



# Know Your Needs Better: Towards Structured Understanding of Marketer Demands with Analogical Reasoning Augmented LLMs

Junjie Wang\*  
Zhejiang University  
Ant Group  
Hangzhou, China  
wangjj2018@zju.edu.cn

Dan Yang\*  
Ant Group  
Hangzhou, China  
luoyin.yd@antgroup.com

Binbin Hu\*  
Ant Group  
Hangzhou, China  
bin.hbb@antfin.com

Yue Shen  
Ant Group  
Hangzhou, China  
zhanying@antfin.com

Wen Zhang†  
Zhejiang University  
Hangzhou, China  
zhang.wen@zju.edu.cn

Jinjie Gu†  
Ant Group  
Hangzhou, China  
jinjie.gujj@antfin.com

## Abstract

In this paper, we explore a new way for user targeting, where non-expert marketers could select their target users solely given demands in natural language form. The key to this issue is how to transform natural languages into practical structured logical languages, i.e., the structured understanding of marketer demands. In practical scenarios, the demands of non-expert marketers are often abstract and diverse. Considering the impressive natural language processing ability of large language models (LLMs), we try to leverage LLMs to solve this issue. To stimulate the LLMs' reasoning ability, the chain-of-thought (CoT) prompting method is widely used, but existing methods still have some limitations in our scenario: (1) Previous methods either use simple "Let's think step by step" spells or provide fixed examples in demonstrations without considering compatibility between prompts and concrete questions, making LLMs ineffective when the marketers' demands are abstract and diverse. (2) Previous methods are often implemented in closed-source models or excessively large models, which is not suitable in industrial practical scenarios. Based on these, we propose ARALLM (i.e., Analogical Reasoning Augmented Large Language Models) consisting of two modules: Analogical Reasoning based Prompting and Reasoning-Augmented Multi-Task Model Distillation. To be specific, we first construct a reasoning library consisting of several high-quality and topic-rich reasoning examples. Then, we adopt a retrieval-based method to conduct analogical reasoning with the help of the reasoning library. The experimental results show that this prompting strategy achieves better performance than the ordinary prompting method. Beyond that, we distill knowledge from super LLMs (GPT-3.5) to fine-tune smaller

student LLMs in a multi-task training paradigm, enabling the models to be easily deployed in practical environments. Part of our data and code can be found at <https://github.com/alipay/Analogic-Reasoning-Augmented-Large-Language-Model>.

## CCS Concepts

• Computing methodologies → Natural language processing.

## Keywords

User Targeting, Large Language Models, Analogical Reasoning, Knowledge Distillation, Chain of Thought

## ACM Reference Format:

Junjie Wang, Dan Yang, Binbin Hu, Yue Shen, Wen Zhang, and Jinjie Gu. 2024. Know Your Needs Better: Towards Structured Understanding of Marketer Demands with Analogical Reasoning Augmented LLMs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*, August 25–29, 2024, Barcelona, Spain. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3637528.3671583>

## 1 Introduction

Recently, the practice of user targeting has gained significant attention in real-world applications (e.g., Alipay and WeChat), as highlighted by a variety of studies [19, 23, 28, 32, 33]. This approach has the excellent potential to attract high-quality users for specific campaigns, aligning with the goals of marketers to enhance conversions and minimize operational expenses. Roughly speaking, current methods devoted to user targeting mainly fall into two lines: *model-based methods* (Figure 1 (a)) that perform expansion of seed users/entities with well-designed neural networks [28, 32, 33] and *rule-based methods* (Figure 1 (b)) that manually group users with different tags based on domain knowledge [19, 23].

Unfortunately, both of the above prior studies have primarily emphasized the intricate architectures for effectively and efficiently gathering targeted users, while commonly ignoring the natural and significant gap between marketers' demand and the capability of current models. In particular, current user targeting pipelines mainly force the marketers to decompose their demands into multiple tags/entities, e.g., "Preference", "Resident City" and "User Age Group" which are characterized in Figure 1. The process incurs several weaknesses: i) unfriendly to marketers; ii) time-consuming

\*Equal contributions

†Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

KDD '24, August 25–29, 2024, Barcelona, Spain

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0490-1/24/08

<https://doi.org/10.1145/3637528.3671583>

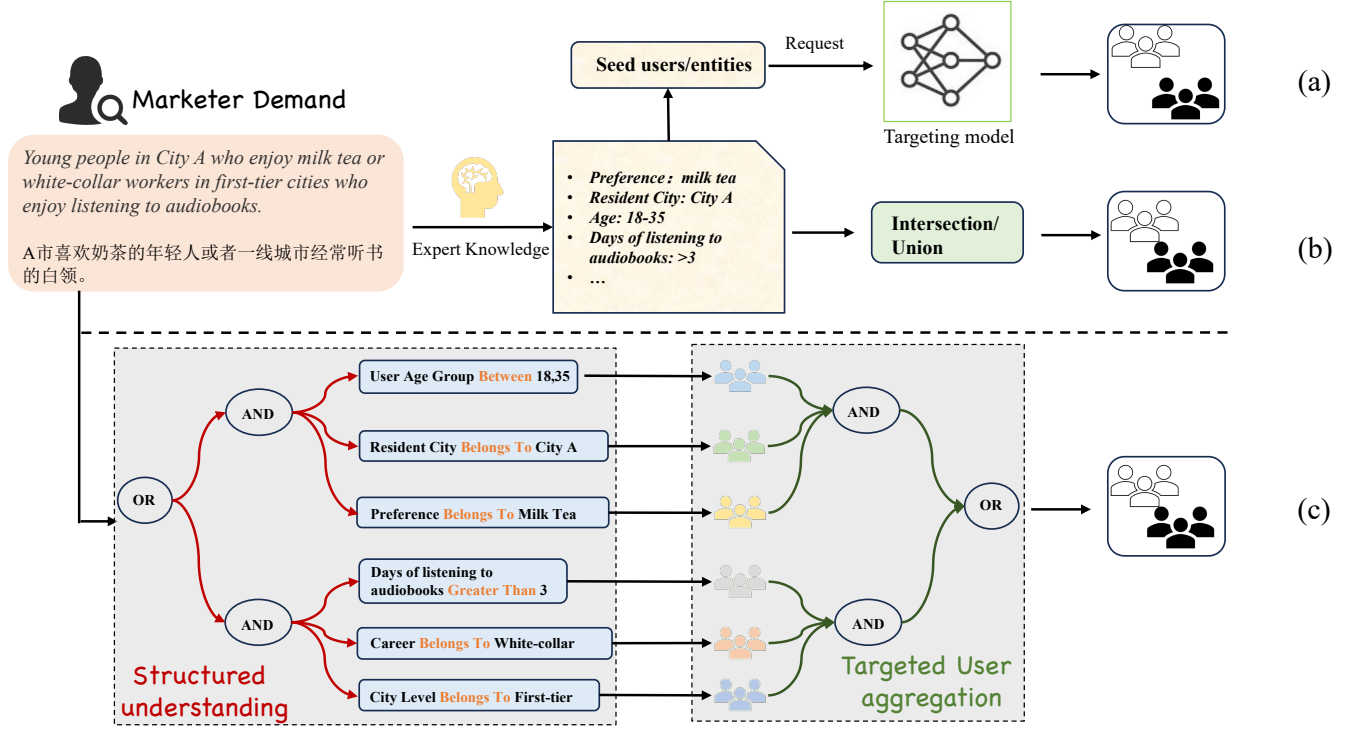


Figure 1: The comparison of user targeting approach.

