OptiSeq: Ordering Examples On-The-Fly for In-Context Learning

Rahul Atul Bhope[†*], **Praveen Venkateswaran**[*], **K. R. Jayaram**[*], **Vatche Isahagian**[*], **Vinod Muthusamy**[*], **Nalini Venkatasubramanian**[†]
[†] University of California, Irvine, USA [*] IBM Research AI, USA
rbhope@uci.edu praveen.venkateswaran@ibm.com jayaramkr@us.ibm.com
vatchei@ibm.com vmuthus@us.ibm.com nalini@ics.uci.edu

Abstract

Developers using LLMs and LLM-based agents in their applications have provided plenty of anecdotal evidence that in-context-learning (ICL) is fragile. In this paper, we show that in addition to the quantity and quality of examples, the order in which the in-context examples are listed in the prompt affects the output of the LLM and, consequently, their performance. While prior work has explored improving ICL through dataset-dependent techniques, we introduce OptiSeq, a purely inference-time, dataset-free optimization method that efficiently determines the best example order. OptiSeq leverages log probabilities of LLM-generated outputs to systematically prune the search space of possible orderings and recommend the best order(s) by distinguishing orderings that yield high levels of accuracy and those that underperform. Extensive empirical evaluation on multiple LLMs, datasets, and prompts demonstrates that OptiSeq improves accuracy by 5.5 - 10.5 *percentage points* across multiple tasks.

1 Introduction

The use of in-context learning (ICL) with large language models (LLMs) has become a popular approach to achieve impressive performance in many NLP tasks (Raffel et al., 2020; Radford et al., 2019). In ICL, models are prompted during inference with task-specific examples that help condition the generated output. Unlike fine-tuning, it does not require updates to the model parameters, which offers many benefits with ever-increasing model sizes and capabilities (Brown et al., 2020). It has been shown that prompting LLMs without fine-tuning is often sensitive to prompt design (Shin et al., 2020; Chen et al., 2021). In particular, the quality, quantity, and permutation of examples can all significantly impact performance (Zhao et al., 2021). To address this, previous work has primarily focused on selecting high-quality examples from a candidate set (Yang et al., 2023; Liu et al., 2021) and determining the optimal number of these examples (Zhang et al., 2023; Agarwal et al., 2024).

Existing solutions to mitigate prompt sensitivity caused by example ordering at *inference-time* are limited. Figure 1 illustrates this using an API sequence generation task on the ToolBench dataset (Qin et al., 2023). Given three in-context examples, LLM predictions vary significantly across the six possible orderings, with only one specific order (*order 2*) yielding the correct answer. This variability in precision and recall underscores how reordering examples alters the input context, influencing token probabilities and ultimately affecting model performance. Most prior approaches rely on a precomputed strategy that assumes access to a predefined set of examples and a fixed label space (Lu et al., 2021; Guo et al., 2024). However, our setting is inherently online, requiring dynamic ordering decisions at inference time without training or validation data. Since orderings cannot be precomputed, the limited number of examples further complicates the problem.

A common approach to selecting examples at inference-time is to generate embeddings of candidate examples using a model like Sentence-BERT (Reimers, 2019) and retrieve the top-k most similar examples for a given test instance, ranking them based on distance or similarity. However, **there is a distinction between ranking examples (determining how relevant they are to our test case) and ordering them (deciding how to arrange them in the prompt)**. While finding relevant examples through ranking is valuable, it does not tell us the best way to order them in the prompt. Furthermore, top-k is dependent on the quality of embeddings and can lead to suboptimal performance if the distances are too close. Recent efforts leverage additional in-domain validation datasets

[*]Part of this work was done during Rahul Atul Bhope's internship at IBM Research.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 24864–24887
November 4-9, 2025

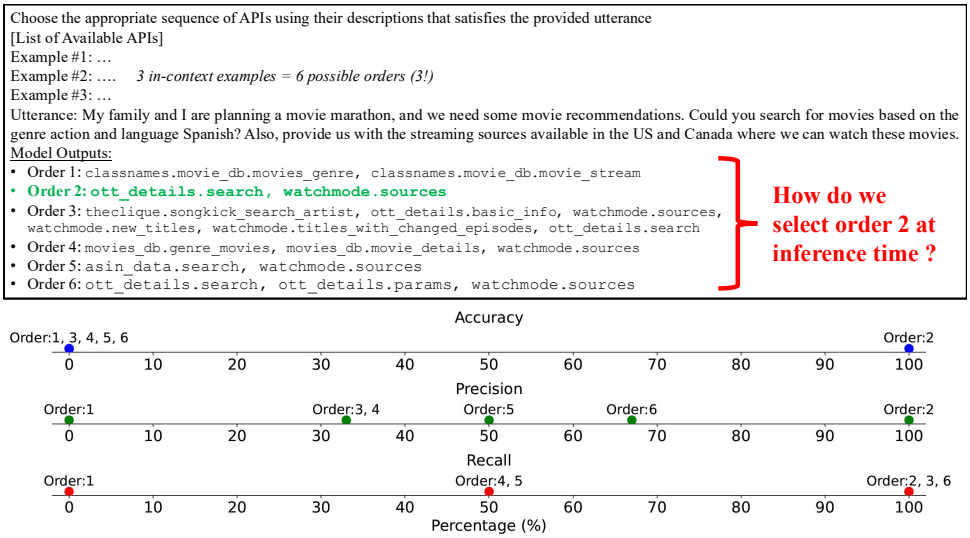


Figure 1: `llama-3-8b-instruct` performance variation with 6 in-context example orderings using 3 examples.

to perform offline evaluation of different orders, which is often not feasible in real-world scenarios where limited data is available (Perez et al., 2021). They also develop mutual information- or entropy-based heuristics (Sorensen et al., 2022; Lu et al., 2021; Guo et al., 2024) on these validation datasets, but are computationally limited to tasks like single-class classification assuming label-balanced tasks and do not generalize to generation tasks. Moreover, as we show in Section 2, the optimal order of the examples varies for the test samples within and between tasks and between different LLMs, finding and selecting this optimal order is very challenging in production settings.

An effective real-world solution needs to be (a) non-reliant on the availability of additional validation/training data, (b) generalizable to different tasks, and (c) performant across different LLMs and number of examples. In this paper, we introduce `OptiSeq`, a novel approach for selecting the optimal order of in-context examples at run-time and make the following contributions:

- We present a study of *example-order* sensitivity in ICL in an inference-time setting (Section 2).
- We describe the design and implementation of OptiSeq, which evaluates few-shot ICL performance across order permutations and then selects the best order by leveraging the model's ability to distinguish between outputs (Sections 3, 5).
- We propose EOptiSeq, a variant of OptiSeq that evaluates fewer permutations at inference time, achieving lower accuracy gains but improving efficiency (Sections 3, 5).
- We present a detailed empirical evaluation of OptiSeq and EOptiSeq (Section 5), across two tasks: API sequence generation and classification, five datasets, and five LLMs (8B–70B parameters). OptiSeq improves accuracy by 10.5 percentage points over random selection, 6.5 percentage points over Top-K selection and 5.5 percentage points over recent baselines (Lu et al., 2021; Guo et al., 2024).

2 Sensitivity of ICL to Example Ordering

In this section, we present an analysis of the impact of in-context example ordering on performance under inference-time settings.

Naive ICL fails to distinguish between correct and incorrect outputs. Figure 2 illustrates an example from the ToolBench dataset, where Naive ICL is applied using the standard prompt structure. We evaluate all six in-context example permutations using `llama-3-8b-instruct` and compare their generated API sequences against the ground truth. Orders 1, 5, and 6 produce correct sequences, while Orders 2, 3, and 4 fail. To analyze these failures, we compute the logarithmic probabilities of each output sequence as given by the LLM. In Figure 2 we can see Naive ICL assigns similar log probabilities to both correct and incorrect sequences across all orders. Ideally, an LLM should assign higher probabilities to correct sequences, effectively distinguishing them from incorrect ones.

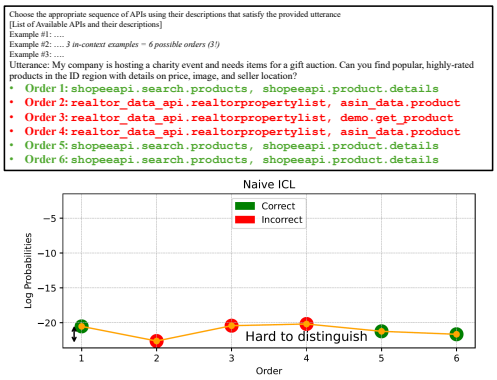


Figure 2: Naive ICL exhibits higher overlap in log probabilities for correct and incorrect outpus, making distinction harder.

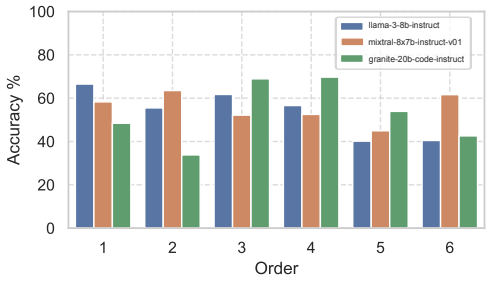


Figure 3: AG News classification accuracy across different orders and models

Optimal order varies across instances and models. We evaluate six example orderings using three AG News (Zhang et al., 2015) articles as in-context examples across three models. Figure 3 shows that optimal ordering is not universal—it varies across test instances and models. A fixed order fails to generalize, necessitating an adaptive approach without relying on training data or precomputed orderings. Moreover, an order that performs well for one model (e.g., `llama-3-8b-instruct`) may underperform for another (e.g., `granite-20b-code-instruct`), highlighting the need for a model-aware, instance-specific ordering strategy without assuming dataset-wide fairness or requiring labeled data.

Impact of choosing the wrong order can be significant depending on the dataset or task. We evaluate five datasets (Section 4.1) across two tasks—API sequence generation and text classification—by measuring accuracy variations across all possible example orderings in a three-shot setting. Figure 4 demonstrates that example ordering plays a crucial role in performance, with accuracy fluctuating by $\approx$ 12 percentage points for API generation and $\approx$ 17 percentage points for classification. This suggests that the model is influenced by example order than by the examples themselves.

