Optimizing Few-Shot Learning: From Static to Adaptive in Qwen2-7B

Wenhao Liu
Beijing University of Posts and Telecommunications
Haidian Qu, Beijing Shi, China
wenhaoliu@bupt.edu.cn

Tianxing Bu
Peking University
Haidian Qu, Beijing Shi, China
butianxing@stu.pku.edu.cn

Erchen Yu
Dalian University of Technology
Dalian Shi, China

Dailin Li
Dalian University of Technology
Dalian Shi, China

Ding Ai
Beijing University of Posts and Telecommunications
Haidian Qu, Beijing Shi, China

Zhenyi Lu
Huazhong University of Science and Technology
Wuhan Shi, China
luzhenyi529@gmail.com

Haoran Luo
Beijing University of Posts and Telecommunications
Haidian Qu, Beijing Shi, China
luohaoran@bupt.edu.cn

Abstract

Under resource-constrained conditions, our research team focused on two primary algorithm: Qwen2-7B-Few-Shots and Qwen2-7B-Adaptive-Few-Shots. These initiatives centered on official development set benchmarks and our proprietary benchmarks in (track1 and track2), evaluated through the KDD leaderboard.

The Qwen2-7B-Few-Shots algorithm leveraged in-context learning methods, specifically analogical few-shot. It progressed from an initial phase to employing multi-agent strategies (Tot and AutoReact), ultimately achieving end-to-end Chain of Thought (COT) few-shot learning. Experimental results demonstrated the efficacy of static few-shots.

The Qwen2-7B-Adaptive-Few-Shots algorithm focused on adaptive learning. Although time constraints prevented its inclusion on the leaderboard, future directions include batch inference and data collection via crawler construction. The project evolved from an initial knowledge graph RAG to end-to-end few-shot learning, and finally to end-to-end COT few-shot learning.

Future work encompasses: 1) Fine-tuning to enhance performance; 2) Improving the COT format with SymbCot; 3) Adding multimodal information

CCS Concepts

• **Information systems**;

Keywords

Recommending Systems, In-context Learning, KDD Cup

ACM Reference Format:
Wenhao Liu, Tianxing Bu, Erchen Yu, Dailin Li, Ding Ai, Zhenyi Lu, and Haoran Luo. 2024. Optimizing Few-Shot Learning: From Static to Adaptive in Qwen2-7B. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Amazon KDD Cup 2024 Workshop).* ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

Amazon KDD Cup 2024 Workshop, August 25–29, 2024, Barcelona, Spain

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Online shopping has become an indispensable service in the lives of modern citizens. To provide better shopping experiences to users, machine learning has been extensively used to understand various entities in online shopping, such as queries, browsing sessions, etc. to infer the user's search and shopping intentions. However, few studies explore online shopping tasks under the multi-task, few-shot learning scenario. In practice, online shopping creates a massive multi-task learning problem that often involves a joint understanding of various shopping entities, such as products, attributes, queries, purchases, etc. Moreover, new shopping entities and tasks constantly emerge over time as a result of business expansion or new product lines, creating few-shot learning problems on these emerging tasks.

Large language models (LLM) emerge as promising solutions to the multi-task, few-shot learning problem in online shopping. Many studies have underscored the ability of a single LLM to perform various text-related tasks with state-of-the-art abilities and to generalize to unseen tasks with only a few samples or task descriptions. Therefore, by training a single LLM for all shopping-related machine learning tasks, we mitigate the costs for task-specific engineering efforts, and for data labeling and re-training upon new tasks. Furthermore, LLMs can improve the customers' shopping

experiences by providing interactive and real-time shopping recommendations.

Track2 in the Amazon KDD Cup 2024 competition aims to evaluate the model's ability to understand the complex implicit knowledge in the domain of online shopping and to apply the knowledge to perform various types of reasoning. Various implicit knowledge exists in different categories of products and plays a crucial role in the browsing and shopping behaviours of customers.

From the computational standpoint,we adopt low-resource approaches like few-shot learning and in-context learning leveraging the ability of the LLMs to solve this track and also design an adaptive RAG method to improve the abilty of in-context learning and finally get the 8th place on the leaderboard.

2 Our Approch

2.1 Data Collection

Given that the official dataset provides only a limited development set (devset), we observed that high scores achieved on this devset do not necessarily translate to high performance on the private testset. To address this issue, we aimed to construct a larger devset designed to be distributed similarly to the official private dataset. To this end, we developed a distributed web crawler system specifically targeted at collecting the most recent product information from the Amazon website.Here is the picture of the system: Fig1.Due to limited resource,we only gather 500000 products on Amazon.

