SciPrompt: Knowledge-Augmented Prompting for Fine-Grained Categorization of Scientific Topics

Anonymous ACL submission

Abstract

Prompt Learning has emerged as a pivotal method for harnessing the knowledge inherent in pre-trained language models (PLMs) to address downstream tasks in Natural Language Processing (NLP). Also, it has been demonstrated that traditional topic classification tasks can attain performance levels comparable to those of low-resource scenarios through prompt tuning. However, cross-domain, fine-grained topic classification with few-shot prompts remain largely unexplored, particularly due to the difficulty of manually selecting class label terms without domain expertise. We introduce SciPrompt, a framework designed to automatically retrieve topic-related terms and phrases as supplementary knowledge and then select semantically correlated domain-specific label terms within scientific contexts for topic classification. Furthermore, we propose a new verbalization strategy that leverages the correlation scores to enhance signal provision during model tuning. We expand our investigation to include multi-class classification across up to 53 sub-domain publication types. Our model outperforms previous low-resource prompting systems in scientific topic classification tasks under few-shot and zero-shot settings.

1 Introduction

Pre-trained language models (PLMs) are developed by extensively training on large text corpora, which enables PLMs to implicitly store knowledge that can be leveraged for domain-specific or downstream tasks. However, traditional fine-tuning, which requires additional training on domain- or task-specific data (Ovadia et al., 2023), faces challenges in limited data scenarios and thus may not be for few-shot or zero-shot settings. Recent work has turned to prompt-tuning (Sanh et al., 2021; Ding et al., 2022a; Gu et al., 2022; Schick and Schütze, 2020; Liu et al., 2023) as an effective alternative, which uses templates to guide the model in generating specific outcomes, addressing tasks like text classification (Schick and Schütze, 2020; Hu et al., 2021; Chen et al., 2022; Gao et al., 2021; He et al., 2022) and summarization (Liu et al., 2022) under low-resource conditions.

Despite these developments, most existing studies on topic classification have not considered fine-grained classification or domain knowledge injection. The complexity of scientific language and the scarcity of labeled data in sub-domains make it challenging to classify cross-domain scholarly articles with limited training samples and manually created verbalizers (Schick and Schütze, 2020). While some efforts have been made to integrate knowledge into the verbalization (Hu et al., 2021), there remains a gap in research on constructing wide-ranged verbalizers. In this study, we introduce a prompt-tuning approach enriched with domain knowledge, proposing a new strategy for retrieving adaptive label terms without manual intervention. We enhance our approach in low-data scenarios with external knowledge to expand the verbalizer's label terms (Hu et al., 2021) from token-level to term spans. Drawing on the rationale of Natural Language Inference (NLI) tasks, we leverage semantic similarity to select relevant scientific phrases from external knowledge bases (KBs) based on topic names.

We also introduce an automated and domain-adapted filtering strategy to isolate topic-related phrases pertinent to input class names. This marks a departure from previous verbalization methods (Hu et al., 2021; Ding et al., 2022b), which solely rely on probability mapping for class prediction and are further limited to single-token mappings. Given the complexity of scientific domain terminology (see Appendix C for more details), we refine the traditional verbalization approach by integrating knowledge and employing a weight-aware label term mapping function, thus creating a more nu-

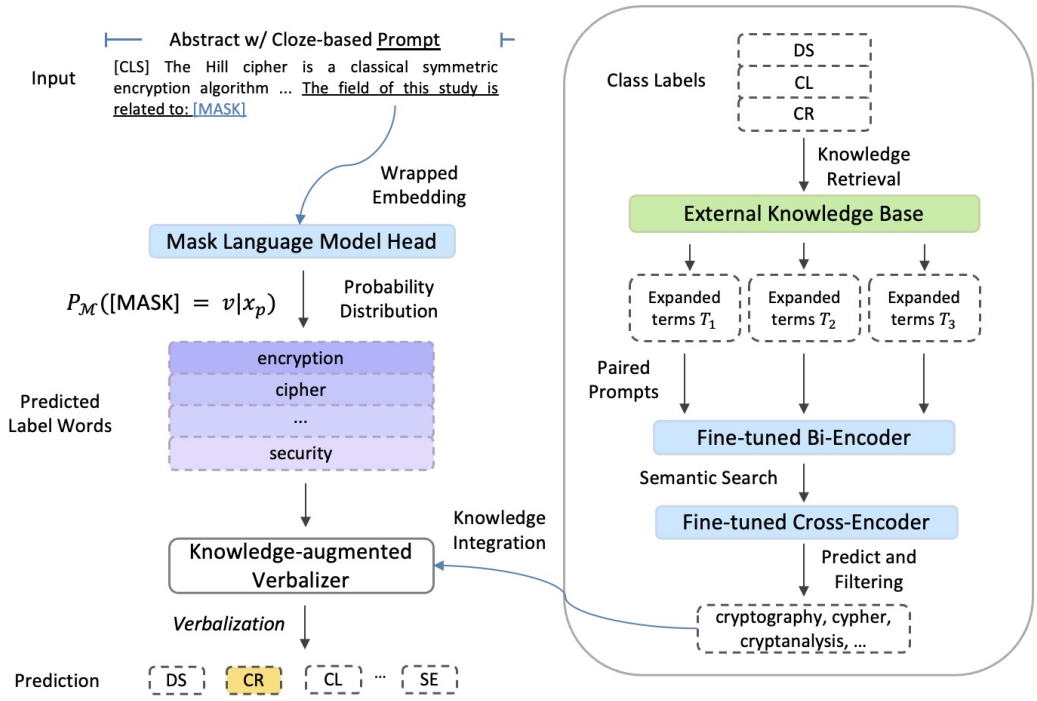

Figure 1: Overall framework of SciPrompt.

anced projection of the model's predictions to topic probabilities.

