Se^2: *Se*quential Example *Se*lection for In-Context Learning

Anonymous ACL submission

Abstract

The remarkable capability of large language models (LLMs) for in-context learning (ICL) needs to be activated by demonstration examples. Prior work has extensively investigated the selection of examples for ICL, predominantly following to a "select then organize" paradigm, such approaches often neglect the internal relationships between examples and exist an inconsistency between the training and inference. In this paper, we formulate the problem as a ***se***quential ***se***lection problem and introduce Se^2, a sequential-aware method that leverages the LLM's feedback on varying context, aiding in capturing inter-relationships and sequential information among examples, significantly enriching the contextuality and relevance of ICL prompts. Meanwhile, we utilize beam search to seek and construct example sequences, enhancing both quality and diversity. Extensive experiments across 23 NLP tasks from 8 distinct categories illustrate that Se^2 markedly surpasses competitive baselines and achieves over 42% relative improvement over random selection. Further in-depth analysis show the effectiveness of proposed strategies, highlighting Se^2's exceptional stability and adaptability across various scenarios. Our code will be released to facilitate future research.

1 Introduction

Large Language Models (LLMs) (Radford et al., 2019; Brown et al., 2020; Zhang et al.; Touvron et al., 2023a) have demonstrated remarkable capabilities in handling a wide range of problems and tasks through in-context learning (ICL). ICL allows these models to learn from a limited number of examples without the need for parameter updates. However, as highlighted by (Wei et al., 2022c; Wang et al., 2022; Wei et al., 2022b), the effectiveness of ICL heavily depends on the careful selection of demonstration examples. The demonstration selection is crucial because, as Liu et al. (2021); Rubin et al. (2021); Su et al. (2022); Zhang et al. (2022); Cheng et al. (2023) have shown, there can be significant variance in performance based on the examples used.

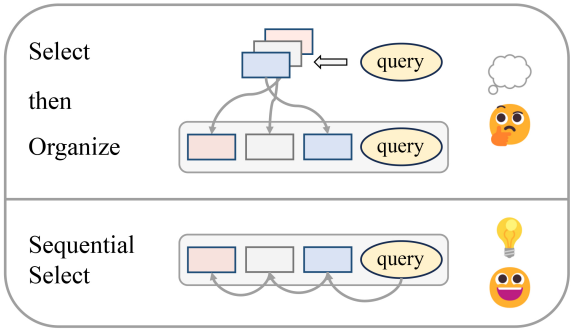


Figure 1: The "select then organize" paradigm and ours sequential example selection.

Given the diverse nature of NLP tasks, researchers have proposed various heuristic criteria for example selection, including entropy (Lu et al., 2022; Wu et al., 2023), influences (Nguyen and Wong, 2023), uncertainty (Diao et al., 2023), and so on. Early approaches focused on retrieving examples based on similarity between the input and the examples (Liu et al., 2022; Su et al., 2022). More recent efforts (Rubin et al., 2021; Cheng et al., 2023) have explored selecting examples based on their contribution to output performance. The above methods select examples and concatenate them together, following the "select then organize" paradigm. However, as shown in Figure 1, these methods often overlook the impact of the relationship between examples. To address this, sequential selection methods have been introduced, with Zhang et al. (2022) proposing a reinforcement learning (RL) approach to formalize the process. Yet, this approach faces limitations in modeling and applicability due to the constraints of the RL framework.

Selecting an optimal sequence of examples from

a large candidate pool is an NP-hard problem. To overcome this challenge, we propose a novel method for example selection, called Se^2, which selects the ideal sequence by modeling the conditional probability of the example sequence given the current context. Se^2 employs a sequential-aware model that utilizes feedback from a LLM across a diverse set of prompting examples as training data. It enables the model to formulate the interrelationships and sequential information between examples. During inference, we use beam search to construct a sequence of examples, which can enhance the quality and diversity of prompt.

We empirically validated the effectiveness of Se^2 at 23 popular tasks, including question answering, paraphrase detection and so on. Se^2 outperformed competitive baselines, achieving a 42% relative improvement over random selection. Our quantitative evaluations demonstrate the advantages of Se^2, showing a significant performance boost with little variance from the sequential training pattern and improved demonstration sequence quality through beam search. Moreover, Se^2 exhibits strong transferability, enhancing larger LLMs with feedback from smaller scoring LLMs. Case study also reveals Se^2's ability to identify example sequences with inherent logical relationships.

Overall, our contributions are summarized as follows:

- In this paper, we explore a novel sequential example selection paradigm for ICL and underscore the importance of a sequential approach in the selection process.
- We propose Se^2, a sequence-aware method that can adeptly handle sequential relationships, generating ideal example sequences as in-context prompt.
- Through extensive experimentation on 23 popular benchmarks, Se^2 demonstrates significant performance improvements, highlighting its ability to uncover and leverage the intrinsic connections between selected examples.

2 Preliminary

In-context learning (ICL) is a pivotal capability of Large Language Models (LLMs), enabling these models to undertake a variety of tasks by observing task prompts without updating parameters. In-context prompts typically comprise sequences of multiple examples. Formally, a K-shot prompt for ICL consist of K input-output pairs. Given a test sample (x_{test}, y_{test}), LLMs predicts $\hat{y}$ based on the in-context prompt and input x_{test}:

$$\hat{y} = \text{LLM}(e_K \oplus, ..., \oplus e_1 \oplus x_{test}) \quad (1)$$

where $e_i = (x_i, y_i)_{i=1}^{K}$ represents a demonstration example from a example pool $\mathcal{E}$, K denotes the shot number and $\oplus$ is the concatenation operation. Our objective is to optimize the in-context prompt by seeking the ideal example sequence $\{e_1, \ldots, e_K\}$ in $\mathcal{E}$ for x_{test} , aiming to make the LLM's prediction $\hat{y}$ match the ground truth y_{test}.

3 Method

In this work, we depart from the traditional "select then organize" paradigm and propose a novel sequential example selection method, namely Se^2, to construct in-context prompts. This section outlines our method, including example scoring, sequence construction, training, and the inference pipeline.

3.1 Example Scoring

As described in Section 2, LLM generates $\hat{y}$ conditioned on prompts composed of examples. Prior work (Rubin et al., 2022; Cheng et al., 2023) indicates that selecting examples solely based on semantic similarity dose not yeild optimal performance. An intuitive and more generalized approach involves scoring by the LLM itself. Our method is adaptable to various NLP tasks, encompassing both Natural Language Understanding (NLU) and Natural Language Generation (NLG). We apply the following scoring functions for them. Given a data instance (x, y) and an example e, we measures the benefit of e for (x, y) by:

NLU

$$\text{S}_{\text{NLU}}(x, y, e) = \frac{\text{LH}(y \mid e \oplus x)}{\sum_{y' \in \mathcal{Y}} \text{LH}(y' \mid e \oplus x)}, \quad (2)$$

where the label space is $\mathcal{Y}$, $\text{LH}(\cdot \mid \cdot)$ is the per-token conditional likelihood of the LLM.