for manual deconstruction of demands; and iii) unpromising performance due to partial tags/entities.

Therefore, in this paper, we take the initial stride towards enhancing current user targeting systems by probing into the structured understanding of marketers' demands in an automatic way, *enabling the system to know your needs better solely based on natural language inputs*. As shown in Figure 1 (c), the marketers only need to input their native demands like "Young people in City A who enjoy milk tea or white-collar workers in first-tier cities who enjoy listening to audiobooks" and our new approach can instantly conduct structured decomposition of marketers' demands and thus provide the user targeting cards. After verifying the cards, they can export the target users with just one click and the system will automatically run the existing targeted user aggregation modules.

**The core of this new approach lies in how to transform natural languages into structured forms**, or so-called *structured understanding of marketer demands*. To address this issue, we first draw upon existing logical and programming languages to design a logical expression that offers both readability and practical applicability for the structured representation of marketers' demands. We name it as SELL, *i.e.*, Structured and Editable Logical Language, which mainly consists of Keys (*e.g.*, Resident City), Values (*e.g.*, City A), Operators (*e.g.*, Belongs To) and Intersection/Union symbols (*i.e.*, AND, OR). It forms the basis of the user targeting cards shown in Figure 1(c). The main concern of our work is transforming the Natural Language into SELL (NL2SELL). For example, the natural language "Young people in City A who enjoy milk tea or white-collar workers in first-tier cities who enjoy listening to audiobook"

can be translated into SELL as "((Resident City#Belongs To# City A) AND (User Age Group#Between#18,35) AND (Preference#Belongs To#Milk Tea)) OR ((City Level#Belongs To#First-tier) AND (Days of listening to audiobooks#Greater Than#3) AND (Career#Belongs To#White-collar))".

Previous research [7, 15, 17, 29] has highlighted the remarkable abilities of Large Language Models (LLMs) in language translation tasks. LLMs are renowned for their robust natural language processing and zero-shot capabilities, which render them possibly effective in NL2SELL. Consequently, we opt to employ LLMs to tackle this task. However, there are still some challenges in using LLMs in our scenario: **(1) The challenge of reasoning accuracy**. In practical user targeting scenarios, understanding of marketer's demands could be very challenging. For example, a marketer might want to market products related to education, targeting parents of middle school students who are focused on education. Therefore, the demand she inputs into the system is a simple sentence "Parents of middle school students", and we need to use existing tags to convert this demand into a structured expression, such as "(Marital Status#Belongs#True) AND (User Child Age#Between#12,15) AND (Preference#Belongs To#Education)". However, the key "Marital Status, User Child Age, and Preferences" do not directly appear in the marketer's demand, which requires the LLMs to possess a high level of language comprehension and reasoning abilities. Recently, it [13, 26] has been pointed out that prompting LLMs with chain-of-thought (CoT) could enhance the reasoning ability of LLMs. These methods either provide fixed reasoning examples as demonstrations through few-shot learning or tell the model "Let's think step

by step” for zero-shot learning. Though effective, they ignore the compatibility between prompts and specific questions. Considering the diversity of marketers’ demands themselves, zero-shot learning or having fixed reasoning examples in few-shot learning may not effectively stimulate the LLMs’ reasoning ability [22]. Ideally, we can customize a reasoning process for each question to induce the model to reason better, but it is time-consuming and laborious. (2) **The challenge of reasoning speed and resource consumption.** Although the reasoning ability of super-large language models can be developed through prompt engineering, it is unacceptable to deploy excessively huge LLMs or API-based LLMs in practical situations. For extremely huge LLMs, excessive parameters will consume a lot of resources, and the reasoning speed is also slower. For API-based LLMs such as GPT-3.5, both the security of private data and the cost of API calls are thorny issues. Our model needs to be deployed in real ToC (To Customer) scenarios, where it will generate at least hundreds of tasks daily. Therefore, there is an urgent requirement to increase the inference speed, reduce the cost of inference, and ensure data security. Thus, smaller, free, and white-box LLMs are preferred.

To address these challenges, we propose a framework called Analogic Reasoning Augmented Large Language Model (ARALLM) to tackle the NL2SELL task, which consists of two modules: Analogical Reasoning based Prompting and Reasoning-Augmented Multi-Task Model Distillation. (1) **Analogical Reasoning based Prompting.** Creating a CoT for each new targeting demand is impractical and time-consuming (considering the need for deployment). Thus, we propose Analogical Reasoning based Prompting. The fundamental concept of it is that when two targeting demands are similar, they might follow similar reasoning steps and have similar conditional expression structures. Therefore, we construct a compact yet reliable reasoning library to serve as a reference for numerous demands that have unknown reasoning steps. This approach helps reduce the effort of manually writing reasoning steps for every demand. Additionally, we can offer assistance for any unfamiliar demands by employing analogical reasoning with retrieval. (2) **Reasoning-Augmented Multi-Task Model Distillation.** Since the white-box and smaller model is preferred, we distill knowledge from teacher LLMs (GPT-3.5) to construct an NL2SELL dataset with over 10,000 pieces of training samples and then propose a novel multi-task training method with reasoning augmentation to improve the reasoning ability of student LLMs.

We summarize our contributions as follows: (1) We explore a new way for marketer-friendly user targeting, which focuses on the structured understanding of marketers’ raw demands. (2) We propose an innovative framework called ARALLM, which is composed of the Analogical Reasoning based Prompting module and the Reasoning-Augmented Multi-Task Distillation module, to address the above task. (3) We demonstrate the superiority of the proposed ARALLM framework through extensive experiments on real-world datasets and make this LLMs-based framework operate in real industry scenarios.

