Abstract

In-context learning has been extensively validated in large language models. However, the mechanism and selection strategy for in-context example selection, which is a crucial ingredient in this approach, lacks systematic and in-depth research. In this paper, we propose a data compression approach to the selection of in-context examples. We introduce a two-stage method that can effectively choose relevant examples and retain sufficient information about the training dataset within the in-context examples. Our method shows a significant improvement of an average of 5.90% across five different real-world datasets using four language models.

1 Introduction

Drawing inspiration from recent research that regards Large Language Models (LLMs) as an efficient means of compressing pre-training datasets, and the notion that In-Context Learning (ICL) can be seen as fine-tuning on example datasets (Dai et al., 2022), we assume that LLMs can achieve data re-compression through ICL. In other words, an effective training dataset compression method can aid in the selection of in-context examples. Looking at the matter from another perspective, it is evident that fine-tuning the entire dataset would yield the best results. However, in the case of ICL, we typically choose only a few examples as LLM prompts due to the limitations of input window length. By employing data compression techniques, we can ensure that the majority of data information is preserved in the in-context examples, which is also the aim of dataset pruning.

Based on the aforementioned analysis, we propose to utilize the influence function (Koh and Liang, 2017), which has exhibited efficacy in dataset pruning, to select examples for ICL. However, recent studies on ICL have revealed that the relevance between the query source and in-context examples is critical for ICL. Furthermore, the influence function requires the gradient of parameters, which is computationally expensive and inefficient. To tackle the aforementioned issues, we suggest a two-stage method (I^2CL). Firstly, relevant examples for the query input are recalled, which ensures the correlation between the examples and the query source. Second, our meta-gradient-based influence function is utilized to calculate the influence score for each recalled example. Finally, based on the influence score, in-context examples are selected from the recalled examples. Notably, our framework compresses important information from the training set into the in-context examples, thereby enhancing the performance of ICL. Additionally, our framework is data-independent, relies solely on a small number of model parameters, and does not require the training of any additional models. Numerous experiments indicate that I^2CL shows a significant improvement of an average of 5.90% on five different real-world datasets using multiple language models.

2 Background

2.1 In-Context Learning

The ICL scenario of LLMs can be regarded as a conditional text generation problem. Concretely, the probability of generating a target $y$ is conditioned on the context $C$, which includes $k$ examples and the source $x$. Therefore, the probability can be expressed as:

$$p_{LLM}(y \mid C, x) = \prod_{t=1}^{T} p(y_t \mid C, x, y_{<t})$$

where LLM denotes the parameters of the large language model, and $C = \ldots$
{x_1, y_1, x_2, y_2, \ldots, x_k, y_k} is a context string concatenating k training instances. For example, (x_k, y_k) is concatenated with the special character, e.g., “\n” or “Sentence: x; Sentiment y.” which is denoted as p_k. In this paper, we have different example sets C at different stages, C_1 in the first stage, and C_2 in the second stage, where C_2 is a subset of C_1.

Dai et al. (2022) explains language models as meta-optimizers and understands ICL as a kind of implicit finetuning:

\[ \tilde{F}_{ICL}(q) = W_{ZSL}q + \sum_i \left( W_K x_i \otimes (W_K x_i)^\top \right) q \]
\[ = W_{ZSL}q + \Delta W_{ICL}q \]
\[ = (W_{ZSL} + \Delta W_{ICL}) q. \]

where ZSL denotes the zero-shot learning, which only contains the source x; x ∈ R^d is the input representation of a query token t, and q = W_Q x ∈ R^d is the attention query vector; x’ ∈ R^d denotes the input representations of the example’s token; W_Q, W_K, W_V ∈ R^{d×d} are the projection matrices for computing the attention queries, keys, and values, respectively. Dai et al. (2022) regards W_V X’ as some meta-gradients, which are used to compute the updated matrix ΔW_{ICL}.

2.2 Dataset Pruning

Investigating the data redundant problem not only helps to improve the training efficiency but also helps us understand the representation ability of small data and how many training samples are required and sufficient for a learning system. (Yang et al., 2022) proposed to use the Influence Function to accurately and fast estimate the parameter change caused by weighting an example p for the training dataset. The influence of weighting p on the parameters is given by:

\[ \mathcal{I}_{\text{param}}(p) = \left. \frac{d\hat{\delta}_p}{d\delta} \right|_{\delta=0} = -H^{-1}_\hat{\theta}\nabla \theta L(p, \hat{\theta}), \]

where \[ H_\hat{\theta} = \frac{1}{n} \sum_{p_i \in D} \nabla^2 \theta L(p_i, \hat{\theta}) \] is the Hessian and positive definite by assumption. \[ \mathcal{I}_{\text{param}}(p) \in \mathbb{R}^N, \] N is the number of network parameters, D is the original dataset. After getting the weighting of each example p, (Yang et al., 2022) propose generalization-guaranteed pruning or cardinality-guaranteed pruning to get the final compressed dataset D.