2.2 Dataset

In this track, we were given a dev set with 16 items belonging to 3 tasks. Given that only a small portion of the tasks (3 out of 8) were provided in the dev set, reasonable assumptions had to be made for the remaining tasks. We brainstorm with the Large Language Model (LLM), providing it with few-shot examples to help hypothesize what these tasks might entail. We generated approximately 20 potential tasks and filtered them through human expert judgment. Ultimately, 10 tasks were retained, we use GPT-4 to synthesize a candidate set which consists of these tasks and with the format of the dev set, of approximately 30000 items.

2.3 Prompt Engineering

Given that our team consists solely of students without the computational support of a company, we face significant constraints in terms of computational power and resources. As such, executing large-scale pretraining tasks on Large Language Models (LLMs) is not only impractical but also uneconomical. Hence, our strategy is built upon a foundation of LLMs, which are known for their robust knowledge reasoning capabilities. This strategy incorporates a meticulously designed Prompt Engineering and in-context learning samples to construct a reasoning scheme tailored for this task.

Multi-stage processing flow. At the onset of the competition, we formulated a reasoning process based on Prompt Engineering, which involved multi-stage prompt design: intent recognition, necessary knowledge introduction, chain-of-thought (Cot) reasoning, and structured output. This formed the preliminary, yet incomplete, version of our final solution.

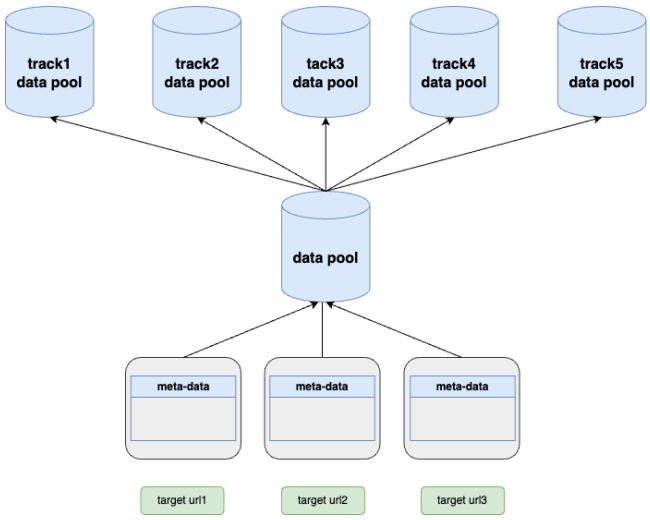

Figure 1: The overall architecture of our crawler system. Initially, all retrieved data is consolidated into a centralized data pool where it undergoes merging and preliminary cleansing. This consolidation phase ensures the consistency and integrity of the data. Subsequently, the processed data is intelligently routed to the appropriate processing tracks based on predefined rules or machine learning models, ensuring that each piece of data is directed to the most suitable processing pipeline for further analysis and storage. This approach not only enhances the efficiency and speed of data handling but also maintains the flexibility and scalability of the system.

Exploring single and multi-agent frameworks. Subsequently, we explored the application of Agent or Multi-Agent frameworks to this track. Specifically, we replicated the classic single Agent framework, React, and the multi-agent framework, AutoAct, and applied both to our established benchmark for reasoning experiments. Utilizing Agent and Multi-Agent frameworks is a brilliant method for knowledge reasoning tasks in this competition, as Agents can autonomously determine reasoning paths through task planning, automation of task decomposition, and tool invocation. In a Multi-agent framework, multiple agents can collaboratively control the reasoning process from various perspectives and capabilities, leading step-by-step to the final answer, exhibiting greater accuracy than few-shot Cot. However, the downside of Agent and Multi-agent lies in their time-consuming nature. Due to hardware and time restrictions imposed by the competition organizers, we were compelled to abandon this strategy.

Two-stage strategy of intent recognition and tailored prompt crafting. Ultimately, we refocused our strategy on Prompt Engineering and In-context Learning. First, we distinguish between question types. Second, as the Retrieval question data was not disclosed in the Dev Set, we deduced that Retrieval questions predominantly involve common sense reasoning and implicit multi-hop reasoning tasks, rather than mathematical reasoning. For both question types, we adopted a two-stage strategy for prompt crafting:

In the first stage, we enabled the LLM to perform intent recognition on the input question, determining whether the question required mathematical reasoning or non-mathematical common sense or implicit multi-hop reasoning. By directing different reasoning tasks into distinct branches during the intent recognition stage, we could tailor the generation of Cot thinking processes and draft more targeted in-context learning samples for each question category, thus enhancing the reasoning process.

The prompt template of intent recognition is as follows.