## 2 Method

In this work, we **focus on how to convert a sentence of marketer’s demand  $d$  into a standard structured expression SELL**,

while how to parse and execute the SELL and return the target users from the database will be handled by the internal system, which does not constitute the core of this article. Inspired by the analogical reasoning [6, 24, 30] in the classic artificial intelligence field and CoT prompting methods in LLMs, we first propose a novel prompting method enhanced by analogical reasoning and then adopt a reasoning-augmented multi-task training strategy to fine-tune smaller LLMs.

### 2.1 Design of SELL

We first explain the design of SELL to help readers understand our task better. The SELL is mainly composed of four elements: **Keys**, **Values**, **Operators**, and **Intersection and Union symbols**:

(1) **Keys** are a series of tags that describe the features of the user, such as *Gender*, *Monthly Income*, *Pet Owning*, and so on.

(2) **Values** are the fillings of the corresponding keys. They can be generally divided into three types: numerical type, string type, and boolean type.

(3) **Operators** represent the relationship between keys and values. For numerical values, seven types of operators can be used to connect them with corresponding keys: *Equal To*, *Greater Than*, *Less Than*, *Not Equal To*, *Not Greater Than*, *Not Less Than*, and *Between*. For values of string type and boolean type, there are two types of operators: *Belongs To* and *Not Belongs To*. Keys, operators, and values can form a basic **conditional expression** in SELL, formatted as “(key#operator#value)”, such as “(User Marital Status#Belongs#True)”. A conditional expression represents a cluster of target users.

(4) **Intersection and Union symbols** are used to take the intersection or union of multiple conditional expressions (targeted users). We use the symbols *AND* and *OR* to denote the intersection and union operations, respectively.

Compared to the popular database languages SQL and logic language FOL, the advantages of SELL are: (1) SELL is simpler in expression and has much fewer syntax symbols, making it easier for non-programming experts to understand. (2) Since SELL can be transformed into a clear tree structure, marketers can easily modify them even if the generated results have flaws. Due to space limitations, we recommend readers to read more details about SELL in the supplementary materials.

### 2.2 Prompting With Analogical Reasoning

The core idea of analogical reasoning in the NL2SELL task is that if two demands are similar, they may share similar reasoning steps, as well as similar logical structures. Thus, we construct a small but high-quality reasoning library to provide references for numerous demands with unknown reasoning steps, minimizing the cost of manually writing reasoning steps for each of them.

**2.2.1 Construction of Reasoning Library.** To construct a reasoning library, we first randomly collect  $N_{RL}$  marketer demands from real-world scenarios and write their corresponding answers based on our expert knowledge to form Q&A pairs  $\mathcal{R} = \{(d_e, s_e)\}_{e=1}^{N_{RL}}$ , where  $d_e$  is the demand,  $s_e$  is the corresponding SELL expressions. The collected marketer’s demands encompass services that are popular on Alipay, including education, technology, reading, travel, financial management, insurance, government affairs, and public welfare, thus ensuring that the reasoning library constructed based

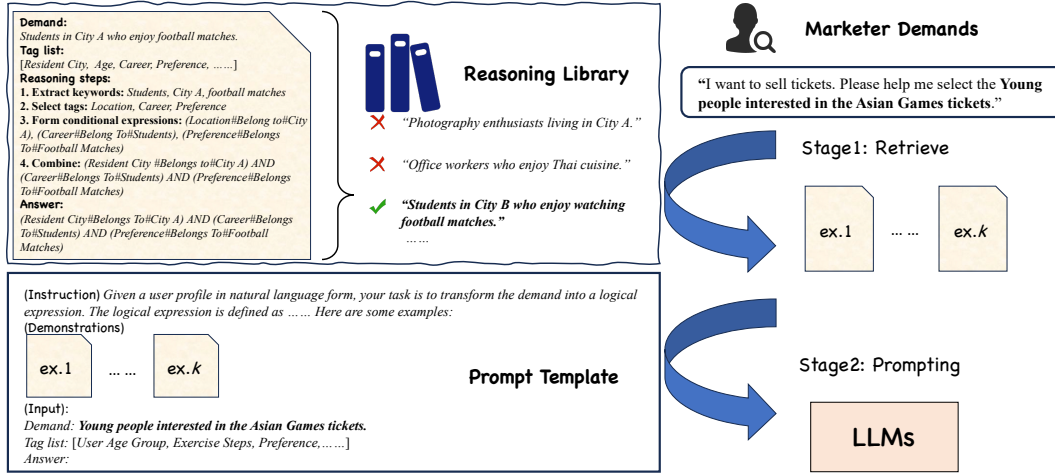


Figure 2: Overview of analogical reasoning based prompting methods.

on this can cover as many demand topics as possible. Then we use a small amount of manual effort to write corresponding reasoning details for 10% random selected Q&A pairs  $\in \mathcal{R}$  to construct the seed examples in the format  $(d_e, r_e, s_e)$ , where  $r_e$  represent the reasoning details. We generally summarize the procedure of solving the NL2SELL problem as four steps:

- (1) Extract keywords: Extract the keywords from the demand.
- (2) Select tags: Select the most relevant tags from the tag list based on the keyword as the key in SELL.
- (3) Form conditional expressions: Based on the selected keywords and contextual information, fill in the corresponding operators and values to form conditional expressions.
- (4) Combine: Combine conditional expressions using intersection or union symbols.

What we need to do is to fill in the details of these four steps.

For the remaining data in  $\mathcal{R}$ , we utilize the OpenAI gpt-3.5-turbo-1106 (GPT-3.5) API to help us fill the corresponding reasoning steps, and our seed examples serve as demonstrations in the prompt. Since we have standardized the overall framework of reasoning, it is not difficult for GPT-3.5 to complete this part of the reasoning steps, cases are shown in supplemental materials. After being verified by experts, we harvest a batch of solid reasoning details to form a reasoning library  $\mathcal{R} = \{(d_e, r_e, s_e)\}_{e=1}^{N_{RL}}$ .

**2.2.2 Retrieval-Based Analogical Prompt Construction.** After the construction of the reasoning library, we can provide references for unknown demands through analogical reasoning. As shown in Figure 2, given a marketer demand  $d_p$  that needs to predict the corresponding SELL expression, we first retrieve similar demands  $d_e$  in  $\mathcal{R}$  based on their text embedding similarity:

$$\text{sim}(d_p, d_e) = \cos(\mathbf{E}(d_p), \mathbf{E}(d_e)), d_e \in \mathcal{R} \quad (1)$$

where  $\mathbf{E}$  is the embedding model in which we use BGE-large-zh [27, 31] off the shelf as it is well-performed in the Chinese text retrieval field. The top  $k$  demands in  $\mathcal{R}$  that get the highest similarity score when compared with  $d_p$  will be fetched as analogical examples in this stage, along with their corresponding  $r$  and  $s$ .