Our approach is composed of three essential stages: prediction, retrieval, and filtering. Initially, we use a cloze-based prompt and an input abstract to guide the masked language model (MLM) to generate label words to fill the `[MASK]`. Then, we use each topic name as a query to fetch pertinent domain phrases from external KBs. To filter the potentially irrelevant data gathered in the retrieval stage, we employ both bi-encoder and cross-encoder models, which have been fine-tuned using the SciNLI dataset (Sadat and Caragea, 2022), enabling us to pinpoint the most relevant domain phrases. We set various thresholds and adopt a label term calibration approach to remove irrelevant knowledge terms during filtering. Finally, with the polished set of knowledge-enriched domain terms, we incorporate them into the verbalization function to convert the MLM's outputs into specific topic categories. For the final mapping of scores from label terms to class labels, we opt for either an average loss (Hu et al., 2021) or a semantic score-weighted average loss, enhancing the precision of our probability projections for the augmented verbalizer. Our methodology extends beyond mere token-to-token verbalization, encompassing token-to-term spans that enrich semantic meaning of domain knowledge. This broader scope allows for an advanced interpretation of language.

2 Related Work

Our work is inspired by previous studies on knowledge incorporation and label terms refinement.

2.1 Knowledge-Powered Prompting for Text Classification

Pattern-Exploiting Training (PET) framework (Schick and Schütze, 2021a,b), which initially investigates how prompt templates could guide language models to tackle classification tasks (Han et al., 2021; Ding et al., 2022b; Min et al., 2022; Wang et al., 2022a; Zhang et al., 2022; Wang et al., 2022b), has inspired research into knowledge incorporation. Specifically, Hu et al. (2021) incorporate knowledge in the verbalizing process to help MLMs predict masked tokens more accurately. AdaPrompt (Chen et al., 2022) is another knowledge incorporation method that leverages both task and prompt characteristics to adaptively retrieve external knowledge for continual pre-training of PLMs. However, classifying scientific articles presents various challenges that previous methods have not addressed, including complex semantic structures and a wide range of topics in scientific domain (Eykens et al., 2021; Khadhraoui et al.,

Datasets	# Training Abstracts	# Classes	Avg. Length	Test
arXiv	50000	53 (sub)	129	5300
SDPRA 2021	28000	7	155	2800
S2ORC	60000	19	136	5700

Table 1: Datasets Statistics.

2022) as well as a scarcity or the imbalance of labeled data across various disciplines(Cunha et al., 2021).

2.2 Label Terms Refinement

Prior research on prompt-based classification has utilized verbalizer modules to map MLM predictions to specific classes. However, relying solely on a single label word often results in low accuracy. Schick and Schütze (2021a) introduce an automatic verbalizer search that identifies suitable label words from training data and language models. This approach has been further explored to enhance model performance (Gao et al., 2021; Shin et al., 2020; Liu et al., 2021), although these methods typically demand extensive training data, making them less suitable for low-resource scenarios. To address these challenges, approaches such as the multiple label words summing method (Wang et al., 2022a) or manually expanding the verbalizer (Shin et al., 2020) have limitations when classifying fine-grained categories without expert intervention. Recently, external KBs have been applied to enrich the verbalizer by sourcing class-related label words (Hu et al., 2021; Chen et al., 2022).

3 Methodology

Our SciPrompt framework introduces a comprehensive approach for topic classification tasks with knowledge augmentation and filtering. The overall framework of SciPrompt is shown in Figure 1.

3.1 Cross-Domain and Fine-Grained Scientific Topic Classification

MLM $\mathcal{M}$ (e.g., SciBERT) is created through randomly masking tokens in the context and train the model to predict the masked positions. Similarly, the prompt-tuning method leverages a cloze- or prefix-based prompt template, reformulating the input into a masked language modeling task. Such a strategy enables $\mathcal{M}$ to predict the masked position, facilitating the execution of downstream tasks based on the outputs of $\mathcal{M}$. Building upon Hu et al. (2021), our framework employs a few-shot prompting strategy that conceptualizes the scientific topic classification as an N-way K-shot classification task, where N indicates the number of classes and K is the number of labeled examples per class.

We provide a limited number of labeled examples for each class to tune $\mathcal{M}$. We construct training $\mathcal{D}_{train}$ and validation set $\mathcal{D}_{val}$ following the previous studies (Gao et al., 2021; Perez et al., 2021; Wang et al., 2022a; Hu et al., 2021) with n examples per class. For the few-shot setting, given a cloze-based prompt template $\mathcal{P}_t$ and an input abstract a_n, where $a_n \in \mathcal{D}_{train}$, $\mathcal{M}$ will predict label word l to fill into the `[MASK]` position in the prompt template. After that, the verbalizer function f_v will map the predicted label word l onto pre-defined label term set $\mathcal{L}$ to classify into class, i.e., $\mathcal{L} \rightarrow \mathcal{Y}$. We denote a cross-entropy loss (Gao et al., 2021) to update the parameters of $\mathcal{M}$ through verbalization outputs. For instance, the prompt is designed as "`[Abstract]`. The field of this article is related to: `[MASK]`". The $\mathcal{M}$ will predict suitable label word l to fill into the `[MASK]`. Then, the f_v will calculate the probability of classifying l into a topic y_i, where $y_i \in \mathcal{Y}$:

$$P(y_i \,|\, a_n) = P(\texttt{[MASK]} = \mathcal{M}(l) \,|\, a_n), \qquad (1)$$

where $l \in \mathcal{L}$. In the zero-shot setting, given $\mathcal{M}$ can directly generate a label word to fill into `[MASK]`, we use the output of $\mathcal{M}$ as the final label word and input into the verbalizing function to calculate class probabilities without tuning loss updates.

3.2 Scientific Knowledge Retrieval

Predicting masked words with an MLM involves generating a range of potential label words, each with varying probabilities of matching a specific class. Enhancing the verbalizer with a more extensive set of label terms is a proven strategy to improve the accuracy of word-to-class mapping (Hu et al., 2021; Chen et al., 2022; Wang et al., 2022a; Shin et al., 2020). To facilitate this, we leverage two external KBs, `Related Words`[1] and `Reverse Dictionary`[2] for topic retrieval. `Related Words` identifies relevant terms using vector similarity and resources like word embeddings and ConceptNet. `Reverse Dictionary`, which acts as a word search engine, finds terms based on definitions or phrases. `Reverse Dictionary` is particularly useful in complex scientific fields, where straightforward labels from `Related Words` may not suffice

[1] https://relatedwords.org
[2] https://reversedictionary.org

given a domain-specific span (e.g., Networking and Internet Architecture). We set class labels $C = \{y_1, y_2, ..., y_n\}$ as queries to retrieve from `Related Words` $\mathcal{G}_{RW}$.