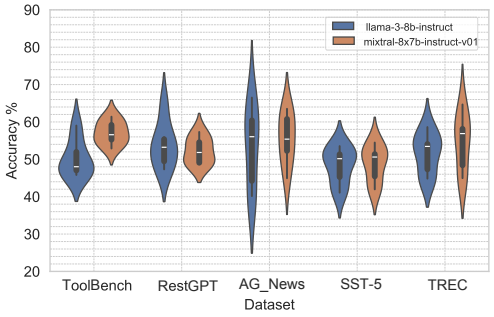


Figure 4: Variation in the average accuracy across all permutations of three examples for five datasets

3 Methodology

The effectiveness of in-context learning in large language models (LLMs) depends on how well the model utilizes contextual cues, which in turn is influenced by the ordering of examples. Section 2 demonstrates that the optimal order is shaped by both the characteristics of the examples and the model itself, making it difficult to predict solely from the inputs. This leads to a key question: *How does the ordering of in-context examples affect an LLM's ability to distinguish between possible outputs?* The ordering of examples provides a signal that influences the LLM's predictions, often leading the model to generate different outputs with comparable log probabilities across different orders as seen in Figure 2. This suggests that LLMs treat the entire prompt—content and order—as a holistic sequence, generating the output they are most confident in given that specific sequence. If we can evaluate the generated output in a way that removes the influence of ordering context, we would be able to better distinguish which outputs are inherently more likely to be correct, independent of example order.

3.1 OptiSeq

Building on this insight, we introduce `OptiSeq`, an *example-free* approach that optimizes in-context example ordering by leveraging the log probabilities of LLM-generated outputs (Figure 5). The process consists of the following steps:

- **Generate outputs for all permutations:** Given a task instruction $\mathcal{I}$ with in-context ex-

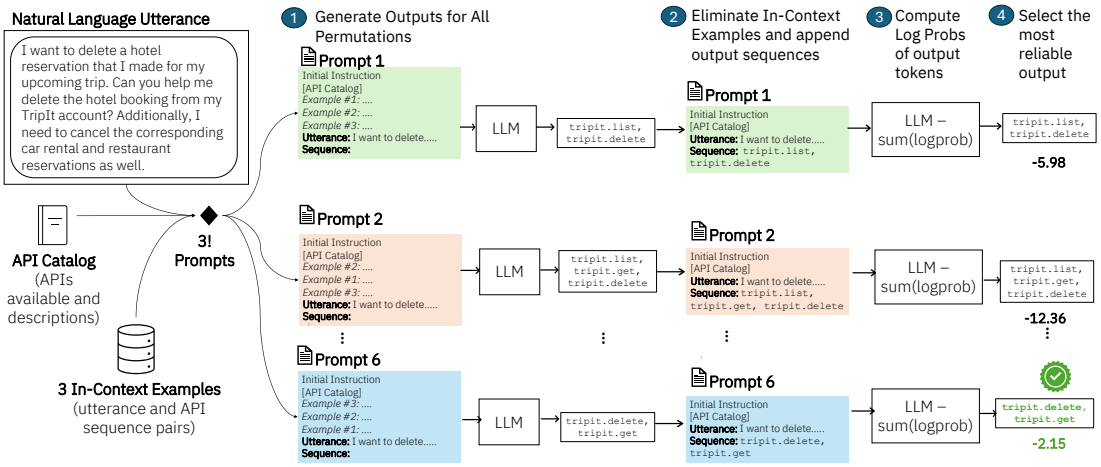


Figure 5: OptiSeq Overview

amples $\mathcal{E}$, we construct $k = |\mathcal{E}|!$ prompts, each corresponding to a unique ordering of examples. These prompts are fed into the LLM to generate candidate outputs $o_k \in \mathcal{O}$ given by:

$$\log P(o_k \mid \mathcal{I}, \mathcal{E}_k) = \sum_{i=1}^{n} \log P(x_{ik} \mid \mathcal{I} \oplus \mathcal{E}_k \oplus x_{jk<ik}) \tag{1}$$

where x_{ik} is the i^{th} token of output o_k, $\mathcal{E}_k$ is the k^{th} example permutation and $x_{jk<ik}$ represents preceding tokens providing autoregressive context.

- **Eliminate In-context examples:** For each candidate output, we modify the prompt by removing the in-context examples while retaining only the task instructions $\mathcal{I}$.
- **Append candidate outputs:** Each generated output o_k is appended to its corresponding modified prompt.
- **Compute log probabilities:** Using the same LLM, we compute the sum of the log probabilities of the output tokens o_k, conditioned only on the task instructions:

$$\Phi_k = \sum_{i=1}^{n} \log P(x_{ik} \mid \mathcal{I} \oplus x_{jk<ik}) \forall o_k \in \mathcal{O}$$

- **Select the optimal ordering:** The order k^* that maximizes the sum of log probabilities is chosen as the optimal in-context example ordering.

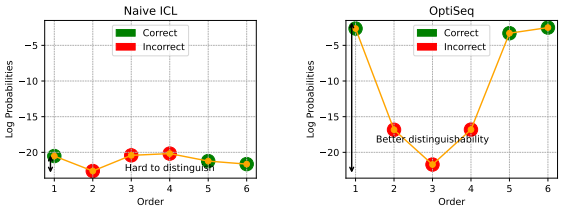
$$k^* = \underset{k}{\operatorname{argmax}}\, \Phi_k$$

Algorithm 1 `OptiSeq`

Inputs: Task instruction $\mathcal{I}$, In-context examples $\mathcal{E}$, Large Language Model M
Outputs: Optimal example ordering $\mathcal{E}_{k^*}$

```
Construct k = |E|! permutations E_k
for each permutation E_k do
    o_k ← M(I, E_k) // Generate outputs for all orderings
end for
for each generated output o_k do
    Φ_k ← Σ_{i=1}^{n} log P(x_ik | I ⊕ x_jk<ik) // Compute log probs without examples
end for
k* = argmax_k Φ_k    // Select optimal ordering
return E_k*
```

This results in better distinguishability among outputs at inference-time as seen in Figure 6 for the same utterance from Figure 2. The full algorithm is demonstrated in Algorithm 1.



(a) (b)

Figure 6: Comparison of log probabilities with and without OptiSeq optimization

3.2 EOptiSeq

While OptiSeq enhances distinguishability by computing log probabilities without in-context examples, it still requires evaluating multiple example orderings to find the most effective one. However, exhaustively searching over all $|\mathcal{E}|!$ permutations is computationally costly for a higher number of

examples. To prune the permutation search space during inference-time we propose EOptiSeq .

EOptiSeq optimizes in-context example ordering at inference time without exhaustive search of all $|\mathcal{E}|!$ permutations. Given x_{test} and examples $\{e_1, \ldots, e_E\}$, it computes embeddings via Sentence-BERT (Reimers, 2019):

$$\mathbf{v}_i = \text{SBERT}(e_i), \quad \mathbf{v}_{\text{test}} = \text{SBERT}(x_{\text{test}}) \quad (2)$$

and calculates cosine similarities:

$$s_i = \frac{\mathbf{v}_i \cdot \mathbf{v}_{\text{test}}}{\|\mathbf{v}_i\| \|\mathbf{v}_{\text{test}}\|} \quad (3)$$

The top-$\mathcal{E}$ examples are selected and the highest-ranked one based on cosine-similarity is anchored first, requiring only $(\mathcal{E}-1)!$ permutations for the remaining examples. This approach reduces evaluations from $\mathcal{E}!$ to $(\mathcal{E}-1)!$. This strategy is inspired by (Liu et al., 2024b), which shows that placing the most similar (highest-contextual-relevance) example in the first position results in the highest accuracy. The final ordering is selected using *example-free* log prob computation as in OptiSeq.

4 Evaluation

We evaluate our proposed methodologies OptiSeq and EOptiSeq across two tasks, five datasets, and five Large Language Models across three LLM families. To ensure consistency across all experiments, we utilize greedy decoding and implement 3-shot ICL for all models and datasets. This 3-shot approach allows for efficient batching of LLM inferences for permutations with in-context examples, while balancing the trade-off between larger context sizes and increased latency, which is crucial for inference-time applications. Details of our experimental setup are provided below.

4.1 Datasets

We evaluate our approach on *two* different tasks: API sequence generation and text classification. API generation requires the model to generate a sequence from a set of API candidates (i.e.) multi-label prediction, resulting in a large combinatorial solution space that prior approaches do not address (Guo et al., 2024; Lu et al., 2021). For text classification we use (i) AG News (Zhang et al., 2015), (ii) SST5 (Socher et al., 2013), and (iii) TREC (Hovy et al., 2000), while for the API generation task we use (iv) RestGPT (Song et al., 2023), and (v) ToolBench (Qin et al., 2023).

4.2 Models

We evaluate these datasets across five models from three model families with a diverse range of parameters ranging from 8B to 70B – (i) `llama-3-8b-instruct` and (ii) `llama-3-70b-instruct` (Touvron et al., 2023), (iii) `granite-13b-instruct-v2` and (iv) `granite-20b-code-instruct` (IBM, 2023), and (v) `mixtral-8x7b-instruct-v01` (MistralAI, 2023), which uses the mixture of experts approach. This heterogeneity in model parameters and architecture allows for a comprehensive assessment of performance across varying scales of computational complexity and capability.

4.3 Comparative Techniques

We compare OptiSeq and EOptiSeq against random order selection and Top-k order selection. In random selection, an order is selected at random for each test instance. In Top-k, the cosine similarity is calculated between each in-context example and the test instance, and the examples are arranged in decreasing order based on their similarity scores.