NLG

$$\hat{y} = \text{LLM}(e \oplus x) \quad (3)$$

$$\text{S}_{\text{NLG}}(x, y, e) = \text{metric}(y, \hat{y}), \quad (4)$$

where $\text{metric}(\cdot)$ is the task-specific metric (e,g., Rouge (Lin, 2004)) to compare the prediction $\hat{y}$ and ground truth y.

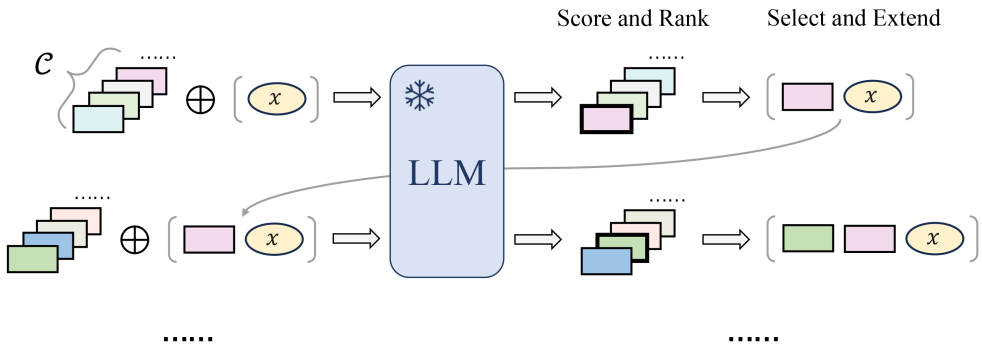


Figure 2: The process of Sequence Construction, mainly includes sampling examples, scoring and ranking examples for dynamic input, and selecting and extending example sequence.

3.2 Context Sequence Construction

Previous methods (Cheng et al., 2023; Rubin et al., 2022) concatenated different examples with input x of data instance (x, y) and scored them separately, but this process ignored the sequential information and inherent connections among examples. Additionally, it was not consistent with the K-shot ICL setting during inference. To address this, we construct the example sequence and find reasonably supervised signals to align this setting. Specifically, we sample L candidate examples from the example pool $\mathcal{E}$ as $\mathcal{C}$, where L is the sample size. We score and rank $\mathcal{C}$ using frozen LLM. In the next step, we select a example e_c from scored $\mathcal{C}$ based on its rank:

$$p(\text{rank}) = \frac{f(\text{rank})}{\Sigma_{\text{rank}'=1}^{L} f(\text{rank}')} \tag{5}$$

$$f(\text{rank}) = \exp(-\text{rank}) \tag{6}$$

where $\text{rank} \in [1, L]$, the higher-scoring example is more likely to be selected and the diversity of the data is preserved. We iteratively update the context input to $e_c \oplus x$, continuing the process of sampling, scoring, ranking and extending context sequences. Algorithm 1 shows the context sequence construction procedure. Through this process, we approximate the optimal example sequence incrementally and obtain the example selection signals for dynamic context inputs, thereby establishing the foundation for the training process.

Algorithm 1 Context Sequence Construction

Input: data instance (x, y), example pool $\mathcal{E}$, shot number K, sample size L
Output: Training Data $\mathcal{D}$ for (x, y)

```
for i=1 to K do
    C ← Random Sample L examples from E
    Score and rank C for (x, y) based on formula 2 and 4
    Append {x, y, C} and their scores to D
    Select e_c from C based on formula 5
    x ← e_c ⊕ x
end for
```

3.3 Training

After completing the sequence construction, we obtained feedback from the LLM on candidate examples for varying context inputs. We then initialize two encoders $E_e(\cdot)$ for examples and $E_x(\cdot)$ for inputs. The aim is to maximize the bi-encoder score between the current input and its most effective example. Accordingly, the top-1 example with the highest score is selected as the current input's positive example e^+. For the set of negative samples $\mathcal{N}$, we sample B examples from $\mathcal{E}$ and choose the bottom-B candidates as hard-negative examples to enhance the model's discriminatory capability. The training loss is defined using InfoNCE (van den Oord et al., 2019):

$$\mathcal{L}(x, e^+, \mathcal{N}) = -\log \frac{\exp(\text{sim}(x, e^+))}{\sum_{e' \in \{\mathcal{N} \bigcup e^+\}} \exp(\text{sim}(x, e'))} \tag{7}$$

where scores are computed via inner product: $\text{sim}(\cdot, \cdot) = E_e(\cdot)^\top E_x(\cdot)$. Observing the few-shot example sequence data allows the model to discern the optimal item for dynamic input sequences.

3.4 Inference

During inference, we encode the entire example pool $\mathcal{E}$ using the trained $E_e(\cdot)$ and index them. For a test input x_{test}, we identify beneficial examples—those with the highest inner product scores, as illustrated in Figure 3. However, greedy search may not be globally optimal for sequential selec-

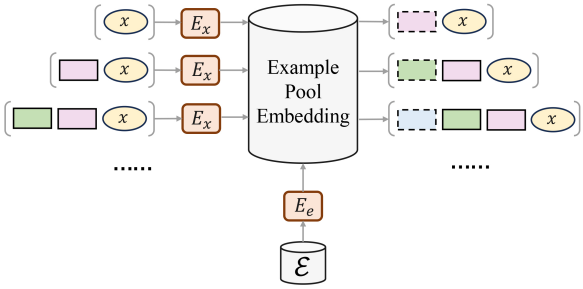


Figure 3: The process of searching for example sequences during inference using beam search. We draw the case of $w = 1$ to illustrate this briefly.

tion. Therefore, we utilize beam search to increase the searching space. We compute the representation of x_{test} with $E_x(\cdot)$ and search for the top-w examples, where w is beam size. These examples are then concatenated with the current inputs as new context sequence, aligning with the training process. This process is repeated, encoding inputs and seeking examples to maintain the w best candidate sequences until each includes K examples. The final predictions are generated by LLM conditioned on the in-context prompts, and evaluated using the corresponding metric for each task.

4 Experiments

We conducted extensive experiments on a wide range of NLP tasks. The results and analysis of the experiments demonstrate the effectiveness and advantages of our method.

4.1 Experiment Settings

Task and Dataset We utilized a total of 23 tasks across 8 distinct categories, including Paraphrase Detection, Common Reasoning, Natural Language Inference, Story Generation, Data-to-Text Generation, Question Answering, Sentiment Analysis, and Text Summarization, based on references (Wei et al., 2022a; Cheng et al., 2023; Li et al., 2023). The prompt templates, and dataset details for different tasks are provided in Appendix A.

Implementation Details We use GPT-Neo-2.7B (Black et al., 2022) as the scoring and inference LLM for most experiments, in line with its widespread adoption in prior research (Rubin et al., 2022; Cheng et al., 2023). Both encoders were initialized with "BERT-base-uncased" (Devlin et al., 2019). For computational efficiency, we randomly select up to 10k data points for each task to construct the training data and example pool while maintaining class balance in classification tasks. Training instances lacking positive e^+ were filtered out.