When $\mathcal{G}_{RW}$ fails to produce terms with similarity scores above zero, we use `Reverse Dictionary`, denoted as $\mathcal{G}_{RD}$, for additional phrase retrieval. Each retrieved term is assigned a single relevance score. Initially, we adopt the same threshold (i.e., threshold = 0) as KPT (Hu et al., 2021) for term retrieval based on topic names. Subsequently, we impose two additional thresholds for further selection of retrieved terms (§3.3). Utilizing these KBs enables the compilation of knowledge-enhanced term sets $\mathcal{T}_i = t_1, t_2, ..., t_m$ for each dataset, where $i \in n$ and t represents the retrieved label terms. It is important to note that the number of terms m may vary for each class.

3.3 Domain Adaptive Model Tuning

To effectively select relevant label words for each class from the retrieved raw terms, it is crucial to employ a method that is both customized and adaptable to specific domains. Inspired by Chen et al. (2022), who use a pre-trained NLI model to filter through label words generated by an MLM, we introduce a strategy for accurately identifying domain-related label terms through the infusion of knowledge. We leverage a new scientific NLI dataset $\mathcal{D}_{SciNLI}$ introduced by Sadat and Caragea (2022), consisting of labeled sentence pairs (s_i, s_j) from scholarly articles in the fields of NLP and computational linguistics. This dataset is used to fine-tune both a cross-encoder model $\mathcal{M}_{ce}$ and a bi-encoder pre-trained NLI model $\mathcal{M}_{be}$. The training labels are defined as "*Entailment*", "*Contradiction*", framing it as a binary classification task. The inference from the fine-tuned NLI models is represented as follows:

$$\mathcal{M}'(s_i, s_j) = \begin{cases} > 0 & \text{if } s_i \text{ entails } s_j \\ < 0 & \text{if } s_i \text{ contradicts } s_j \end{cases},$$

where $\mathcal{M}'$ denotes either $\mathcal{M}_{ce}$ or $\mathcal{M}_{be}$.

3.4 Semantic Knowledge Filtering

For each retrieved scientific topic phrase, we merge it with a standard prompt (see Appendix D), encode the sentences using a fine-tuned $\mathcal{M}_{be}$, and use these as queries for semantic searches. We calculate semantic similarity scores, and cross scores for each topic using $\mathcal{M}_{be}$ and a classifier from $\mathcal{M}_{ce}$, selecting relevant sentences based on predefined thresholds ($\mu_{be} = 0.5$, $\mu_{ce} = 0.1$). As these scores also help in predicting label words, we apply this method in both few-shot and zero-shot scenarios.

Following KPT (Hu et al., 2021), we also apply knowledge terms calibration with full training set to directly remove irrelevant class terms in the verbalizer that are less likely being predicted by $\mathcal{M}$. The retrieved topic terms for each class with lower probabilities (i.e., less than 0.5) produced by $\mathcal{M}$ will be removed. The probability of t is:

$$\hat{P}_{\mathcal{M}}(\texttt{[MASK]} = t|a_n) \propto \frac{P_{\mathcal{M}}(\texttt{[MASK]} = t|a_n)}{prior(p_t)}, \tag{2}$$

where $prior(p_t)$ is the prior probability of label term t.

3.5 Weighted Verbalizer Transformation

Given that label terms may be tokenized into multiple tokens, we adopt a "mean" method to average the tokens of a label word (Ding et al., 2022b), considering all parts of a term significant. The mean logits for a tokenized label word in a verbalizer are calculated as:

$$v_{y_i} = \frac{\sum_{i \in \mathcal{N}}(v_i \cdot v_{mask})}{|v_{mask}|}, \tag{3}$$

where v_{y_i} is the average logits for the tokenized label term, $\mathcal{N}$ represents the count of tokens from the tokenizer, v_i denotes the logits of each token ($v_i \in \mathcal{L}$), and v_{mask} indicates the position of the tokenized logit.

Adopting the verbalizer structure from Ding et al. (2022b), we introduce a verbalization approach that maps $\mathcal{M}$'s output to specific classes y_i, using predefined semantic scores w_l as weights for each label term. This method aims to enhance the accuracy of classifying $\mathcal{M}$'s predictions l into topic y_i:

$$\begin{aligned} P(y_i|a_n) &= \arg\max_{y_i \in \mathcal{Y}} s(v_{y_i}|h_{mask}, w_l) \\ &= \frac{\exp\left(v_{y_i} \cdot h_{mask} \cdot w_l\right)}{\sum_{y \in \mathcal{Y}} \exp\left(v_y \cdot h_{mask} \cdot w_l\right)}, \end{aligned} \tag{4}$$

where $s(v_{y_i}|h_{mask}, w_l)$ calculates $\mathcal{M}$'s scores for output v_{y_i} at the `[MASK]` position, with v_{y_i} as the label term embeddings, and h_{mask} the hidden states at the `[MASK]` position.

3.6 Vector-Based Verbalizer Mapping

Incorporating the retrieved knowledge terms into the verbalizer is crucial to making accurate predictions and eliminating noise simultaneously. Moving beyond simple summing (Wang et al., 2022a) or

weighted averaging (Hu et al., 2021) of label words, WARP model (Hambardzumyan et al., 2021) uses vector representations for class mapping, distinct from conventional single word projection. These methods, however, which typically rely on a limited label words, may not suit our broader needs. Building on the uniqueness of our span-level verbalizer, we refine the verbalization in SciPrompt$_{Soft}$ by drawing inspiration from the soft verbalizer concept introduced by WARP. We aggregate all retrieved terms per topic with semantic scores into a vector for topic probability prediction and optimize it during tuning.

4 Experiments

This section presents the experimental settings of SciPrompt across three scientific datasets for topic classification in few-shot and zero-shot scenarios.