We also compare against recent baselines LocalE (Lu et al., 2021) and Influence score (Guo et al., 2024). LocalE computes output token probabilities from the first ICL task in OptiSeq and their entropy for each example order ϕ: $Ent(\phi) = -\sum_y P(y|C_\phi) \log P(y|C_\phi)$, selecting the order with median entropy to balance model confidence. The Influence score measures each order's deviation from expected probability: $I(x_t, \phi) = P(y|x_t, C_\phi) - \frac{1}{|\Phi|} \sum_{\phi' \in \Phi} P(y|x_t, C_{\phi'})$, capturing the ordering's relative impact on prediction. However, these methods rely on corpus-level properties: LocalE needs label fairness assumptions and artificial development sets, while Influence score assumes implicit and fair label distribution across orderings. In contrast, OptiSeq and EOptiSeq operate without validation datasets, performing purely inference-time optimization without corpus-level assumptions, making direct comparisons challenging.

Additionally, we focus on example ordering, not selection, ensuring all techniques operate on identical in-context examples to isolate ordering effects. This approach avoids unfair comparisons that would arise from different example sets. OptiSeq optimizes ordering using log probabilities, independent of specific examples or dataset heuristics, ensuring broad applicability and consistency

across various example sets.

4.4 Metrics

For the classification tasks, we report the Accuracy in % for the entire dataset. For the API sequence generation task, we report the following metrics:

- Accuracy: Represents the fraction of test cases where the generated API sequence exactly matches the ground truth (in correct order), compared to the ground truth sequence.
- Recall: For each test utterance, this metric represents the fraction of correctly predicted APIs (ignoring order) compared to the total number of APIs in the ground truth.
- Precision: For each test utterance, this metric represents the fraction of correctly predicted APIs (ignoring order) compared to the total number of APIs in the predicted sequence.

5 Results and Discussions

OptiSeq shows improved performance over Top-K and random selection. Table 1 highlights experimental results for different tasks. OptiSeq achieves an average improvement of 10.5% points over random selection, 9.05% points over Top-K, 6.5 % over LocalE and 5.5 % over Influence score for the API sequence generation task. For text classification task, OptiSeq demonstrates an approximate improvement of 6% points compared to random selection and Top-K and 4 % over LocalE and Influence score. LocalE and Influence take into account example ordering while measuring log probability, which reduces distinguishability among order permutations. OptiSeq evaluates all permutations and then analyzes the output sequences in an *example-free* setting, which leads to performance improvements due to better distinguishability. EOptiSeq, which builds on principles of OptiSeq and Top-K, performs marginally better than Top-K but worse than OptiSeq. This is attributed to the fact that it evaluates fewer permutations than OptiSeq but still uses zero-shot inference to improve ICL.

The strategic ordering of a smaller number of examples in OptiSeq can significantly enhance performance compared to using a larger set of examples in a random/Top-K order. As highlighted in Figure 7, ordered 3 shot ICL using OptiSeq performs better than 4 and 5 shot ICL using a random order and Top-K. On average, for the API sequence generation task, OptiSeq performs better than random and top-k selection by 5.07% points and 2.1% points for classification task (shown in Figure 12). Adding more examples does not guarantee better performance, especially given context-length limitations. Exceeding the model's input window can lead to prompt truncation and degraded performance at inference-time. (Liu et al., 2023) demonstrates that more examples may introduce noise or redundancy, limiting generalization. OptiSeq offers a robust solution for ICL by focusing on order optimization, which remains effective even when adding more examples is infeasible.

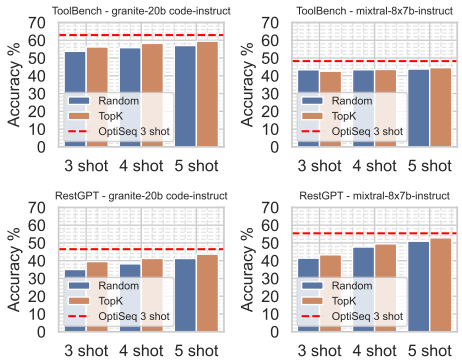


Figure 7: Comparing 3, 4, and 5 shot random/Top-K selection strategy with 3 shot OptiSeq (in red) .

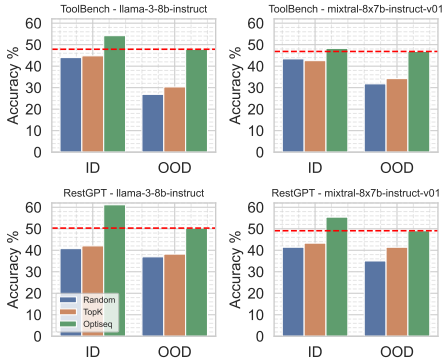


Figure 8: Comparing In-distribution with Out-of-distribution performance. Here ToolBench uses in-context examples from RestGPT and vice-versa. Results shown for 2 models.

The strategic ordering of out-of-distribution (OOD) examples using OptiSeq can lead to better performance compared to using in-distribution (ID) examples in a random/Top-K order. We evaluate the 3 shot ICL API sequence generation task for ID and OOD examples. Here,

Dataset	Model	Random	Top-K	OptiSeq	EOptiSeq	LocalE	Influence
Tool Bench	llama-3-8b-instruct	43.99	44.80	**54.18**	51.12	48.77	50.30
	llama-3-70b-instruct	58.24	59.06	**68.43**	65.37	63.03	64.56
	granite-20b-code-instruct	53.76	56.21	**62.93**	60.28	60.32	61.54
	granite-13b-instruct-v2	43.42	44.41	**47.76**	46.37	43.31	44.64
	mixtral-8x7b-instruct-v01	43.38	42.56	**48.26**	46.23	46.90	42.26
RestGPT	llama-3-8b-instruct	40.76	42.04	**61.15**	52.32	48.46	51.09
	llama-3-70b-instruct	54.14	57.96	**69.42**	64.34	61.48	64.97
	granite-20b-code-instruct	35.03	39.49	**46.49**	42.43	38.15	38.15
	granite-13b-instruct-v2	24.84	24.20	**30.57**	28.02	25.87	26.71
	mixtral-8x7b-instruct-v01	41.41	43.31	**55.41**	51.59	43.17	42.15
AGNews	llama-3-8b-instruct	72.13	73.89	**78.54**	75.64	74.44	76.53
	llama-3-70b-instruct	75.00	76.50	**81.00**	78.00	77.50	79.03
	granite-20b-code-instruct	62.90	66.31	**73.94**	69.27	61.99	65.92
	granite-13b-instruct-v2	59.81	58.69	**61.36**	59.98	59.42	60.52
	mixtral-8x7b-instruct-v01	85.45	85.97	**89.60**	86.37	86.78	87.62
SST-5	llama-3-8b-instruct	53.10	53.10	**57.10**	54.10	54.87	54.00
	llama-3-70b-instruct	55.00	55.50	**60.00**	56.00	57.20	56.80
	granite-20b-code-instruct	27.80	30.50	**32.70**	30.70	28.66	29.16
	granite-13b-instruct-v2	46.90	47.90	**50.10**	48.90	47.15	47.45
	mixtral-8x7b-instruct-v01	52.90	53.90	**55.70**	54.90	54.38	54.85
TREC	llama-3-8b-instruct	60.60	63.20	**67.80**	66.40	63.80	64.20
	llama-3-70b-instruct	62.00	65.00	**70.00**	68.00	65.60	66.00
	granite-20b-code-instruct	50.60	53.40	**58.00**	55.60	50.98	51.34
	granite-13b-instruct-v2	40.00	43.20	**46.40**	45.20	40.36	40.69
	mixtral-8x7b-instruct-v01	62.20	66.80	**73.20**	70.60	68.94	69.40

Table 1: Updated accuracy (%) comparison of various models across datasets. Bold values indicate the highest accuracy, and underlined values represent the second-highest accuracy.

the ToolBench dataset uses examples from RestGPT, and RestGPT uses examples from ToolBench. Figure 8 shows that the performance drops when we use OOD examples across all techniques. The drop is significantly high for random and Top-K selection. On average OptiSeq using OOD examples performs better than Top-K using ID examples by 5.35% points and random selection using ID examples by 6.15% points.

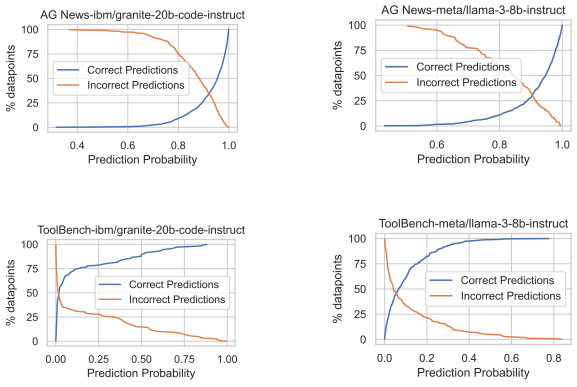


Figure 9: Probability distribution for correct and incorrect predictions, reflecting the model's confidence.

OptiSeq enhances accuracy using confidence metrics. Figure 9 shows that correct predictions generally have higher probabilities, indicating greater model confidence. The distribution of correct predictions skews towards higher confidence levels, while incorrect predictions tend to have lower probabilities. This pattern demonstrates the potential of using confidence metrics to improve model accuracy, possibly by filtering or adjusting predictions based on their confidence levels. The correlation metrics can be seen in Table 4.

OptiSeq improves instance level predictions for API sequence generation. Table 2 shows the Precision and Recall metrics of instance level predictions at run-time for API sequence generation task. OptiSeq and EOptiSeq achieve an average improvement of 3.52% in Precision and 3.21% points in Recall over Random or Top-K, respectively and 3.01 % in Precision and 3.37 % in Recall over recent baselines. This indicates that our approaches induce the inclusion of more relevant APIs in the sequence compared to baselines.

