Large Language Models Might Not Care What You Are Saying: Prompt Format Beats Descriptions

Chenming Tang[†] **Zhixiang Wang**[†] **Hao Sun** **Yunfang Wu**[*]
National Key Laboratory for Multimedia Information Processing, Peking University
MOE Key Laboratory of Computational Linguistics, Peking University
School of Computer Science, Peking University
{tangchenming, ekko}@stu.pku.edu.cn {2301213218, wuyf}@pku.edu.cn

Abstract

With the help of in-context learning (ICL), large language models (LLMs) have achieved impressive performance across various tasks. However, the function of descriptive instructions during ICL remains under-explored. In this work, we propose an ensemble prompt framework to describe the selection criteria of multiple in-context examples, and preliminary experiments on machine translation (MT) across six translation directions confirm that this framework boosts ICL performance. But to our surprise, LLMs might not care what the descriptions actually say, and the performance gain is primarily caused by the ensemble format, since it could lead to improvement even with random descriptive nouns. We further apply this new ensemble framework on a range of commonsense, math, logical reasoning and hallucination tasks with three LLMs and achieve promising results, suggesting again that designing a proper prompt format would be much more effective and efficient than paying effort into specific descriptions.

1 Introduction

In-context learning (ICL) boosts the performance of large language models (LLMs) across numerous natural language processing (NLP) tasks, where LLMs are presented with in-context examples containing input and ground truth output (Brown et al., 2020; Dong et al., 2023). Many works have verified the vital role of in-context examples in ICL (Wang et al., 2023; Wei et al., 2023). However, Min et al. (2022) find that ground truth labels might not be the key to ICL performance on classification tasks.

The selection of in-context examples has been proven significant to the performance of ICL (Rubin et al., 2022) and there have been various works on in-context example selection (Agrawal et al., 2023; Li et al., 2023; Ye et al., 2023). Besides diverse approaches of selecting examples, no existing work has tried to explicitly tell LLMs *in what way those specific examples are selected*. We hypothesize that if LLMs are prompted with instructions describing the properties of selected in-context examples, they might learn better from these examples, since instruction following is one of LLMs' most important qualities nowadays (Ouyang et al., 2022; Peng et al., 2023; Zhang et al., 2024). Tang et al. (2024) prompt LLMs with examples selected based on both word-level and syntax-level criteria for machine translation (MT) for better ICL performance. This inspires us to tell LLMs where different in-context examples come from when they are selected by multiple methods.

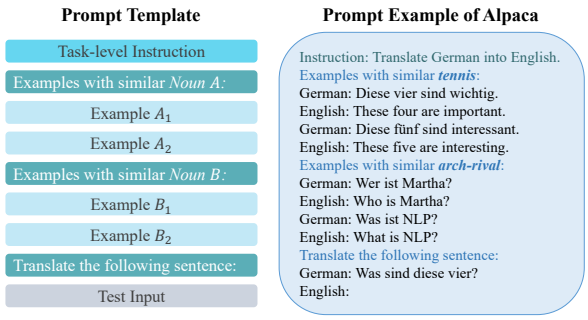


Figure 1: Template and Alpaca's example of *Ensemble*.

In our experiments on MT, we first select in-context examples based on lexical and syntactic similarity for each test input separately. Then we combine both to construct the complete set of examples, with half word-level examples and half syntax-level examples. Further, we devise a novel ensemble prompt framework (as shown in the left part "Prompt Template" of Figure 1), adding example-level instructions to describe that the following examples are with similar words or similar syntax.

Experimental results on MT demonstrate that adding such ensemble prompt framework does improve LLMs' performance over conventional

[*] Corresponding author.
[†] These authors contributed equally.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 26–48
November 4-9, 2025

prompts. Meanwhile, we find that when the example-level descriptions do not correspond to the source of in-context examples or are completely nonsense, LLMs still benefit from the prompt. These surprising results indicate that in fact LLMs might not care what the descriptions say and are more sensitive to the prompt format. In other words, a proper format can be much more effective than well-designed descriptions in ICL.

To further verify the superiority of the ensemble framework, we present empirical evaluations on commonsense, math, logical reasoning and hallucination benchmarks (including nine datasets in total) across three small-scale LLMs (Alpaca, Llama3 and Mistral) and one large-scale LLM (GPT-3.5). The novel prompt framework is able to achieve promising results even with the descriptive nouns in the prompt being random nouns, further suggesting that a proper prompt format would be much more effective and efficient compared with laborious design of detailed and specific descriptions.

There are a few studies very related to our work. Min et al. (2022) find that the labels of in-context examples do not need to be correct for classification tasks. Wei et al. (2023) find that larger language models learn from in-context examples even when the labels are flipped or unrelated. Srivastava et al. (2024) demonstrate that optimizing examples is less effective in some tasks given a high-quality task instruction. Our work is different from the above in that we focus on the meaning of **descriptions** rather than **labels** or **examples** in ICL and our finding is that the format of prompts is more important than carefully designed descriptions.

Our contributions can be summarized as follows:

- For the first time, we specifically analyze the effect of prompt descriptions on ICL performance and find that LLMs might not care what users actually say in descriptions, while they are more sensitive to the prompt format.
- We present a simple yet effective prompt framework that is proven feasible on MT through comprehensive experiments across six translation directions. Promising experimental results on three LLMs further verify the superiority of the novel framework on a range of commonsense, math, logical reasoning and hallucination tasks.

Our code is available at `https://github.com/JamyDon/Format-Beats-Descriptions`.

2 Prompting LLMs for MT

Primarily, we focus on MT, a typical generation task. Recently, various approaches of selecting in-context examples have been proposed for MT (Agrawal et al., 2023; Kumar et al., 2023; Tang et al., 2024). However, no existing work has tried to make LLMs aware of *in what way those specific in-context examples are selected*.

We assume that LLMs would perform better when they are told the reasons for selecting those examples. Tang et al. (2024) select examples based on a combination of word-level and syntax-level criteria, which inspires us to present an ensemble prompt framework to make LLMs clearly know the reasons behind example selection. In addition, to have a comprehensive idea of whether LLMs really know what is said in the descriptions, we design some prompt variants that are less meaningful or completely nonsense.

2.1 In-context Example Selection for MT

For word-level examples, we simply select them using BM25 (Bassani, 2023). For syntax-level examples, we use the top-k polynomial algorithm proposed by Tang et al. (2024) to convert dependency trees into polynomials and compute syntactic similarity based on the Manhattan distances (Craw, 2017) between polynomial terms. For brevity, we denote the syntax-level algorithm by "Polynomial".

To combine word-level and syntax-level examples, we simply concatenate them. For example, the first and the remaining half of examples are selected by BM25 and Polynomial respectively.

2.2 A New Ensemble Prompt Framework

To maintain consistency, all our MT experiments use four in-context examples.

First of all, we use the most regular prompt without any example-level descriptions as baseline (referred to as *Vanilla*), which is shown in Figure 2.

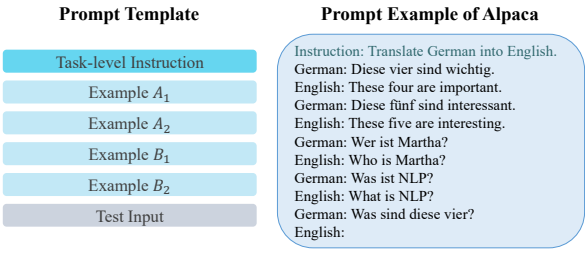


Figure 2: Template and Alpaca's example of *Vanilla*.

In the template, "Task-level Instruction" instructs

the model to do the current task (MT here). "Example A_i" and "Example B_i" denote the i-th example from selection approach A (*e.g.*, BM25) and B (*e.g.*, Polynomial) respectively, all containing both source language inputs and target language translations. "Test Input" refers to the source language input of the test sample, which requires the LLM to translate it into the target language.

Then, we add example-level descriptions for examples from different selection approaches and explicitly instruct the LLM to translate the test input. This prompt framework is referred to as *Ensemble* and is shown in Figure 1 as presented in Section 1. "Noun A" and "Noun B" describe the examples from selection A and B respectively. For example, the two nouns can be "words" and "syntax" to properly describe examples selected by BM25 and Polynomial respectively. In this way, we can conveniently control the example-level descriptions to tell the LLM why those examples are used.

2.3 Experimental Setup

Language	ISO Code	Dataset	#Pairs (M)
German	DE	Europarl	1.8
French	FR	Europarl	1.9
Russian	RU	ParaCrawl	5.4

Table 1: Data statistics.

2.3.1 Datasets

We perform evaluation on the *devtest* set of FLORES-101 (Goyal et al., 2022), which contains 1012 sentences with translations in 101 languages. We experiment between English and three common languages: German, French and Russian. We use Europarl (Koehn, 2005) for German and French and ParaCrawl (Bañón et al., 2020) for Russian as example database, from which we select in-context examples. Detailed statistics are in Table 1.

2.3.2 Evaluation Metrics

We report COMET (Rei et al., 2020) scores from `wmt20-comet-da` [1], which is considered a superior metric for MT today (Kocmi et al., 2021).