In addition, to help the LLMs better organize answers, we should also inform the LLMs of the tags in the marketing database as explicit knowledge. To address the issue of an excessive number of tags, we still use BGE as a retriever to retrieve the most relevant  $n$  tags of every demand from the marketing database to construct a part of the prompt. The top  $n$  tags that get the highest similarity scores when compared with demand will form a small-scale tag list  $T = \{t_x\}_{x=1}^n$  to provide a reference for selecting keys in SELL.

Therefore, the basic organizational form of the final prompt  $x_p$  for predicting is :

$$x_p = \text{concat}(\mathcal{I}, \{(d_e, T_e, r_e, s_e)\}_{e=1}^k, (d_p, T_p)) \quad (2)$$

where  $\mathcal{I}$  is the instruction which simply describes the task,  $k$  is the number of analogical examples,  $T_e$  and  $T_p$  are the most relevant tag lists of the  $d_e$  and  $d_p$  respectively.

Compared to the ordinary CoT method, our approach exhibits several main differences: (1) The NL2SELL task is more challenging. Unlike solving mathematical problems or common sense reasoning tasks, the training data for LLMs don't inherently include the syntax of SELL. This highlights the importance of appropriate reasoning steps. (2) We place a greater emphasis on the compatibility between prompts and questions. We believe that using an analogical reasoning based prompting method can make LLMs more adaptable to specific problems, rather than relying on fixed few-shot examples, which could potentially limit the model's performance.

## 2.3 Reasoning-Augmented Multi-Task Model Distillation

To train a private small model for deployment, we first distill knowledge from teacher LLM (GPT-3.5) to construct an NL2SELL dataset with over 10,000 pieces of training samples and then propose a multi-task training method with reasoning augmentation.

**2.3.1 Knowledge Distillation.** We collect nearly 10,000 demands from our daily marketing environment. Following the analogical reasoning method mentioned above, we construct input prompts for each demand in the form of Equation (2), where analogical

examples come from the previously constructed reasoning library. Subsequently, we call the GPT-3.5 API to generate SELL answers based on the above prompts, which will then be further verified and corrected through manual crowdsourcing. We denote this part of data comes from *demand to answer* paradigm.

It can be seen that the above distillation method relies on a *large number of native demands and tedious manual calibration*. At the same time, due to the different frequencies of user tags used in actual marketing activities, the distilled data has a *long tail distribution*. This means that some demands based on unpopular user tags only exist in a small amount of data, resulting in poor response to such demands. To address the above two issues, we propose a distillation approach based on *answer to demand*, which can be further divided into two stages: **answer generation** and **demand generation**.

- **Answer generation:** We first construct a series of logical templates of SELL. The content of logical templates is obtained from the marketing database through random sampling. Specifically, we first randomly sample a tag from the database as a key. Then we sample its operator based on the type of keys. Finally, we randomly select a value from all possible values of that key, thus constructing a unit conditional expression in the form of  $(key\#operator\#value)$ . Different sampled conditional expressions can be combined by union and intersection to form different types of targeted users. To ensure that the synthesized answers are logically reasonable, we will limit the number of conditional expressions in the logical template, and abandon overly complex intersection and union operations.

- **Demand generation:** Considering the strong ability of GPT-3.5, we provide some  $(d_e, s_e)$  examples in  $\mathcal{R}$  as demonstrations to guide GPT-3.5 to write demands given above synthesized answers:

$$\text{concat}((d_e, s_e), s_i) \xrightarrow{\text{GPT-3.5}} d_i \quad (3)$$

where  $s_i$  is the synthesized SELL answers. The language fluency and logical rationality of the generated demand will be checked.

We denote this part of data distilled through the *answer to demand* pipeline. We obtain 1,200 pieces of data in this way and combine them with data coming from the *demand to answer* pipeline to form the final training data  $\mathcal{D}_{train} = \{(d_i, s_i)\}_{i=1}^N$  for small model, where  $N$  is the number of training data.

**3.3.2 Reasoning-Augmented Multi-Task Fine-tune.** The training objective of the normal distillation is to minimize the loss  $L$ :

$$L = \frac{1}{N} \sum_{i=1}^N CE(f(d_i, s_i), (d_i, s_i)) \in \mathcal{D}_{train} \quad (4)$$

where  $f$  represents the model,  $CE$  is the cross-entropy loss function.

It [9, 12] has been pointed out that fine-tuning LLMs using data combined with reasoning steps can enhance the reasoning capabilities of the LLMs. Inspired by [9], we first prompt GPT-3.5 to generate intermediate reasoning steps of  $\mathcal{D}_{train}$  based on  $\mathcal{R}$ :

$$\text{concat}((d_e, r_e, s_e), (d_i, s_i)) \xrightarrow{\text{GPT-3.5}} r_i \quad (5)$$

where  $(d_e, r_e, s_e) \in \mathcal{R}$ ,  $(d_i, s_i) \in \mathcal{D}_{train}$ ,  $r_i$  is the corresponding generated reasoning steps of  $(d_i, s_i)$ . Subsequently, we define a multi-task training strategy in which not only  $s_i$  but also  $r_i$  are used as supervised labels to train the model.

**Table 1: Data statistics.** Number( $d$ ), MaxLen( $d$ ), MinLen( $d$ ) and AvgLen( $d$ ) denote the number, the maximum string length, the minimum string length and the average string length of the demands.

	Number( $d$ )	MaxLen( $d$ )	MinLen( $d$ )	AvgLen( $d$ )
$\mathcal{R}$	150	42	7	16.5
$\mathcal{D}_{train}$	11329	207	1	16.8
$\mathcal{D}_{test}$	170	48	3	17.0

To be specific, we denote the input of these two tasks as:

$$x_{i_s} = \text{concat}(\mathcal{I}_s, \{(d_e, T_e, r_e, s_e)\}_{e=1}^k, (d_i, T_i)) \quad (6)$$

$$x_{i_r} = \text{concat}(\mathcal{I}_r, \{(d_e, T_e, r_e, s_e)\}_{e=1}^k, (d_i, T_i)) \quad (7)$$

where the only difference between them is the content of instruction  $\mathcal{I}_s$  and  $\mathcal{I}_r$  which guide the model to generate the SELL answer or reasoning steps respectively. The corresponding training objectives of these two tasks are:

$$L_s = \frac{1}{N} \sum_{i=1}^N CE(f(x_{i_s}), s_i) \quad (8)$$

$$L_r = \frac{1}{N} \sum_{i=1}^N CE(f(x_{i_r}), r_i) \quad (9)$$

The final loss function of multi-task fine-tuning is defined as:

$$L_{total} = L_s + L_r \quad (10)$$

## 3 Experiments

### 3.1 Experimental Settings

**3.1.1 Benchmarks.** We manually construct an NL2SELL testing benchmark for predicting, which contains 170 pieces of data based on expert knowledge in the format  $\mathcal{D}_{test} = \{(d_p, s_p)\}_{p=1}^{170}$ . The demands in  $\mathcal{D}_{test}$  do not exist in the training set  $\mathcal{D}_{train}$  or reasoning library  $\mathcal{R}$ , eliminating the issue of label leakage. Table 1 shows the statistics of these 3 parts of data<sup>1</sup>. It should be noted that all training data generated by GPT-3.5 are strictly verified by well-trained volunteers, and we provide some examples in the supplementary materials. We will open source all these datasets on request.