3 Method

3.1 Recall

Given the training dataset D and the query source x, we use BM25 (Robertson et al., 2009) to retrieve a set of relevant examples C_1 for x. For each example p_j in D, the BM25 score, denoted as R(p_j, x), is computed. This score reflects the relevance of example p_j to the query x. Specifically:

\[ R_j = \text{BM25}(p_j, x), \]

where R_j is the relevance score of example p_j with respect to query x.

Subsequently, we form the set C_1 which consists of the top-N examples with the highest relevance scores:

\[ C_1 = \{p_j | j = 1, 2, \ldots, N\}, \]

where N is the number of examples we wish to recall for the given query x.

3.2 Influence-Aware Rerank

For each p in C_1, we calculate the input representation of tokens p as P and the meta-gradient \[ G_p = W_V P. \] To compute \[ T_{\text{param}}(p) \] in Eq. (2), we require the Hessian of p for the parameter W_V, which necessitates the computation of second-order derivatives. However, we only have access to first-order derivatives approximations of the parameters. Considering that LLMs typically employ cross-entropy loss and maximum likelihood estimation (MLE) for fine-tuning, we have opted to employ the Fisher matrix as an approximation of the Hessian (Barshan et al., 2020). The key to the approximation process is as follows:

\[ \nabla^2 f(x) \approx \nabla f(x)\nabla f(x)^\top \]

Then, combining the Eq. (2) with Eq. (5), the expression of the influence function for p is:

\[ T_{\text{param}}(p) = -H^{-1}_{\hat{\theta}} G_p, \]

where \[ H_{\hat{\theta}} = \frac{1}{n} \sum_{p_i \in D} G_p G_p^\top. \]

The score of C_1 is determined by a combination of the influence score and the relevance score, represented as:

\[ S = \{||T_{\text{param}}(p_1)||^2_F + R_1, \ldots, ||T_{\text{param}}(p_N)||^2_F + R_N\}. \]

Finally, the K in-context learning examples in C_2 are chosen by:

\[ C_2 = \{p_i | i \in I\}, \] where \[ I = \arg \max_{|I| \leq K} \frac{1}{N} \] \[ S. \]
4 Experiments

In this section, we empirically verify the efficiency of our approach. The source code and all experiments have been shared at https://github.com/ke-01/ICL.

4.1 Experiments setup

This section introduces the detailed information about our experiments.

Models. We use the open source GPT2 model family (Radford et al., 2019) (i.e., GPT2-Small, GPT2-Medium, GPT2-Large, GPT2-XL) as a representative of large models to verify the effectiveness of our method.

Datasets. We use five datasets spanning four tasks: linguistic analysis, hate speech detection, tweet classification, and semantic similarity. Specifically, we employ Linguistic Acceptability dataset (Cola) (Warstadt et al., 2019), online hate speech detection dataset (Ethos and Ethos-disability) (Mollas et al., 2022), Tweet_eval_stance_feminist and Tweet_eval_stance_hillary (Barbieri et al., 2020) from Twitter, and Sentences Involving Compositional Knowledge dataset (Sick) (Marelli et al., 2014). We use Accuracy and $F_1$ score as evaluation metrics. Detailed dataset statistics and the prompt templates used can be found in Appendix A.2 and Appendix A.1.

Implementation Details. In the study, we chose $K = 3$ and $K = 4$ demonstrations to contrast example selection methods from training data. We set $N = 100$ for all models. Sentences were either truncated or supplemented to have a uniform length at 50% of the average sentence length. Although using multiple transformer layers’ meta-gradient might be beneficial, considering the time efficiency, we used the first layer and obtained higher accuracy than baseline models.

Baselines. Considering the model proposed in this paper is unsupervised and requires no training, it possesses a higher generalizability and operational efficiency compared to models that undergo supervised training. To ensure a fair comparison, our primary baseline is the unsupervised BM25-based In-Context Example Selection. Previous work (Wang et al., 2023; Gupta et al., 2023) has demonstrated that BM25 constitutes a robust baseline for demonstration selection, hence we juxtapose our methodology against BM25. The demonstrations selected by the BM25 are utilized across all GPT2 models.

4.2 Overall Performance

Tables 1 and 2 display results for three and four ICL examples, respectively. Observing the last two rows, our method consistently outperforms across all models and datasets. Using three and four examples, we surpass BM25 by averages of 5.17% and 6.64% in all metrics. Specifically, accuracy sees improvements of 6.33% and 7.80% over BM25. This underscores our approach’s superiority. We found
some higher model performance with three ICL examples compared to four, which can be explained by overfitting and example quality. Overfitting in few-shot learning means too many examples leads to adaptation to specific instances rather than general patterns, reducing accuracy on unseen data. Furthermore, if the additional fourth example is of lower quality or relevance, it can degrade model performance.