2.3.3 Language Models

We experiment with two LLMs commonly used in MT: $\text{XGLM}_{7.5\text{B}}$ (Lin et al., 2022) and Alpaca (Taori et al., 2023). XGLM is a multilingual language model with 7.5B parameters supporting 30 languages that is frequently used in MT. Alpaca is a 7B LLM instruction-tuned from LLaMA (Touvron et al., 2023).

2.3.4 Example Selection

To maintain consistency, all our MT experiments use four in-context examples (4-shot). We evaluate different ways of selecting examples for comparison. Note that if all four examples are selected by the same method, the first two are considered examples from A and the last two are considered from B in the *Ensemble* template in Figure 1.

Random: The four examples are randomly sampled from the example database. We report the average result of three different random seeds.

BM25: We retrieve the top-4 matching examples for each test input using BM25 (Bassani, 2023).

Polynomial: It is rather time-consuming to retrieve examples from databases containing millions of data using the Polynomial algorithm. Following Tang et al. (2024), we instead re-rank the top-100 examples retrieved by BM25 using Polynomial and the top-4 are used as final in-context examples.

BM25 + Polynomial: To combine examples with both lexical and syntactic similarity, we simply concatenate examples from BM25 and Polynomial. Specifically, the first two examples are from BM25 and the remaining two are from Polynomial.

Polynomial + BM25: The first two examples are from Polynomial and the remaining two are from BM25.

2.3.5 Prompts

We design various prompts to explore whether LLMs can benefit from explicit descriptions of examples and whether they really understand the meaning of descriptions.

***Vanilla*:** The normal prompt without any example-level descriptions as shown in Figure 2.

***Ensemble* (Word + Syntax):** Shown in Figure 3a, Noun A and Noun B are "words" and "syntax" respectively, which semantically corresponds to BM25 + Polynomial examples but is converse to Polynomial + BM25.

***Ensemble* (Syntax + Word):** Shown in Figure 3b, Noun A and Noun B are "syntax" and "words" respectively, which semantically matches Polynomial + BM25 examples but mismatches BM25 + Polynomial.

Different *Ensemble* (Word + Syntax): Shown in Figure 3c, Noun A and Noun B are still "words" and "syntax" respectively but the qualifier "simi-

[1] https://huggingface.co/Unbabel/wmt20-comet-da

lar" is replaced with "different". This aims to find out whether LLMs pay attention to the meaning of "different/similar" and care the semantics of descriptions.

***Ensemble* (Word + Semantics):** Shown in Figure 3d, Noun A and Noun B are "words" and "semantics" respectively, which does not semantically match any of our example selection methods.

***Ensemble* (Random + Random):** Shown in Figure 3e, for each input, Noun A and Noun B are different random English nouns sampled using Wonderwords [2], aiming to explore LLMs' understanding of descriptions.

2.4 Main Results

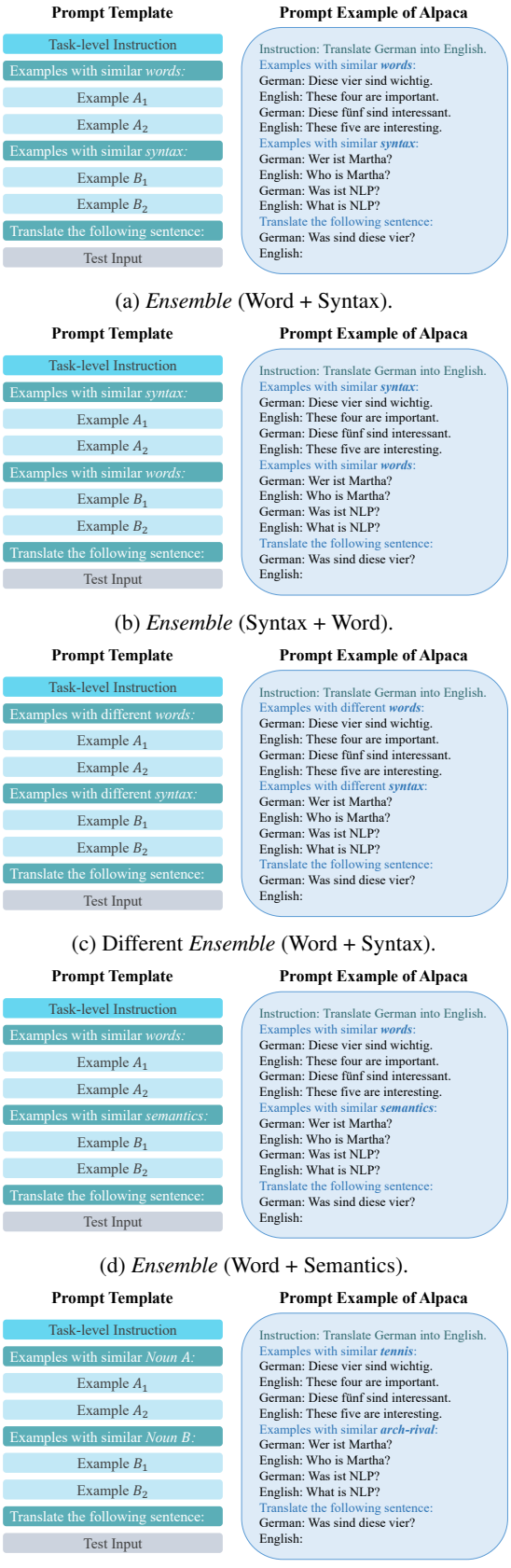
To give a quick view of LLMs' MT performance, Table 2 shows the COMET scores of *Vanilla* baselines averaged over six translation directions.

Example Selection	XGLM	Alpaca
Random	54.07	55.42
BM25	55.00	56.27
Polynomial	55.52	56.13
BM25 + Polynomial	56.17	56.18
Polynomial + BM25	56.18	55.49

Table 2: Results of *Vanilla* baselines of XGLM and Alpaca with different example selection methods, averaged over six translation directions.

Main results are shown in Figure 4. For convenient comparison, we present the performance gain of different *Ensemble* prompts over *Vanilla* with different selections of in-context examples and the results are averaged over six translation directions. For detailed results of different translation directions, please refer to Appendix B.

As can be seen from the results, those **"correct"** prompts, exactly corresponding to the selection of in-context examples (*e.g.*, *Ensemble* (Word + Syntax) with BM25 + Polynomial examples and *Ensemble* (Syntax + Word) with Polynomial + BM25 examples), do bring some help as expected. However, when the prompt does not correspond to the selection of examples (*i.e.*, is **"incorrect"**), the performance improves as well and sometimes even more than those "correct" cases. For example, on XGLM with BM25 + Polynomial examples, *Ensemble* (Syntax + Word) improves more than *Ensemble* (Word + Syntax), even though the former is completely reversed. On Alpaca with BM25 +



(a) *Ensemble* (Word + Syntax).

(b) *Ensemble* (Syntax + Word).

(c) Different *Ensemble* (Word + Syntax).

(d) *Ensemble* (Word + Semantics).

(e) *Ensemble* (Random + Random).

Figure 3: Templates and Alpaca's examples of *Ensemble* prompts.

[2]https://github.com/mrmaxguns/wonderwordsmodule

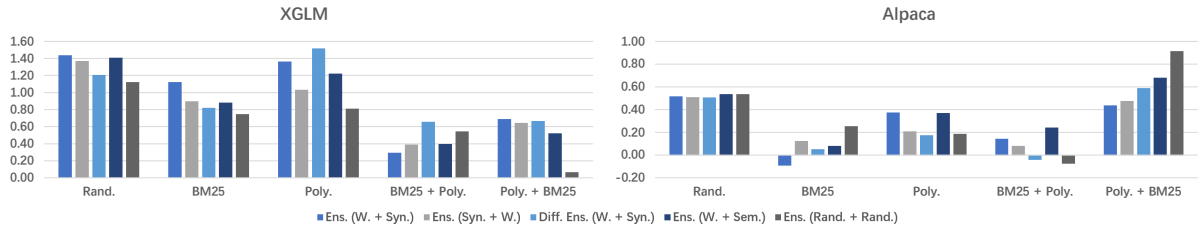


Figure 4: Main results on XGLM and Alpaca, showing the performance gain of different prompts over the *Vanilla* prompt, averaged over all six translation directions. Each cluster presents the results of a selection of in-context examples and each bar in it presents the result of a prompt. "Ens.", "W.", "Syn.", "Sem.", "Diff.", "Rand.", "Poly." refer to "*Ensemble*", "Word", "Syntax", "Semantics", "Different", "Random", "Polynomial", respectively.

Polynomial examples, *Ensemble* (Word + Semantics) improves more than *Ensemble* (Word + Syntax), albeit the examples with similar syntax do not necessarily bear similar semantics. More interestingly, Different *Ensemble* (Word + Syntax), telling the LLM that the in-context examples are with different properties, is able to beat "correct" prompts sometimes (*e.g.*, on XGLM with BM25 + Polynomial examples and Alpaca with Polynomial + BM25 examples).