We configured the sample size $L = 50$, the (hard) negative number $B = 20$, the shot number $K = 3$, and the beam size $w = 3$ to balance the performance and efficiency. The candidate example sequence with the highest score within the beam is selected as the in-context prompt. For comprehensive details on training hyperparameters and additional experimental specifics, refer to Appendix B.

Method Comparison We compared our method with prior competitive ICL methods using GPT-Neo-2.7B as inference LLM, including: **1. Random Selection**: We randomly sample demonstrations from the example pool, repeating 10 times. We report the average performance as "Random" and the best performance as "Best-of-10". **2. BM25** (Robertson and Zaragoza, 2009): A commonly used sparse retriever that extends TF-IDF to rank relevant examples for the test input. **3. SBERT** (Reimers and Gurevych, 2019): A dense retriever by computing sentence embedding. Specifically, we follow Rubin et al. (2022) to take "paraphrase-mpnet-basev2" as encoder. **4. Instructor**[1] (Su et al., 2023): A new competitive model for computing text embeddings given instructions. It trained on 330 tasks and achieved SOTA on 70 diverse embedding tasks. We compared ours with the released instructor-base model [2]. **5. Active Example Selection (AES)**[3] (Zhang et al., 2022): An RL-based method to select examples. For fair comparison, we extended its tasks and re-trained it with GPT-Neo-2.7B. **6. UPRISE**[4] (Cheng et al., 2023): A recently proposed prompt retriever for cross-task and different LLMs. We re-trained task-specific UPRISE that aligns settings for fair comparison.

Notably, the parameters of our encoders are identical to SBERT and UPRISE, so there is no extra burden on deployments and applications.

4.2 Main Results

Table 1 present the overall results of ICL methods at various benchmarks. We found that random sampling does lead to sizable gains compared to

[1] https://instructor-embedding.github.io/

[2] We use the new version checkpoint released on 2024/01/21: https://huggingface.co/hkunlp/instructor-base

[3] https://github.com/ChicagoHAI/active-example-selection. GPT-2 Medium is used in the original paper.

[4] https://github.com/microsoft/LMOps/tree/main/uprise

Method	Paraphrase Detection						Common Reasoning		
	MRPC		QQP		PAWS	Average	COPA	HellaSwag	Average
	Acc.	F1	Acc.	F1	Acc.		Acc.	Acc.	
Zero-shot	46.6	46.0	48.4	42.1	51.8	47.0	67.0	54.5	60.8
Random	61.4	73.1	45.5	47.2	50.4	55.5	71.8	53.7	62.8
Best-of-10	69.4	81.3	63.2	53.9	55.8	64.7	74.0	54.4	64.2
BM25	58.8	70.2	55.3	55.7	49.8	58.0	68.0	54.8	61.4
SBERT	58.8	70.1	58.1	55.9	49.0	58.4	67.0	55.0	61.0
Instructor	58.8	70.0	56.4	57.4	49.1	58.3	71.0	**55.2**	63.1
AES	64.2	N/A	63.0	N/A	51.7	N/A	N/A	N/A	N/A
UPRISE	76.0	84.2	77.9	74.3	49.2	72.3	72.0	54.3	63.2
Se^2	**77.9**	**85.6**	**79.2**	**75.6**	**58.4**	**75.3**	**76.0**	54.6	**65.3**

Method	Natural Language Inference						Story Generation		
	MNLI-m	MNLI-mm	QNLI	SNLI	RTE	Average	Roc Story	Roc Ending	Average
	Acc.	Acc.	Acc.	Acc.	Acc.		Rouge-L	Rouge-L	
Zero-shot	35.3	36.6	50.9	35.3	34.3	38.5	5.8	3.2	4.5
Random	35.9	35.5	51.6	33.7	51.7	41.7	14.6	17.4	16.0
Best-of-10	41.3	40.2	53.2	35.0	54.5	44.8	19.8	17.4	18.6
BM25	36.8	36.8	51.9	38.0	53.4	43.4	10.7	17.5	14.1
SBERT	37.1	38.1	52.7	39.5	48.4	43.2	10.4	17.4	13.9
Instructor	39.4	40.1	53.5	40.3	49.5	44.6	11.6	17.4	14.5
AES	43.2	29.5	61.5	35.0	47.5	43.3	N/A	N/A	N/A
UPRISE	62.9	64.8	72.5	75.5	55.2	66.2	18.2	**18.0**	18.1
Se^2	**69.8**	**69.8**	**80.2**	**78.4**	**56.0**	**70.8**	**20.4**	17.8	**19.1**

Method	Data-to-Text Generation			Question Answering			
	CommonGen	E2E NLG	Average	ARC-C	ARC-E	OBQA	Average
	Rouge-L	Rouge-L		Acc.	Acc.	Acc.	
Zero-shot	14.2	7.6	10.9	29.5	48.3	43.0	40.3
Random	28.8	43.1	36.0	30.8	56.9	45.3	44.3
Best-of-10	33.9	48.7	41.3	31.8	60.5	47.2	46.5
BM25	29.9	47.9	38.9	31.8	61.6	47.2	46.9
SBERT	29.9	42.6	36.3	32.0	62.9	47.2	47.4
Instructor	30.0	43.3	36.6	32.4	64.0	47.6	48.0
AES	N/A	N/A	N/A	N/A	N/A	N/A	N/A
UPRISE	33.0	51.9	42.5	32.9	**64.1**	49.8	**48.9**
Se^2	**34.6**	**53.4**	**44.0**	**33.3**	63.3	**50.0**	**48.9**

Method	Sentiment Analysis				Text Summarization			
	SST-2	SST-5	Sent140	Average	AGNews	Gigaword	AESLC	Average
	Acc.	Acc.	Acc.		Acc	Rouge-L	Rouge-L	
Zero-shot	52.4	28.2	64.3	48.3	38.4	1.4	1.8	13.9
Random	52.9	22.5	66.2	47.2	35.4	18.1	3.1	19.9
Best-of-10	57.3	27.7	73.3	52.8	47.8	21.8	9.2	26.3
BM25	65.6	27.5	73.3	55.5	81.7	23.3	11.6	38.9
SBERT	72.5	22.6	73.5	56.2	85.0	19.4	7.8	37.4
Instructor	80.8	20.1	**85.5**	62.1	87.3	19.9	10.3	39.2
AES	82.8	20.9	69.1	57.6	78.2	N/A	N/A	N/A
UPRISE	78.8	52.6	84.4	71.9	91.4	**25.8**	13.4	43.5
Se^2	**89.0**	**52.7**	83.8	**75.2**	**91.6**	**25.8**	**14.0**	**43.8**

Table 1: Main results on various tasks. We bold the best results. The column "Average" is the mean performance of the same category task. "N/A" means that the task or metric is not applicable to the corresponding method.