5 Related Work

5.1 In-context Learning

In-context learning (ICL) has emerged as a fresh approach in natural language processing (NLP), where large models predict based solely on contexts supplemented by several examples (Dong et al., 2022; Shin et al., 2022; Zhang et al., 2023; Bai et al., 2023). Numerous studies have sought to modify, improve, and comprehend ICL, encompassing topics like prompt tuning (Kim et al., 2022; Wang et al., 2022a; Mishra et al., 2022), intrinsic mechanism analysis (Chan et al., 2022; Li et al., 2023; Garg et al., 2022), evaluations (Srivastava et al., 2023; Wang et al., 2022b), and its use across various fields (Sun, 2023; Sun et al., 2024a,b), among others.

5.2 Demonstration Selection

The goal of demonstration selection is to identify optimal examples for ICL. (Liu et al., 2022) demonstrated that choosing the nearest neighbors as in-context examples is an effective approach. The used distance measures include the pre-set L2 distance or the cosine similarity based on sentence embeddings. They introduced KATE, an unsupervised kNN retriever for in-context example selection. (Rubin et al., 2022) suggested a two-phase retrieval process for demonstration selection. For a given input, it initially employs an unsupervised retriever (like BM25) to retrieve similar candidate examples and then uses a supervised retriever, EPR, to pick demonstrations from these candidates. Recent studies indicate that LLMs exhibit strong sensitivity to the examples chosen, resulting in significant performance variations (Nie et al., 2022), dependency on example sequence (Lu et al., 2022), and at times, an insensitivity to the actual labels (Min et al., 2022).

Our research focuses on reducing training overhead and condensing crucial data from the training set into in-context examples, which in turn amplifies the ICL’s effectiveness.

6 Conclusion

In summary, inspired by LLMs and ICL’s potential, we devised a two-stage method using the influence function for optimal in-context example selection. Our approach ensures relevance with the query source and efficiently determines influence scores. The result is an enhancement in ICL performance, with our experiments validating our model’s effectiveness. Our framework stands out due to its data-independent nature and minimal re-
liance on model parameters.

7 Limitation and Future Work
Given resource constraints and page limitations, we provide limited validation in this paper. Our model is a model-agnostic and free-training approach that can be applied to various in-context learning selection models. In the future, we will validate the effectiveness of our model on more large-scale language models, baselines, and datasets.

8 Acknowledgements
This work was funded by the National Key R&D Program of China (2023YFA1008704), the National Natural Science Foundation of China (No. 62376275, 62377044), Beijing Key Laboratory of Big Data Management and Analysis Methods, Major Innovation & Planning Interdisciplinary Platform for the “Double-First Class” Initiative, PCC@RUC, funds for building world-class universities (disciplines) of Renmin University of China. Supported by the Outstanding Innovative Talents Cultivation Funded Programs 2024 of Renmin University of China.

References


Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.


A APPENDIX FOR REPRODUCIBILITY

A.1 Dataset

In our experiments, we employ Linguistic Acceptability dataset (Cola) (Warstadt et al., 2019), online hate speech detection dataset (Ethos and Ethos-disability) (Mollas...
**Table 3: Template examples.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Example</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sick</td>
<td>[S1] The question is: [S2] True or False? Answer: [Label]</td>
<td>True, Not sure, False</td>
</tr>
<tr>
<td>Cola</td>
<td>Review: [S1] Acceptability: [Label]</td>
<td>unacceptable, acceptable</td>
</tr>
<tr>
<td>Ethos-disability</td>
<td>There is a hate based on disability in [S1] True or False? Answer: [Label]</td>
<td>False, True</td>
</tr>
<tr>
<td>Tweet_eval-stance_feminist</td>
<td>Tweet: [S1] Sentiment: [Label]</td>
<td>none, against, favor</td>
</tr>
<tr>
<td>Tweet_eval-stance_hillary</td>
<td>Tweet: [S1] Sentiment: [Label]</td>
<td>none, against, favor</td>
</tr>
</tbody>
</table>

**Table 4: Dataset statistics.**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Training Size</th>
<th>Valid Size</th>
<th>Label class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sick</td>
<td>4439</td>
<td>495</td>
<td>3</td>
</tr>
<tr>
<td>Cola</td>
<td>8551</td>
<td>1043</td>
<td>2</td>
</tr>
<tr>
<td>Ethos-disability</td>
<td>346</td>
<td>87</td>
<td>2</td>
</tr>
<tr>
<td>Tweet_eval-stance_feminist</td>
<td>597</td>
<td>67</td>
<td>3</td>
</tr>
<tr>
<td>Tweet_eval-stance_hillaryt</td>
<td>620</td>
<td>69</td>
<td>3</td>
</tr>
</tbody>
</table>

et al., 2022), Tweet_eval-stance_feminist and Tweet_eval-stance_hillary (Barbieri et al., 2020) from Twitter, and Sentences Involving Compositional Knowledge dataset (Sick) (Marelli et al., 2014). Table 4 shows more detailed statics of the datasets.

### A.2 Templates.

Based on (Min et al., 2022; Yoo et al., 2022), we employ minimal templates. For the GPT-2 series model, we distinguish between input and label using a space, and this is also applied between demonstration examples. Table 3 presents templates from five datasets, where blue represents the fixed section, and black varies depending on the example content.