PICL data ($R(z_0) \oplus z_0$) based on GPT-J (6B) and the corresponding model performance.

losses begin to increase. From the results, we conclude that when the corpus is small, pre-training with the constructed data hurts the performance because the search library is too small to find paragraphs sharing the same intrinsic tasks. Training on small data for multiple epochs also causes overfitting. When the corpus contains more than 80K paragraphs (0.1%), adding more data constantly improves the performance, which is consistent with the scaling law (Kaplan et al., 2020).

Data Comparison We compare the usefulness of different pre-training data to enhance the ICL ability. In addition to the final model performance, we borrow the thoughts for designing the filtering criterion in Section 2.2 to measure the usefulness of a pre-training instance by computing the perplexity using a reference large PLM: GPT-J (Wang and Komatsuzaki, 2021) with 6B parameters. Lower perplexity means the correlation within the instance is higher and is intuitively more helpful for enhancing the ICL ability. In Figure 6(b), we show the perplexity and the final model performance of 3 pre-training data: original full documents before being split into paragraphs (Full Doc), concatenation of randomly selected paragraphs (Rand $\oplus$ z_0), and the concatenated same-intrinsic-task paragraphs gathered using the retrieval method in PICL *before filtering* ($R(z_0) \oplus z_0$). We can see that the data constructed by retrieval has much lower perplexity and correspondingly yields higher accuracy scores, which verifies its usefulness. In Appendix F, we present several examples of the retrieved paragraphs and the corresponding intrinsic tasks.

5 Related Work

In-Context Learning Recently, in-context learning (ICL), where models perform tasks simply conditioning on instructions or the concatenation of examples in the context (Brown et al., 2020), has been found promising for using PLMs in various application scenarios. To this end, there emerge many works to improve the ICL performance by calibrating the model predictions (Zhao et al., 2021; Han et al., 2022; Holtzman et al., 2021; Min et al., 2022a), selecting or reordering demonstrations (Rubin et al., 2022; Liu et al., 2022; Lu et al., 2022), designing pre-training tasks (Chen et al., 2022a), and breaking the context length limits (Hao et al., 2022). However, the underlying mechanism of ICL is poorly understood (Min et al., 2022c). Therefore, some works propose mathematical frameworks to reveal how ICL works (Xie et al., 2021; Olsson et al., 2022; Elhage et al., 2021), or investigate the pre-training data to explain ICL's good performance (Chan et al., 2022; Shin et al., 2022).

Multi-Task Fine-tuning for Cross-Task Generalization Fine-tuning PLMs on a large collection of downstream tasks enables generalization to unseen tasks under zero-shot (Wei et al., 2022; Sanh et al., 2022; Ouyang et al., 2022; Chung et al., 2022) and few-shot (Min et al., 2022b; Chen et al., 2022b; Mishra et al., 2022; Garg et al., 2022) scenarios. However, the performance of multi-task fine-tuning is largely restricted by the diversity of the annotated training tasks (Gu et al., 2022b), which requires massive human efforts to scale up. In addition, direct training on downstream tasks easily brings undesired bias. In this work, we propose to meta-train the model with the intrinsic tasks automatically collected from the large-scale general corpus, which is easier to scale up and introduces little bias.

Pre-training Data Programming The conventional pre-training paradigm trains the model on plain-text corpora with the language modeling objective (Radford et al., 2018, 2019; Brown et al., 2020). Recently works have found that carefully designed pre-training instances can further boost specific abilities like prompt adaption (Gu et al., 2022a), reasoning (Razeghi et al., 2022), or sentence representation (Levine et al., 2021). Our work studies constructing pre-training instances to improve the PLM's ICL ability while still maintaining its generalization on various NLP tasks.

6 Conclusion

This paper presents PICL, a framework that exploits the in-context learning ability of PLMs by pre-training models on concatenations of text paragraphs sharing the same "intrinsic tasks" gathered from the large-scale general corpus. In PICL, models learn to perform various intrinsic tasks conditioning on their context while preserving their generalization due to the little bias of the pre-training data. Extensive experiments show that PICL improves the ICL performance on various datasets against several baselines, enabling a 770 M model to outperform a larger model with about 4x parameters while maintaining good generalization across a wide range of tasks. For future work, we would like to consider adding human instructions to our pre-training framework to enhance more abilities of PLMs like zero-shot instruction following.

Limitations

One limitation of our paper is that the exact distribution of the intrinsic tasks in the original corpus and the constructed data is still unknown. Knowing the distribution can offer a better interpretation of the effectiveness of PICL, even of the strong performance of large language models. Besides, although we can find many constructed instances that share obvious intrinsic tasks (see Appendix F), there still exist some instances where the intrinsic tasks are hard to identify. How to better evaluate the contribution of these instances to the ICL ability or designing better filtering approaches to select more informative data for ICL is worth studying.