4.1 Datasets

We use three publicly available datasets in English for experiments: SDPRA 2021 (Reddy and Saini, 2021), arXiv (Meng et al., 2019), and S2ORC (Lo et al., 2019). SDPRA 2021 collects scientific articles within computer science across 7 categories. arXiv (Meng et al., 2019) includes abstracts sourced from the arXiv website[3] across 53 subcategories, and S2ORC gathers academic papers across 19 disciplines. For the S2ORC data, we only select abstracts with a single discipline label through the Semantic Scholar Public API[4]. The statistics and category examples of these datasets are displayed in Table 1 and Appendix C.

The SDPRA 2021 dataset serves as a pilot for developing SciPrompt and testing the incorporation of scientific knowledge into prompt tuning. In our experiments, abstracts shorter than 30 words are excluded to remove invalid abstracts, leading to final training and test sizes of 25,110 and 2,790 for SDPRA; 49,300 and 5,300 for arXiv; and 60,000 and 5,700 for S2ORC. We utilize sub categories for arXiv and parent categories for both SDPRA and S2ORC for topic classification tasks.

4.2 Experimental Settings

SciPrompt is built upon the OpenPrompt framework (Ding et al., 2022b), utilizing SciBERT (Beltagy et al., 2019) as the backbone MLM. We apply a consistent prompt template across all experiments (see Appendix D for more details). All experiments are conducted through a 32 GB Tesla V100 GPU. The experimental details is shown in Appendix B.

In the few-shot setting, we benchmark SciPrompt alongside standard fine-tuning, simplified prompt-tuning, and KPT (Hu et al., 2021). Standard fine-tuning takes all labeled training examples as input to tune the MLM for topic classification. We take the final representation of the `[CLS]` token as the output vector of the abstract (Cohan et al., 2020). Simplified prompt-tuning (PT) with manually defined verbalizer (Ding et al., 2022b) only takes each lowercase topic name as a seed word for verbalization. We apply the same setting as in SciPrompt, including a unified prompt template, MLM, and model hyper-parameters. KPT (Hu et al., 2021) leverages external knowledge to enrich the verbalizer with additional word relevance and frequency filtering strategies. The original experimental settings are applied for KPT, including label word retrieval, refinement, and verbalization. All experiments use the same MLM (i.e., SciBERT) for equal comparison. Besides, training and validation examples per class (Ding et al., 2022b; Hu et al., 2021; Wang et al., 2022a) are uniform, conducting tests with 1, 5, 10, 20, and 50 shots across all datasets and reporting accuracy as evaluation metric. We evaluate model performance across five random seeds to account for variability (Hu et al., 2021; Ding et al., 2022b).

For the zero-shot setting, we sample approximately 10% of each dataset for testing, ensuring adequate representation for each topic. Additionally, random seeds are applied in KPT, where it samples an unlabeled support set of 200 examples for the calibration of label words. We also include the latest iterations of ChatGPT[5] and open-sourced Llama 2[6] for zero-shot classification using the same instruction. For ChatGPT, we use `gpt-3.5-turbo-instruct`, which contains 175 million model parameters developed by OpenAI. We apply `llama-2-70b-chat` as the backbone model for Llama 2 through Replicate API[7].

5 Results and Analysis

We report our main experimental results and model analysis in this section.

[3] https://arxiv.org/
[4] https://www.semanticscholar.org/product/api
[5] https://openai.com/chatgpt
[6] https://llama.meta.com/
[7] https://replicate.com/blog/run-llama-2-with-an-api

Examples	Method	SDPRA 2021	arXiv	S2ORC	Avg.
1	Fine-tuning$_{SciBERT}$	12.72 $\pm$ 3.70	2.03 $\pm$ 0.21	4.76 $\pm$ 0.85	6.50 $\pm$ 1.59
	Prompt-tuning$_{Manual}$	**71.68** $\pm$ 4.73	34.95 $\pm$ 1.45	40.88 $\pm$ 1.92	49.17 $\pm$ 2.7
	KPT	50.74 $\pm$ 3.03	32.18 $\pm$ 1.08	43.20 $\pm$ 1.33	42.04 $\pm$ 1.81
	SciPrompt	64.42 $\pm$ 3.64	**40.57** $\pm$ 1.60	**47.92** $\pm$ 1.67	**50.97** $\pm$ 2.30
	SciPrompt$_{Soft}$	62.65 $\pm$ 4.94	31.06 $\pm$ 1.74	29.94 $\pm$ 1.94	41.22 $\pm$ 2.87
5	Fine-tuning$_{SciBERT}$	16.45 $\pm$ 4.35	2.36 $\pm$ 0.55	5.63 $\pm$ 1.37	8.15 $\pm$ 2.09
	Prompt-tuning$_{Manual}$	83.46 $\pm$ 1.41	47.58 $\pm$ 1.68	49.53 $\pm$ 0.88	60.19 $\pm$ 1.32
	KPT	77.71 $\pm$ 3.34	53.68 $\pm$ 1.69	50.40 $\pm$ 1.84	60.60 $\pm$ 2.29
	SciPrompt	81.81 $\pm$ 3.34	56.36 $\pm$ 0.95	**52.12** $\pm$ 1.59	**63.43** $\pm$ 1.96
	SciPrompt$_{Soft}$	**83.7** $\pm$ 2.86	**58.01** $\pm$ 0.94	47.44 $\pm$ 1.60	63.05 $\pm$ 1.80
10	Fine-tuning$_{SciBERT}$	17.44 $\pm$ 4.5	3.14 $\pm$ 1.15	6.31 $\pm$ 0.81	8.96 $\pm$ 2.15
	Prompt-tuning$_{Manual}$	85.60 $\pm$ 0.81	50.86 $\pm$ 2.89	52.15 $\pm$ 0.98	62.87 $\pm$ 1.56
	KPT	83.82 $\pm$ 0.72	61.83 $\pm$ 0.83	52.91 $\pm$ 0.66	66.19 $\pm$ 0.74
	SciPrompt	84.71 $\pm$ 0.89	62.37 $\pm$ 0.57	**53.65** $\pm$ 0.22	66.91 $\pm$ 0.56
	SciPrompt$_{Soft}$	**85.96** $\pm$ 0.60	**63.42** $\pm$ 0.50	52.41 $\pm$ 0.30	**67.26** $\pm$ 0.47
20	Fine-tuning$_{SciBERT}$	17.16 $\pm$ 3.90	3.53 $\pm$ 0.86	7.29 $\pm$ 1.32	9.33 $\pm$ 2.03
	Prompt-tuning$_{Manual}$	87.76 $\pm$ 0.70	52.92 $\pm$ 2.72	54.32 $\pm$ 0.89	65.00 $\pm$ 1.44
	KPT	87.74 $\pm$ 0.51	66.25 $\pm$ 0.73	54.67 $\pm$ 0.43	69.55 $\pm$ 0.56
	SciPrompt	**87.95** $\pm$ 0.41	66.59 $\pm$ 0.64	**55.49** $\pm$ 0.56	**70.01** $\pm$ 0.54
	SciPrompt$_{Soft}$	87.90 $\pm$ 0.51	**66.86** $\pm$ 0.46	54.70 $\pm$ 0.42	69.82 $\pm$ 0.46
50	Fine-tuning$_{SciBERT}$	27.50 $\pm$ 9.48	11.07 $\pm$ 1.93	12.02 $\pm$ 2.22	16.86 $\pm$ 4.54
	Prompt-tuning$_{Manual}$	88.93 $\pm$ 0.57	60.63 $\pm$ 1.32	56.08 $\pm$ 0.29	68.55 $\pm$ 0.73
	KPT	88.93 $\pm$ 0.37	69.95 $\pm$ 0.63	56.50 $\pm$ 0.81	71.79 $\pm$ 0.60
	SciPrompt	**88.99** $\pm$ 0.75	69.89 $\pm$ 0.63	**56.66** $\pm$ 0.49	**71.85** $\pm$ 0.62
	SciPrompt$_{Soft}$	88.97 $\pm$ 0.71	**70.15** $\pm$ 0.52	56.02 $\pm$ 0.60	71.71 $\pm$ 0.61
Full Set	Fine-tuning (Full) *	90.71	54.58	53.74	66.34