**3.1.2 Baseline.** We first use GPT-3.5 to test the effectiveness of the analogical reasoning based prompting method. For comparison, we propose four additional variants of prompts as baselines:

- **Zero-shot:** Only given the instruction  $\mathcal{I}$ , tag list  $T_p$  and a test demand  $d_p$ .

- **Fixed few-shot:** Given the instruction  $\mathcal{I}$ ,  $k$  fixed examples  $\{(d_e, s_e)\}_{e=1}^k \in \mathcal{R}$  for all  $d_p \in \mathcal{D}_{test}$ , tag list  $T_p$  and a  $d_p$ .

- **Fixed few-shot + RS (Reasoning Steps):** Given the instruction  $\mathcal{I}$ ,  $k$  fixed examples  $\{(d_e, r_e, s_e)\}_{e=1}^k \in \mathcal{R}$  for all  $d_p \in \mathcal{D}_{test}$ , tag list  $T_p$  and a  $d_p$ .

- **Random few-shot + RS:** Given the instruction  $\mathcal{I}$ ,  $k$  randomly sampled examples  $\{(d_e, r_e, s_e)\}_{e=1}^k \in \mathcal{R}$  for every  $d_p \in \mathcal{D}_{test}$ , tag

<sup>1</sup>The dataset does not contain any Personal Identifiable Information (PII). The dataset is desensitized and encrypted. Adequate data protection was carried out during the experiment to prevent the risk of data copy leakage.



list  $T_p$  and a  $d_p$ . Different  $d_p$  has different randomly sampled examples.

In the knowledge distillation stage, we choose ChatGLM2-6B-32K [5] and Baichuan2-13B-Chat [1] as student LLMs, considering their better comprehension ability in dealing with long Chinese texts compared to other models. In addition to fine-tuning methods mentioned in Section 2.3.1, we also propose three other training strategies as baselines:

- - MT: Training LLMs without multi-task (MT) strategy. The model input is (6) and the training objective is (8).
- - RS: Training LLMs with multi-task strategy but the reasoning steps  $r_e$  of examples in (6) and (7) are not given in the inputs, in which way we can explore the impact of analogical reasoning.
- Normal: Training LLMs in a most normal way, in which the input of the model is the simple combination of instruction  $I$  and a demand  $d_i$ , and the supervised label is  $s_i$  only.

**3.1.3 Evaluation metrics.** To conduct a more reasonable and comprehensive evaluation, we evaluate the output of the LLMs on the NL2SELL task from 4 perspectives:

- Structure accuracy (Structure Acc.): The logic structure’s accuracy of the predicted SELL expression. We remove the keys, operators, and values in the SELL, leaving only intersection and union symbols and separators as logic structures. For example, the structure of “((Location#Belongs To#Hangzhou) OR (Location#Belongs To#Shanghai)) AND (Pet Owning#Belongs To#True)” is “((##) OR (##)) AND (##)”. We use the mean value of the Levenshtein distance (L) [14] and the Ratcliff/Obershelp similarity (R/O) [2] to evaluate the structure accuracy compared to label SELL.

- Overall accuracy (Overall Acc.): We use the ScaREBLEU [20] score to evaluate the overall similarity of the output and label texts as the overall accuracy. We set this as the main metric to evaluate the quality of predicted SELL answers.

- Human evaluation (Human Eval.): We conduct more fine-grained evaluations through human evaluation. Annotators who receive good training will score the accuracy of the key, value, and logic between 0 and 10, and finally provide an overall score.

- GPT4 evaluation (GPT4 Eval.): We use the most advanced LLM gpt-4-turbo-preview (GPT-4) as the judges. To be specific, we prompt the GPT-4 to score the predicted answers from 0 to 10 and provide different scoring examples as a reference for GPT-4.

**3.1.4 Implementation details.** We set the number of analogical examples  $k$  as 3. The size of the reasoning library  $N_{RL}$  is 100. In the fine-tuning stage, ChatGLM2-6B-32K and Baichuan2-13B-Chat are tuned with LoRA [10] on 1 and 4 A100 (80G). We set the optimizer, maximum context size, batch size, and learning rate to Adam, 4096, 8, and  $5e-5$  respectively. For testing, a single 80G A100 is used. We use the code framework from LLaMA-Factory<sup>2</sup> [8].

## 3.2 Analogical Prompting result on GPT-3.5

**3.2.1 Main Results.** Table 2 displays the results of different prompting methods on GPT-3.5, where ARAP represents our Analogical Reasoning Augmented Prompting. The experimental results demonstrate that: (1) ARAP significantly outperforms other prompting

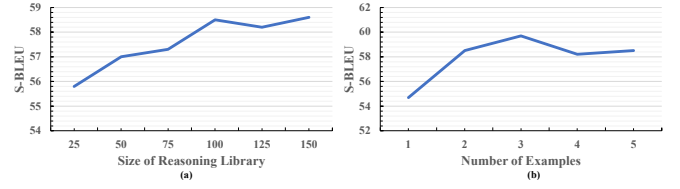


Figure 3: The impact of the size of parameter settings.

methods in various evaluation metrics, especially in terms of structural accuracy which improves by over 7%, proving that analogical examples provide a good logical structure for reasoning. (2) Even randomly selected few-shot reasoning examples are better than fixed ones. This indicates that fixed reasoning examples for all inputs are often suboptimal, demonstrating the necessity of establishing a reasoning library to provide diverse reasoning samples. (3) Reasoning steps are essential in prompting. When comparing results between “Fixed few-shot” and “Fixed few-shot + RS”, we found that reasoning steps have a significant active impact on reasoning accuracy, which is similar to the trend in other previous research works [26].