Beam Size	Candidates	PD.	CR.	NLI.	SG.	DTG.	QA.	SA.	TS.
$w = 1$	/	74.96	64.84	**70.96**	18.78	43.93	48.63	74.99	43.81
$w = 2$	top-1	74.99	63.79	70.65	19.01	43.90	48.63	75.04	43.71
	avg	74.91	64.30	70.61	19.03	43.87	48.34	75.05	43.85
$w = 3$	top-1	**75.34**	**65.30**	70.84	**19.10**	**44.00**	**48.87**	**75.20**	43.81
	avg	75.32	65.15	70.79	19.00	43.85	48.36	75.19	**44.20**

Table 2: The average performance of the 8 categories of tasks at different beam size.

zero-shot. This demonstrates the necessity of providing examples for LLMs in downstream tasks. Compared to random sampling, Se^2 shows over 42% average relative improvement. BM25 and SBERT have a significant gain over random sampling. The text embedding model Instructor benefits from extensive training data and performs better than BM25 and SBERT. It shows that the appropriate demonstraion selection is beneficial. In comparison, Se^2 achieves an average relative improvement of 25%, but with fewer parameters and less training data. This demonstrates the effectiveness of our data construction and modeling approach.

Meanwhile, AES, UPRISE and Se^2 perform better due to using LLM's feedback to find appropriate examples. Compared with UPRISE, Se^2 benefits from modeling dynamic context inputs, capturing relationships between examples, as well as the beam search strategy, showing significantly better performance on the majority of tasks. The result shows that the sequential selection is also essential for K-shot ICL. Se^2 and AES all consider the sequence formulation. However, the features and model of AES are relatively simple, making it difficult to capture sufficient semantic information, resulting in poor performance on some tasks, and its applicability is limited. In contrast, Se^2 achieves better performance by sequential model training and can be applied to all tasks.

4.3 Analysis

In this section, we delve into evaluating the effectiveness of Se^2's strategy, examining the impact of search strategy, the stability of selected examples, the transferability across models, and providing a case study for a tangible demonstration of Se^2's advantages.

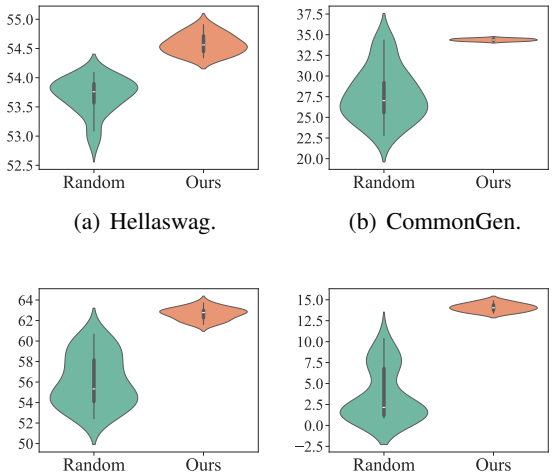
On the Effect of Search Strategy In our method, we employ beam search to seek candidate demonstration example sequences, with beam size denoted as w. In this section, We investigate the effect of our search strategy by setting $w = [1, 2, 3]$, where $w = 1$ corresponds to a greedy search strategy. Performance evaluations of searched candidate sequences are depicted in Table 2, with "top-1" indicating results from the highest scoring prompts among w candidates, and "avg" representing the average performance across w candidate prompts. We report the average performance of the 8 categories of tasks. Results demonstrate that as beam size w increases, enlarging the search space, there's a notable improvement in performance, signifying Se^2 found more optimal candidate example sequences. Moreover, "top-1" performance tends to be better than "avg", indicating Se^2's ability to accurately identify and prioritize the most effective candidates, thereby affirming the strategy's efficacy. Overall, Se^2's approximate search strategy successfully identifies superior example sequences, enhancing downstream task performance.

On the stability of selected examples As suggested by Lu et al. (2022); Zhao et al. (2021a), ICL performance can vary significantly, with different in-context examples causing fluctuations from near state-of-the-art (SOTA) performance to random guessing, and different orderings may lead to strong fluctuations in the performance.

In this section, we explore the stability of Se^2 across 4 different ategories of tasks, HellaSwag, Roc Story, ARC-E, and AESLC. We randomly sample 5 distinct example sets for each task, conducting a full permutation on them and recording $5 * 3! = 30$ outcomes. We also perform a full permutation on 3 candidate sequences obtained through Se^2 with $w = 3$, recording $3 * 3! = 18$ results. Figure 4 displays the performance distribution, highlighting that random demonstration selection results in substantial performance variability, aligning with prior findings (Lu et al., 2022; Li and Qiu, 2023). In contrast, Se^2 not only enhances per-

Inference Model	Method	1-shot	2-shot	4-shot	8-shot	16-shot
GPT2-XL	BM25	51.40	51.38	52.26	52.53	53.14
	UPRISE	64.29	65.83	66.07	66.31	66.23
	Se^2	67.19	67.66	68.46	68.26	68.57
GPT-Neo-2.7B	BM25	51.69	52.62	54.22	55.24	55.27
	UPRISE	66.93	66.98	66.92	67.10	66.84
	Se^2	69.11	69.64	69.60	69.72	69.63
LLaMA2-7B	BM25	57.31	57.87	60.19	63.34	65.44
	UPRISE	69.36	69.17	71.09	72.01	72.40
	Se^2	71.79	72.12	72.97	73.27	73.67

Table 3: Se^2's average performance across 17 NLU tasks from 1-shot to 16-shot on LLMs with parameter scales 1.5B to 7B.



(a) Hellaswag. (b) CommonGen.

(c) ARC-E. (d) AESLC.

Figure 4: The performance distribution of Se^2 and random sampling when tuning in-context prompts.

	HellaSwag	CommonGen	ARC-E	AESLC
Original	**54.62**	**34.47**	**62.92**	**14.44**
Permutation	54.55	34.34	62.59	13.99

Table 4: Average performance of original sequences and their full permutations obtained by Se^2.

formance but also ensures significantly greater stability, indicating its selected examples effectively support and describe the tasks. Further analysis of the average performance of the 3 original sequences against their permutations reveals Se^2's original sequences outperform alternative permutations, showcasing the benefits of modeling sequential information and example relationships for ideal sequence generation without the need for ordering reconsideration.

On the transferability of Se^2 In above experiments, we show the results when the inference model and the scoring model are consistent. As the LLM is scaling up, aligning the scoring and inference models is time-consuming and resource-intensive. Thus, We explored Se^2's example selection effectiveness across different LLMs, transferring it to 2 various size model GPT2-XL (1.5B) (Radford et al., 2019) and LLAMA2-7B (Touvron et al., 2023b), comparing against BM25 and UPRISE baselines. Table 3 illustrates that Se^2 consistently outperforms BM25 and UPRISE from 1-shot to 16-shot across models ranging from 1.5B to 7B parameters. Notably, despite Se^2 being trained on 3-shot data, it shows a progressive performance increase with more examples, especially enhancing smaller models due to their greater reliance on high-quality examples for task execution, which is in line with the recent findings (Wang et al., 2023).

Case Study For an intuitive grasp of the Se^2's effectiveness, we present an interesting case on the ARC-C task. In Figure 5, We compare prompts derived from UPRISE and Se^2, highlighting the examples' answers with underscores. UPRISE-selected examples, while relevant to science, were somewhat broad, leading to incorrect model predictions. Conversely, Se^2-selected examples related to gravity and energy conversion to kinetic energy, exhibiting a conceptually progressive relationship that guided the model to the correct answer. This demonstrates how Se^2's sequential formulation fosters a rich, informative relationship between examples, enhancing LLM prediction accuracy.