Surprisingly, no matter how in-context examples are selected and whether the prompts are "correct", *Ensemble* prompts bring improvement in most cases. Even *Ensemble* (Random + Random), in which example-level descriptions are with random nouns and could be completely nonsense (like "examples with similar nobody"), brings improvement in most cases, especially obtaining the most gain on Alpaca with Polynomial + BM25 examples compared with other prompts, correct or incorrect. These results indicate that LLMs might not really take the example-level descriptions into consideration during ICL. In other words, they might not necessarily care what users say in the descriptions.

Compared with proper descriptions, it seems the format of prompts matters more. For example, on Alpaca with Random examples, no matter what the example-level descriptions say, all *Ensemble* prompts bring nearly equal improvement over *Vanilla*. This indicates that *Ensemble* is a superior format compared with *Vanilla* in this case.

To sum up, the experimental results on MT suggest that a proper prompt format leads to better ICL performance of LLMs while a careful design of descriptions might be less effective.

2.5 Ablation Study

To better understand how the *Ensemble* format brings improvement, we perform ablation experiments over the organization of the prompt:

***Ensemble* (Random + Random):** The *Ensemble* prompt with random nouns in the example-level descriptions as described in Section 2.3.

***Single* (Random):** Organized based on Figure 1, but the second description is removed. There is only one example-level description above the four examples, where Noun A is a random noun.

***Single* (Example):** Organized based on Figure 1, but the second description is removed. There is only one example-level description above the four examples, being "Examples:" only, without any further descriptions. This prompt only informs the LLM that the following four instances are examples and does not describe their properties.

***Vanilla* (Translate):** Organized based on Figure 1, but both the two descriptions are removed. The only difference with *Vanilla* is the translation instruction "Translate the following sentence:" before the test input. This prompt only informs the LLM to translate the test input and tells nothing about the in-context examples.

Detailed templates and examples of the above prompts are presented in Appendix A.

Results are presented in Figure 5, showing that removing one or two example-level descriptions or removing the random noun describing the property of in-context examples hurt the performance gain in most cases. On XGLM, only the original *Ensemble* format performs better than *Vanilla*. Alpaca exhibits an abnormal trend when prompted with Polynomial and BM25 + Polynomial examples, where *Ensemble* (Random + Random) cannot outperform other prompts. This may be due to that Alpaca is instruction-tuned and the *Single* or *Vanilla* (Translate) prompts are also friendly to it in some cases because of the post-training stage. But overall, *Single* (Random), *Single* (Example) and *Vanilla* (Translate) still bring less improvement than *Ensemble* (Random + Random) in more than half of the cases.

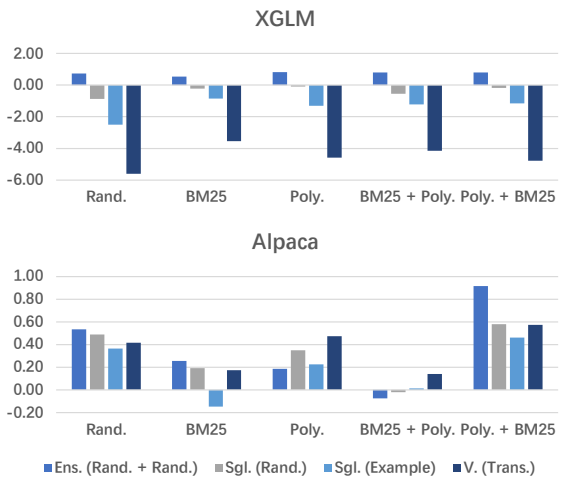


Figure 5: Ablation studies over the organization of the prompt, showing the performance gain of different prompts over *Vanilla*, averaged over all six translation directions. "Rand.", "Poly.", "Ens.", "Sgl.", "V.", "Trans." refer to "Random", "Polynomial", "*Ensemble*", "*Single*", "*Vanilla*", "Translate", respectively.

Ablation experiments suggest that in MT, our proposed *Ensemble* is a relatively superior prompt format, performing better than other variants.

2.6 Analysis via Attention Weights

To have a better idea of the internal mechanism of LLMs when prompted with different prompts, we compute the attention weights between different prompt components. We focus on three components: in-context examples (from A or B, denoted by "Example-A" and "Example-B"), the target position (denoted by "Target") where the model starts to generate predictions (following Wang et al. (2023), we use the final token in the input) and the two descriptive nouns ("Noun-A" and "Noun-B"). We obtain the attention weights averaged over all attention heads from the attention matrix across all the layers. All the results are averaged over all six language directions.

Results comparing *Ensemble* (Word + Syntax) (*EWS*) and *Ensemble* (Random + Random) (*ERR*) on XGLM with BM25 + Polynomial examples are presented in Figure 6 (for results on Alpaca, refer to Appendix C). If the model really cares what the descriptions say, its attention to meaningful descriptive nouns (in *EWS*) should be much greater than those meaningless (in *ERR*). However, in most cases, *EWS* performs no higher than *ERR*, indicating that the model does not really care what the descriptive nouns actually are. "Target to Noun-A" is a special case, where *EWS* is high in shallow layers. But in deeper layers, *EWS* falls behind and *ERR* takes the lead. This shows that the model might pay more attention to the meaningful noun when understanding the context in shallow layers but gradually forgets it when it comes to generation in deeper layers. In a word, the attention weights further confirm our claim that LLMs do not really care what the descriptive nouns are in most cases.

2.7 Discussion

Above results show that LLMs benefit from our *Ensemble* prompts in most cases. However, the benefit comes from a proper format rather than the meaningful descriptions (*e.g.*, "similar words" and "similar syntax"). This demonstrates that LLMs might not care what users say in the descriptions but is more sensitive to the format of prompts. In other words, designing a proper prompt format would be more efficient than paying a lot of effort into looking for a perfect description.

In the next section, we apply *Ensemble* format to more tasks to further verify its generalizability.

3 Generalizing the New Ensemble Prompt Framework to More Tasks

To further verify our conclusion obtained from MT that our proposed *Ensemble* framework improves ICL even when the example-level descriptions are incorrect or meaningless, we perform the comparison between *Vanilla* and *Ensemble* (Random + Random), which we would refer to as *ERR*, on more types of tasks across different language models.

3.1 Experimental Setup

3.1.1 Datasets

We use a total of nine benchmarks, covering four task types: commonsense QA, logical reasoning, arithmetic reasoning, and hallucination detection.

For commonsense QA, we adopt four datasets. The widely-used CSQA (Talmor et al., 2019) features commonsense questions about the world involving complex semantics requiring prior knowledge. StrategyQA (Geva et al., 2021) challenges models to infer implicit reasoning steps using a strategy to answer questions. We also choose two specialized evaluation sets from BIG-bench (Srivastava et al., 2023): Date Understanding, which asks models to infer the date from a context, and Sports Understanding, which involves assessing the plausibility of sentences related to sports.

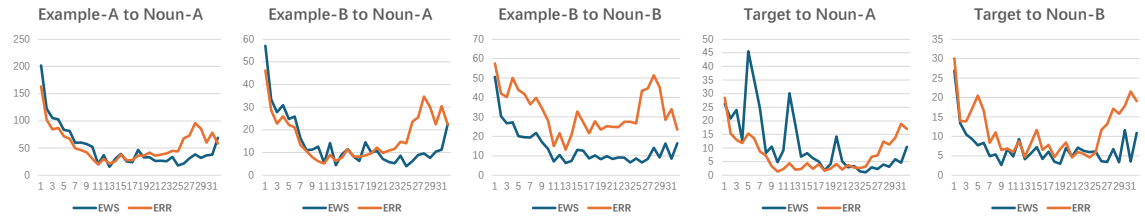


Figure 6: Attention weights (×1e-4) on XGLM of all 32 layers with BM25 + Polynomial examples. *EWS* and *ERR* denotes *Ensemble* (Word + Syntax) and *Ensemble* (Random + Random) respectively.

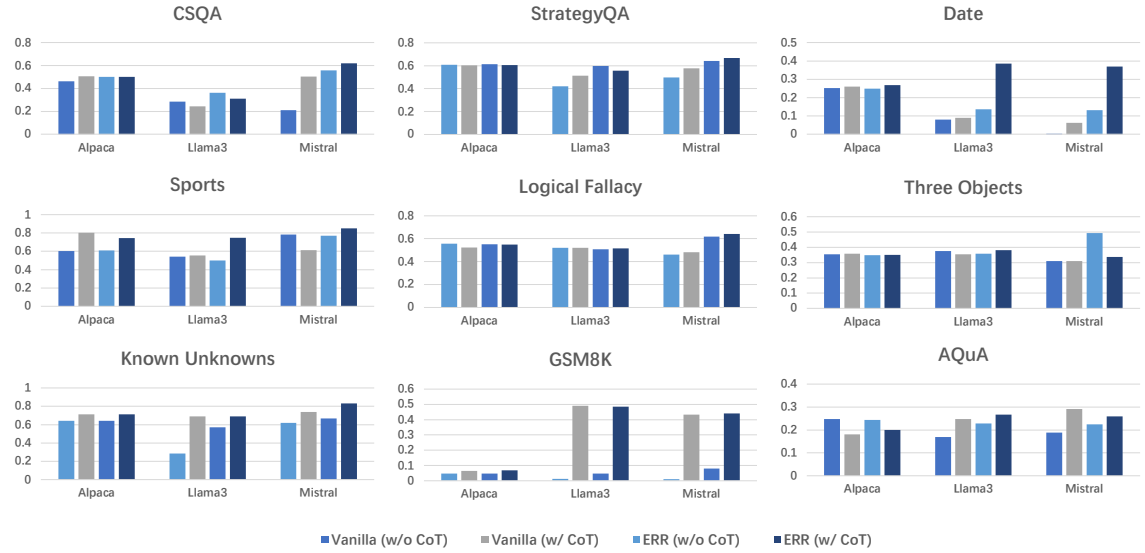


Figure 7: Results on nine datasets across three small-scale models. In the "Date" subplot, the score of Mistral under the *Vanilla* prompt is too low to be a visible bar in the chart.