Our task-semantics encoder inevitably contains some bias because it is trained on downstream datasets, although we have tried to ensure a large number and diversity of the dataset collection. However, the final language model is pre-trained on the general corpus, and we add the full document loss, which eliminates the bias to some extent.

Regarding computing power, we acknowledge that our framework takes relatively large training resources in the retrieval and pre-training process. Therefore, we did not conduct experiments based on extra-large language models.

Acknowledgements

This work was supported by the NSFC projects (Key project with No. 61936010). This work was also supported by the Guoqiang Institute of Tsinghua University, with Grant No. 2020GQG0005.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of ACL*.

Stephen H Bach, Victor Sanh, Zheng-Xin Yong, Albert Webson, Colin Raffel, Nihal V Nayak, et al. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of ACL (demo)*.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. In *Proceedings of ICLR*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of AAAI*.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

Stephanie CY Chan, Adam Santoro, Andrew Kyle Lampinen, Jane X Wang, Aaditya K Singh, Pierre Harvey Richemond, James McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. In *NeurIPS*.

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022a. Improving in-context few-shot learning via self-supervised training. In *Proceedings of NAACL*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of ICML*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. In *Proceedings of ICLR*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022b. Meta-learning via language model in-context tuning. In *Proceedings of ACL*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. *Machine Learning Challenges: Evaluating Predictive Uncertainty*.

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of EMNLP*.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The CommitmentBank: Investigating projection in naturally occurring discourse. In *Proceedings of Sinn und Bedeutung 23*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Manaal Faruqui and Dipanjan Das. 2018. Identifying well-formed natural language questions. In *Proceedings of EMNLP*.

Wikimedia Foundation. 2022. Wikimedia downloads.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. In *Proceedings of NeurIPS*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization (EMNLP2019)*.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. 2019. Openwebtext corpus.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022a. PPT: Pre-trained prompt tuning for few-shor learning. In *Proceedings of ACL*.

Yuxian Gu, Pei Ke, Xiaoyan Zhu, and Minlie Huang. 2022b. Learning instructions with unlabeled data for zero-shot cross-task generalization. In *Proceedings of EMNLP*.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, et al. 2021. Pre-trained models: Past, present and future. *AI Open*.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2022. Prototypical calibration for few-shot learning of language models. *arXiv preprint arXiv:2205.10183*.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of CVPR*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of EMNLP*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of EMNLP*.

Kelvin Jiang, Dekun Wu, and Hui Jiang. 2019. FreebaseQA: A new factoid QA data set matching trivia-style question-answer pairs with Freebase. In *Proceedings of NAACL-HLT*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Proceedings of NeurIPS*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of AAAI*.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In *Proceedings of AAAI*.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. In *Proceedings of EMNLP*.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, and Christian Bizer. 2014. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*.

Yoav Levine, Noam Wies, Daniel Jannai, Dan Navon, Yedid Hoshen, and Amnon Shashua. 2021. The inductive bias of in-context learning: Rethinking pretraining example design. In *Proceedings of ICLR*.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of EMNLP*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of Text Summarization Branches Out (ACL 2004)*.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering (EMNLP 2019)*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures (ACL 2022)*.

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*.

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding indirect answers. In *Proceedings of EMNLP*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of ACL*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of ACL*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022b. MetaICL: Learning to learn in context. In *Proceedings of NAACL*.

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

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of ACL*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of EMNLP*.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, et al. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Proceedings of NeurIPS*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*.

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*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Technical report*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Technical report*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *arXiv preprint arXiv:2202.07206*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*.

Yuanhang Ren, Ye Du, and Di Wang. 2018. Tackling adversarial examples in QA via answer sentence selection. In *Proceedings of the Workshop on Machine Reading for Question Answering (ACL 2018)*.

Anna Rogers, Olga Kovaleva, Matthew Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of AAAI*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of ACL*.

Amrita Saha, Rahul Aralikatte, Mitesh M. Khapra, and Karthik Sankaranarayanan. 2018. DuoRC: Towards complex language understanding with paraphrased reading comprehension. In *Proceedings of ACL*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *Proceedings of ICLR*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of EMNLP*.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset. *arXiv preprint arXiv:1907.01041*.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and Nako Sung. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. In *Proceedings of NAACL-HLT*.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining discourse markers for unsupervised sentence representation learning. In *Proceedings of NAACL-HLT*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Kai Sun, Dian Yu, Jianshu Chen, Dong Yu, Yejin Choi, and Claire Cardie. 2019. DREAM: A challenge data set and models for dialogue-based reading comprehension. *TACL*.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2019a. QUAREL: A dataset and models for answering questions about qualitative relationships. In *Proceedings AAAI*.