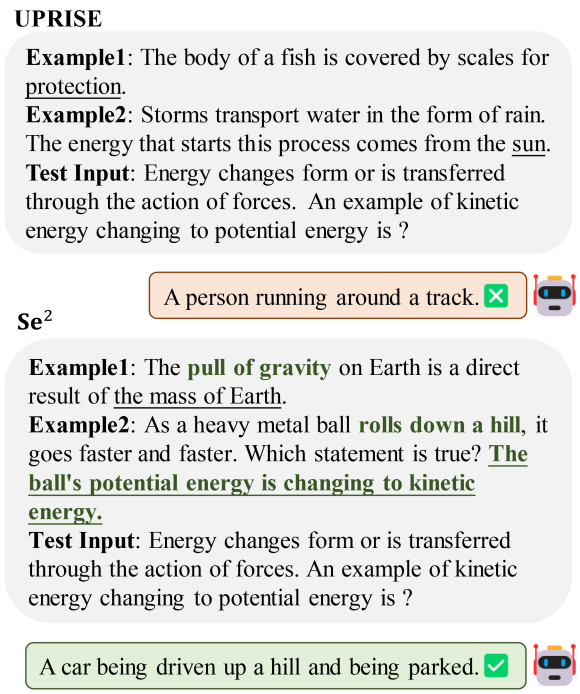


Figure 5: An example on ARC-C where Se^2 helps LLM infer the correct answer, but UPRISE does not.

5 Related Work

Selecting appropriate in-context examples for LLMs is crucial for enhancing performance in downstream tasks. Due to the variety of NLP tasks, various heuristic criteria have been proposed, including entropy (Lu et al., 2022; Wu et al., 2023), diversity (Ye et al., 2023b; Hongjin et al., 2022), influences (Nguyen and Wong, 2023), sensitivity (Chen et al., 2023), and uncertainty (Diao et al., 2023). Another research line posits that beneficial examples are those semantically similar to the test inputs, leading Agrawal et al. (2023) to adopt BM25 (Robertson and Zaragoza, 2009) to retrieve examples in machine translation, Liu et al. (2022); Lee et al. (2022) to retrieve example with a dense encoder (Devlin et al., 2019). However, the assumption that model performance consistently correlates with semantic similarity is not universally reliable. Consequently, methods like EPR (Rubin et al., 2022), CEIL (Ye et al., 2023a), UDR (Li et al., 2023), and UPRISE (Cheng et al., 2023) utilize LLMs to score examples so as to select examples that the model truely prefers.

Despite these advancements, existing methods often overlook the internal relationships between examples, a factor that significantly influences model performance (Zhang et al., 2022). Meanwhile, ICL prompts are often sequences composed of multiple examples, modeling such a sequence is a kind of NP-hard problem. Zhang et al. (2022) introduced a RL algorithm to develop generalized policies for example selection. However, this method simply utilizes the number of examples as the feature representation and MLP as the Q-network within the constraints of RL settings, struggles to capture the nuanced semantic features of natural language. This paper proposes a novel approach to the sequential example selection paradigm for ICL, demonstrating a more effective method for enhancing ICL performance than previously considered.

6 Conclusion

In this work, we explore a new sequential example selection paradigm for ICL and propose Se^2, a sequential-aware method that can end-to-end select ideal example sequences for test inputs. Se^2 leverage the feedback from LLMs across a broad spectrum of inputs and examples. This approach not only facilitates the modeling of sequential information and the intrinsic connections between examples but also empowers the construction of prompts through the beam search strategy. Through extensive experimentation, Se^2 demonstrated superior performance over established baselines, highlighting its ability to generate more effective prompts through beam search.

Our analysis revealed Se^2's robustness in example selection, contributing to its stability and adaptability across different tasks and LLMs. This work underscores the importance of sequential example selection in improving ICL, shedding new light on ICL.

Limitations

Our research primarily utilized GPT-Neo-2.7B (Black et al., 2022) for experiments, constrained by computational resource limitations. This choice, while effective, also meant confronting the model's sequence length limitations, potentially leading to the truncation of overly long inputs. Recent advancements, such as those by Ge et al. (2024), have explored innovative approaches for compressing inputs and prompts to expand the usable context window, presenting a valuable direction for future exploration. Furthermore, evolving from a BERT-based encoder to a more capable decoder-only model represents another promising avenue for enhancing our framework's capacity and effectiveness. We see exploring this in our future work.

Additionally, the presence of various biases within LLMs, as identified in studies by Zhao et al. (2021b); Fei et al. (2023), poses a challenge. Since our method relies on feedback from LLMs, it's conceivable that our results could be influenced by these inherent biases. Addressing this, we recognize the exploration of strategies to achieve fair and explainable outcomes from these complex models as an essential and promising area of research.

Our discussions aim to inspire further investigation within the community, encouraging advancements that address these limitations and propel the field of NLP research forward.

References

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1.

Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, and Peter Clark. 2021. Think you have solved direct-answer question answering? try arc-da, the direct-answer ai2 reasoning challenge.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. Gpt-neox-20b: An open-source autoregressive language model.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2023. On the relation between sensitivity and accuracy in in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 155–167, Singapore. Association for Computational Linguistics.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. UPRISE: Universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12318–12337, Singapore. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.

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

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14014–14031, Toronto, Canada. Association for Computational Linguistics.

Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

SU Hongjin, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.

Young-Jun Lee, Chae-Gyun Lim, and Ho-Jin Choi. 2022. Does GPT-3 generate empathetic dialogues? a novel in-context example selection method and automatic evaluation metric for empathetic dialogue generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 669–683, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. Association for Computational Linguistics.

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 the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

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 the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, Montréal, Canada. Association for Computational Linguistics.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*.

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

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

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 the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023. Learning to retrieve in-context examples for large language models. In *EACL 2024*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022b. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022c. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1423–1436, Toronto, Canada. Association for Computational Linguistics.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023a. Compositional exemplars for in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39818–39833. PMLR.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Veselin Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2023b. Complementary explanations for effective in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4469–4484, Toronto, Canada. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Rui Zhang and Joel Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 446–456, Florence, Italy. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models, 2022. *URL https://arxiv. org/abs/2205.01068*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021a. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021b. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Appendices

A Overview of tasks and datasets

- **Paraphrase Detection**: MRPC (Dolan and Brockett, 2005), QQP (Wang et al., 2018), and Paws Wiki (Zhang et al., 2019).
- **Commonsense Reasoning**: COPA (Roemmele et al., 2011), and HellaSwag (Zellers et al., 2019).
- **Natural Language Inference**: MNLI-m/mm (Williams et al., 2018), QNLI (Rajpurkar et al., 2018), SNLI (Bowman et al., 2015), and RTE (Bentivogli et al., 2009).
- **Story Generation**: Roc Story and Roc Ending (Mostafazadeh et al., 2016).
- **Data-to-Text Generation**: CommonGen (Lin et al., 2020), and E2E NLG (Dušek et al., 2019).
- **Question Anwering**: ARC-c/e (Bhakthavatsalam et al., 2021), and OBQA (Mihaylov et al., 2018).
- **Sentiment Analysis**: SST-2/5 (Socher et al., 2013), and Sentiment140 (Go et al., 2009).
- **Text Summarization**: AESLC (Zhang and Tetreault, 2019), AGNews (Zhang et al., 2015), and Gigaword (Napoles et al., 2012).