OptiSeq Overheads: OptiSeq uses batched inference to reduce evaluation latency. To assess the efficiency of batched inference, we compared single and batched inference times on an NVIDIA A5000 GPU averaged across 100 runs for each setting (single and batched) for AG News using `llama-3-8b-instruct`. Initial runs were discarded for warm-up to mitigate initialization

Dataset	Model	Random		TopK		OptiSeq		EOptiSeq		LocalE		Influence	
		Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Tool Bench	llama-3-8b-instruct	71.47	70.58	71.63	70.45	77.75	77.12	75.77	74.58	74.33	73.14	75.61	74.18
	llama-3-70b-instruct	82.57	82.08	83.04	82.95	85.03	84.07	84.79	83.88	83.68	82.72	84.23	83.11
	granite-20b-code-instruct	75.83	75.18	77.35	77.10	81.78	81.64	80.72	79.34	81.07	80.16	81.31	79.96
	granite-13b-instruct-v2	72.03	74.85	73.20	76.29	76.55	78.58	75.16	77.04	69.72	74.40	71.18	74.85
	mixtral-8x7b-instruct-v01	74.16	76.15	73.78	76.33	81.85	84.21	80.12	81.97	79.28	80.99	67.72	68.24
RestGPT	llama-3-8b-instruct	75.43	75.04	75.74	75.62	78.69	78.72	77.52	77.11	75.59	75.33	76.68	75.97
	llama-3-70b-instruct	74.70	75.35	74.22	75.15	77.64	78.07	76.79	77.13	75.18	75.23	76.43	77.45
	granite-20b-code-instruct	74.76	74.23	75.28	74.84	76.27	76.01	75.86	74.79	75.51	75.01	73.71	72.77
	granite-13b-instruct-v2	74.12	74.56	74.68	75.12	76.23	76.78	75.89	76.34	73.56	73.71	74.39	71.97
	mixtral-8x7b-instruct-v01	74.56	74.84	75.24	75.76	77.53	77.47	77.10	77.32	75.51	76.01	73.71	72.77

Table 2: Precision (Prec.) and Recall (Rec.) in % for different models and datasets.

overhead and early measurement variability. Single Prompt Inference: 13.44s per prompt. Batched Inference (6 or 3! Prompts): 16.87s for the full batch. Batching significantly reduces per-prompt latency (from 13.44s to 2.81s), making inference more efficient. This reduces the latency of OptiSeq to ≈ 2 sequential single-prompt LLM calls. The average hit rate stabilizes around $\sim$1.5$\times$as seen in Figure 13.

6 Related Work

Methods for Optimizing Example Ordering: (Xu et al., 2024) formulates example ordering as an optimization problem. Using label proportion, it improves accuracy and reduces model miscalibration across classification tasks. (Zhang et al., 2024) Batch-ICL aggregates meta-gradients from independent computations, making the model agnostic to example order while improving performance and reducing computational costs. (Wu et al., 2022) Proposes a select-then-rank framework for self-adaptive ICL, achieving significant performance gains by dynamically optimizing example orders. Inspired by how humans learn, (Liu et al., 2024c) gradually increases example complexity, improving instance and corpus-level performance through curriculum ordering. Unlike batch or curriculum-based approaches, OptiSeq performs instance-specific optimization rather than applying a general rule.

Example Selection and Ranking Techniques: (Gupta et al., 2023) Selects diverse and informative examples using BERTScore-Recall, significantly outperforming independent ranking methods. DEmO (Guo et al., 2024) identifies optimal example orders for individual instances through label fairness and content-free metrics. (Liu et al., 2024a) formulates example selection as a sequential process using beam search to optimize inter-relationships and diversity among examples. EXPLORA (Purohit et al., 2024) improves task-specific exemplar selection for complex reasoning tasks by efficiently estimating scoring function parameters, reducing computational cost while enhancing performance. CEIL (Ye et al., 2023) models example selection as a subset selection problem using Determinantal Point Processes and contrastive learning to optimize example interactions across diverse NLP tasks. OptiSeq goes beyond static or sequential ranking by dynamically testing every possible example order and using log probabilities to determine the best sequence.

Theoretical Insights and Adaptive Strategies in ICL (Chandra et al., 2024) demonstrates that dynamically adjusting the number of in-context examples improves task-specific performance over fixed hyperparameters. (Zhao et al., 2024) examines the limitations of ICL for instruction-following tasks and identifies key parameters for alignment. (Long et al., 2024) employs adversarial learning to iteratively refine prompts, significantly improving performance across diverse tasks. OptiSeq avoids the complexity of adversarial learning or fine-tuning, providing an inference-time method for ICL.

7 Conclusion

This study highlights the impact of in-context example ordering on LLM performance. OptiSeq significantly enhances accuracy by optimizing example orderings. By evaluating all possible orderings and selecting the highest confidence score based on input log probabilities, OptiSeq consistently improved accuracy by 5.5 to 10.5 percentage points over baselines. This improvement was observed across various generation and text classification tasks for three different model families with diverse parameter ranges, demonstrating OptiSeq's robustness and versatility in enhancing ICL.

Limitations

OptiSeq achieves better results than Top-K and Random order selection but requires evaluating $k = |\mathcal{E}|!$ permutations of prompts, which introduces computational challenges as the number of in-context examples grows. Our experiments confirm that fixed example orderings struggle to generalize across tasks, instances, and model architectures (Section 2). This limitation arises from the strong dependence between the optimal ordering, the characteristics of the examples, and the model-specific biases. Additionally, our work focuses on instance-specific adaptive ordering, which optimizes example sequences for individual inputs. While this approach maximizes performance for a given instance, we recognize that it does not inherently address cross-instance or cross-model generalization. A promising future direction is exploring methods using meta-learning, or domain adaptation to learn transferable ordering strategies that can be applied across various instances and models without repeated optimization. While our current approach evaluates all factorial permutations, this becomes impractical in many-shot settings (e.g., > 50 examples) as seen in Agarwal et al. (2024), where the ordering problem remains relevant. In such scenarios, search-based strategies like beam search can be employed to efficiently prune the permutation space and reduce computational overhead. Furthermore, while the approach relies on logarithmic probability evaluations for optimal permutation selection, not all LLM platforms and APIs services currently support token-level log-probability computation. However, as models continue to evolve and LLM platforms expand to include more granular scoring features, the applicability and efficiency of OptiSeq are likely to improve, paving the way for broader adoption in real-world scenarios.

References

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. 2024. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*.

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.

Manish Chandra, Debasis Ganguly, and Iadh Ounis. 2024. One size doesn't fit all: Predicting the number of examples for in-context learning.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2021. Meta-learning via language model in-context tuning. *arXiv preprint arXiv:2110.07814*.

Qi Guo, Leiyu Wang, Yidong Wang, Wei Ye, and Shikun Zhang. 2024. What makes a good order of examples in in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14892–14904.

Shivanshu Gupta, Matt Gardner, and Sameer Singh. 2023. Coverage-based example selection for in-context learning. *arXiv preprint arXiv:2305.14907*.

Eduard H Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. 2000. Question answering in webclopedia. In *TREC*, volume 52, pages 53–56.

IBM. 2023. Granite: Scaling language models with ibm's efficient architecture. *IBM Research Journal*. Available at https://research.ibm.com/granite.

Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. 2024a. se2: Sequential example selection for in-context learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5262–5284.

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*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024b. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

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

Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. 2024c. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *arXiv preprint arXiv:2402.10738*.

Do Long, Yiran Zhao, Hannah Brown, Yuxi Xie, James Zhao, Nancy Chen, Kenji Kawaguchi, Michael Shieh, and Junxian He. 2024. Prompt optimization via adversarial in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7308–7327.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

MistralAI. 2023. Mixtral: A diverse and scalable instruction-tuned language model. *Mistral AI Technical Report*. Available at https://mistral.ai/mixtral.

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

Kiran Purohit, Raghuram Devalla, Krishna Mohan Yerragorla, Sourangshu Bhattacharya, Avishek Anand, et al. 2024. Explora: Efficient exemplar subset selection for complex reasoning. *arXiv preprint arXiv:2411.03877*.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Yifan Song, Weimin Xiong, Dawei Zhu, Wenhao Wu, Han Qian, Mingbo Song, Hailiang Huang, Cheng Li, Ke Wang, Rong Yao, et al. 2023. Restgpt: Connecting large language models with real-world restful apis. *arXiv preprint arXiv:2306.06624*.

Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama 3: Open and efficient foundation language models. *arXiv preprint arXiv:2307.09288*.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2022. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *arXiv preprint arXiv:2212.10375*.

Zhichao Xu, Daniel Cohen, Bei Wang, and Vivek Srikumar. 2024. In-context example ordering guided by label distributions. *Preprint*, arXiv:2402.11447.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5443–5456.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

Kaiyi Zhang, Ang Lv, Yuhan Chen, Hansen Ha, Tao Xu, and Rui Yan. 2024. Batch-icl: Effective, efficient, and order-agnostic in-context learning. *arXiv preprint arXiv:2401.06469*.

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

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2023. What makes good examples for visual in-context learning? *Advances in Neural Information Processing Systems*, 36:17773–17794.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Is in-context learning sufficient for instruction following in llms? *arXiv preprint arXiv:2405.19874*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

A Appendix: Additional Figures

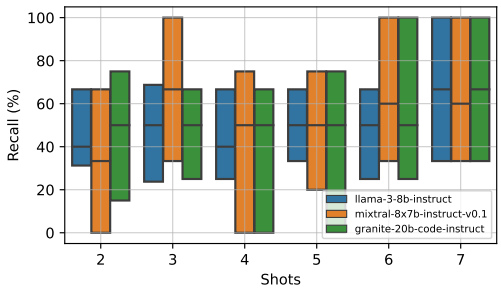
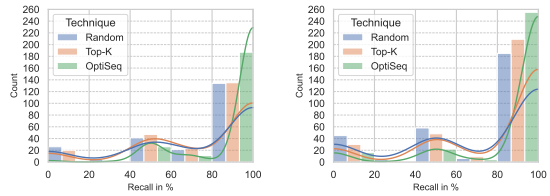


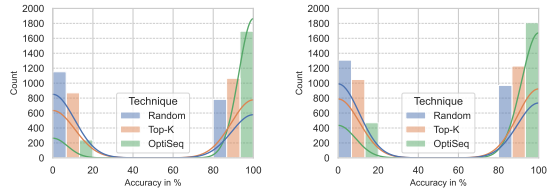
Figure 10: Variation in API recall for different number of in-context examples for ToolBench dataset

Increasing model size or number of examples does not mitigate order sensitivity. We randomly sample 100 test instances from the ToolBench dataset (Qin et al., 2023). We vary the number of in-context examples between $2-7$, and use LLMs of varying sizes. We observe that adding more examples does not mitigate the prompt sensitivity, as illustrated in Figure 10.