Oyvind Tafjord, Matt Gardner, Kevin Lin, and Peter Clark. 2019b. QuaRTz: An open-domain dataset of qualitative relationship questions. In *Proceedings of EMNLP*.

Niket Tandon, Bhavana Dalvi, Keisuke Sakaguchi, Peter Clark, and Antoine Bosselut. 2019. WIQA: A dataset for "what if..." reasoning over procedural text. In *Proceedings of EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of ACL*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks. In *Proceedings of EMNLP*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *Proceedings of ICLR*.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text (ACL 2017)*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *TACL*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. In *Proceedings of ICLR*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of EMNLP*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of EMNLP*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of ACL*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of NeurIPS*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of NAACL-HLT*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of ICML*.

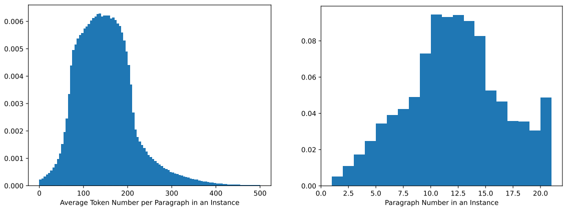
Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of ICCV*.

A Details of the Pre-training Corpus

This section presents details of the data processing of the pre-training corpus and its statistics.

Data Processing Our pre-training corpus is a merge of OpenWebText (Gokaslan et al., 2019), WikiCorpus (Foundation, 2022), and BookCorpus (Zhu et al., 2015), downloaded from the HuggingFace datasets repository[1]. We first split each document in the corpus into paragraphs with "\n". To avoid training with too short paragraphs, we concatenate a paragraph with previous paragraphs if the token number after concatenation is lower than 128. We also exclude paragraphs longer than 500 tokens because they are not likely to fit into an instance with more than 1 paragraph. The filtering process in Section 2.3 drops about 24% instances. The licenses of all corpora allow for scientific research.

Statistics We plot the distribution of the mean paragraph length per instance in Figure 7(a) and the distribution of the paragraph number per instance in Figure 7(b). The average paragraph length is 150.0, and the average paragraph number in an instance is 11.7. We can see that the model sees various demonstration numbers in PICL pre-training.



(a) Paragraph Length (b) Paragraph Number

Figure 7: Pre-training data statistics. **(a)**: the distribution of the average paragraph length per instance. **(b)**: the distribution of the paragraph number per instance.

Stage	Name	Values
Retrieve	learning rate	1e-4, **5e-5**, 1e-5
	batch size	16, 32, **64**
	hard negatives	0, 1, **4**
Pre-train	learning rate	1e-5, 5e-6, **1e-6**, 2e-7
	batch size	64, **128**, 256, 512
	warmup	0, **1000**, 5000

Table 4: Searching intervals of hyper-parameters.

B Details of the Evaluation Data

B.1 Few-shot Text Classification

The details of each text classification dataset and the corresponding prompt in evaluation are listed in Table 6. All datasets are downloaded from the HuggingFace datasets repository[1]. We simplify the evaluation prompts as much as possible to reduce the effect of prompt engineering. Following previous works (Brown et al., 2020; Sanh et al., 2022), the model is evaluated by the *ranking score* strategy, where we compare the perplexity of each classification label under the model and choose the label with the lowest perplexity. The licenses of all datasets allow for scientific research.

B.2 Instruction Following

The original test split of the benchmark SuperNaturalInstructions (Wang et al., 2022) contains 119 tasks. We exclude tasks that appear in the training tasks of the task-semantics encoder or whose input length is too long to fit in the context of our model. Our final evaluation includes 105 tasks. A full list of the tasks is shown in Table 8. We use the same template to combine few-shot examples with task instructions as Wang et al. (2022). The license of this benchmark is Apache License 2.0.

C Details of the Downstream Training Data

The downstream datasets we use to train the task-semantics encoder are a merge of the training data used in (Sanh et al., 2022) and the HR→LR setting in (Min et al., 2022b). All datasets are downloaded from the HuggingFace datasets repository [1] and all prompts come from the PromptSource library (Bach et al., 2022)[2]. We exclude datasets from the sentiment classification task, the topic classification task, and the natural language inference task because they are included in our text classification evaluation. We finally get a collection of 37 datasets, as listed in Table 5. The licenses of all datasets allow for scientific research.

D More Experimental Details

All model checkpoints we used come from the HuggingFace models repository[3]. The searching interval of each hyper-parameter is listed in Table 4.