Table 2: Experimental results of few-shot settings. We report mean accuracy (%) and standard deviation after running 5 times over 6 learning shots. * represents using full labeled training set. SciBERT serves as the foundational MLM in all experimental settings.

Methods	SDPRA 2021	arXiv	S2ORC	Avg.
Prompt-tuning$_{Manual}$	62.97	20.81	32.93	38.90
KPT	41.50 $\pm$ 3.00	20.83 $\pm$ 0.18	38.42 $\pm$ 0.66	33.58 $\pm$ 1.28
ChatGPT (*gpt-3.5-turbo-instruct*)	79.43	54.51	46.95	60.30
Llama 2 (*llama-2-70b-chat*)	62.04	26.98	40.30	43.11
SciPrompt	51.97	22.28	41.30	38.52

Table 3: Performance of zero-shot setting. Only KPT is reported through mean accuracy (%) and standard deviation (§4.2). We apply the same instruction to ChatGPT and Llama v2 to conduct classification inference without labeled examples under test sets.

5.1 Main Results

We highlight the performance of SciPrompt against baseline models across three scientific datasets in both few-shot and zero-shot settings, focusing on fine-grained cross-domain scientific topic classification tasks. The experimental performances are listed in Table 2. Results are averaged over five runs as the same as KPT (Hu et al., 2021) to counteract sampling randomness, reported as mean accuracy with standard deviation.

Few-shot Results. Overall, SciPrompt achieves the best average accuracy of three datasets in all settings. Specifically, SciPrompt and SciPrompt$_{soft}$ excel in low-data scenarios (e.g., 1-shot and 5-shot), particularly within the arXiv and S2ORC datasets, often surpassing baseline models. SciPrompt also outperformes KPT by 8.93% in the 1-shot setting and 2.83% in the 5-shot setting. As the number of training examples increases, the margin of improvement over baseline models narrows. Notably,

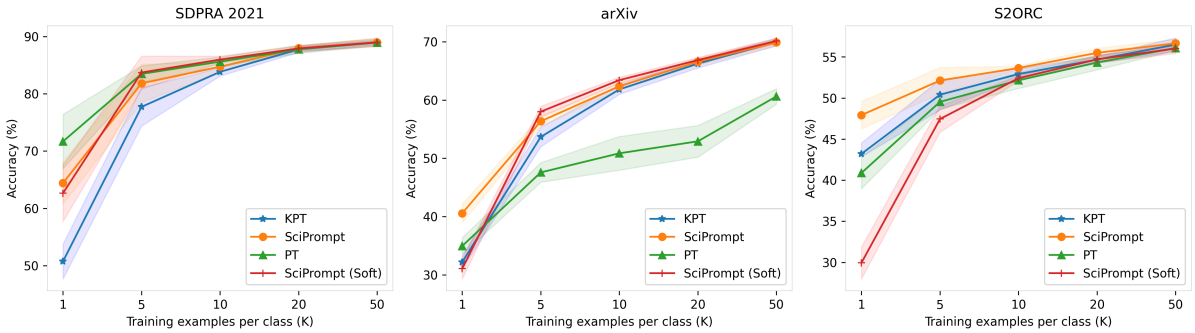

Figure 2: Performance comparison of SciPrompt and KPT under different training shots K over three datasets. We report mean accuracy as line and standard deviation as shadow.

Method	**K=1**	**K=5**	**K=10**	**K=20**	**K=50**	**Avg.**	**Zero-shot**
KPT	32.18±1.08	53.68±1.69	61.83±0.83	66.25±0.73	69.95±0.63	56.78	20.83
SciPrompt	40.57±1.60	56.36±0.95	62.37±0.57	66.59±0.64	69.89±0.63	**59.16**	**22.28**
w/o *CL*	40.19±1.46	55.84±0.98	62.32±0.50	66.45±0.61	69.92±0.64	58.94	21.87
w/o *SS*	38.70±0.86	55.19±0.80	62.48±0.59	66.70±0.77	69.73±1.01	58.56	6.17
w/o *SS+CL*	38.36±0.86	54.76±0.86	62.25±0.56	66.54±0.81	69.86±0.92	58.35	5.62
w/o *FL+CL*	29.77±0.74	50.13±0.88	59.57±0.97	65.77±0.47	69.55±0.70	54.96	3.77
SciPrompt$_{Soft}$	31.06±1.74	58.01±0.94	63.42±0.50	66.86±0.46	70.15±0.52	57.90	-
w/o *CL*	38.65±0.90	58.33±1.62	63.64±0.66	67.05±0.55	70.41±0.56	59.62	-
w/o *SS*	41.49±1.38	58.36±0.99	63.70±0.75	67.26±0.75	70.20±0.21	**60.20**	-
w/o *SS+CL*	42.22±1.32	57.72±1.46	63.53±0.57	67.03±0.78	70.35±0.49	60.17	-
w/o *FL+CL*	37.50±1.31	57.66±1.49	63.63±0.49	67.13±0.93	70.24±0.49	59.23	-

Table 4: Ablation study of SciPrompt regarding mean accuracy and standard deviation in arXiv dataset under few-shot and zero-shot settings.