Further, we explore the impact of parameter settings, such as the size of reasoning library  $N_{RL}$  and the number of analogical examples  $k$ . As shown in Figure 3 (a), as the size of the reasoning library increases, the overall accuracy gradually increases, but the rate slows down, indicating that when the inference library reaches a certain scale, it can already provide references for most marketer’s demands. In Figure 3 (b), it can be observed that the overall accuracy first increases and then decreases when the number of examples in prompts increases. This reflects that when engaging in context-based learning, providing more examples to LLMs is not always better. When an excessive number of examples are input, irrelevant examples may introduce noise, and an overly lengthy context can also impose a greater burden on the model’s understanding.

**3.2.2 Case Study.** Table 3 is a case we obtained from GPT-3.5 with different versions of prompts. Although we provide the same tag list for a specific demand across different prompt versions, there are significant variations in the selection of key-value pairs. For the first demand “Company white-collar workers who enjoy drinking Starbucks”, all baseline versions ignore the tag of “Career”, indicating that the model cannot smoothly map the description “white-collar workers” to “Career”, although this is common sense. Meanwhile, in both fixed few-shot versions, the model treats “company white-collar workers” as the value of the “Resident City”, which is due to the answers of fixed examples in prompts containing similar expressions such as (Resident city#Belongs To#Hangzhou), resulting in the model being inappropriately imitated. While in ARAP, we retrieve similar demands like “Young people who enjoy swimming or female white-collar workers who enjoy reading in Shanghai” and its reasoning steps from the reasoning library, thus helping the LLM analogically transform the “white-collar workers” into conditional expression “(Career#Belongs To#White-collar)” as they share the similar description “white-collar workers”.

<sup>2</sup><https://github.com/hiyouga/LLaMA-Factory>

**Table 2: NL2SELL results on GPT-3.5. The best results are in bold, while the second are underlined.**

	Overall Acc.	Structure Acc.			Human Eval.				GPT4 Eval.
	S-BLEU	L	R/O	Mean	Key	Value	Logic	Overall	GPT-S
Zero-shot	37.0	0.225	0.738	0.482	0.549	0.638	0.597	0.628	5.87
Fixed few-shot	42.0	0.368	0.719	0.543	0.613	0.692	0.637	0.735	6.54
Fixed few-shot + RS	52.8	0.657	0.771	0.714	<u>0.635</u>	0.701	0.654	0.734	6.65
Random few-shot + RS	56.5	<u>0.727</u>	0.791	0.759	0.626	<b>0.731</b>	0.670	0.772	<u>6.74</u>
ARAP	<b>58.5</b>	<b>0.779</b>	<b>0.809</b>	<b>0.794</b>	<b>0.653</b>	<u>0.715</u>	<b>0.678</b>	<b>0.774</b>	<b>6.84</b>

**Table 3: Case study on GPT-3.5 with different versions of prompt. Bad responses are marked with a wavy underline.**

Demand: <i>Company white-collar workers who enjoy drinking Starbucks</i>	
Zero-shot	(Preference#Belongs To#Starbucks)
Fixed few-shot	(Preference#Belongs To#Starbucks) AND (Resident City#Belongs To# <u>Company white-collar</u> )
Fixed few-shot + RS	(Preference#Belongs To# <u>Enjoy drinking Starbucks</u> ) AND (Resident City#Belongs To# <u>Company white-collar</u> )
Random few-shot + RS	(Preference#Belongs To#Starbucks)
ARAP	( <b>Preference#Belongs To#Starbucks</b> ) AND ( <b>Career#Belongs To#White-collar</b> )

### 3.3 Knowledge distillation results on smaller LLMs

**3.3.1 Main Results.** Table 4 shows the testing results on fine-tuned LLMs with the knowledge distilled from GPT-3.5, where ARAFT represents fine-tuning (FT) in the reasoning-augmented multi-task paradigm mentioned in section 2.3.2. All testing prompts are the same as ARAP prompts in GPT-3.5. As is shown: (1) Both two models show comparable capability to GPT-3.5, especially on Baichuan2-13B-Chat, where the score of S-BLEU is improved by over 5% compared to GPT-3.5 after fine-tuning. This demonstrates the superiority of our distillation and training methods. (2) Compared with the results of the single-task training strategy (-MT), the multi-task training strategy brings improvement to the reasoning performance of the LLMs. Although during testing, we limit the model to only output the final answers without reasoning steps, adding training tasks that predict reasoning steps can enhance the models’ reasoning abilities and robustness. (3) Compared with the results without reasoning steps in the input corpus (-RS), we can conclude that providing explicit reasoning steps in training input is beneficial. If there are no reasoning steps in the input corpus but a multi-task training strategy is used, it will increase the difficulty of the training task and lead to poor performance, which is particularly evident in Baichuan2-13B-Chat. (4) The result of fine-tuning using only the normal demand and answer pairs is not satisfactory, which indicates that it is challenging for LLMs to learn the data patterns solely based on the Q&A pairs in situations where tasks are difficult. Therefore, it is necessary to design appropriate demonstrations and training tasks to fine-tune the LLMs.

**3.3.2 Ablation Study.** As mentioned in section 2.3.1, there are two sources of distilled knowledge we use when fine-tuning, *i.e.*, knowledge distilled from *demand to answer* and *answer to demand* approach. The knowledge distilled from *demand to answer* is the most commonly used knowledge distillation approach, and what we mainly want to explore is the effectiveness of the knowledge distilled from *answer to demand*. To verify this, we conduct ablation experiments on the best-performing models “Baichuan2-13B-Chat” with two training settings (*i.e.*, ARAFT and ARAFT - RS). Table 5 shows the effect of the knowledge distilled from *answer to demand* (a2d), it can be found that when removing this part of data during the fine-tuning, the performance on both ARAFT and ARAFT - RS significantly decreases, although this part of data only accounts for about one-tenth of the training data. This indicates that adding samples with a more uniform distribution of logic and tags to the training corpus can improve the robustness of the LLMs, and our distillation method from *answers to demand* achieves this.

## 4 Application

**Model Deployment.** The LLM tuned in section 3.3 has been deployed online for application using an A10 with a memory capacity of 24G. To achieve optimal online inference performance, the deployment involves a total of 8 cards, including both the pre-production and online environments. Meanwhile, the retriever service based on BGE has also been deployed on an A10 GPU.