[1] https://huggingface.co/datasets/

[2] https://github.com/bigscience-workshop/promptsource

[3] https://huggingface.co/models

E More Results

E.1 Results on Larger Base Model

We test PICL based on the GPT2-xLarge (Radford et al., 2019) with 1.5B parameters. From the results in Figure 7, we can see that PICL is also applicable to larger models, outperforming the baselines based on the same-sized model on most datasets.

E.2 Instruction Following

We present the performance comparison between PICL and MetaICL per evaluation task in Figure 8. PICL outperforms MetaICL on 77 / 105 tasks, indicating that PICL ensures the better generalization of the trained model. The name of each task is also listed in Figure 8. We can see that the top three tasks where MetaICL performs the best are:

- `doqa_movies_isanswerable`,
- `glue_entailment_classification`,
- `tweetqa_classification`,

which are all “Yes/No” classification tasks. The top three tasks where PICL performs the best are:

- `winogrande_question_modification_object`,
- `plausible_result_generation`,
- `winowhy_reason_plausibility_detection` ,

which are text generation, text generation, and “Correct/Wrong” classification tasks respectively. The four tasks where PICL and MetaICL have the same scores are:

- `bard_analogical_reasoning_containers`,
- `copa_commonsense_cause_effect`,
- `winogrande_answer_generation`,
- `bard_analogical_reasoning_trash_or_treasure`,

which belong to commonsense reasoning and word analogy tasks.

F Case Studies

In Table 9 and 10, we present several cases of the retrieved paragraphs and the corresponding intrinsic tasks. We can see that there exists a large range of intrinsic tasks in the constructed data and many of them do not appear in the training data of the task-semantics encoder, which shows the generalization of the encoder.