The detailed datasets statistical information are shown in Table 6 for . We mainly use instruction templates from FLAN (Wei et al., 2022a) to convert task datasets into natural language instructions, except for the SST5, Roc Story, and Roc Ending tasks. Each task dataset corresponds to approximately seven templates. For SST5, ROC Story, and ROC Ending tasks which are not included in FLAN, we use a single simple template, readers can refer to Table 9 for details.

B HyperParameters and Implementation Details

The subscripts in Section 3.3 range from 1 to L, but in the actual $\exp(\cdot)$ calculation, they range from 0 to $L-1$. It is fine to use any other probability.

We extended the available tasks to AES and redid the task-specific training, but AES is currently limited in the tasks and metrics it can support. Specifically, AES requires the model to predict a single target token, the tasks that contain multiple tokens in the answer, such as multiple choice with textual content, and generation tasks are not supported for the time being. We report them as "N/A" in Table 1.

Hyperparameter	Assignment
Computing Infrastructure	8 V100-32GB GPUs
Number of epochs	6
Batch size per GPU	8
Maximum sequence length	512
Maximum learning rate	5e-6, 1e-5, 3e-5
Optimizer	Adam
Adam epsilon	1e-8
Adam beta weights	0.9, 0.999
Learning rate scheduler	warmup linear
Weight decay	0.0
Warmup steps	1000
Learning rate decay	linear

Table 5: Hyperparameter settings

In Section 4.3, we found that some models do not follow the prompts as expected in the generation task, resulting in incomparable performance. For fair comparison, we report the average performance across all NLU tasks.

C Settings and Performance of Different Work

In the experiments in this paper, we compare Se^2 with representative SOTA methods. Due to variations in the language models, tasks, instruction templates, training and testing datasets, as well as evaluation metrics used by different methods, and due to limitations in computational resources, it is hard to include all related work in the comparison. So we collected the settings and reported performance of previous work in Table 7 and Table 8 for reference only.

Category	Task	Train	Test	Metric
Paraphrase Detection	MRPC	3,668	408	Accuracy
	QQP	363,846	40,430	Accuracy
	PAWS	49,401	8,000	Accuracy
Common Reasoning	COPA	400	100	Accuracy
	HellaSwag	39,905	10,042	Accuracy
Natural Language Inference	MNLI-m	392,702	9,815	Accuracy
	MNLI-mm	392,702	9,832	Accuracy
	QNLI	104,743	5,463	Accuracy
	SNLI	549,367	9,824	Accuracy
	RTE	2,490	277	Accuracy
Story Generation	Roc Story	87,526	9,799	Rouge-L
	Roc Ending	87,906	9807	Rouge-L
Data-to-Text Generation	CommonGen	67,389	4,018	Rouge-L
	E2E NLG	33,525	1,847	Rouge-L
Question Answering	ARC-C	1,117	1,165	Accuracy
	ARC-E	2,241	2,365	Accuracy
	OBQA	4,957	500	Accuracy
Sentiment Analysis	SST-2	67,349	872	Accuracy
	SST-5	8,534	2,210	Accuracy
	Sent140	1,600,000	359	Accuracy
Text Summarization	AGNews	120,000	7,600	Accuracy
	Gigaword	2,044,465	730	Rouge-L
	AESLC	13,181	1,750	Rouge-L

Table 6: The categories, statistics, split and evaluation metrics of each dataset.

Method	Mainly utilized LMs	Shot Number
UDR (Li et al., 2023)	GPT-Neo-2.7B	8
CEIL (Ye et al., 2023a)	GPT-Neo-2.7B	50
AES (Zhang et al., 2022)	GPT-2-M	4
TopK+MDL (Wu et al., 2023)	GPT-2-XL	8
LENS (Li and Qiu, 2023)	GPT-2-L	8 for SST-2 and AGNews, 10 for SST-5
GlobalE (Lu et al., 2022)	GPT-2/3 Series	2 for AGNews, 4 for others
Ours	GPT-Neo-2.7B	3

Table 7: The setting of related method.

Method	E2E NLG	Commongen	Roc Story	Roc Ending
UDR (Li et al., 2023)	32.6(BLEU-4)	27.1(BLEU-3)	17.6(BLEU-1)	24.7(BLEU-1)
Ours	53.4(Rouge-L)	34.6(Rouge-L)	20.4(Rouge-L)	17.8(Rouge-L)

Method	COPA	Hellaswag	AGNews	SST-2	SST-5	MRPC	QNLI	SNLI	RTE
UDR (Li et al., 2023)	72.8	/	91.5	92.4	50.5	/	/	83.6	65.3
CEIL (Ye et al., 2023a)	/	43.2	/	/	47.05	80.15	85.41	/	/
AES (Zhang et al., 2022)	/	/	70.8	81	/	/	/	/	/
TopK+MDL (Wu et al., 2023)	/	/	87.94	91.51	40.27	/	61.43	58.77	/
LENS (Li and Qiu, 2023)	/	/	77.9	86.3	44.9	/	/	/	/
GlobalE (Lu et al., 2022)	/	/	78.1	80.2	43.2	/	/	/	51.3
Ours	76	54.6	91.6	89	52.7	77.9	80.2	78.4	56

Table 8: The performance of related method on the same task.

Task Category: Paraphrase Detection

Task: MRPC
Prompt Templates:
("Here are two sentences: {sentence1} {sentence2} Do they have the same meaning?", "{answer}"),
("Here are two sentences: {sentence1} {sentence2} Are the two sentences saying the same thing?", "{answer}"),
("{sentence1} {sentence2} Do the above sentences mean the same thing?", "{answer}"),
("{sentence1} {sentence2} Please tell me if the sentences above mean the same.", "{answer}"),
("{sentence1} {sentence2} Are these sentences conveying the same meaning?", "{answer}"),
("{sentence1} {sentence2} If the first sentence is true, is the second one also true?", "{answer}"),
("{sentence1} {sentence2} Are these two sentences paraphrases of each other?", "{answer}"),
("Do the following two sentences have the same meaning? {sentence1} {sentence2}", "{answer}"),
("Do these two sentences mean the same thing? {sentence1} {sentence2}", "{answer}"),
("Do these sentences have the same meaning? {sentence1} {sentence2}", "{answer}"),