> There is a shopping knowledge reasoning task. Your job is to determine the type of reasoning task associated with the user's question.
> Below are two types of reasoning tasks, and you need to identify which task the question belongs to.
> <Numerical Reasoning>: Involve numerical information and include numerical reasoning.
> <CommonSense Reasoning>: Revolves around common sense knowledge of the product.
> rules:
> If the question belongs to <Numerical Reasoning>, generate: A
> If the question belongs to <CommonSense Reasoning>, generate: B
> Here gives two example:
> Example 1:
> Question: Which of the following product categories best complement the product type tabletop game?
> Answer:B
> Example 2:
> Question: The product 'Anker USB C Hub, 655 USB-C Hub (8-in-1), with 2 USB-A 10 Gbps Data Ports, 100W Power Delivery, 4K HDMI, 1 Gbps Ethernet, microSD and SD Card Slots, 3.5 mm AUX, for MacBook, and More (Earthy White)' appears on an e-commerce website. It is a multi-port hub. How many usb ports are there?
> Answer:A
> Caution: Please just respond the letter A or B, without any other words and symbol.
> Begin!
> Question: question
> Answer:

After intent recognition, we needed LLM to perform reasoning and produce a formatted output for the final answer. At this step, we crafted prompts, including the task goal, input-output templates, and well-designed static few-shots. The input was the Question, and the output consisted of Thinking steps and Answer. Empirical evidence suggests that providing the LLM with input-output templates enhances its ability to follow instructions. For the multiple-choice questions, we selected three examples each of mathematical and non-mathematical reasoning; for retrieval questions, we chose three examples of retrieval reasoning. For each example, we crafted clear and concise Cot reasoning and model output for answers, serving as a reference for the LLM in generating the reasoning process. We controlled the format of the model output by separating the [Ans] from the Cot reasoning process. During model reasoning, we used regular expressions to capture the answer.

The prompt template of few-shots COT reasoning in numeric reasoning is as follows, which is similar with commonsense and implicit reasoning.

> You are a helpful online shopping expert in numerical reasoning. Your role is to read the given shopping numerical reasoning multiple-choice question, and give reasonable thinking steps to answer the question.
> The goal of multiple-choice question is to choose the most suitable option in the candidate options.
> Please use the following format:
> Question: the input question and options
> Thinking steps: think step by step how to solve the question. Finally, find the indice of the answer.
> To help you understand, here are two examples for numeric reasoning.
> Example 1:
> Question: The product 'Simply Asia Garlic Basil Singapore Street Noodles, 9.24 oz (Pack of 6)' appears on e-commerce website. What is the total weight of the noodles? 0. 8 ounce 1. 55.44 ounce 2. 14.19 ounce 3. 60 ounce Answer:
> Thinking steps: The product is a pack of 6 noodle packs, each weighing 9.24 oz. There are 6 packs, so the total weight would be 6 multiplied by 9.24 oz, equaling 55.44 ounce. [Ans]: The indice of answer is 0. 8 ounce.
> Example 2:
> Question: The product 'Anker USB C Hub, 655 USB-C Hub (8-in-1), with 2 USB-A 10 Gbps Data Ports, 100W Power Delivery, 4K HDMI, 1 Gbps Ethernet, microSD and SD Card Slots, 3.5 mm AUX, for MacBook, and More (Earthy White)' appears on an e-commerce website. It is a multi-port hub. How many usb ports are there? 0. It cannot be inferred. 1. 1 2. 2 3. 3 Answer:
> Thinking steps: The USB C hub comes with 2 USB-A 10 Gbps data ports and a 100W Power Delivery. According to standard convention, Power Delivery is a kind of USB port, so we consider this as a USB port as well. Therefore, there are a total of 3 USB ports. [Ans]: The indice of answer is 3. 3.
> Caution: Thinking steps should be limited in 4 simple sentenses.
> When reasoning, please think about the attribute and feature of the product.
> Begin!
> Question: question
> Thinking steps:

Our method achieved great results in KDD CUP track2, with a total score of 0.75 on the offline test set and a score of 0.7221 on the second-stage online test set.