For logical reasoning task, we choose Logical Fallacy and Three Objects (a subset of Logical Deduction) from Big-bench (Srivastava et al., 2023). Logical Fallacy aims to test the model's ability to identify whether there are fallacies in a given logical reasoning, and Three Objects requires the model to infer the order of a sequence of objects from a set of minimal conditions.

To explore the performance of *ERR* on math word problems, we adopt the following two datasets: GSM8K (Cobbe et al., 2021), which consists of high quality free-response grade school math problems, and AQuA (Ling et al., 2017), containing the algebraic word problems in the form of multiple-choice questions.

In addition, to explore whether *ERR* could alleviate LLMs' hallucination, we choose Known Unknowns from Big-bench (Srivastava et al., 2023).

The number of test inputs for each dataset is listed in Table 3. Details of splitting training set (example database) and test set are in Appendix D.

Dataset	Test Inputs
CSQA	1221
StrategyQA	1012
Date	365
Sports	996
Logical Fallacy	1012
Three Objects	296
Known Unknowns	42
GSM8K	1319
AQuA	254

Table 3: Number of test inputs for each dataset.

3.1.2 Evaluation Metric

These nine datasets are either in the form of multiple-choice questions or free-response questions with standard answers, so we use accuracy as the metric for all of them.

3.1.3 Language Models

We experiment with both instruction-tuned and non-instruction-tuned models to see whether our findings could extend to different kinds of mod-

els. We evaluate three frequently used open source LLMs with around 7B parameters, including Alpaca (Taori et al., 2023), Llama3 (Llama Team, 2024), and Mistral (Jiang et al., 2023), among which Llama3 is a base model before instruction tuning. To assess the effect of *ERR* on more powerful models, we also evaluate GPT-3.5 (Ouyang et al., 2022) [3]. We use Llama-3.1-8B, Mistral-7B-Instruct-v0.2 and gpt-3.5-turbo-0125 [4] for Llama3, Mistral and GPT-3.5 respectively.

3.1.4 Example Selection

Note that randomly selected examples combined with *ERR* have already brought non-trivial improvements to MT. Therefore, for each dataset discussed in this section, we randomly select a uniform set of examples (4-shot) for all test inputs without applying any carefully designed selection method, in order to focus on and verify the simple yet effective and universal nature of *ERR*.

3.1.5 Prompts

We compare *ERR* with *Vanilla* across different datasets and LLMs. Given that these tasks usually involve reasoning, on which chain-of-thought (CoT) is commonly utilized (Wei et al., 2022), we experiment both without CoT ("w/o CoT", which are identical to the original templates) and with CoT ("w/ CoT"). This allows us to examine both the orthogonality and compatibility with CoT of *ERR* , as well as assess its performance across various models and tasks. Specifically, we evaluate *Vanilla* (w/o CoT), *Vanilla* (w/ CoT), *ERR* (w/o CoT), and *ERR* (w/ CoT). Due to space constraints, examples of prompt templates discussed in this section are provided in Appendix E.

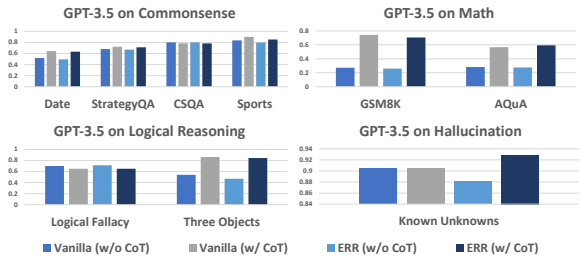


Figure 8: Results of the four types of tasks on GPT-3.5.

[3]We choose this model because it is a commonly used cost-effective API-based LLM and a *de facto* black box baseline.

[4]https://openai.com/api/

3.2 Results of Small-scale Models

Results across all nine datasets and three small-scale models (Alpaca, Llama3 and Mistral) are illustrated in Figure 7. Detailed results are presented in Appendix B.

The results demonstrate that *ERR* (w/ CoT), achieved by integrating CoT with our proposed prompt framework, either significantly outperforms or matches *Vanilla* (w/ CoT) in 25 out of 27 experiments (covering nine datasets and three models). The exceptions are Alpaca on the Sports dataset and Mistral on the AQuA dataset, where *ERR* (w/ CoT) shows somewhat lower performance compared to *Vanilla* (w/ CoT). When CoT is not incorporated, *ERR* generally performs much better than or on par with *Vanilla*, except for the Sports dataset with Llama3, where *ERR* performs a little poorer.

Surprisingly, *ERR* (w/o CoT) sometimes even surpasses *Vanilla* (w/ CoT), suggesting that the *ERR* framework alone can offer more improvements than CoT. This highlights the value of *ERR* and reaffirms that the format plays a crucial role in enhancing LLMs' problem-solving capabilities. In terms of models, the performance of *ERR* on Alpaca is far less impressive than on Llama3 and Mistral, which may be because Alpaca has strong instruction-following capabilities and is more robust to different prompts.

In summary, without using any carefully designed selection methods, directly filling the randomly selected examples into the *ERR* framework brings significant improvement to various reasoning tasks and even alleviates the hallucination of models in most cases, no matter how meaningless and incorrect the example-level descriptions are. Moreover, *ERR* can work perfectly with CoT. Therefore, at least for relatively small models, this simple yet effective trick is worth introducing into prompt engineering for various tasks.

We also experiment with Llama2 (Llama Team, 2023) and the results are in Appendix F. The overall trend is consistent with Llama3.

3.3 Results of GPT-3.5

As shown in Figure 8, *ERR* performs similarly to *Vanilla* across every dataset using GPT-3.5. Although the *ERR* format does not bring significant improvement to these tasks with GPT-3.5 and Alpaca (as shown in Figure 7), the fact remains that the incorrect or meaningless example-level descriptions caused by random nouns do not

have much negative impact on GPT-3.5, a sufficiently powerful model, or Alpaca, which has strong instruction-following capabilities. In some cases, it even slightly improves performance (*e.g.*, *ERR* (w/ CoT) outperforms *Vanilla* (w/ CoT) on AQuA and Known Unknowns). In other words, LLMs might not care what users actually say to describe the provided examples while they are more sensitive to the format of prompts, which is in line with our findings obtained from MT.

3.4 Discussion

Based on the experiments conducted on both small-scale and large-scale models, we can conclude that *ERR* is a simple yet practical and universal prompt framework. It can enhance problem-solving capabilities of small models and be applied to large models without the risk of performance degradation due to the meaningless noise within it. In other words, there might be less need to meticulously select examples or design detailed descriptions. Instead, you can uniformly and efficiently apply *ERR* to various tasks with different models.

As analyzed in Section 2.6, the *ERR* framework can work because LLMs pay less attention to the descriptive nouns while being more sensitive to the overall prompt format. We conjecture that the underlying reason could be that LLMs have been presented with many patterns similar to *ERR* during pre-training and thus perform better when presented with *ERR* prompts (Chen et al., 2024). However, due to lack of access to the pre-training process of LLMs (either open-source or close-source), we cannot further validate our conjecture more solidly and our understanding of the deeper mechanism remains limited to superficial analysis, which is one of the limitations of this work.

4 Related Work

In-context Example Selection Rubin et al. (2022) suggest that LLMs' ICL performance strongly depends on the selection of in-context examples. In consequence, many works have been trying to explore ways of selecting better in-context examples in recent years. Li et al. (2023) train a unified in-context example retriever across a wide range of tasks. Ye et al. (2023) select examples based on both relevance and diversity, with the help of determinantal point processes. Agrawal et al. (2023) ensure n-gram coverage to select better examples for MT. Kumar et al. (2023) train an in-context example scorer for MT based on several features. Tang et al. (2024) combine both word-level and syntax-level coverage when selecting examples for MT.

Mechanism of In-context Learning With the popularity of ICL, there have been numerous studies on analyzing the mechanism of ICL. One stream of these studies focuses on explaining the essence of ICL, relating ICL to gradient descent (Von Oswald et al., 2023), implicit Bayesian inference (Xie et al., 2022), induction heads completing token sequences based on similar context (Olsson et al., 2022), generation maintaining coherency (Sia and Duh, 2023), creation of task vectors based on in-context examples (Hendel et al., 2023), *etc.* The other stream focuses on the role of in-context examples, especially labels of these examples. Min et al. (2022) find that ground truth labels are not necessary and LLMs perform fairly well even with random labels. Wang et al. (2023) find that label words play the role of anchors that aggregating information of the whole examples and serve as a reference for LLMs' final predictions. Wei et al. (2023) find that larger language models can override semantic priors and learn from in-context examples with flipped labels or semantically-unrelated labels.