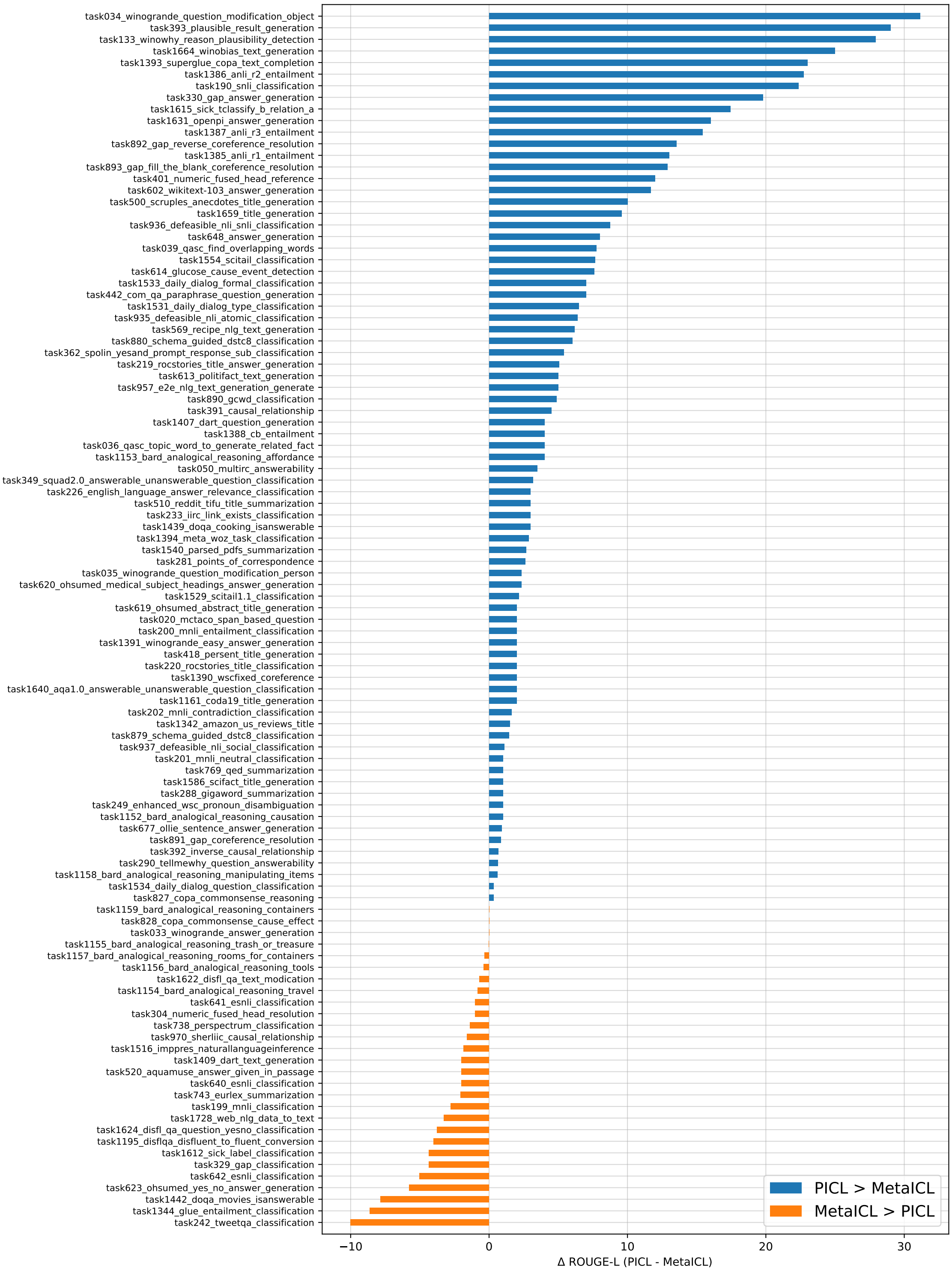


Figure 8: Per-task results of the comparison between PICL and MetaICL.

COS-E (Aggarwal et al., 2021)
QuaRTz (Tafjord et al., 2019b)
CosmosQA (Huang et al., 2019)
SciQ (Welbl et al., 2017)
Quoref (Dasigi et al., 2019)
Hotpot-QA (Yang et al., 2018)
Wiki-Bio (Lebret et al., 2016)
MRPC (Dolan and Brockett, 2005)
art (Bhagavatula et al., 2019)
Freebase_QA (Jiang et al., 2019)
liar (Wang, 2017)
swag (Zellers et al., 2018)
DBpedia (Lehmann et al., 2014)

DREAM (Sun et al., 2019)
Social-IQA (Sap et al., 2019)
QASC (Khot et al., 2020)
Wiki-Hop (Welbl et al., 2018)
ROPES (Lin et al., 2019)
Wiki-QA (Yang et al., 2015)
SAMSum (Gliwa et al., 2019)
PAWS (Zhang et al., 2019)
circa (Louis et al., 2020)
google_wellformed_query (Faruqui and Das, 2018)
piqa (Bisk et al., 2020)
tab_fact (Chen et al., 2019)

QuAIL (Rogers et al., 2020)
WiQA (Tandon et al., 2019)
QUAREL (Tafjord et al., 2019a)
Adversarial-QA (Ren et al., 2018)
DuoRC (Saha et al., 2018)
Common-Gen (Lin et al., 2020)
XSum (Narayan et al., 2018)
QQP (Sharma et al., 2019)
discovery (Sileo et al., 2019)
HellaSwag (Zellers et al., 2019)
scitail (Khot et al., 2018)
yahoo_answer_topics[4]

Table 5: The references of the 37 downstream datasets used to train the task-semantics encoder.

Dataset	Task	Prompt	Label Space
SST2	Sent. CLS	Sentence: {sentence} Label: {label}	Negative / Positive
SST5	Sent. CLS	Sentence: {sentence} Label: {label}	Terrible / Bad / Neutral / Good / Great
MR	Sent. CLS	Sentence: {sentence} Label: {label}	Negative / Positive
RTE	NLI	Passage: {premise} Question: {hypothesis} Answer: {label}	Yes / No
CB	NLI	Passage: {premise} Question: {hypothesis} Answer: {label}	Yes / No / Maybe
SUBJ	Subj. CLS	Input: {text} Type: {label}	Objective / Subjective
AgNews	Topic CLS	Sentence: {text} Label: {label}	World politics / Sports / Business / Science and technology

Table 6: Details of the text classification datasets. "Sent. CLS" stands for "Sentiment Classification". "NLI" stands for "Natural Language Inference". "Subj. CLS" stands for "Subjectivity Classification". "Topic CLS" stands for "Topic Classification".

Shot	Method	SST2	SUBJ	MR	RTE	AgNews	CB	SST5	Average
GPT-xlarge	VanillaICL	$74.9_{9.7}$	$65.2_{10.0}$	$61.9_{6.5}$	$50.4_{0.4}$	$65.6_{4.8}$	$67.8_{5.6}$	$32.4_{4.6}$	$59.7_{2.4}$
	MetalICL	$71.1_{2.0}$	$64.9_{7.6}$	$66.8_{6.3}$	$\mathbf{60.0}_{2.8}$	$66.2_{5.4}$	$64.4_{1.6}$	$34.6_{3.7}$	$61.2_{1.3}$
	PICL	$\mathbf{86.9}_{2.8}$	$\mathbf{72.5}_{7.3}$	$\mathbf{76.2}_{4.6}$	$54.0_{2.7}$	$\mathbf{67.1}_{6.0}$	$\mathbf{70.0}_{4.6}$	$\mathbf{38.0}_{4.2}$	$\mathbf{66.4}_{1.6}$
GPT-*Neo*	VanillaICL	$75.0_{7.5}$	$65.4_{2.9}$	$71.4_{13.3}$	$49.8_{1.8}$	$65.6_{2.8}$	$60.0_{2.1}$	$32.1_{5.4}$	$59.9_{1.2}$
	MetalICL	$80.1_{5.8}$	$55.6_{9.5}$	$73.1_{9.0}$	$\mathbf{57.5}_{3.9}$	$64.2_{3.4}$	$\mathbf{65.5}_{6.4}$	$32.8_{4.7}$	$61.3_{1.4}$
	PICL	$\mathbf{86.4}_{1.0}$	$\mathbf{68.6}_{5.7}$	$\mathbf{83.6}_{2.4}$	$50.2_{0.7}$	$\mathbf{67.5}_{1.2}$	$63.1_{3.7}$	$\mathbf{35.7}_{3.6}$	$\mathbf{65.0}_{1.1}$