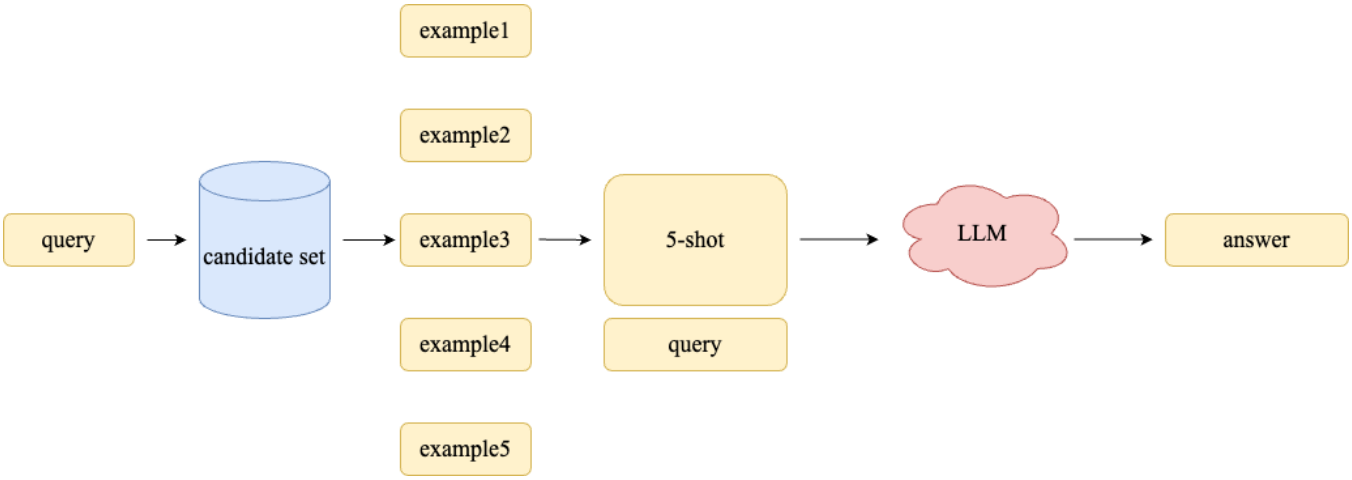

Figure 2: The overall architecture of our system. For each query, the system first retrieves 5 examples from the candidate set based on semantic similarity, then the system reranks these examples and concates them with the query for LLM to answer the query.

Method	$score_{track1}$	$score_{track2}$
zero-shot	0.67	0.67
few-shot	0.73	0.72
fine-tune	-	0.728
Adaptive RAG	0.77	-

Table 1: Comparison of the performance between different methods

2.4 Adaptive RAG

In this track, we design a low-resource approach, where the model learns from few-shot examples retrieved from the candidate set. We use vector retrieval to retrieve the five samples with the highest semantic similarity. For text embedding, we employ bge-large-en, which maps each piece of text to a 1024-dimensional vector.

In the field of vector retrieval, we adopted open-source vector databases to compress the retrieval time and meet the competition requirements. We tested the Chroma, Milvus, Faiss and Weaviate vector databases. We also considered using the GraphRAG technology, but due to time constraints, we ultimately adopted the Weaviate vector database compressing the time for retrieving data from 1.8 seconds to 0.24 seconds. We do not submit this edition because we have overcome the time limit after the competition.

In the field of text re-ranking, we tested llama-index, RerankGPT and some reranking models like bge-reranker-base. However, due to the time constraints of the competition, we ultimately abandoned the re-ranking part. The score on the dev set is shown in table1

3 Future Work

In this section, we outline the directions for future research that build upon the current findings and address some of the limitations encountered during our study.

Fine-Tuning for Enhanced Performance. One promising direction involves further fine-tuning our model to improve its performance, particularly in scenarios with less satisfactory results. Specifically, we aim to focus on enhancing the model's accuracy based on 8th place, which was fine-tuned on a thousand large instruction sets. This will involve collecting additional data to fine-tune our text embedding model.

Symbolic Chain-of-Thought. Another area of interest is the development and implementation of an improved symbolic chain-of-thought (symbCot) format. The symbCot approach has shown promise in providing a structured and interpretable representation of the model's reasoning process. However, there is room for improvement in terms of clarity, efficiency, and ease of use. We plan to refine the symbCot format by incorporating feedback from users and experts, as well as experimenting with logical representations that could enhance the clarity and utility of the output. This will not only benefit researchers but also practitioners who rely on the interpretability of models for decision-making processes.By addressing these two areas, we anticipate that our contributions will not only improve the technical capabilities of our model but also advance the broader field of machine learning, particularly in the areas of interpretability and practical applicability.

Multimodal Models Enhancement for Complex Reasoning in E-commerc. Our future work aims to improve complex reasoning in AI models, focusing on implicit reasoning in e-commerce contexts. We plan to integrate multimodal data from platforms like Amazon into advanced models such as GPT-4V. By combining product images, titles, and query-related information, we'll develop reasoning processes based on Chain of Thought and Tree of Thoughts methodologies. This approach is expected to enhance the model's performance in complex problem-solving through structured reasoning. We will explore the effectiveness and limitations of this method across various applications, advancing multimodal AI in sophisticated reasoning tasks.