(a) ToolBench dataset on `llama-3-8b-instruct`.

(b) ToolBench dataset on `granite-20b-code-instruct`.



(c) AG News dataset on `llama-3-8b-instruct`.

(d) AG News dataset on `granite-20b-code-instruct`.

Figure 11: Distribution of Recall values for ToolBench (top row) and AG News (bottom row) datasets.

OptiSeq improves the number of correct predictions as compared to Random and Top-K selection. We sample test cases for different tasks and observe the performance spread. Figure 11 shows the distribution of Recall values for API Sequence generation task and Accuracy values for the classification task. OptiSeq performs better than Random and Top-K selection – shifts further towards the 100% – by increasing the number of correctly predicted sequences.

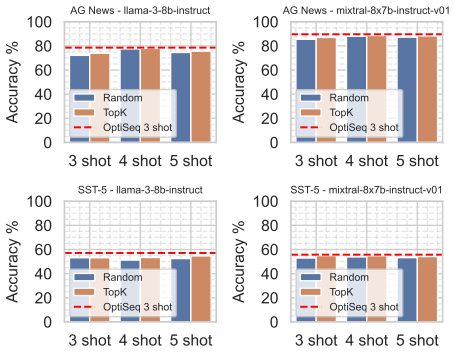


Figure 12: Comparing 3, 4, and 5 shot random/Top-K selection strategy with 3 shot OptiSeq (in red) for the classification task.

Dataset	Labels	Type
ToolBench	100	Multi-class
RestGPT	75	Multi-class
AG_news	4	Single-class
SST-5	5	Single-class
TREC	6	Single-class

Table 3: Number of labels for each dataset

Figure 12 illustrates the performance of OptiSeq on classification tasks, comparing it to 3-, 4-, and 5-shot in-context learning (ICL) using both random order and Top-K selection of examples. The results demonstrate that OptiSeq, by strategically ordering a smaller set of examples, outperforms approaches using larger sets of examples in random or Top-K order.

B Appendix: Dataset Labels

Table 3 presents the number of possible labels for each dataset in our study. This information is crucial for understanding the complexity of the classification tasks the model must perform.

As shown in the table, ToolBench and RestGPT are multi-class classification tasks with 100 and 75 possible labels, respectively. These datasets present more complex classification challenges due to their higher number of potential outcomes. In contrast, AG_news, SST-5, and TREC are single-class classification tasks with fewer labels (4, 5, and 6, respectively), representing comparatively simpler classification problems.

The variation in the number of labels and classification types across these datasets allows for a comprehensive evaluation of our model's performance across different levels of task complexity.

C Appendix: Correlation between probability and accuracy

Table 4 presents the correlation metrics between predicted rankings and ground truth orderings across two datasets (AG_News and ToolBench) using two large language models: LLaMA-3-8B-Instruct and Granite-20B-Code-Instruct. Both Spearman and Kendall's τ coefficients show strong, statistically significant correlations (p-value = 0), indicating that the predicted orderings are well-aligned with the ground truth across tasks. This can also be seen in Figure 9.

Dataset	Model	Spearman	P	Kendall-τ	P
AG_News	llama-3-8b-ins.	0.644	0	0.526	0
	granite-20b-code-ins.	0.629	0	0.514	0
ToolBench	llama-3-8b-instruct	0.542	0	0.443	0
	granite-20b-code-instruct	0.544	0	0.445	0

Table 4: Correlation metrics (Spearman and Kendall-τ, along with respective P(P-values)) for different models on AG_News and ToolBench datasets.

D Appendix: OptiSeq Inference-time hit as compared to single inference

To compare the inference time of **OptiSeq** to a **single-inference** baseline, we define the hit rate as:

$$\text{Hit Rate} = \frac{\text{OptiSeq Inference Time}}{\text{Single Inference Time}} - 1$$

A value of 0 indicates that OptiSeq matches baseline performance. A **positive** value indicates a **slower** OptiSeq (i.e., time overhead), whereas a **negative** value indicates a **speedup**. For 100 runs of OptiSeq, we discard the first 3 to account for warm-up effects and reduce variability due to initialization and system overhead. The average hit rate stabilizes around $\sim$1.5$\times$ as seen in Figure 13. for 3-shot inference.

E Representativeness of the evaluation setting

OptiSeq operates *strictly after* retrieval: it takes a fixed pool of k examples and returns the permutation $\pi^\star = \arg\max_{\pi \in S_k} \log P_\theta(\hat{y}_\pi \mid \text{input})$. Because this objective depends only on *relative* answer log-likelihoods, its effectiveness is *orthogonal* to **(i)** *which selector* produced the pool (random, BM25, DPP, learned retriever), **(ii)** *how many* shots are in the pool (the optimal k-ordering is always a feasible candidate in the $k!$-pool), and **(iii)** *task family* (classification, tool invocation).

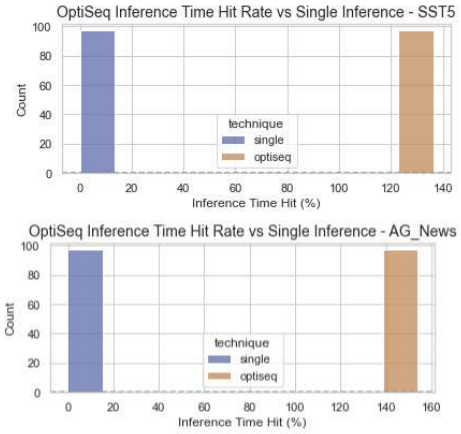


Figure 13: Comparing inference hit rate of OptiSeq vs Single Inference Time

F Appendix: Prompt structure and sample inputs and outputs

We present the prompt structures and sample inputs and outputs for each dataset we experimented with.

F.1 ToolBench Dataset

The ToolBench dataset (Qin et al., 2023) is a specialized benchmark designed to evaluate large language models (LLMs) on their ability to predict and sequence API calls in multi-tool reasoning tasks. It includes diverse natural language instructions paired with ground-truth API sequences across various domains and tools. Each data point typically contains an instruction, a catalog of available APIs, a set of in-context examples, and an evaluation metric (e.g., accuracy, precision, and recall). The dataset is tailored to assess the impact of in-context learning, permutation ordering, and tool usage alignment with human-designed workflows. ToolBench supports research on improving multi-tool coordination, mitigating biases in tool selection, and optimizing task-specific API predictions. For the ToolBench dataset in particular, we evaluate with *G1 – single-tool*, *G2 – intra-category* multi-tool, and *G3 – intra-collection multi-tool* parts of the dataset (Qin et al., 2023). We report aggregated results for all three parts of ToolBench to provide a concise summary. This approach enhances clarity, improves statistical robustness, and demonstrates the model's ability to generalize across tasks and scenarios. The prompt structure can be seen in 5. We can see instances of in-context ordering sensitivity and identification of the relevant sequences using log probs in OptiSeq in 6.

- **Task Description:**
  - "I will ask you to perform a task. Your job is to come up with a sequence of APIs in a comma-separated list in the format that will perform the task. Start the list with « and end it with ». Do not include anything other than the API name. Use the APIs below to answer the question posed to you. Avoid the use of any other text unless specified."
- **APIs Available:**
  - `asin_data.category`: Retrieve category results from Amazon.
  - `asin_data.offers`: Retrieve seller offers for a product on Amazon.
  - `asin_data.reviews`: Retrieve customer reviews for a product on Amazon.
  - `asin_data.search`: Retrieve search results for an Amazon domain.
  - `asin_data.product`: Retrieve details of a single product on Amazon.
  - `keyword_analysis.popularsitesforquery`: Get popular sites for a search query.
  - `keyword_analysis.similarqueries`: Get similar queries for a search query.
  - `spellout.rulesets`: List available rule sets for a given language.
  - `immersiverouletteapi.statistics`: Get statistics of wheel results.
  - `diffbot.article_api`: Extract clean article text from web pages.
  - `covid_19_india.get_details`: Get coronavirus updates for India.
  - `realtor_data_api_for_real_estate.realtorpropertylist`: Get Realtor Property List.
  - `generate_linkedin_leads.get_available_locations`: Get available locations for LinkedIn leads.
  - `virtual_number.get_all_countries`: Get list of available countries.
  - ...
- **Examples:**
  - **Utterance:** "I am a fitness enthusiast and I want to buy a fitness tracker. Can you suggest some top-rated fitness trackers available on Amazon along with their features and prices?"
  - **Sequence:** «`asin_data.search, asin_data.product`»
  - **Utterance:** "I'm a football enthusiast and I want to know more about Lionel Messi's career. Can you provide me with information about Messi's clubs, managers, teammates, and referees?"
  - **Sequence:** «`theclique.transfermarkt_search, theclique.transfermarkt_details`»
  - **Utterance:** "I want to plan a surprise birthday party for my friend. Can you suggest popular sites and main keywords for the search query 'birthday party ideas'?"
  - **Sequence:** «`keyword_analysis.popularsitesforquery, keyword_analysis.querykeywords`»
- **Test Utterance:**
  - **Utterance:** "I want to explore trending content on social media. Can you provide me with the current trending feed of videos? I would like to limit the output to 20 records. Please include the direct URLs to the videos and their statistics. Additionally, if possible, I would like to filter the feed based on a specific hashtag, such as #summer."
  - **Sequence:**