5 Conclusion

In this work, we analyze the effect of descriptive instructions in prompts during ICL and propose an *Ensemble* prompt framework describing the properties of in-context examples selected by different methods. Experimental results on MT indicate that while LLMs are sensitive to prompt formats, they might not care the actual meaning of the descriptions and the framework improves LLMs' performance even with meaningless descriptions compared with the conventional prompt. We further apply the *Ensemble* framework to four other NLP tasks and find that it achieves promising results, especially on small-scale models. These results suggest that rather than working hard on well-designed descriptions, making use of a proper prompt format would be more effective and efficient.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62076008).

Limitations

First, since there are so many open-source LLMs in the world nowadays, it is impossible to experiment with all existing models and thus our work only employ several commonly-used LLMs. Second, since we do not have access to the pre-training or post-training process of LLMs (either open-source or close-source), our analysis of the mechanism of ICL could be somewhat superficial. The behavior of LLMs can be highly subject to their training data, which we have no access to. Lastly, although we reveal that *ERR* is a superior prompt format for several models, it could still be a local optimum and how to effectively search for a best prompt format for different models and tasks is still under-explored, which we leave for future work.

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*.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Elias Bassani. 2023. retriv: A Python Search Engine for the Common Man.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. 2024. Parallel structures in pre-training data yield in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8582–8592, Bangkok, Thailand. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Susan Craw. 2017. Manhattan distance. *Encyclopedia of Machine Learning and Data Mining*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333, Singapore. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*.

Aswanth Kumar, Ratish Puduppully, Raj Dabre, and Anoop Kunchukuttan. 2023. CTQScorer: Combining multiple features for in-context example selection for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*.

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.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation : Learning to solve and explain algebraic word problems. *Preprint*, arXiv:1705.04146.

Llama Team. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *Preprint*, arXiv:2304.03277.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*.

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.

Suzanna Sia and Kevin Duh. 2023. In-context learning as maintaining coherency: A study of on-the-fly machine translation using large language models. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 173–185, Macau SAR, China. Asia-Pacific Association for Machine Translation.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Pragya Srivastava, Satvik Golechha, Amit Deshpande, and Amit Sharma. 2024. NICE: To optimize in-context examples or not? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5494–5510, Bangkok, Thailand. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Chenming Tang, Zhixiang Wang, and Yunfang Wu. 2024. SCOI: Syntax-augmented coverage-based in-context example selection for machine translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9840–9855, Singapore. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023. Larger language models do in-context learning differently. *Preprint*, arXiv:2303.03846.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. 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.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

A Prompts for MT Ablation Study

Templates and prompt examples of *Ensemble* (Random + Random), *Single* (Random), *Single* (Example), *Vanilla* (Translate) are shown in Figure 9-12.

B Full Experimental Results

Full results of MT are presented in Table 4 and 5. Full results of other tasks in Section 3.2 are presented in Table 6.

C Attention Weights on Alpaca

Figure 13 presents the attention weights on Alpaca. For example-to-noun attention weights, *ERR* is

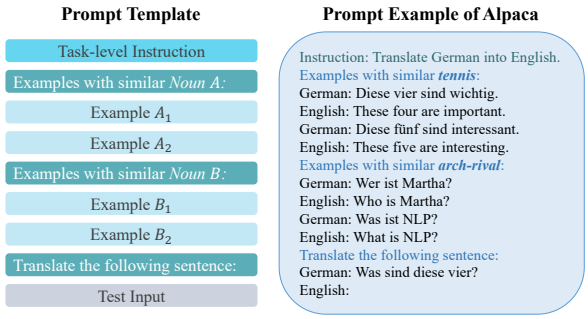


Figure 9: Template and example of *Ensemble* (Random + Random).

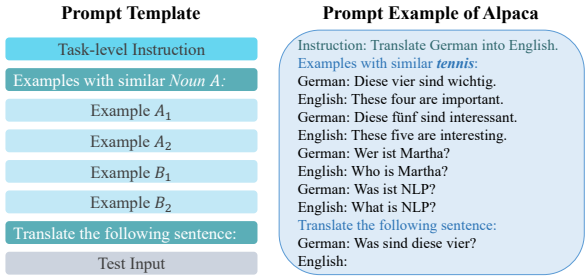


Figure 10: Template and example of *Single* (Random).

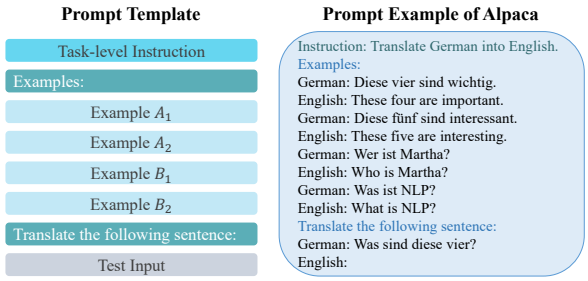


Figure 11: Template and example of *Single* (Example).

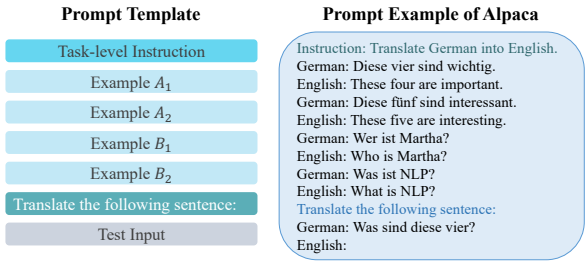


Figure 12: Template and example of *Vanilla* (Translate).

Prompt	Selection-A	Selection-B	Into EN			Out of EN			Avg.
			DE	FR	RU	DE	FR	RU	
Vanilla	Random	Random	63.65	71.40	52.37	40.34	54.58	42.07	54.07
	BM25	BM25	64.32	71.83	51.15	42.53	56.58	43.58	55.00
	Polynomial	Polynomial	64.29	71.25	53.47	43.23	55.15	45.73	55.52
	BM25	Polynomial	65.34	72.01	53.74	43.41	56.48	46.04	56.17
	Polynomial	BM25	64.66	72.04	53.04	44.37	56.56	46.43	56.18
Ensemble (Word + Syntax)	Random	Random	65.07	72.41	54.14	43.09	56.25	42.07	55.50
	BM25	BM25	66.02	72.81	54.13	43.77	56.81	43.19	56.12
	Polynomial	Polynomial	65.98	72.66	54.67	44.37	58.13	45.49	56.88
	BM25	Polynomial	66.08	72.58	54.29	43.51	57.07	45.26	56.47
	Polynomial	BM25	65.89	73.03	54.06	43.78	57.60	46.87	56.87
Ensemble (Syntax + Word)	Random	Random	65.04	72.37	54.15	42.95	56.27	41.85	55.40
	BM25	BM25	66.20	72.41	53.73	43.39	57.24	42.40	55.90
	Polynomial	Polynomial	66.16	72.55	54.66	44.75	56.82	44.38	56.55
	BM25	Polynomial	65.99	72.77	54.21	44.05	57.26	45.09	56.56
	Polynomial	BM25	65.96	72.96	54.12	43.47	57.51	46.94	56.83
Diff. *Ensemble* (Word + Syntax)	Random	Random	65.09	72.53	53.84	42.65	56.15	41.38	55.27
	BM25	BM25	65.95	72.33	54.06	43.42	57.27	41.88	55.82
	Polynomial	Polynomial	65.98	72.58	54.76	44.60	58.36	45.95	57.04
	BM25	Polynomial	66.04	72.49	54.75	44.03	57.70	45.95	56.83
	Polynomial	BM25	66.17	73.17	54.10	44.17	57.48	46.02	56.85
Ensemble (Word + Semantics)	Random	Random	65.09	72.26	54.14	42.64	56.11	42.61	55.48
	BM25	BM25	66.32	72.25	53.89	43.29	57.02	42.51	55.88
	Polynomial	Polynomial	66.36	72.46	54.62	44.17	57.24	45.61	56.74
	BM25	Polynomial	65.80	72.74	54.40	44.06	56.86	45.54	56.57
	Polynomial	BM25	66.06	72.99	53.47	43.24	57.55	46.93	56.71
Ensemble (Random + Random)	Random	Random	65.37	72.54	54.01	42.59	56.22	40.41	55.19
	BM25	BM25	66.41	72.72	54.24	43.08	56.59	41.44	55.75
	Polynomial	Polynomial	66.28	72.56	54.70	43.81	57.42	43.22	56.33
	BM25	Polynomial	65.99	72.77	54.04	44.85	57.25	45.39	56.72
	Polynomial	BM25	65.73	73.26	53.73	43.27	56.78	44.73	56.25

Table 4: Full MT results of XGLM.