Table 7: 4-shot text classification results based on GPT2-XL (1.5B) and GPT-*Neo* (2.7B). We report the average accuracy scores and the standard deviations across 5 random seeds for selecting demonstrations. The best scores on each dataset are in **boldface**.

Task Category	List of Tasks	
	Evaluation Tasks (105)	
Coreference Resolution	task893_gap_fill_the_blank_coreference_resolution task648_answer_generation task891_gap_coreference_resolution task892_gap_reverse_coreference_resolution task1390_wscfixed_coreference task330_gap_answer_generation task249_enhanced_wsc_pronoun_disambiguation	task1664_winobias_text_generation task304_numeric_fused_head_resolution task033_winogrande_answer_generation task401_numeric_fused_head_reference task133_winowhy_reason_plausibility_detection task329_gap_classification task1391_winogrande_easy_answer_generation
Textual Entailment	task641_esnli_classification task202_mnli_contradiction_classification task1387_anli_r3_entailment task890_gcwd_classification task936_defeasible_nli_snli_classification task201_mnli_neutral_classification task1516_imppres_naturallanguageinference task970_sherliic_causal_relationship task935_defeasible_nli_atomic_classification task1388_cb_entailment task190_snli_classification task640_esnli_classification	task1529_scitail1.1_classification task1344_glue_entailment_classification task738_perspectrum_classification task1612_sick_label_classification task1386_anli_r2_entailment task1385_anli_r1_entailment task1615_sick_tclassify_b_relation_a task199_mnli_classification task937_defeasible_nli_social_classification task1554_scitail_classification task200_mnli_entailment_classification task642_esnli_classification
Cause Effect Classification	task1393_superglue_copa_text_completion task828_copa_commonsense_cause_effect task827_copa_commonsense_reasoning task392_inverse_causal_relationship	task391_causal_relationship task614_glucose_cause_event_detection task393_plausible_result_generation
Title Generation	task288_gigaword_summarization task619_ohsumed_abstract_title_generation task569_recipe_nlg_text_generation task602_wikitext-103_answer_generation task510_reddit_tifu_title_summarization task1342_amazon_us_reviews_title task220_rocstories_title_classification task219_rocstories_title_answer_generation	task1161_coda19_title_generation task500_scruples_anecdotes_title_generation task1586_scifact_title_generation task769_qed_summarization task743_eurlex_summarization task418_persent_title_generation task1659_title_generation task1540_parsed_pdfs_summarization
Dialogue Act Recognition	task880_schema_guided_dstc8_classification task1394_meta_woz_task_classification task1533_daily_dialog_formal_classification task1534_daily_dialog_question_classification	task1531_daily_dialog_type_classification task362_spolin_yesand_prompt_response_sub_classification task879_schema_guided_dstc8_classification
Answerability Classification	task1439_doqa_cooking_isanswerable task242_tweetqa_classification task233_iirc_link_exists_classification task520_aquamuse_answer_given_in_passage task050_multirc_answerability task1624_disfl_qa_question_yesno_classification	task1640_aqa1.0_answerable_unanswerable_question_classification task1442_doqa_movies_isanswerable task290_tellmewhy_question_answerability task226_english_language_answer_relevance_classification task349_squad2.0_answerable_unanswerable_question_classification task020_mctaco_span_based_question
Data to Text	task1728_web_nlg_data_to_text task1407_dart_question_generation task677_ollie_sentence_answer_generation	task1409_dart_text_generation task957_e2e_nlg_text_generation_generate task1631_openpi_answer_generation
Keyword Tagging	task036_qasc_topic_word_to_generate_related_fact task613_politifact_text_generation	task620_ohsumed_medical_subject_headings_answer_generation task623_ohsumed_yes_no_answer_generation
Word Analogy	task1159_bard_analogical_reasoning_containers task1152_bard_analogical_reasoning_causation task1156_bard_analogical_reasoning_tools task1153_bard_analogical_reasoning_affordance	task1154_bard_analogical_reasoning_travel task1155_bard_analogical_reasoning_trash_or_treasure task1157_bard_analogical_reasoning_rooms_for_containers task1158_bard_analogical_reasoning_manipulating_items
Overlap Extraction	task039_qasc_find_overlapping_words	task281_points_of_correspondence
Question Rewriting	task035_winogrande_question_modification_person task034_winogrande_question_modification_object task1622_disfl_qa_text_modication	task1195_disflqa_disfluent_to_fluent_conversion task442_com_qa_paraphrase_question_generation
	Excluded Tasks (14)	
	task1356_xlsum_title_generation task645_summarization task402_grailqa_paraphrase_generation task671_ambigqa_text_generation task1345_glue_qqp_question_paraprashing task232_iirc_link_number_classification task1562_zest_text_modification	task670_ambigqa_question_generation task760_msr_sqa_long_text_generation task1598_nyc_long_text_generation task121_zest_text_modification task1557_jfleg_answer_generation task1358_xlsum_title_generation task102_commongen_sentence_generation