Task: QQP
Prompt Templates:
("{question1} {question2} Would you say that these questions are the same?", "{answer}"),
("{question1} {question2} Do those questions have the same meaning?", "{answer}"),
("{question1} {question2} Are these two questions inquiring about the same information?", "{answer}"),
("{question1} {question2} Please tell me if those questions are the same.", "{answer}"),
("{question1} {question2} Are these two questions paraphrases of each other?", "{answer}"),
("First question: {question1} Second question: {question2} Are these two questions asking the same thing?", "{answer}"),
("Question 1: {question1} Question 2: {question2} Are questions 1 and 2 asking the same thing?", "{answer}"),
("Question 1: {question1} Question 2: {question2} Would the answer to these two questions be the same?", "{answer}"),
("Are the following two questions the same? {question1} {question2}", "{answer}"),
("Do these questions have the same meaning? {question1} {question2}", "{answer}"),

Task: PAWS
Prompt Templates:
("{sentence1} {sentence2} Do these sentences mean the same thing?", "{answer}"),
("{sentence1} {sentence2} Are these two sentences paraphrases of each other?", "{answer}"),
("1. {sentence1} 2. {sentence2} Are these two sentences paraphrases of each other?", "{answer}"),
("(1) {sentence1} (2) {sentence2} Do these two sentences mean the same thing?", "{answer}"),
("Sentence 1: {sentence1} Sentence 2: {sentence2} Do these two sentences convey the same information?", "{answer}"),
("Do these two sentences from wikipedia have the same meaning? {sentence1} {sentence2}", "{answer}"),
("Same meaning? {sentence1} {sentence2}", "{answer}"),
("Are these paraphrases? {sentence1} {sentence2}", "{answer}"),
("Do these mean the same? {sentence1} {sentence2}", "{answer}"),
("Please check if these have the same meaning. Answer ÿesïf they do, otherwise ño. {sentence1} {sentence2}", "{answer}"),

Task Category: Paraphrase Detection

Task: COPA
Prompt Templates:
("{premise}Ẅhat is the {question}?", "{answer}"),
("Here is a premise: {premise}Ẅhat is the {question}?", "{answer}"),
("{premise}Ẅhat is the {question} of the preceding sentence?", "{answer}"),
("{premise}Ẅhat is a plausible {question}?", "{answer}"),
("Based on the following sentence, what is the {question}? {premise}", "{answer}"),
("{premise}{question}: ", "{answer}"),
("What is the {question} of the following sentence? {premise}", "{answer}"),
("Answer the following question about this sentence: {premise}Ẅhat is the {question}?", "{answer}"),

Task: HellaSwag
Prompt Templates:
("What happens next in this paragraph? {context}", "{answer}"),
("Continue writing the next sentence in this paragraph: {context}", "{answer}"),

("Continue writing the next sentence. {context}", "{answer}"),
("This is a test of commonsense. Complete the next sentence: {context}", "{answer}"),
("Write the next sentence in this paragraph: {context}", "{answer}"),
("How does the next paragraph end? {context}", "{answer}"),
("What most naturally follows? {context}", "{answer}"),
("What happens next? {context}", "{answer}"),
("What is the most logical next event? {context}", "{answer}"),
("Write the next sentence in the following story. {context}", "{answer}"),

Task Category: Natural Language Inference

Task: MNLI
Prompt Templates:
("Premise: {premise}Hypothesis: {hypothesis}Does the premise entail the hypothesis? Yes, No, or Maybe?", "{answer}"),
("Premise: {premise}Hypothesis: {hypothesis}Is the hypothesis entailed by the premise? Yes, No, or Maybe?", "{answer}"),
("Here is a premise: {premise}Here is a hypothesis: {hypothesis}Is it possible to conclude that if the premise is true, then so is the hypothesis? Yes, No, or Maybe?", "{answer}"),
("Sentence 1: {premise}Sentence 2: {hypothesis}Is this second sentence entailed by the first sentence? Yes, No, or Maybe?", "{answer}"),
("Sentence 1: {premise}Sentence 2: {hypothesis}If the first sentence is true, then is the second sentence true? Yes, No, or Maybe?", "{answer}"),
("Based on the premise {premise}, can we conclude the hypothesis {hypothesis}is true? Yes, No, or Maybe?", "{answer}"),
("Premise: {premise}If this premise is true, what does that tell us about whether it entails the hypothesis {hypothesis}? Yes, No, or Maybe?", "{answer}"),
("Premise: {premise}Based on this premise, is the hypothesis {hypothesis}true? Yes, No, or Maybe?", "{answer}"),
("If {premise}, can we conclude that {hypothesis}? Yes, No, or Maybe?", "{answer}"),
("{premise}Does it follow that {hypothesis}? Yes, No, or Maybe?", "{answer}"),

Task: QNLI
Prompt Templates:
("Does the sentence {sentence}answer the question {question}?", "{answer}"),
("Does the sentence {sentence}provide a valid answer to the question {question}?", "{answer}"),
("Is {sentence}a good answer to the question {question}?", "{answer}"),
("Does {sentence}correctly answer the question of {question}?", "{answer}"),
("Does {sentence}contain the correct answer to {question}?", "{answer}"),
("Q: {question} A: {sentence} Does the answer correctly answer the question?", "{answer}"),
("Question: {question} Answer: {sentence} Is the question answered in a satisfactory fashion?", "{answer}"),
("Question: {question} Is {sentence} a good answer to this question?", "{answer}"),
("Question: {question} Is {sentence}the correct answer?", "{answer}"),

Task: SNLI
Prompt Templates:
("If {premise}, does this mean that {hypothesis}? Yes, No, or Maybe?", "{answer}"),
("If {premise}, can we conclude {hypothesis}? Yes, No, or Maybe?", "{answer}"),
("If {premise}, does it logically follow that {hypothesis}? Yes, No, or Maybe?", "{answer}"),
("Based on the sentence {premise}, is the sentence {hypothesis}a true sentence? Yes, No, or Maybe?", "{answer}"),
("Premise: {premise} Hypothesis: {hypothesis} Can we conclude that the hypothesis is true if the premise is true? Yes, No, or Maybe?", "{answer}"),
("Premise: {premise} Hypothesis: {hypothesis} Given the premise, can we conclude the hypothesis? Yes, No, or Maybe?", "{answer}"),
("Here is a premise: {premise}Here is a hypothesis: {hypothesis}. Does the premise tell us whether the hypothesis is true? Yes, No, or Maybe?", "{answer}"),
("Is it possible to conclude that {premise}if {hypothesis}? Yes, No, or Maybe?", "{answer}"),
("Is the premise {premise}true if {hypothesis}? Yes, No, or Maybe?", "{answer}"),

Task: RTE
Prompt Templates:
("{premise} Based on the paragraph above can we conclude that {hypothesis}? Yes, No, or Maybe?", "{answer}"),
("{premise} Based on that paragraph can we conclude that this sentence is true? {hypothesis} Yes, No, or Maybe?", "{answer}"),