Table 5: Prompt Structure for API Sequencing Task using ToolBench

Utterance	Generated Sequence	Order	Precision	Recall	Accuracy	Log Prob OptiSeq
Utterance: I need to retrieve the pending messages from my device with ID 123456. Please provide the pending messages using my TrumpetBox Cloud API KEY.	trumpetbox_cloud.devices_getasingledeviceinfofromaccount, trumpetbox_cloud.messages_getpendingmessagesfromaccount	1	**100.0**	**100.0**	**100.0**	**-1.54**
	trumpetbox_cloud.messages_getpendingmessagesfromaccount	2	100.0	50.0	0.0	-3.55
	trumpetbox_cloud.devices_getasingledeviceinfofromaccount, trumpetbox_cloud.messages_getpendingmessagesfromaccount	3	**100.0**	**100.0**	**100.0**	**-1.54**
	trumpetbox_cloud.messages_getpendingmessagesfromaccount	4	100.0	50.0	0.0	-3.55
	trumpetbox_cloud.messages_getpendingmessagesfromaccount	5	100.0	50.0	0.0	-3.55
	trumpetbox_cloud.devices_getasingledeviceinfofromaccount, trumpetbox_cloud.messages_getpendingmessagesfromaccount	6	**100.0**	**100.0**	**100.0**	**-1.54**
Utterance: I need to track the performance of my family's crypto investments. Can you provide me with a list of our current positions for Bitcoin, Ethereum, and Ripple, along with their historical data and market trends?	crypto_grana.list.position.for.each.crypto, crypto_grana.list.histories	1	**100.0**	**100.0**	**100.0**	**-1.67**
	crypto_grana.list.position.for.each.crypto	2	100.0	50.0	0.0	-5.61
	crypto_grana.list.position.for.each.crypto	3	100.0	50.0	0.0	-5.61
	crypto_grana.list.position.for.each.crypto, crypto_grana.list.histories	4	**100.0**	**100.0**	**100.0**	**-1.67**
	crypto_grana.list.position.for.each.crypto	5	100.0	50.0	0.0	-5.61
	crypto_grana.list.position.for.each.crypto, crypto_grana.list.histories	6	**100.0**	**100.0**	**100.0**	**-1.67**
Utterance: I run a company that organizes 4D lottery events. Can you give me the list of available 4D companies? It would also be helpful to get the past results from January 1, 2020, to March 31, 2020. Finally, I need the 4D results for Magnum on January 29, 2020.	4d_results.get.4d.company.list, 4d_results.get.past.results.(1.year), 4d_results.get.4d.results	1	**100.0**	**100.0**	**100.0**	**-3.49**
	4d_results.get.4d.company.list, 4d_results.get.past.results.(1.year), 4d_results.get.4d.results	2	**100.0**	**100.0**	**100.0**	**-3.49**
	4d_results.get.4d.company.list, 4d_results.get.past.results.(1.year), 4d_results.get.4d.results	3	**100.0**	**100.0**	**100.0**	**-3.49**
	4d_results.get.4d.company.list, 4d_results.get.past.results.(1.year), 4d_results.get.4d.results	4	**100.0**	**100.0**	**100.0**	**-3.49**
	4d_results.get.4d.company.list, 4d_results.get.past.results.(1.year)	5	100.0	66.67	0.0	-4.55
	4d_results.get.4d.company.list, 4d_results.get.past.results.(1.year), 4d_results.get.4d.results	6	**100.0**	**100.0**	**100.0**	**-3.49**
Utterance: I'm a weather enthusiast and I'm interested in studying aviation weather data. Can you provide me with the most recent TAFs for the next 24 hours? I also want to see the most recent METARs from the past 2 hours. Please include the temperature, dew point, and wind direction in both reports.	aviation_weather_center.most_recent.tafs, aviation_weather_center.most_recent.metars	1	0.0	0.0	0.0	-6.18
	aviation_weather_center.most.recent.tafs, aviation_weather_center.most.recent.metars	2	**100.0**	**100.0**	**100.0**	**-4.17**
	aviation_weather_center.most.recent.tafs, aviation_weather_center.most.recent.metars	3	**100.0**	**100.0**	**100.0**	**-4.17**
	aviation_weather_center.most.recent.tafs, aviation_weather_center.most.recent.metars	4	**100.0**	**100.0**	**100.0**	**-4.17**
	aviation_weather_center.most.recent.tafs, aviation_weather_center.most.recent.metars	5	**100.0**	**100.0**	**100.0**	**-4.17**
	aviation_weather_center.most_recent.tafs, aviation_weather_center.most_recent.metars	6	0.0	0.0	0.0	-6.18
Utterance: I'm developing an art events app and I need a list of all genres of the events. Can you provide me with this information? It would be helpful if you could also give me a list of all locations where art events take place.	art_openings_italy.get.all.genres, art_openings_italy.get.all.locations	1	**100.0**	**100.0**	**100.0**	**-1.12**
	art_openings_italy.get.all.genres, art_openings_italy.get.all.locations	2	**100.0**	**100.0**	**100.0**	**-1.12**
	art_openings_italy.get.all.genres, art_openings_italy.get.all.locations	3	**100.0**	**100.0**	**100.0**	**-1.12**
	art_openings_italy.get.all.genres, art_openings_italy.get.all.locations	4	**100.0**	**100.0**	**100.0**	**-1.12**
	art_openings_italy.get.all.genres, art_openings_italy.get.all.locations	5	**100.0**	**100.0**	**100.0**	**-1.12**
	art_openings_italy.get_all.genres, art_openings_italy.get_all.locations	6	0.0	0.0	0.0	-3.85
Utterance: I'm planning a family vacation and need to check the availability of a specific product. Can you provide me with the details of the product 'XYZ' including its price, stock quantity, and ID? Additionally, I want to find a client with the name 'Mark' and his contact details.	realtor_data_api_for_real_estate.realtorpropertylist, realtor_data_api_for_real_estate.realtoragentlist	1	0.0	0.0	0.0	-8.30
	capacitacionangular.productos, capacitacionangular.cliente	2	**100.0**	**100.0**	**100.0**	**-6.69**
	realtor_data_api_for_real_estate.realtorpropertylist, realtor_data_api_for_real_estate.realtoragentlist	3	0.0	0.0	0.0	-8.30
	capacitacionangular.productos, capacitacionangular.cliente	4	**100.0**	**100.0**	**100.0**	**-6.69**
	capacitacionangular.productos, capacitacionangular.cliente	5	**100.0**	**100.0**	**100.0**	**-6.69**
	realtor_data_api_for_real_estate.realtorpropertylist, realtor_data_api_for_real_estate.realtoragentlist	6	0.0	0.0	0.0	-8.30
Utterance: I'm planning a vacation to Bali and I want to explore the most popular Instagram profiles from the area. Can you provide me with the Instagram profiles of famous travelers who have visited Bali recently? Additionally, I would like to see their stories.	access_instagram.instagram.endpoint_copy, access_instagram.instagram.endpoint	1	**100.0**	**100.0**	**100.0**	**-3.09**
	access_instagram.instagram.endpoint_copy	2	100.0	50.0	0.0	-3.39
	access_instagram.instagram.endpoint_copy	3	100.0	50.0	0.0	-3.39
	access_instagram.instagram.endpoint_copy, access_instagram.instagram.endpoint	4	**100.0**	**100.0**	**100.0**	**-3.09**
	access_instagram.instagram.endpoint_copy, access_instagram.instagram.endpoint	5	**100.0**	**100.0**	**100.0**	**-3.09**
	access_instagram.instagram.endpoint_copy, access_instagram.instagram.endpoint	6	**100.0**	**100.0**	**100.0**	**-3.09**

Table 6: Generated Sequences and Metrics for the ToolBench Dataset with `meta/llama-3-8b-instruct`

F.2 RestGPT Dataset

The RestBench (Song et al., 2023) dataset is a high-quality test set designed to evaluate large language models (LLMs) on task execution in two primary domains: the TMDB movie database and the Spotify music player. It includes natural language queries and instructions that require models to reason about and generate API call sequences for tasks such as retrieving movie details, searching for music tracks, creating playlists, and handling user preferences. Each data point comprises a user query, a structured API catalog, and ground-truth API sequences, with a focus on multi-step reasoning and alignment with user intents. RestBench serves as a robust benchmark for assessing the capabilities of LLMs in handling complex domain-specific workflows, demonstrating their potential in real-world applications across entertainment platforms. The prompt structure can be seen in 7. We can see instances of in-context ordering sensitivity and identification of the relevant sequences using log probs in OptiSeq in 8.

- **Task Description:**
  - "I will ask you to perform a task. Your job is to come up with a sequence of APIs in a comma-separated list in the format that will perform the task. Start the list with « and end it with ». Do not include anything other than the API name. Use the APIs below to answer the question posed to you. Avoid the use of any other text unless specified."
- **APIs Available:**
  - `SearchCollection`: GET /search/collection
  - `CollectionImages`: GET /collection/collection_id/images
  - `SearchPerson`: GET /search/person
  - `PersonMovieCredits`: GET /person/person_id/movie_credits
  - `SearchMovie`: GET /search/movie
  - `MovieCredits`: GET /movie/movie_id/credits
  - `MovieTopRated`: GET /movie/top_rated
  - `SearchCompany`: GET /search/company
  - `CompanyImages`: GET /company/company_id/images
  - `PersonImages`: GET /person/person_id/images
  - `MovieSimilar`: GET /movie/movie_id/similar
  - `MovieReviews`: GET /movie/movie_id/reviews
  - `MovieRecommendations`: GET /movie/movie_id/recommendations
  - `PersonTvCredits`: GET /person/person_id/tv_credits
  - `SearchTv`: GET /search/tv
  - `TvRecommendations`: GET /tv/tv_id/recommendations
  - `Trending`: GET /trending/media_type/time_window
  - `.....`
- **Examples:**
  - **Utterance:** "Give me the number of movies directed by Sofia Coppola."
  - **Sequence:** «`SearchPerson, PersonMovieCredits`»
  - **Utterance:** "Who was the lead actor in the movie The Dark Knight?"
  - **Sequence:** «`SearchMovie, MovieCredits`»
  - **Utterance:** "Who directed the top-1 rated movie?"
  - **Sequence:** «`MovieTopRated, MovieCredits`»
- **Test Utterance:**
  - **Utterance:** "I'm watching the tv series The Last Of Us and I need some more recommendations"
  - **Sequence:**