Table 8: A full list of the evaluation tasks we use and the tasks we exclude from the original test split of SUPER-NATURALINSTRUCTIONS (Wang et al., 2022).

ID	Paragraphs	Intrinsic Task
1	• Marko Jovanovski Marko Jovanovski (born 24 July 1988) is a Macedonian professional footballer who plays as a goalkeeper for Akademija Pandev. • Andreas Paraskevas Andreas Paraskevas (; born 15 September 1998) is a Cypriot footballer who plays as a goalkeeper for Doxa Katokopias. • Evripidis Giakos Evripidis Giakos (; born 9 April 1991) is a Greek professional footballer who plays as an attacking midfielder for Super League 2 club AEL.	World Knowledge Completion
2	• The Hive scouting teams had been infiltrating our space for several weeks, sending three- and six-man teams in. In short, they were making me look bad on the home world. • And for good measure, Walker ordered the Wisconsin National Guard to prepare to intervene in case of any strike action by unions. In a word, Walker wants the destruction of organized labor in Wisconsin. • "Scientists began examining him... he was covered in tattoos consisting of lines and dots,... 80 percent of the points correspond to those used in acupuncture today." This means the Prince of Wales ought to start listening to scientists.	Intent Identification
3	• ln(x) $\approx$ π 2 M (1,2^2 - m / x) - m ln(2). {\displaystyle \ln(x)\approx {\frac {\pi }{2M(1,2^{2-m}/x)}}-m\ln(2).} • sin(x) + 1 3 sin (3 x) + 1 5 sin (5 x) + $\cdots$. {\displaystyle \sin(x)+{\frac {1}{3}}\sin(3x)+{\frac {1}{5}}\sin(5x)+\dotsb.} • c q (n) = Σ d $\mid$ q μ (q d) η d (n). {\displaystyle c_{q}(n)=\sum_{d\mid q}\mu \left({\frac{q}{d}}\right)\eta_{d}(n).}	Latex Equation Translation
4	• How did Japan stumble on for another nine years, borrowing trillions of yen and squandering those trillions on make-work bridges to nowhere and lavish social spending? Answer: its citizens self-funded its deficits by saving trillions and investing those trillions in government debt. • How did our country thrive without income taxes for 126 years? Answer: federal spending was significantly lower than it is today. In the early 1900s, government spending accounted for roughly 7% of our GDP; today, federal spending accounts for around 35% of our GDP. • What was Trump's biggest persuasion problem in the election? Answer: His opponents did a great job of framing him as some kind of Hitler.	Question Answering
5	• An isopycnal is a line of constant density. An isoheight or isohypse is a line of constant geopotential height on a constant pressure surface chart. Isohypse and isoheight are simply known as lines showing equal pressure on a map. Temperature and related subjects • Once theory is applied to a mechanical design, physical testing is often performed to verify calculated results. Structural analysis may be used in an office when designing parts, in the field to analyze failed parts, or in laboratories where parts might undergo controlled failure tests. Thermodynamics and thermo-science • Complex numbers often generalize concepts originally conceived in the real numbers. For example, the conjugate transpose generalizes the transpose, hermitian matrices generalize symmetric matrices, and unitary matrices generalize orthogonal matrices. In applied mathematics Control theory	Topic Classification
6	• (speaking to Elder Fortie): Is this something you always wanted to do? ELDER FORTIE: Nope. It's not. SEVERSON: So why are you here? ELDER FORTIE: Because the idea of having an empty seat in heaven troubles me. SEVERSON: Sister Waymith is from Sweet, Idaho. • (speaking to Steve Allen): Are there any countries in particular that you're really zeroing in on, you'd really like to make some inroads? ALLEN: Yeah, the United States of America, North America. We'd like to make more inroads here. SEVERSON: Inroads like the church has made south of the border. Mexico, in particular, has been fertile ground for Mormon missionaries. • (to Elder Russell): Why are you learning Mandarin if you're going to Canada? ELDER RUSSELL: I guess there's a sizable population up there. I mean, everyone deserves to hear our message, so we'll go worldwide wherever they are. SEVERSON: This group is leaving soon for Ukraine. First, they had to be considered worthy of serving a mission.	Dialogue in a Script

Table 9: Cases of the retrieved paragraphs and the corresponding intrinsic tasks.