SciPrompt exceeds full-set fine-tuning by an average of 0.57%, 3.67%, and 5.51%, respectively, in 10, 20, and 50 shots. Despite variability in performance improvements across different training sizes, our method consistently achieves the highest accuracy in arXiv and S2ORC datasets across all configurations.

Zero-shot Results. We report the results in Table 3. ChatGPT leads in performance across all datasets. However, SciPrompt outperforms traditional PT and KPT in the arXiv dataset by 1.47% and 1.45%, respectively, and in the S2ORC dataset by 8.37% and 2.88%. These results highlight SciPrompt's effectiveness in leveraging domain-specific knowledge for scientific topic classification, even without labeled training data. Although SciPrompt's accuracy is lower than Llama 2, the margin of difference is significantly reduced when compared to both ChatGPT and Llama 2. The average accuracy of ChatGPT is higher than SciPrompt by 21.78%, and Llama 7's is higher by 4.59%. However, against Llama 2 (`meta/llama-2-7b`), SciPrompt outperforms it by 7.98%. Note that zero-shot testing for SciPrompt$_{soft}$ is infeasible since it needs trainable tokens in the decoding layer during model tuning.

5.2 Cross-Domain Visualization

In few-shot settings, SciPrompt outperforms baseline models in fine-grained classification tasks on the arXiv and S2ORC datasets, as shown in Figure 2. SciPrompt$_{Soft}$, despite lower overall performance, excels in the arXiv dataset in most cases, underscoring the advantage of learnable and adaptive domain-specific verbalizer in high-granularity tasks. Traditional prompt-tuning method achieves poorer performance in fine-grained tasks, especially in minimal data scenarios like 1-shot and 5-shot, where our models demonstrate significant improvements. Notably, the accuracy of all models starts to surge from 5 shots across three datasets.

5.3 Ablation Study

Our ablation study on the arXiv dataset (Table 4) demonstrates the advantages of our models over KPT, with a 1.45% increase in zero-shot accuracy. SciPrompt and SciPrompt$_{Soft}$ outperform KPT by 2.38% and 3.42%, respectively, in terms of average

accuracy under the few-shot setting. We examine the impact of removing full-size calibration ("w/o *CL*"), semantic scores ("w/o *SS*"), and both ("w/o *SS+CL*"), finding that both components improve the performance, especially in the zero-shot setting where their absence lowers accuracy by 0.41% ("w/o *CL*") and 16.11% ("w/o *SS*") compared to SciPrompt, underlining the critical role of *SS* in bolstering the model's effectiveness.

Interestingly, SciPrompt$_{Soft}$ performs better without *SS* than when both components are included. Removing both *SS* and *CL* yields the best 1-shot performance, suggesting less intervention optimizes model tuning in low-data contexts. Furthermore, comparing setups without pre-filtering and calibration ("w/o *FL+CL*") to those with pre-filtering shows an accuracy increase by 3.39% and 0.94% for SciPrompt and SciPrompt$_{Soft}$ respectively, highlighting the effectiveness of pre-filtering of augmented verbalizer for topic classification. The ablation study of SDPRA and S2ORC also shows the same pattern as in arXiv.

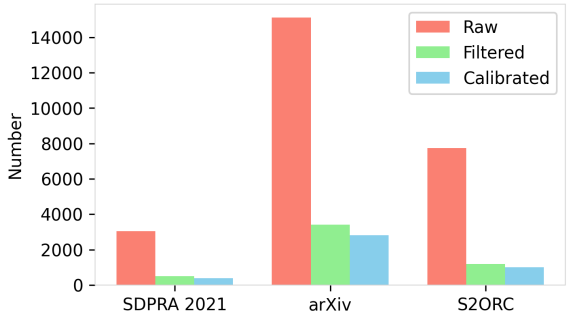

Figure 3: The number of verbalizer terms across three datasets.

5.4 Calibration of Domain Knowledge

Figure 3 compares the verbalizer label term counts across datasets: "Raw" reflects the initial count after knowledge retrieval from two KBs (§3.2); "Filtered" shows counts post-semantic filtering (§3.4), reducing terms by 84%, 77%, and 85%; "Calibrated" involves removing low-likelihood terms before model tuning. Appendix A and Table 6 reveal that NLI filtering and calibration enhance the model's accuracy in few-shot and zero-shot settings, linking domain-relevant spans in the verbalizer to improved classification.

5.5 Model Tuning Efficiency

Table 5 shows SciPrompt*Soft* reduces GPU memory usage by 16.5% for SDPRA 2021, 38% for arXiv, and 46.2% for S2ORC compared to SciPrompt's full-size label term calibration. Although SciPrompt achieves higher average accuracy in the few-shot setting on the S2ORC dataset (see Table 6 in Appendix A), SciPrompt$_{Soft}$ outperforms SciPrompt on SDPRA 2021 and arXiv datasets, suggesting SciPrompt$_{Soft}$ can achieve competitive results with less GPU usage. Moreover, while ChatGPT and Llama 2 exhibit superior performance in a zero-shot setting, as shown in Table 3, it is worth noting that these language models are either mainly for commercial use or require substantial GPU resources, incurring higher costs or more time. For instance, in S2ORC dataset, our model not only cuts down the combined training and testing (inference) time by 93% compared to the Llama 2 70B model but also enhances accuracy by 1% over Llama 2, highlighting the efficiency and effectiveness of our approach.