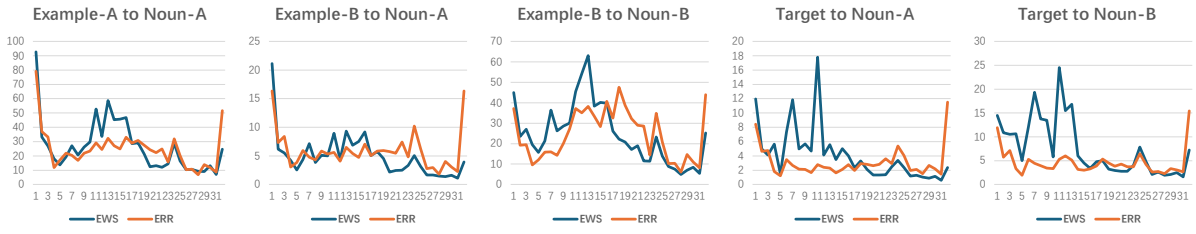


Figure 13: Attention weights (×1e-4) on Alpaca of all 32 layers with BM25 + Polynomial examples. EWS and *ERR* denotes *Ensemble* (Word + Syntax) and *Ensemble* (Random + Random) respectively.

Prompt	Selection-A	Selection-B	Into EN			Out of EN			Avg.
			DE	FR	RU	DE	FR	RU	
Vanilla	Random	Random	69.88	76.46	57.80	42.52	56.61	29.25	55.42
	BM25	BM25	69.08	76.41	58.52	43.65	57.34	32.63	56.27
	Polynomial	Polynomial	69.65	75.79	58.77	43.55	56.60	32.39	56.13
	BM25	Polynomial	69.34	76.02	58.38	43.31	56.79	33.21	56.18
	Polynomial	BM25	69.03	76.11	57.84	42.20	55.81	31.93	55.49
Ensemble (Word + Syntax)	Random	Random	69.86	76.64	57.57	43.51	57.25	30.79	55.94
	BM25	BM25	69.44	76.21	57.60	44.41	58.16	31.26	56.18
	Polynomial	Polynomial	69.86	76.06	58.26	44.46	57.06	33.27	56.50
	BM25	Polynomial	69.58	76.09	57.98	43.84	57.57	32.85	56.32
	Polynomial	BM25	69.41	76.32	57.78	42.79	58.18	31.07	55.93
Ensemble (Syntax + Word)	Random	Random	69.80	76.73	57.53	43.50	57.31	30.70	55.93
	BM25	BM25	69.46	76.15	57.63	44.58	58.46	32.09	56.40
	Polynomial	Polynomial	69.76	76.11	58.14	44.11	56.64	33.24	56.33
	BM25	Polynomial	69.60	76.11	57.91	43.88	57.81	32.22	56.26
	Polynomial	BM25	69.64	76.11	57.77	42.42	58.43	31.41	55.96
Diff. *Ensemble* (Word + Syntax)	Random	Random	69.77	76.63	57.46	43.67	57.48	30.55	55.93
	BM25	BM25	69.44	76.33	57.70	44.24	58.48	31.76	56.33
	Polynomial	Polynomial	69.75	76.07	58.12	44.09	57.31	32.46	56.30
	BM25	Polynomial	69.54	76.21	57.68	43.61	57.49	32.27	56.13
	Polynomial	BM25	69.49	76.23	57.65	42.79	58.20	32.09	56.08
Ensemble (Word + Semantics)	Random	Random	69.88	76.70	57.51	43.30	57.35	30.98	55.96
	BM25	BM25	69.44	76.20	57.65	44.47	57.88	32.48	56.35
	Polynomial	Polynomial	69.82	76.10	58.35	44.00	57.25	33.45	56.50
	BM25	Polynomial	69.57	76.23	58.12	44.07	57.73	32.79	56.42
	Polynomial	BM25	69.49	76.17	57.99	43.20	58.49	31.66	56.17
Ensemble (Random + Random)	Random	Random	69.77	76.61	57.38	43.52	57.55	30.90	55.95
	BM25	BM25	69.46	76.24	57.58	44.40	58.25	33.23	56.53
	Polynomial	Polynomial	69.63	75.93	57.77	44.36	56.73	33.45	56.31
	BM25	Polynomial	69.55	76.01	57.86	42.75	57.80	32.63	56.10
	Polynomial	BM25	69.74	76.04	57.73	43.33	58.30	33.27	56.40

Table 5: Full MT results of Alpaca.

Model	Template	Performance									
		Date	SgyQA	CSQA	Sports	LF	TO	GSM8K	AQuA	KU	Avg.
LLaMA-2-7B	*Vanilla* w/ CoT	3.01	50.99	21.70	50.20	48.72	30.74	**22.29**	21.65	45.24	32.73
	ERR w/ CoT	**45.21**	59.78	**55.36**	**79.42**	**56.03**	**41.55**	21.15	**23.62**	**85.71**	**51.98**
	Vanilla w/o CoT	0.27	47.33	19.25	54.42	48.91	30.07	0.91	23.23	52.38	30.75
	ERR w/o CoT	27.67	**59.88**	23.91	16.77	1.58	36.49	5.91	22.44	69.05	29.30
LLaMA-3.1-8B	*Vanilla* w/ CoT	9.04	51.28	24.41	55.32	**52.08**	35.47	**49.05**	24.80	**69.05**	41.17
	ERR w/ CoT	**38.63**	55.53	31.04	**74.60**	51.58	**38.18**	48.52	**26.77**	**69.05**	**48.21**
	Vanilla w/o CoT	7.95	41.90	28.42	54.22	51.98	37.50	1.14	16.93	28.57	29.84
	ERR w/o CoT	13.70	**59.68**	**36.12**	50.00	50.79	35.81	4.70	22.83	57.14	36.75
Alpaca-7B	*Vanilla* w/ CoT	26.03	60.38	**50.53**	**80.12**	52.27	**35.81**	6.44	18.11	**71.43**	**44.57**
	ERR w/ CoT	**26.85**	60.47	50.04	74.40	54.74	35.14	**6.75**	20.08	**71.43**	44.43
	Vanilla w/o CoT	25.21	60.77	46.27	60.24	**55.63**	35.47	4.70	**24.80**	64.29	41.93
	ERR w/o CoT	24.93	**61.17**	50.12	61.04	55.14	34.80	4.78	24.41	64.29	42.30
Mistral-7B	*Vanilla* w/ CoT	6.30	57.71	50.37	61.24	48.22	31.08	43.21	**29.13**	73.81	44.56
	ERR w/ CoT	**36.99**	**66.60**	**61.83**	**84.94**	**64.03**	33.78	**43.97**	25.98	**83.33**	**55.72**
	Vanilla w/o CoT	0.27	49.70	20.97	78.21	46.05	31.08	1.06	18.90	61.90	34.24
	ERR w/o CoT	13.15	64.13	55.86	77.11	61.86	**49.32**	7.96	22.44	66.67	46.50

Table 6: Full results of Date Understanding (Date), StrategyQA (SgyQA), CSQA, Sports Understanding (Sports), Logical Fallacy (LF), Three Objects (TO), GSM8K, AQuA and Known Unknowns (KU).

close to EWS. For target-to-noun attention weights, EWS is higher in shallow layers but falls behind *ERR* in deeper layers, especially in the last layer. This demonstrates that Alpaca might pay more attention to the meaningful words ("word" and "syntax") when understanding the context in shallow layers but gradually forgets them when it comes to generation in the deeper layers. In short, EWS performs no higher than *ERR* in most cases.

D Dataset Details for Reasoning Tasks

We list the details of splitting training set (example database) and test set for our conducted reasoning tasks, covering four types and nine datasets. We set random seed for all possible shuffling and sampling operations to 42. Note that we experiment with **4-shot** for all datasets.

D.1 Datasets Fetched from Exclusive Source

- CSQA (Talmor et al., 2019): `https://www.tau-nlp.org/commonsenseqa`. We follow the official split and select the training set as our example database and the dev set as our test set. Because the training set itself is randomly divided from the whole dataset, we directly select examples from it in order.
- GSM8K (Cobbe et al., 2021): `https://github.com/openai/grade-school-math`. We select the `test.jsonl` as our test set and the `train.jsonl` as our example database and randomly sample four examples from it.
- AQuA (Ling et al., 2017): `https://github.com/google-deepmind/AQuA`. We select the `test.json` as our test set. Since the original training set is relatively large, for simplicity, we directly copy the four examples listed in the supplementary materials of Wei et al. (2022) and we ensure that these four examples do not appear in the test set.

D.2 Datasets Fetched from The Big-bench

For StrategyQA (Geva et al., 2021), Date, Sports, Logical Fallacy, Three Objects, and Known Unknowns, we fetched them from the Big-bench (Srivastava et al., 2023). Each of them has a `task.json`. We randomly shuffle the `task.json` and split it to a training set and a test set. Then we select examples from the training set in order.

Specifically, the principle for splitting the training and test sets is as follows: If the total number of samples exceeds 1,012 a lot, we retain 1,012 samples as the test set and use the remainder as the training set. Otherwise, we select four examples for the training set and use the rest for testing. For the Sports and Logical Fallacy datasets, which have only two possible answers (similar to binary classification), we first separate the positive and negative examples, shuffle them individually, and then construct the test set and training set. The test set is composed of an equal number of positive and negative examples, with the remaining samples used as the training set.