("{premise} Can we draw the following conclusion? {hypothesis} Yes, No, or Maybe?", "{answer}"),
("{premise} Does this next sentence follow, given the preceding text? {hypothesis} Yes, No, or Maybe?", "{answer}"),
("{premise} Can we infer the following? {hypothesis} Yes, No, or Maybe?", "{answer}"),
("Read the following paragraph and determine if the hypothesis is true: {premise} Hypothesis: {hypothesis} Yes, No, or Maybe?", "{answer}"),
("Read the text and determine if the sentence is true: {premise} Sentence: {hypothesis} Yes, No, or Maybe?", "{answer}"),
("Can we draw the following hypothesis from the context? Context: {premise} Hypothesis: {hypothesis} Yes, No, or Maybe?", "{answer}"),
("Determine if the sentence is true based on the text below: {hypothesis} {premise} Yes, No, or Maybe?", "{answer}"),

Task Category: Story Generation

Task: Roc Story
Prompt Templates:
("Beginning of the story: {question} Please continue the story: ", "{target}"),

Task: Roc Ending
Prompt Templates:
("Beginning of the story: {question} Please write the end of the story: ", "{target}"),

Task Category: Data-to-Text Generation

Task: CommonGen
Prompt Templates:
("Concepts: {concepts}. Write a sentence that includes all these words.", "{target}"),
("Keywords: {concepts}. What is a sentence that includes all these keywords?", "{target}"),
("Here are some concepts: {concepts}. What is a sentence about these concepts?", "{target}"),
("Produce a sentence which mentions all of these concepts: {concepts}.", "{target}"),
("Write a sentence about the following things: {concepts}.", "{target}"),
("Generate a sentence that includes all the following words: {concepts}.", "{target}"),

Task: E2E NLG
Prompt Templates:
("Attributes: {meaning_representation}. Produce a detailed sentence about this restaurant.", "{target}"),
("Data: {meaning_representation}. Can you generate a sentence about this data?", "{target}"),
("Data: {meaning_representation}. What is a sentence that describe this data?", "{target}"),
("Here are some keywords about a restaurant: {meaning_representation}. Write a sentence that describes the following attributes of a restaurant.", "{target}"),
("Here is some data about a restaurant: {meaning_representation}. Write a sentence that includes the following data about a restaurant.", "{target}"),
("Sentence: {meaning_representation}. Can you represent the content in this sentence in data form?", "{target}"),
("Write a sentence about a restaurant with all the following attributes: {meaning_representation}.", "{target}"),
("Write a sentence that is about a restaurant with all the following properties: {meaning_representation}.", "{target}"),
("Produce a detailed sentence about a restaurant using the following words: {meaning_representation}.", "{target}"),
("Generate a descriptive sentence about a restaurant using the following words: {meaning_representation}.", "{target}"),

Task Category: Question Answering

Task: ARC
Prompt Templates:
("{question}", "{answer}"),
("Question: {question} Answer:", "{answer}"),
("Question: {question} What is the correct answer to the question from the following choices?", "{answer}"),
("Q: {question} What is the correct answer to this question?", "{answer}"),
("What is the answer? {question}", "{answer}"),
("Answer the question {question}", "{answer}"),
("{question} Pick the answer from these options.", "{answer}"),

Task: OBQA
Prompt Templates:
("{fact} {question}", "{answer}"),
("Read this fact: ̈{fact}N̈ow answer this question: ̈{question}̈", "{answer}"),

("Given the fact ̈{fact}̈, what is the answer to the question or completion ̈{question}̈", "{answer}"),
("Knowing that ̈{fact}̈, how would one answer ̈{question}̈", "{answer}"),
("Use evidence from the fact that {fact} to answer this question: ̈{question}̈", "{answer}"),
("Fact: {fact} Question: {question} What's the answer?", "{answer}"),
("Use this fact to answer the question: {fact} {question}", "{answer}"),

Task Category: Sentiment Analysis

Task: SST-2
Prompt Templates:
("Review: ̈{sentence}̈Is this movie review sentence negative or positive?", "{answer}"),
("Short movie review: ̈{sentence}̈Did the critic thinking positively or negatively of the movie?", "{answer}"),
("Sentence from a movie review: ̈{sentence}̈Was the movie seen positively or negatively based on the preceding review?", "{answer}"),
("̈{sentence}̈How would the sentiment of this sentence be perceived?", "{answer}"),
("Is the sentiment of the following sentence positive or negative? ̈{sentence}̈", "{answer}"),
("What is the sentiment of the following movie review sentence? ̈{sentence}̈", "{answer}"),
("Would the following phrase be considered positive or negative? ̈{sentence}̈", "{answer}"),
("Does the following review have a positive or negative opinion of the movie? ̈{sentence}̈", "{answer}"),

Task: SST-5
Prompt Templates:
("Review: ̈{sentence}̈. It was", "{answer}"),

Task: Sentiment140
Prompt Templates:
("{text} What is the sentiment of this tweet?", "{answer}"),
("{text} How would the sentiment of this tweet be described?", "{answer}"),
("{text} Describe the sentiment embodied by this tweet.", "{answer}"),
("Tweet: {text} Predict the sentiment of this tweet.", "{answer}"),
("What is the sentiment of the following tweet? Tweet:{text}", "{answer}"),
("How would one describe the sentiment of this tweet? {text}", "{answer}"),

Task Category: Text Summarization

Task: AGNews
Prompt Templates:
("̈{text}̈What is this text about? World, Sports, Business, or Technology?", "{answer}"),
("̈{text}̈Which topic is this article about? World, Sports, Business, or Technology?", "{answer}"),
("̈{text}̈Which is the best summary of this article? World, Sports, Business, or Technology?", "{answer}"),
("̈{text}̈What is this text about? World, Sports, Business, or Technology?", "{answer}"),
("̈{text}̈What best summarizes the content of the above article? World, Sports, Business, or Technology?", "{answer}"),
("Which is this about? ̈{text}̈World, Sports, Business, or Technology?", "{answer}"),
("Which is an appropriate title for this article? ̈{text}̈World, Sports, Business, or Technology?", "{answer}"),
("Select the topic that this about: ̈{text}̈World, Sports, Business, or Technology?", "{answer}"),

Task: Gigaword
Prompt Templates:
("Write a short summary for this text: {text}", "{summary}"),
("Briefly summarize this sentence: {text}", "{summary}"),
("Generate a short summary this sentence: {text}", "{summary}"),
("What is a shorter version of this: {text}", "{summary}"),
("{text} Write a brief summary in a sentence or less", "{summary}"),
("{text} What is a very short summary of the above text?", "{summary}"),
("{text} Summarize the aforementioned text in a single phrase.", "{summary}"),
("{text} Can you generate a short summary of the above paragraph?", "{summary}"),

Task: AESLC
Prompt Templates:
("What is the subject line for this email? {body}", "{subject}"),
("Write a subject line for this message: {body}", "{subject}"),
("{body} Write a subject line for this email.", "{subject}"),
("Here is an email: {body} What is a potential subject line for this email?", "{subject}"),

("{body} Propose a subject line for this email?", "{subject}"),
("This is the content of an email: {body} What was the subject line for this email?", "{subject}"),
("This is an email {body} What is the subject of this email?", "{subject}"),
("{body} Generate a subject line for this email.", "{subject}"),

Table 9: The prompt templates of tasks.