Method	SDPRA 2021	arXiv	S2ORC
SciPrompt			
w/o CL	9.5%	12.5%	12.4%
w/ CL	29.3%	51.2%	59.6%
SciPrompt$_{Soft}$			
w/o CL	12.3%	12.3%	12.3%
w/ CL	12.8%	13.2%	13.4%

Table 5: The usage percentage of GPU memory during model tuning.

6 Conclusion

In this study, we introduce a knowledge-enhanced prompting framework designed for fine-grained scholarly topic classification using minimal or no abstracts. Recognizing the intricate structures and subtle distinctions within scientific abstracts, we employ a traditional MLM augmented with knowledge injection and semantic filtering. This approach enables the automatic extraction of domain-specific phrases and their integration into a weighted verbalizer for precise classification. Our experimental findings highlight SciPrompt's superior performance over existing models and the standard full-set fine-tuning approach, particularly in tasks requiring high levels of topic granularity. Remarkably, SciPrompt demonstrates competitive accuracy compared with the advanced Llama 2 70B model in the zero-shot setting, showing its potential to categorize scholarly topics with a light-weighted and efficient approach.

7 Limitations

Our study's limitations are outlined as follows: 1) Our external knowledge sources are limited to two non-scientific domain databases for retrieving topic words, potentially missing fine-grained scientific terminologies. Despite the challenge of identifying a universally applicable cross-domain scientific knowledge resource, future efforts should aim to discover more precise terminology databases. 2) We focus solely on a multi-class classification task and exclude abstracts that span multiple scientific sub-domains. Advancing towards a multi-label classification system capable of identifying publications across various domains would enhance the robustness of our approach. 3) Although SciPrompt and SciPrompt$_{Soft}$ generally surpass baseline methods, the enhancements are modest, and results fluctuate, particularly with an increase in labeled training data. Further investigation into the causes of these minimal gains and more comprehensive, interpretable experiments are necessary to bolster model performance. 4) We only consider using classification accuracy and standard deviation as model evaluation metrics. The experimental results can change given other metrics (e.g., Micro F1 and Macro F1). Additionally, while the standard deviation of our methods shrinks as the number of training examples increases, it is imperative that we employ statistical significance testing to draw statistically sensible conclusions on the comparison of system performance against baseline models.

8 Ethics Statement

The datasets and MLM employed in our study are publicly accessible and extensively utilized in the research community. To enhance the quality of our data, we apply heuristic filtering to exclude short-length abstracts across these datasets, acknowledging that this process may impact experimental accuracy. Our methodology includes extracting data from external knowledge bases via public APIs. Furthermore, as we utilize PLMs as the foundation of our approach, it is important to note that the predictive behavior of these models can be challenging to regulate due to the implicit knowledge embedded within the PLMs, which is difficult to explicitly decode. Therefore, caution should be exercised when adapting our method to other tasks, especially in the context of text classification through prompting.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*.

Yulong Chen, Yang Liu, Li Dong, Shuohang Wang, Chenguang Zhu, Michael Zeng, and Yue Zhang. 2022. Adaprompt: Adaptive model training for prompt-based nlp. *arXiv preprint arXiv:2202.04824*.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. *arXiv preprint arXiv:2004.07180*.

Washington Cunha, Vítor Mangaravite, Christian Gomes, Sérgio Canuto, Elaine Resende, Cecilia Nascimento, Felipe Viegas, Celso França, Wellington Santos Martins, Jussara M Almeida, et al. 2021. On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. *Information Processing & Management*, 58(3):102481.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Haitao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2022a. Prompt-learning for fine-grained entity typing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6888–6901.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022b. Openprompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113.

Joshua Eykens, Raf Guns, and Tim CE Engels. 2021. Fine-grained classification of social science journal articles using textual data: A comparison of supervised machine learning approaches. *Quantitative Science Studies*, 2(1):89–110.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. Ppt: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8423.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference*

on Natural Language Processing (Volume 1: Long Papers), pages 4921–4933, Online. Association for Computational Linguistics.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Pan He, Yuxi Chen, Yan Wang, and Yanru Zhang. 2022. Protum: A new method for prompt tuning based on"[mask]". *arXiv preprint arXiv:2201.12109*.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*.

Mayara Khadhraoui, Hatem Bellaaj, Mehdi Ben Ammar, Habib Hamam, and Mohamed Jmaiel. 2022. Survey of bert-base models for scientific text classification: Covid-19 case study. *Applied Sciences*, 12(6):2891.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Xiaochen Liu, Yang Gao, Yu Bai, Jiawei Li, Yinan Hu, He-Yan Huang, and Boxing Chen. 2022. Psp: Pre-trained soft prompts for few-shot abstractive summarization. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6355–6368.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Dan S Weld. 2019. S2orc: The semantic scholar open research corpus. *arXiv preprint arXiv:1911.02782*.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2019. Weakly-supervised hierarchical text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6826–6833.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. 2023. Fine-tuning or retrieval? comparing knowledge injection in llms. *arXiv preprint arXiv:2312.05934*.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

Saichethan Miriyala Reddy and Naveen Saini. 2021. Overview and insights from scope detection of the peer review articles shared tasks 2021. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 73–78. Springer.

Mobashir Sadat and Cornelia Caragea. 2022. Scinli: A corpus for natural language inference on scientific text. *arXiv preprint arXiv:2203.06728*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Han Wang, Canwen Xu, and Julian McAuley. 2022a. Automatic multi-label prompting: Simple and interpretable few-shot classification. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5483–5492.

Jianing Wang, Chengyu Wang, Fuli Luo, Chuanqi Tan, Minghui Qiu, Fei Yang, Qiuhui Shi, Songfang Huang, and Ming Gao. 2022b. Towards unified prompt tuning for few-shot text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 524–536.

Haoxing Zhang, Xiaofeng Zhang, Haibo Huang, and Lei Yu. 2022. Prompt-based meta-learning for few-shot text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1342–1357.

A Experiments of Various Verbalizer Sizes

As presented in Table 6, we document the performance metrics across various verbalizer sizes, in accordance with the configurations outlined in Figure 3. We only report the mean accuracy for each shot for brevity. The findings indicate that the model's performance is enhanced across all scientific domain topic classification datasets in both few-shot and zero-shot scenarios, attributable to the implementation of more sophisticated label term filtering techniques.