Table 7: Prompt Structure for API Sequencing Tasks in RestGPT

Utterance	Generated Sequence	Order	Precision	Recall	Accuracy	Log Prob OptiSeq
Utterance: I just finished watching Titanic and I want some other movie recommendations	`Movie, MovieRecommendations`	1	50.0	50.0	0.0	-18.28
	`SearchMovie, MovieRecommendations`	2	**100.0**	**100.0**	**100.0**	**-6.76**
	`MovieRecommendations`	3	100.0	50.0	0.0	-9.56
	`Movie, MovieRecommendations`	4	50.0	50.0	0.0	-19.59
	`Movie, MovieRecommendations`	5	50.0	50.0	0.0	-18.79
	`Movie, MovieRecommendations`	6	50.0	50.0	0.0	-19.14
Utterance: What dose the lead actor of Titanic look like?	`SearchMovie, MovieCredits, PersonImages`	1	**100.0**	**100.0**	**100.0**	**-5.48**
	`SearchMovie, MovieCredits, PersonImages`	2	**100.0**	**100.0**	**100.0**	**-6.62**
	`SearchMovie, MovieCredits, PersonImages`	3	**100.0**	**100.0**	**100.0**	**-7.25**
	`SearchPerson, PersonImages`	4	50.0	33.33	0.0	-16.95
	`SearchMovie, MovieCredits, PersonImages`	5	**100.0**	**100.0**	**100.0**	**-5.83**
	`SearchPerson, PersonImages`	6	50.0	33.33	0.0	-17.07
Utterance: What is the logo of the Walt Disney?	`SearchCompany, CompanyImages`	1	**100.0**	**100.0**	**100.0**	**-5.09**
	`SearchCompany, CompanyImages`	2	**100.0**	**100.0**	**100.0**	**-5.83**
	`SearchCompany, CompanyImages`	3	**100.0**	**100.0**	**100.0**	**-5.69**
	`SearchCompany, CompanyImages`	4	**100.0**	**100.0**	**100.0**	**-5.32**
	`SearchCompany, CompanyImages, Movie, MovieReleaseDates, andendwitha, SearchMovie, Movie`	5	28.57	100.0	0.0	-94.40
	`SearchCompany, CompanyImages`	6	**100.0**	**100.0**	**100.0**	**-5.32**
Utterance: Who directed the top-1 rated movie?	`MovieTopRated, Movie, Credit`	1	33.33	50.0	0.0	-35.84
	`MovieTopRated, MovieCredits`	2	**100.0**	**100.0**	**100.0**	**-6.34**
	`MovieTopRated, MovieCredits`	3	**100.0**	**100.0**	**100.0**	**-6.89**
	`MovieTopRated, MovieCredits`	4	**100.0**	**100.0**	**100.0**	**-6.62**
	`MovieTopRated, MovieCredits`	5	**100.0**	**100.0**	**100.0**	**-6.95**
	`MovieTopRated, MovieCredits`	6	**100.0**	**100.0**	**100.0**	**-6.71**
Utterance: Who is the director of the movie "Twilight"?	`SearchMovie, MovieCredits`	1	**100.0**	**100.0**	**100.0**	**-5.71**
	`SearchMovie, Mcosy.credits`	2	50.0	50.0	0.0	-54.40
	`SearchMovie, MovieCredits`	3	**100.0**	**100.0**	**100.0**	**-7.08**
	`SearchMovie, MovieCredits`	4	**100.0**	**100.0**	**100.0**	**-6.59**
	`SearchMovie, MovieCredits`	5	**100.0**	**100.0**	**100.0**	**-5.69**
	`SearchMovie, MovieCredits`	6	**100.0**	**100.0**	**100.0**	**-5.83**
Utterance: Who is the most popular person?	`PersonPopular`	1	**100.0**	**100.0**	**100.0**	**-0.22**
	`Trending, PersonPopular`	2	50.0	100.0	0.0	-16.97
	`PersonPopular`	3	**100.0**	**100.0**	**100.0**	**-0.10**
	`PersonPopular`	4	**100.0**	**100.0**	**100.0**	**-0.10**
	`PersonPopular`	5	**100.0**	**100.0**	**100.0**	**-0.80**
	`PersonPopular`	6	**100.0**	**100.0**	**100.0**	**-0.10**
Utterance: Who was the lead actor in the movie The Dark Knight?	`SearchMovie, MovieCredits`	1	**100.0**	**100.0**	**100.0**	**-6.86**
	`SearchMovie, MovieCredits`	2	**100.0**	**100.0**	**100.0**	**-7.35**
	`SearchMovie, MovieCredits, ortrailing`	3	66.67	100.0	0.0	-54.92
	`SearchMovie, MovieCredits`	4	**100.0**	**100.0**	**100.0**	**-7.23**
	`SearchMovie, MovieCredits`	5	**100.0**	**100.0**	**100.0**	**-7.22**
	`SearchMovie, B`	6	50.0	50.0	0.0	-36.12
Utterance: give me a image for the collection Star Wars	`SearchCollection, CollectionImages`	1	**100.0**	**100.0**	**100.0**	**-5.37**
	`SearchCollection, CollectionImages`	2	**100.0**	**100.0**	**100.0**	**-5.69**
	`CollectionImages`	3	100.0	50.0	0.0	-5.75
	`SearchCollection, CollectionImages`	4	**100.0**	**100.0**	**100.0**	**-5.60**
	`SearchCollection, CollectionImages`	5	**100.0**	**100.0**	**100.0**	**-5.55**
	`SearchCollection, CollectionImages`	6	**100.0**	**100.0**	**100.0**	**-5.35**
Utterance: give me a photo belong to the second episode of the first season of the Witcher	`SearchTv, TvSeasonEpisodeImages`	1	**100.0**	**100.0**	**100.0**	**-6.13**
	`SearchTv, TvSeasonEpisodeImages`	2	**100.0**	**100.0**	**100.0**	**-6.36**
	`TvSeasonEpisodeImages,`	3	50.0	50.0	0.0	-20.16
	`SearchTv, TvSeasonEpisodeImages`	4	**100.0**	**100.0**	**100.0**	**-6.36**
	`SearchTv, TvSeasonEpisodeImages`	5	**100.0**	**100.0**	**100.0**	**-6.60**
	`SearchTv, TvSeasonEpisodeImages`	6	**100.0**	**100.0**	**100.0**	**-6.23**
Utterance: give me the number of movies directed by Sofia Coppola	`SearchPerson, PersonMovieCredits`	1	**100.0**	**100.0**	**100.0**	**-6.60**
	`SearchPerson, PersonMovieCredits`	2	**100.0**	**100.0**	**100.0**	**-6.35**
	`SearchPerson, PersonMovieCredits`	3	**100.0**	**100.0**	**100.0**	**-7.14**
	`SearchPerson, PersonMovieCredits, andenditwitha`	4	66.67	100.0	0.0	-55.08
	`SearchPerson, PersonMovieCredits`	5	**100.0**	**100.0**	**100.0**	**-6.72**
	`SearchPerson, PersonMovieCredits, MOVIE_TOP_RATED`	6	66.67	100.0	0.0	-33.02
Utterance: tell me a TV show recently directed by Catherine Hardwicke	`SearchPerson, PersonTvCredits`	1	**100.0**	**100.0**	**100.0**	**-5.31**
	`SearchPerson, PersonTvCredits, Tv, andenditwitha`	2	50.0	100.0	0.0	-52.12
	`SearchCompany, CompanyImages, SearchPerson, PersonTvCredits`	3	50.0	100.0	0.0	-25.36
	`Trending, TvCredits, DetectcurrentTVshow`	4	0.0	0.0	0.0	-60.15
	`SearchPerson, PersonTvCredits`	5	**100.0**	**100.0**	**100.0**	**-5.31**
	`SearchPerson, PersonTvCredits, Trending, movie_type=tv, time_window=week, TvCredits`	6	33.33	100.0	0.0	-59.68

Table 8: Generated Sequences and Metrics for the RestGPT Dataset with `meta/llama-3-8b-instruct`

- **Task Description:**
  - "Classify the following news articles into one of these categories: World, Sports, Business, Sci/Tech."
- **Examples:**
  - **Title:** "Fears for T N pension after talks"
  - **Article:** "Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul."
  - **Category:** `Business`
  - **Title:** "The Race is On: Second Private Team Sets Launch Date for Human Spaceflight (SPACE.com)"
  - **Article:** "SPACE.com - TORONTO, Canada – A second team of rocketeers competing for the $10 million Ansari X Prize, a contest for privately funded suborbital space flight, has officially announced the first launch date for its manned rocket."
  - **Category:** `Sci/Tech`
  - **Title:** "Giddy Phelps Touches Gold for First Time"
  - **Article:** "Michael Phelps won the gold medal in the 400 individual medley and set a world record in a time of 4 minutes 8.26 seconds."
  - **Category:** `Sports`
- **Test Utterance**
  - **Title:** "Prediction Unit Helps Forecast Wildfires (AP)"
  - **Article:** "AP - It's barely dawn when Mike Fitzpatrick starts his shift with a blur of colorful maps, figures and endless charts, but already he knows what the day will bring. Lightning will strike in places he expects. Winds will pick up, moist places will dry and flames will roar."
  - **Category:**