**Application Case.** Figure 4 shows an application case of AR-ALLM in online user targeting. A marketer wants to do marketing for a stage drama performance, so he can input a raw demand and click the “Search” button. Our system will invoke the retriever service to retrieve similar top-k demonstrations and top-n tags from the reasoning library embeddings (RL Embs) and tag embeddings (Tag Embs), respectively. After filling the prompt based on retrieved demonstrations and tags, our system requests the deployed ARAFT model to generate the answer expressed in SELL. The SELL expression will be parsed by our parser and results will be visualized to the marketers on the panel as shown in Figure 4. The marketers can verify and edit the card on the panel after clicking the edit button. Finally, they can click the “Export” button to get the target users.

**Practical Effects.** Our system has been already running for months. We have set two metrics to evaluate the new approach online, including *operation time*, and *number of likes* from marketers. By collecting the operation log, the entire system takes no more than 10 seconds to complete a request and the average marketers’ operation time of the new system is about 3 minutes, which is

**Table 4: NL2SELL results on distilled model Baichuan2-13B-Chat and ChatGLM-6B-32K. The best tuning results of each fine-tuned LLMs are in bold.**

		Overall Acc. S-BLEU	Structure Acc.			Human Eval.				GPT4 Eval.
			L	R/O	Mean	Key	Value	Logic	Overall	GPT-S
<i>GPT-3.5</i>	<i>ARAP</i>	59.7	0.779	0.809	0.794	0.815	0.715	0.678	0.774	6.84
Baichuan2-13B-Chat	ARAFT	<b>62.2</b>	<b>0.822</b>	<b>0.806</b>	<b>0.814</b>	<b>0.779</b>	<b>0.737</b>	<b>0.813</b>	<b>0.735</b>	<b>7.10</b>
	- MT	60.8	0.809	0.794	0.802	0.754	0.731	0.800	0.718	6.98
	- RS	50.1	0.741	0.762	0.751	0.713	0.661	0.754	0.679	6.63
	Normal	54.9	0.812	0.786	0.806	0.754	0.712	0.809	0.715	6.77
ChatGLM2-6B-32K	ARAFT	<b>56.0</b>	<b>0.829</b>	<b>0.773</b>	<b>0.801</b>	<b>0.740</b>	<b>0.715</b>	<b>0.790</b>	<b>0.710</b>	<b>7.13</b>
	- MT	55.3	0.784	0.757	0.770	0.726	0.689	0.768	0.686	6.65
	- RS	55.0	0.775	0.742	0.758	0.688	0.669	0.741	0.668	6.33
	Normal	42.0	0.368	0.719	0.543	0.641	0.603	0.698	0.612	5.72

**Table 5: Ablation study of *answer to demand* distillation approach on Baichuan2-13B-Chat.**

	Use <i>a2d</i> ?	Overall Acc. S-BLEU	Structure Acc.			Human Eval.				GPT4 Eval.
			L	R/O	Mean	Key	Value	Logic	Overall	GPT-S
ARAFT	✓	<b>62.2</b>	<b>0.822</b>	<b>0.806</b>	<b>0.814</b>	<b>0.779</b>	<b>0.737</b>	<b>0.813</b>	<b>0.735</b>	<b>7.10</b>
	✗	58.9	0.811	0.784	0.798	0.752	0.713	0.790	0.712	7.01
ARAFT - RS	✓	<b>50.1</b>	<b>0.741</b>	<b>0.762</b>	<b>0.751</b>	<b>0.713</b>	<b>0.661</b>	<b>0.754</b>	<b>0.679</b>	<b>6.63</b>
	✗	46.1	0.667	0.728	0.697	0.674	0.625	0.700	0.637	6.37

**Table 6: Online A/B testing results.**

Campaign	CTR	Exposure
Digital car	+55.21%	+0.33%
Housing provident fund	+30.56%	-0.21%
Ophthalmic health	+69.17%	+0.01%
Traditional Chinese Medicine	+112.7%	+0.67%
Recharge	+73.89%	+0.20%

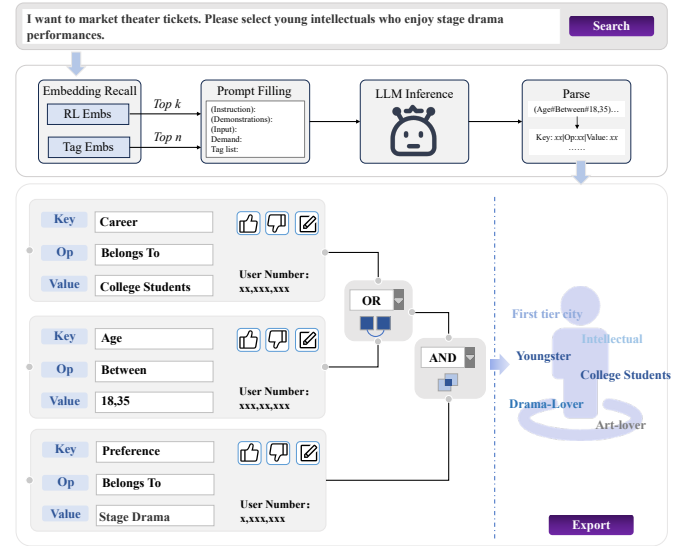
4-10 times shorter than the former on average. Beyond that, we receive likes 1.3 times larger than the former, revealing the operation friendliness of this new way of user targeting.

**Online A/B testing results.** We conduct online A/B experiments to compare our methods with the traditional rule-based methods and use the direct metrics CTR (Click-Through-Rate) and Exposure (the number of users who have been exposed by the campaign) to evaluate them. Results are shown in Table 6. From an end-to-end perspective, our solution has achieved better marketing results than rule-based methods in multiple marketing campaigns.

## 5 Related Work

### 5.1 User targeting

Methods of user targeting mainly can be categorized into two lines: rule-based and model-based methods. The rule-based methods [19, 23] match potential users with specific demographic tags (age, gender, geography) or interests that are targeted by marketers which need marketers to do user profile association or mining. The model-based methods expand a given seed set by calculating the similarity of all pairs between seed users and candidate users [16, 18] or training customized prediction models for each service or

**Figure 4: The application of ARALLM in User Targeting.**

campaign [32, 33]. These methods show good performance in user targeting but need seed sets. Both methods have primarily focused on intricate architectures for effectively and efficiently targeting users, while commonly disregarding the natural and significant gap between marketers' demands and the capabilities of current models, which can be time-consuming and unfriendly.

### 5.2 NL2SQL

Our NL2SELL task is somewhat similar to the task of translating natural language into SQL (NL2SQL).