B Experimental Details

All experiments are conducted with a maximum input length of 256 tokens over 5 epochs, using the same hyper-parameters as KPT Hu et al. (2021), precisely a learning rate 3e-5. Our experiments were performed on a 32 GB Tesla V100 GPU with a batch size of 5. We set the length of `[MASK]` as 16 for decoding and shuffle the training set during model tuning.

C Datasets and Examples of Domain Topic Categories

We present a detailed introduction to datasets, including data collection and fine-grained topic categories.

SDPRA 2021 collects topics of scientific articles within computer science, consisting of abstracts sourced from arXiv and categorized under one of seven predefined domain labels. We combined the training and validation sets, reallocating them into new training (90%) and validation (10%) sets.

arXiv includes abstracts sourced from the arXiv website collected by Meng et al. (2019), categorized into 53 sub-categories and 3 parent categories (i.e., Math, Physics, and CS). We select 100 samples for each category as test set.

S2ORC incorporates academic papers across 19 disciplines. We filter abstracts to those with a single discipline label from the 2023-11-07 release through the Semantic Scholar Public API[8].

[8] https://www.semanticscholar.org/product/api

Detailed topic categories for each dataset are presented in Table 7. As outlined in Section 4.1, parent categories within arXiv are enumerated to elucidate our approach to categorizing fine-grained sub-categories. Our experiments focus exclusively on these sub-categories as our system is designed to tackle multi-class classification tasks.

D Cloze Prompt and Instructions of LLMs

As detailed in Section 5.1, we utilize ChatGPT and Llama 2 to perform a task of scientific topic classification guided by specific instructions. The same instructions are applied to both LLMs to infer the topics from scientific abstracts. It is important to note that we employ a distinct prompt from that used with MLMs due to our observation that the original prompt from SciPrompt fails to yield relevant field names, given the LLMs' limitations in comprehension. Consequently, we have crafted a more elaborate set of instructions to direct the LLMs in classifying topics, employing a projection of pre-defined class names similar to those used in the verbalization.

Instructions of LLMs

Based on the given article's abstract, please classify the abstract to a specific field of study. Only select field words from the following field words I provided. Only select one field name as output for each abstract. Your output should be all in lower cases.\n

Field Words List: logic in computer science, distributed computing, software engineering, data structures and algorithms, computational linguistics... \n

Abstract: For the purpose of developing applications for Post-K at an early stage, RIKEN has developed a post-K processor simulator. This simulator is based on the general-purpose processor simulator gem5...\n

Field of Study:

The "Field Words List" represents the original class names in the dataset. We concatenate the above instructions to LLMs and extract the predictions that appear after "Field of Study:" to evaluate

Paradim	K=1	K=5	K=10	K=20	K=50	Avg.	Zero-Shot
SciPrompt (SDPRA)							
w/o *FL*	45.23	73.20	81.61	87.40	88.94	75.28	25.56
w/ *FL*	61.25	81.33	84.67	87.78	89.05	80.829	34.98
w/ *CL*	63.56	81.57	84.62	88.02	89.02	**81.36**	**51.40**
SciPrompt$_{Soft}$ (SDPRA)							
w/o *FL*	55.00	80.62	83.84	87.91	88.69	79.21	-
w/ *FL*	65.53	83.43	85.26	88.13	88.97	**82.26**	-
w/ *CL*	64.92	81.78	85.46	87.79	89.14	81.82	-
SciPrompt (arXiv)							
w/o *FL*	29.77	50.13	59.57	65.77	69.55	54.96	3.77
w/ *FL*	38.36	54.76	62.25	66.54	69.86	58.35	5.62
w/ *CL*	38.70	55.19	62.48	66.70	69.73	**58.56**	**6.17**
SciPrompt$_{Soft}$ (arXiv)							
w/o *FL*	37.50	57.66	63.63	67.13	70.24	59.23	-
w/ *FL*	42.22	57.72	63.53	67.03	70.35	60.17	-
w/ *CL*	41.49	58.36	63.70	67.26	70.20	**60.20**	-
SciPrompt (S2ORC)							
w/o *FL*	41.27	49.22	52.69	55.30	56.31	50.96	25.25
w/ *FL*	46.00	51.23	53.43	55.25	56.15	52.41	26.11
w/ *CL*	47.55	51.85	53.52	55.32	56.67	**52.98**	**40.79**
SciPrompt$_{Soft}$ (S2ORC)							
w/o *FL*	42.35	50.10	51.89	54.52	56.17	51.01	-
w/ *FL*	46.33	50.24	52.83	54.76	56.17	52.07	-
w/ *CL*	46.34	51.09	53.02	54.59	55.82	**52.17**	-

Table 6: Performance comparison given various number of label terms in verbalizer. We report the mean accuracy after 5 runs for each shot.

the classification performance.

Cloze-Based Prompt Template of MLM

`Abstract`. The field of this study is related to: `[MASK]`.

Datasets	Parent-category	Sub-category
arXiv	Math (25)	numerical analysis, algebraic geometry, functional analysis, number theory, complex variables, applied mathematics, general mathematics, logic, optimization and control, statistics, probability, differential geometry, combinatorics, operator algebras, representation theory, classical analysis, dynamical systems, group theory, quantum algebra, rings and algebras, symplectic geometry, algebraic topology, commutative algebra, geometric topology, metric geometry
	Physics (10)	optics, fluid dynamics, atomic physics, instrumentation and detectors, accelerator physics, general physics, plasma physics, chemical physics, sociophysics, classical physics
	CS (18)	computer vision, game theory, information theory, machine learning, distributed computing, cryptography, networking and internet architecture, computational linguistics, computational complexity, software engineering, artificial intelligence, logic in computer science, systems and control, cryptography and security, data structures and algorithms, programming languages, other computer science, databases
SDPRA 2021	logic in computer science, distributed computing, software engineering, data structures and algorithms, computational linguistics, networking and internet architecture, cryptography	-
S2ORC	engineering, chemistry, computer science, business, political science, environmental science, physics, economics, geography, medicine, psychology, art, materials science, mathematics, sociology, geology, philosophy, biology, history	-

Table 7: Detailed topic categories of three datasets. Note we only classify sub-categories for arXiv.