Table 9: AG News Prompt Structure

Utterance	Generated Sequence	Order	Accuracy	Log Prob OptiSeq
Article: A quot;formidable information and technology management challenge quot; faces the Homeland Security Department, according a report released today by the Government Accountability Office.	World	1	0.0	-1.57
	World	2	0.0	-0.64
	World	3	0.0	-0.77
	World	4	0.0	-1.15
	Sci/Tech	5	**100.0**	**-0.55**
	Sci/Tech	6	**100.0**	**-0.81**
Article: A former executive who was a participant in the wrongdoing that helped cripple Enron testified on Monday, providing the first glimpse through the eyes of a principal of	Business	1	**100.0**	**-0.08**
	World	2	0.0	-1.34
	Business	3	**100.0**	**-0.16**
	World	4	0.0	-2.29
	World	5	0.0	-0.87
	World	6	0.0	-1.49
Article: AOL has kicked off an initiative designed to make it easier for developers to engineer, test and distribute licensed AOL Instant Messenger (AIM) clients for mobile devices.	Sci/Tech	1	**100.0**	**-0.29**
	Sci/Tech	2	**100.0**	**-0.28**
	Sci/Tech	3	**100.0**	**-0.28**
	Business	4	0.0	-0.59
	Sci/Tech	5	**100.0**	**-0.30**
	Business	6	0.0	-0.56
Article: AP - Raymond Goethals, the Belgian soccer coach who led Olympique Marseille to the 1993 European Champions Cup title, died Monday, according to news reports. He was 83.	Sports	1	**100.0**	**-0.17**
	Sports	2	**100.0**	**-0.14**
	World	3	0.0	-1.07
	Sports	4	**100.0**	**-0.24**
	Sports	5	**100.0**	**-0.04**
	Sports	6	**100.0**	**-0.05**
Article: Crude oil prices settled at $49.64 a barrel, up 76 cents as traders expressed concern that recent hurricanes had hurt output in the United States.	Business	1	**100.0**	**-0.14**
	World	2	0.0	-1.04
	World	3	0.0	-1.92
	World	4	0.0	-0.85
	World	5	0.0	-0.89
	World	6	0.0	-1.55
Article: Moises Alou has a right to his opinion, Chicago Cubs manager Dusty Baker said Monday. Alou said everything he needed to say Sunday.	Sports	1	**100.0**	**-0.04**
	World	2	0.0	-2.14
	Sports	3	**100.0**	**-0.08**
	Sports	4	**100.0**	**-0.11**
	Sports	5	**100.0**	**-0.02**
	Sports	6	**100.0**	**-0.02**
Article: NEW YORK – Dale Earnhardt Jr. has trouble remembering those frantic seconds when he escaped from his burning racecar. He believes, however, that his late father figured in his survival.	Sports	1	**100.0**	**-0.10**
	World	2	0.0	-1.09
	Sports	3	**100.0**	**-0.25**
	World	4	0.0	-1.10
	Sports	5	**100.0**	**-0.02**
	Sports	6	**100.0**	**-0.07**
Article: One way or another, Paul Hamm 39;s gold-medal odyssey is about to end. Whether he gets to keep the medal and the title he won a month ago in the Olympic men 39;s gymnastics all-around will be up to the sporting world 39;s highest authority.	Sports	1	**100.0**	**-0.05**
	Sports	2	**100.0**	**-0.12**
	Sports	3	**100.0**	**-0.16**
	World	4	0.0	-1.29
	Sports	5	**100.0**	**-0.03**
	Sports	6	**100.0**	**-0.04**
Article: The role of agents in multimillion-pound football transfer deals came under fresh scrutiny yesterday after Manchester United revealed payments of 11m to middle-men for their help in signing players.	Sports	1	**100.0**	**-0.28**
	World	2	0.0	-1.03
	World	3	0.0	-1.32
	Sports	4	**100.0**	**-0.12**
	Sports	5	**100.0**	**-0.19**
	Sports	6	**100.0**	**-0.16**

Table 10: Generated Sequences and Metrics for the AG News Dataset with `meta/llama-3-8b-instruct`

- **Task Description:**
  - "Classify the sentiment of the following sentences as very negative, negative, neutral, positive, or very positive."
- **Examples:**
  - **Sentence:** "a 93-minute condensation of a 26-episode tv series, with all of the pitfalls of such you'd expect."
  - **Sentiment:** `negative`
  - **Sentence:** "this is a startling film that gives you a fascinating, albeit depressing view of iranian rural life close to the iraqi border."
  - **Sentiment:** `positive`
  - **Sentence:** "but you'll definitely want the t-shirt."
  - **Sentiment:** `neutral`
- **Test Utterance:**
  - **Sentence:** "he just wants them to be part of the action, the wallpaper of his chosen reality."
  - **Sentiment:**

Table 11: SST-5 Prompt Structure

Utterance	Generated Sequence	Order	Accuracy	Log Prob OptiSeq
Sentence: a sudsy cautionary tale .	neutral	1	**100.0**	**-5.69**
	neutral	2	**100.0**	**-5.69**
	neutral	3	**100.0**	**-5.69**
	neutral	4	**100.0**	**-5.69**
	neutral	5	**100.0**	**-5.69**
	negative	6	0.0	-66.24
Sentence: alex nohe 's documentary plays like a travelogue for what mostly resembles a real-life , big-budget nc-17 version of tank girl .	positive	1	0.0	-8.92
	neutral	2	**100.0**	**-6.79**
	negative	3	0.0	-173.75
	positive	4	0.0	-8.92
	positive	5	0.0	-8.92
	negative	6	0.0	-173.75
Sentence: here , thankfully , they are .	positive	1	0.0	-7.76
	neutral	2	**100.0**	**-5.76**
	positive	3	0.0	-7.76
	positive	4	0.0	-7.76
	positive	5	0.0	-7.76
	positive	6	0.0	-7.76
Sentence: hip-hop has a history , and it 's a metaphor for this love story .	positive	1	0.0	-5.67
	neutral	2	**100.0**	**-5.55**
	neutral	3	**100.0**	**-5.55**
	positive	4	0.0	-5.67
	positive	5	0.0	-5.67
	positive	6	0.0	-5.67
Sentence: lucas , take notes .	very negative	1	0.0	-13.19
	neutral	2	**100.0**	**-5.57**
	very negative	3	0.0	-13.19
	positive	4	0.0	-8.38
	very positive	5	0.0	-15.00
	very negative	6	0.0	-13.19
Sentence: taken purely as an exercise in style , this oppressively gloomy techno-horror clambake is impossible to ignore .	positive	1	0.0	-9.48
	neutral	2	**100.0**	**-8.01**
	neutral	3	**100.0**	**-8.01**
	positive	4	0.0	-9.48
	positive	5	0.0	-9.48
	positive	6	0.0	-9.48
Sentence: the cartoon that is n't really good enough to be on afternoon tv is now a movie that is n't really good enough to be in theaters .	very negative	1	**100.0**	**-13.25**
	negative	2	0.0	-113.58
	negative	3	0.0	-113.58
	negative	4	0.0	-113.58
	negative	5	0.0	-113.58
	negative	6	0.0	-113.58
Sentence: the movie 's ripe , enrapturing beauty will tempt those willing to probe its inscrutable mysteries .	positive	1	**100.0**	**-5.41**
	positive	2	**100.0**	**-5.41**
	very positive	3	0.0	-12.19
	positive	4	**100.0**	**-5.41**
	positive	5	**100.0**	**-5.41**
	positive	6	**100.0**	**-5.41**
Sentence: two hours of melodramatic musical married to two hours of underdog sports intrigue , if the picture also shares the weaknesses of both genres , more 's the pity .	neutral	1	**100.0**	**-6.23**
	neutral	2	**100.0**	**-6.23**
	negative	3	0.0	-181.70
	neutral	4	**100.0**	**-6.23**
	neutral	5	**100.0**	**-6.23**
	negative	6	0.0	-181.70

Table 12: Generated Sequences and Metrics for the SST-5 Dataset with `meta/llama-3-8b-instruct`

- **Task Description:**
  - "Classify the type of the following questions into Abbreviation, Entity, Description, Human, Location, or Number."
- **Examples:**
  - **Question:** "How far is it from Denver to Aspen?"
  - **Type:** `Number`
  - **Question:** "What county is Modesto, California in?"
  - **Type:** `Location`
  - **Question:** "Who was Galileo?"
  - **Type:** `Human`
- **Text Utterance:**
  - **Question:** "What is the capital of Yugoslavia?"
  - **Type:**

Table 13: TREC Prompt Structure

Utterance	Generated Sequence	Order	Accuracy	Log Prob OptiSeq
Question: How far is it from Denver to Aspen ?	Number	1	**100.0**	**-5.34**
	Number	2	**100.0**	**-5.34**
	Number	3	**100.0**	**-5.34**
	Abbreviation	4	0.0	-12.68
	Number	5	**100.0**	**-5.34**
	Number	6	**100.0**	**-5.34**
Question: What city had a world fair in 1900 ?	Entity	1	0.0	-9.72
	Location	2	**100.0**	**-7.85**
	Entity	3	0.0	-9.72
	Entity	4	0.0	-9.72
	Entity	5	0.0	-9.72
	Location	6	**100.0**	**-7.85**
Question: What hemisphere is the Philippines in ?	Entity	1	0.0	-11.31
	Entity	2	0.0	-11.31
	Entity	3	0.0	-11.31
	Location	4	**100.0**	**-7.69**
	Entity	5	0.0	-11.31
	Location	6	**100.0**	**-7.69**
Question: What is the average weight of a Yellow Labrador ?	Number	1	**100.0**	**-5.77**
	Number	2	**100.0**	**-5.77**
	Entity	3	0.0	-9.98
	Number	4	**100.0**	**-5.79**
	Number	5	**100.0**	**-5.77**
	Number	6	**100.0**	**-5.77**
Question: What is the temperature at the center of the earth ?	Description	1	0.0	-9.14
	Number	2	**100.0**	**-7.79**
	Description	3	0.0	-9.14
	Description	4	0.0	-9.14
	Description	5	0.0	-9.14
	Description	6	0.0	-9.14
Question: What person 's head is on a dime ?	Human	1	**100.0**	**-7.28**
	Entity	2	0.0	-12.39
	Human	3	**100.0**	**-7.28**
	Human	4	**100.0**	**-7.28**
	Human	5	**100.0**	**-7.28**
	Human	6	**100.0**	**-7.28**
Question: When did Hawaii become a state ?	Human	1	0.0	-9.52
	Number	2	**100.0**	**-7.74**
	Description	3	0.0	-9.07
	Number	4	**100.0**	**-7.74**
	Number	5	**100.0**	**-7.74**
	Number	6	**100.0**	**-7.74**
Question: Who developed the vaccination against polio ?	Human	1	**100.0**	**-10.27**
	Human	2	**100.0**	**-10.27**
	Entity	3	0.0	-56.30
	Human	4	**100.0**	**-10.27**
	Human	5	**100.0**	**-10.27**
	Human	6	**100.0**	**-10.27**
Question: Who was Galileo ?	Human	1	**100.0**	**-6.82**
	Human	2	**100.0**	**-6.82**
	Human	3	**100.0**	**-6.82**
	Entity	4	0.0	-7.44
	Entity	5	0.0	-7.44
	Human	6	**100.0**	**-6.82**

Table 14: Generated Sequences and Metrics for the TREC Dataset with `meta/llama-3-8b-instruct`