E Prompts for Reasoning Tasks Used in this Work

Figure 14-22 show the examples of *ERR* (w/ CoT) prompt for respective datasets. Some tasks contain **Answer Choices**. In order to save space, the blank lines between the options are replaced with spaces in those figures. Each Figure has grey text for reasoning, cyan text for the parts of *ERR* that are unique to *Vanilla*, and italic words in the cyan text representing random nouns. Therefore, deleting the grey text gives *ERR* (w/o CoT), keeping the grey text but deleting the cyan text gives *Vanilla* (w/ CoT), and deleting both the cyan and grey text gives *Vanilla* (w/o CoT).[5] The reasoning is generated by ChatGPT [6]. Note the ChatGPT is not the same as GPT-3.5 we used for experiments.

F Results of Llama2

Results of Llama2-7B-chat-hf (Llama Team, 2023) on the nine datasets are presented in Figure 23. While *ERR* outperforms *Vanilla* with Llama2 across most datasets, its performance on Logical Fallacy and Sports is notably poor. Llama2 almost always responds with confused emojis for Logical Fallacy questions and outputs questions like "plausible or implausible?" for Sports, leading to predominantly incorrect answers. Further investigation into these issues is left for future work.

G Computational Details

G.1 Hardware

Inference of LLMs runs on an NVIDIA A40 GPU (with memory of 48 GB). Other experiments run

[5] When changing "w/ CoT" to "w/o CoT", you may also need to replace "So the answer is ..." with "The answer is ..." for syntactical reasons.

[6] https://chatgpt.com/

Instruction: Read the question and choose the correct answer.
Examples with similar *opportunist*:
Q: The sanctions against the school were a punishing blow, and they seemed to what the efforts the school had made to change?
Answer Choices: (a) ignore (b) enforce (c) authoritarian (d) yell at (e) avoid
A: The answer must be something that reflects a negative impact on the efforts made to change. Of the above choices, only ignore fits this context. So the answer is (a).

Q: Sammy wanted to go to where the people were. Where might he go?
Answer Choices: (a) race track (b) populated areas (c) the desert (d) apartment (e) roadblock
A: The answer must be a place with many people. Of the above choices, populated areas is the best fit. So the answer is (b).

Examples with similar *spokeswoman*:
Q: To locate a choker not located in a jewelry box or boutique where would you go?
Answer Choices: (a) jewelry store (b) neck (c) jewelry box (d) jewelry box (e) boutique
A: The answer must be a place where a choker is typically found. Of the above choices, neck is the best fit. So the answer is (b).

Q: Google Maps and other highway and street GPS services have replaced what?
Answer Choices: (a) united states (b) mexico (c) countryside (d) atlas (e) oceans
A: The answer must be something that was used for navigation before GPS services. Of the above choices, atlas is the best fit. So the answer is (d).

Consider the following question.
Q: What do people aim to do at work?
Answer Choices: (a) complete job (b) learn from each other (c) kill animals (d) wear hats (e) talk to each other
A:

Figure 14: Prompt for CSQA.

Instruction: Read the question and answer yes or no.
Examples with similar *opportunist*:
Q: Does ancient Olympics crown fail to hide tonsure?
A: Tonsure refers to the religious hairstyle in which hair from the top of the head is shaved off. Winners of the ancient Olympics were given a laurel crown to wear. A laurel crown involves leaves arranged in a circle that cover only the edges of the head. So the answer is yes.

Q: Is Olivia Newton-John hyphenated celebrity name with most letters?
A: Olivia Newton-John has sixteen letters in her name. Actress Catherine Zeta-Jones has 18 letters in her name. Actor Joseph Gordon-Levitt has 18 letters in his name. So the answer is no.

Examples with similar *spokeswoman*:
Q: Does a person need to be a parent to become a grandparent?
A: Parents care for their children. When the children grow up and have kids of their own, the parents become grandparents to those kids. A person who is not a parent has no kids, therefore nobody to produce grandchildren for them. So the answer is yes.

Q: Can a human eat an entire 12-lb roast turkey in an hour?
A: A serving of roast turkey is about 1 pound of uncooked turkey. A 12-lb roast turkey would contain about 12 servings of cooked turkey meat. One human cannot eat 12 1-lb servings of turkey in one sitting. So the answer is no.

Consider the following question.
Q: Is nickel dominant material in US 2020 nickels?
A:

Figure 15: Prompt for StrategyQA.

Instruction: Answer question about date inference.
Examples with similar *opportunist*:
Q: Jane thought today is 3/11/2002, but today is in fact Mar 12, which is 1 day later. What is the date one year ago from today in MM/DD/YYYY?
A: Today is Mar 12, which is 03/12/2002. One year ago from today would be the same date but in the previous year, which is 03/12/2001. So the answer is 03/12/2001.

Q: The deadline is Jun 1, 2021, which is 2 days away from now. What is the date 10 days ago in MM/DD/YYYY?
A: Jun 1, 2021 is 2 days away from now, so today is May 30, 2021. 10 days ago from May 30, 2021 is May 20, 2021. So the answer is 05/20/2021.

Examples with similar *spokeswoman*:
Q: Jane was born on the last day of Feburary in 2000. Today is her 16-year-old birthday. What is the date 10 days ago in MM/DD/YYYY?
A: Jane was born on the last day of February in 2000, which is 02/29/2000. Today is her 16-year-old birthday, so today is 02/29/2016. 10 days ago from 02/29/2016 is 02/19/2016. So the answer is 02/19/2016.

Q: It was Sept. 1st, 2021 a week ago. What is the date 10 days ago in MM/DD/YYYY?
A: Sept. 1st, 2021 was a week ago, so today is Sept. 8, 2021. 10 days ago from Sept. 8, 2021 is Aug. 29, 2021. So the answer is 08/29/2021.

Consider the following question.
Q: Jane scheduled 3 apointments with 5 poeple for tomorrow (Tue, 7/9/1972). What is the date one year ago from today in MM/DD/YYYY?
A:

Figure 16: Prompt for Date.

Instruction: Determine whether an artificially constructed sentence relating to sports is plausible or implausible. Answer yes or no.
Examples with similar *opportunist*:
Statement: Jarrett Allen got on base
A: Jarrett Allen is a professional basketball player, not a baseball player. 'Got on base' is a baseball term. So the answer is no.

Statement: Jack Flaherty pitched a shutout in the National League Championship Series
A: Jack Flaherty is a professional baseball player and a pitcher for the St. Louis Cardinals, which is part of the National League. Pitching a shutout in the National League Championship Series is plausible. So the answer is yes.

Examples with similar *spokeswoman*:
Statement: Klaas Jan Huntelaar caught the back shoulder fade
A: Klaas Jan Huntelaar is a professional soccer player. 'Caught the back shoulder fade' is a term used in American football, not soccer. So the answer is no.

Statement: Jacob Markstrom scored in the shootout in the Stanley Cup
A: Jacob Markstrom is a professional ice hockey player and a goaltender in the NHL. Scoring in the shootout in the Stanley Cup is plausible for an ice hockey player. So the answer is yes.

Consider the following statement.
Statement: Deandre Ayton was perfect from the line in the Eastern Conference Finals
A:

Figure 17: Prompt for Sports.

on Intel® Xeon® Gold 6348 CPU (with memory of 256 GB).

G.2 Software

Our OS: Ubuntu 20.04.6 LTS. Our code: Python only. Libraries and packages are specified in the source code.

H Licenses

Artifact	License
XGLM	MIT
Alpaca	Apache-2.0
Llama	Llama Community License Agreement
Mistral	Apache-2.0
COMET	Apache-2.0
FLORES-101	CC-BY-SA-4.0
Europarl	Unknown
ParaCrawl	CC0
CSQA	CC-BY-SA-4.0
StrategyQA	MIT
BIG-bench	Apache-2.0
GSM8K	MIT
AQuA	Apache-2.0

Table 7: Licenses of scientific artifacts we use.

The licenses of the scientific artifacts we use are shown in Table 7.

Instruction: You are an AI. Read the input and give your answer.
Examples with similar *opportunist*:
Input: This AI is identifying whether statements contain fallacies. The AI responds with 'Valid' or 'Invalid' as appropriate. IQ tests were developed to help developmentally delayed children, so they can't be useful for normal adults.
A: The statement contains a fallacy known as the 'genetic fallacy,' which argues that something cannot be useful based on its origin. IQ tests, despite their origin, can still be useful for normal adults. So the answer is Invalid.

Input: This AI is identifying whether statements contain fallacies. The AI responds with 'Valid' or 'Invalid' as appropriate. Robert is faster than Paul. Paul is faster than Anne. Therefore Paul is faster than Robert.
A: The statement contains a logical error. If Robert is faster than Paul, and Paul is faster than Anne, it does not follow that Paul is faster than Robert. So the answer is Invalid.

Examples with similar *spokeswoman*:
Input: Do you think the following argument is 'Valid' or 'Invalid'? Dylan is less reliable than Tim. Charlotte is less reliable than Dylan. Therefore Tim is more reliable than Dylan.
A: The argument is logically consistent. If Dylan is less reliable than Tim, and Charlotte is less reliable than Dylan, it correctly follows that Tim is more reliable than Dylan. So the answer is Valid.