ID	Paragraphs	Intrinsic Task

7	• Now we can log into our Twilio account and set the Message Request URL to our sms route via ngrok: Try the app out by texting into your new Twilio number and you'll get the response back. Displaying Our Messages We're now passing our message to the arduino. The next step is to write the code that examines that message and displays it on our LCD. Let's lay the foundation for our app: # include < Wire. h > # include "rgb_lcd.h" rgb_lcd lcd ; void setup () { Serial. begin (9600); // set up the LCD's number of columns and rows: lcd. begin (16, 2); lcd. setCursor (0, 1); // Print a message to the LCD. lcd. print ("Ricky's Pager"); delay (1000); } void loop () { }
	• Now we're ready to track allocations. The first step is to "hijack" our 3 memory functions we defined in the first part (lines 4, 11 and 17): void* _Malloc(tU32 Size, tU32 AllocType, const tChar* Desc, const tChar* File, tU32 Line) { void* Result = malloc(Size); RegisterAlloc(Result, Size, AllocType, Desc, File, Line); return Result; } void* _Realloc(void* Ptr, tU32 Size, const tChar* File, tU32 Line) { void* Result = realloc(Ptr, Size); UpdateAlloc(Ptr, Result, Size, File, Line); return Result; } void _Free(void* Ptr) { UnregisterAlloc(Ptr); return free(Ptr); }
	• Here we use the gulp.src API to specify our input files. One thing to note is that we need to specify a reporter for JSHint. I'm using the default reporter, which should be fine for most people. More on this can be found on the JSHint website. Compress Images Next, we'll set up image compression: gulp. task ('images', function () { return gulp. src ('src/images/**/*'). pipe (imagemin ({ optimizationLevel : 3, progressive : true, interlaced : true })). pipe (gulp. dest ('dist/assets/img')). pipe (notify ({ message : 'Images task complete' })); });
		Code Generation

8	• Note: GP = Games played; W = Wins; L = Losses; T = Ties; OTL = Overtime loss; • SOL = Shootout loss; GF = Goals for; GA = Goals against; Pts = Points National Conference • ERA = Earned run average; SO = Strikeouts; +/- = Plus/Minus; PIM = Penalty minutes; GS = Games Started;	Word Abbriviation
9	• He strode toward her, barely slowly when he reached her. One arm slid around her waist and the other along her shoulders. She'd barely registered his touch before his mouth descended upon hers. • He lifted his head and stared at her. His face paled, and for the first time she noticed a spattering of orange freckles on his nose and across his cheekbones. He didn't speak. Just stared. • He continued rocking her gently, steadily. Her body's tremors calmed and her sobs quietened. He removed his white handkerchief from his trouser pocket, wiping her face. She didn't look at him and kept her eyes lowered.	Vivid Description
10	• Even the sugary cereals, they said, are of nutritional value because they contain vitamins and minerals. Research shows that 40 percent of U.S. children consume their milk via cereal, said Sutherland of Kellogg. General Mills cites data from the Journal of the American Dietetic Association that says people who frequently eat cereal, including kids who eat sweetened ones, tend to have healthier body weights than those that don't. • Lead's toxicity has long been known, and most of the uses that led to human exposure, like the manufacture of lead paint, have been banned for decades. Lead ammunition consumed only about 3 percent of the 6.4 million tons of lead used worldwide in 2000, according to a 2003 report by the Nordic Council of Ministers. • One reason Tesla has pushed the technology so aggressively is that its battery packs store more than three times the energy of its competitors' electric-car batteries. As a result, they require more power to charge quickly, says Arindam Maitra, a senior project manager at the Electric Power Research Institute.	Scientific Evidence Generation

Table 10: Cases of the retrieved paragraphs and the corresponding intrinsic tasks.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The section named "Limitations"

☑ A2. Did you discuss any potential risks of your work?
The section named "Limitations"

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 3, Appendix B

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
The document to use our code and pre-trained model is in the supplementary materials.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We reported the number of examples, details of train/dev splits, and paragraph lengths of the pre-training corpus.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3, Section 4, Appendix

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.