Input: Do you think the following argument is 'Valid' or 'Invalid'? The charity is run by lots of famous celebrities, so it must do good work.
A: The statement contains a fallacy known as 'appeal to authority' or 'appeal to celebrity,' which argues that something must be good because it is associated with famous people. The quality of the charity's work is independent of its celebrity endorsements. So the answer is Invalid.

Consider the following input.
Input: Do you think the following argument is 'Valid' or 'Invalid'? Laura is less witty than Sara. Sara is less witty than Kim. Sara is less witty than David. Therefore laura is less witty than Kim.
A:

Figure 18: Prompt for Logical Fallacy.

Instruction: Read the statement and choose the correct answer.
Examples with similar *opportunist*:
Statement: In a golf tournament, there were three golfers: Amy, Dan, and Mel. Mel finished above Amy. Dan finished below Amy.
Answer Choices: (a) Amy finished last. (b) Dan finished last. (c) Mel finished last.
A: Mel finished above Amy, and Dan finished below Amy. Therefore, the order from highest to lowest is Mel, Amy, Dan. So, Dan finished last. So the answer is (b).

Statement: In a golf tournament, there were three golfers: Amy, Eli, and Eve. Eve finished above Amy. Eli finished below Amy.
Answer Choices: (a) Amy finished last. (b) Eli finished last. (c) Eve finished last.
A: Eve finished above Amy, and Eli finished below Amy. Therefore, the order from highest to lowest is Eve, Amy, Eli. So, Eli finished last. So the answer is (b).

Examples with similar *spokeswoman*:
Statement: On a shelf, there are three books: a white book, a green book, and an orange book. The green book is to the right of the white book. The orange book is the rightmost.
Answer Choices: (a) The white book is the leftmost. (b) The green book is the leftmost. (c) The orange book is the leftmost.
A: The orange book is the rightmost. The green book is to the right of the white book, making the white book the leftmost. So, the white book is the leftmost. So the answer is (a).

Statement: On a shelf, there are three books: a red book, a gray book, and a white book. The white book is to the left of the gray book. The red book is the second from the left.
Answer Choices: (a) The red book is the leftmost. (b) The gray book is the leftmost. (c) The white book is the leftmost.
A: The red book is second from the left, and the white book is to the left of the gray book. Therefore, the order from left to right is white, red, gray. So, the white book is the leftmost. So the answer is (c).

Consider the following statement.
Statement: In an antique car show, there are three vehicles: a motorcyle, a limousine, and a convertible. The motorcyle is newer than the limousine. The convertible is newer than the motorcyle.
Answer Choices: (a) The motorcyle is the oldest. (b) The limousine is the oldest. (c) The convertible is the oldest.
A:

Figure 19: Prompt for Three Objects.

Instruction: Read the question and choose the proper answer.
Examples with similar *opportunist*:
Statement: How many people watched Seinfeld when it was on the air?
Answer Choices: (a) 76.3 million (b) Unknown
A: The number of people who watched 'Seinfeld' during its original run is well-documented. The series finale alone was watched by 76.3 million people in the U.S. So the answer should be (a).

Statement: When was Abraham Lincoln born?
Answer Choices: (a) February 12, 1809 (b) Unknown
A: The birthdate of Abraham Lincoln is a well-known historical fact. He was born on February 12, 1809. So the answer should be (a).

Examples with similar *spokeswoman*:
Statement: In the year 2020, how many people in California were homeless?
Answer Choices: (a) 161,548 people (b) Unknown
A: The number of homeless people in California in 2020 is recorded in official statistics. According to the U.S. Department of Housing and Urban Development, there were 161,548 homeless people in California in 2020. So the answer should be (a).

Statement: How many fish were eaten by birds in the year 2100 BCE?
Answer Choices: (a) 450,000 (b) Unknown
A: There are no historical records or scientific data that can accurately determine the number of fish eaten by birds in the year 2100 BCE. So the answer should be (b).

Consider the following question.
Statement: Who is a famous whistler?
Answer Choices: (a) Ronnie Ronalde (b) Unknown
A:

Figure 20: Prompt for Known Unknowns.

Instruction: Read the question and give your answer.
Examples with similar *opportunist*:
Q: In a field of 500 clovers, 20% have four leaves and one quarter of these are purple clovers. Assuming these proportions are exactly correct, how many clovers in the field are both purple and four-leaved?
Reasoning: 1): There are 500/5= <<500/5=100>>100 four leaf clovers. 2): There are 100/4= <<100/4=25>>25 purple four leaf clovers.
A: The answer is 25.

Q: Eustace is twice as old as Milford. In 3 years, he will be 39. How old will Milford be?
Reasoning: 1): Eustace's current age must be 39 years old - 3 years = <<39-3=36>>36 years old. 2): So Milford's current age must be 36 years old / 2 = <<36/2=18>>18 years old. 3): So in 3 years, Milford will be 18 years old + 3 years = <<18+3=21>>21 years old.
A: The answer is 21.

Examples with similar *spokeswoman*:
Q: Each yogurt is topped with 8 banana slices. One banana will yield 10 slices. If Vivian needs to make 5 yogurts, how many bananas does she need to buy?
Reasoning: 1): To make the yogurts, Vivian needs 5 x 8 = <<5*8=40>>40 banana slices. 2): She needs to buy 40 / 10 = <<40/10=4>>4 bananas.
A: The answer is 4.

Q: Remi prepared a tomato nursery and planted tomato seedlings. After 20 days, the seedlings were ready to be transferred. On the first day, he planted 200 seedlings on the farm. On the second day, while working alongside his father, he planted twice the number of seedlings he planted on the first day. If the total number of seedlings transferred to the farm on these two days was 1200, how many seedlings did his father plant?
Reasoning: 1): On the second day, he planted 2 * 200 seedlings = <<2*200=400>>400 seedlings. 2): The total number of seedlings Remi planted on the two days is 400 seedlings + 200 seedlings = <<400+200=600>>600 seedlings. 3): If the total number of seedlings transferred from the nursery was 1200 after the second day, Remi's father planted 1200 seedlings - 600 seedlings = <<1200-600=600>>600 seedlings.
A: The answer is 600.

Consider the following question.
Q: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?

Figure 21: Prompt for GSM8K. For this dataset, we let LLMs first generate reasoning and then answer under the "w/ CoT" setting.

Instruction: Read the question and choose the correct answer.
Examples with similar *opportunist*:
Q: John found that the average of 15 numbers is 40. If 10 is added to each number then the mean of the numbers is?
Answer Choices: (a) 50 (b) 45 (c) 65 (d) 78 (e) 64
A: If 10 is added to each number, then the mean of the numbers also increases by 10. So the new mean would be 50. The answer is (a).

Q: If a / b = 3/4 and 8a + 5b = 22, then find the value of a.
Answer Choices: (a) 1/2 (b) 3/2 (c) 5/2 (d) 4/2 (e) 7/2
A: If a / b = 3/4, then b = 4a / 3. So 8a + 5(4a / 3) = 22. This simplifies to 8a + 20a / 3 = 22, which means 44a / 3 = 22. So a is equal to 3/2. The answer is (b).

Examples with similar *spokeswoman*:
Q: A person is traveling at 20 km/hr and reached his destiny in 2.5 hr then find the distance?
Answer Choices: (a) 53 km (b) 55 km (c) 52 km (d) 60 km (e) 50 km
A: The distance that the person traveled would have been 20 km/hr * 2.5 hrs = 50 km. The answer is (e).

Q: How many keystrokes are needed to type the numbers from 1 to 500?
Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788
A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. 9 + 90(2) + 401(3) = 1392. The answer is (b).

Consider the following question.
Q: The original price of an item is discounted 22%. A customer buys the item at this discounted price using a $20-off coupon. There is no tax on the item, and this was the only item the customer bought. If the customer paid $1.90 more than half the original price of the item, what was the original price of the item?
Answer Choices: (A) $61 (B) $65 (C) $67.40 (D) $70 (E) $78.20
A:

Figure 22: Prompt for AQuA.

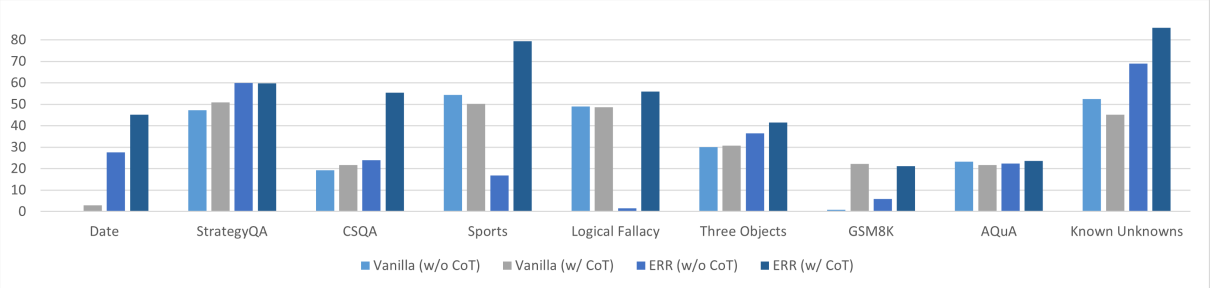


Figure 23: Results of Llama2 on the nine datasets.