NL2SQL is an important and challenging task in helping non-expert users to better manipulate databases. Graph neural networks (GNNs) are popular in NL2SQL research [4, 11, 25], which focus on modeling the relationship between the question and the schema in database system. Recently, large pre-trained language models (PLMs), such as T5 [21] and GPT-3 [3], have shown strong translating ability in NL2SQL tasks. Due to the extensive knowledge injection during the pretrain phase, PLM-based methods achieve better results compared to GNN-based methods by fine-tuning them with a small amount of data. Some works also explore solving NL2SQL tasks with in-context learning [7, 15], it has been pointed out that LLMs show strong few-shot or zero-shot abilities even without any training data, which start a new direction for future research on NL2SQL field. The LLM-based methods in NL2SQL bring inspiration to our works.

## 6 Conclusion and future work

In this paper, we provide a novel user targeting approach by leveraging LLMs to gain a structured understanding of marketers' demands. We first define a new language SELL to express the needs of marketers more clearly. Subsequently, we propose an analogical reasoning augmented framework, ARALLM, which consists of analogical reasoning based prompting and reasoning-augmented multi-task model distillation. The experimental results on GPT-3.5 show that our analogical reasoning based prompting significantly outperforms other baseline prompting methods in the NL2SELL task. In addition, we distill a large-scale dataset using GPT-3.5 and train the student LLMs using a multi-task training approach, which is successfully used for online deployment.

Since we primarily focus on the NL2SELL task in practical application scenarios, we have not yet applied the ARALLM framework to other reasoning tasks, such as coding or other structured language translation (e.g., HTML, JSON), which will be explored as future work. At the same time, there are still labor costs in the construction of reasoning libraries and SELL datasets, we will explore more automated construction methods in the future.

## Acknowledgments

This work is funded by NSFC62306276, NSFCU23B2055 and NSFCU19B2027. This work is supported by the Fundamental Research Funds for the Central Universities (226-2023-00138). This work was supported by Ant Group.

## References

- [1] Baichuan. 2023. Baichuan 2: Open Large-scale Language Models. *arXiv preprint arXiv:2309.10305* (2023). <https://arxiv.org/abs/2309.10305>
- [2] Paul E Black. 2004. Ratcliff/Obershelp pattern recognition. *Dictionary of algorithms and data structures* 17 (2004).
- [3] 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 (2020), 1877–1901.
- [4] Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. 2021. LGEQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. <https://doi.org/10.18653/v1/2021.acl-long.198>
- [5] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 320–335.
- [6] Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive science* 7, 2 (1983), 155–170.
- [7] Zihui Gu, Ju Fan, Nan Tang, Lei Cao, Bowen Jia, Sam Madden, and Xiaoyong Du. 2023. Few-shot Text-to-SQL Translation using Structure and Content Prompt Learning. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–28.
- [8] hiyouga. 2023. LLaMA Factory. <https://github.com/hiyouga/LLaMA-Factory>.
- [9] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301* (2023).
- [10] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).
- [11] Binyuan Hui, Ruiying Geng, Qiyu Ren, Binhua Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, Pengfei Zhu, and Xiaodan Zhu. 2021. Dynamic Hybrid Relation Exploration Network for Cross-Domain Context-Dependent Semantic Parsing. *Proceedings of the ... AAAI Conference on Artificial Intelligence, Proceedings of the ... AAAI Conference on Artificial Intelligence* (May 2021).
- [12] Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The CoT Collection: Improving Zero-shot and Few-shot Learning of Language Models via Chain-of-Thought Fine-Tuning. *arXiv preprint arXiv:2305.14045* (2023).
- [13] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.
- [14] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.
- [15] Aiwei Liu, Xuming Hu, Lijie Wen, and Philip S. Yu. 2023. A comprehensive evaluation of ChatGPT's zero-shot Text-to-SQL capability. (Mar 2023).
- [16] Haishan Liu, David Pardoe, Kun Liu, Manoj Thakur, Frank Cao, and Chongzhe Li. 2016. Audience expansion for online social network advertising. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 165–174.
- [17] Xuantao Lu, Jingping Liu, Zhouhong Gu, Hanwen Tong, Chenhao Xie, Junyang Huang, Yanghua Xiao, and Wenguang Wang. 2022. Parsing Natural Language into Propositional and First-Order Logic with Dual Reinforcement Learning. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 5419–5431. <https://aclanthology.org/2022.coling-1.481>
- [18] Qiang Ma, Eeshan Wagh, Jiayi Wen, Zhen Xia, Robert Ormandi, and Datong Chen. 2016. Score look-alike audiences. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 647–654.
- [19] Ashish Mangalampalli, Adwait Ratnaparkhi, Andrew O Hatch, Abraham Bagherjeiran, Rajesh Parekh, and Vikram Pudi. 2011. A feature-pair-based associative classification approach to look-alike modeling for conversion-oriented user-targeting in tail campaigns. In *Proceedings of the 20th international conference companion on World wide web*. 85–86.
- [20] Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*. Association for Computational Linguistics, Belgium, Brussels, 186–191. <https://www.aclweb.org/anthology/W18-6319>
- [21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *arXiv: Learning, arXiv: Learning* (Oct 2019).
- [22] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633* (2021).
- [23] Jianqiang Shen, Sahin Cem Geyik, and Ali Dasdan. 2015. Effective audience extension in online advertising. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2099–2108.
- [24] Peter D Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research* 33 (2008), 615–655.
- [25] Bailin Wang, Richard Shin, Xiaodong Liu, Olexandr Polozov, and Matthew Richardson. 2020. RAT-SQL: Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. <https://doi.org/10.18653/v1/2020.acl-main.677>
- [26] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.
- [27] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. *arXiv:2309.07597* [cs.CL]
- [28] Dan Yang, Binbin Hu, Xiaoyan Yang, Yue Shen, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. 2023. Who Would be Interested in Services? An Entity Graph Learning System for User Targeting. *arXiv preprint arXiv:2305.18780* (2023).

- [29] Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. 2023. Harnessing the Power of Large Language Models for Natural Language to First-Order Logic Translation. arXiv:2305.15541 [cs.CL]
- [30] Zhen Yao, Wen Zhang, Mingyang Chen, Yufeng Huang, Yi Yang, and Huajun Chen. 2023. Analogical inference enhanced knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4801–4808.
- [31] Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve Anything To Augment Large Language Models. arXiv:2310.07554 [cs.LR]
- [32] Yongchun Zhu, Yudan Liu, Ruobing Xie, Fuzhen Zhuang, Xiaobo Hao, Kaikai Ge, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to Expand Audience via Meta Hybrid Experts and Critics for Recommendation and Advertising. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4005–4013.
- [33] Chenyi Zhuang, Ziqi Liu, Zhiqiang Zhang, Yize Tan, Zhengwei Wu, Zhining Liu, Jianping Wei, Jinjie Gu, Guannan Zhang, Jun Zhou, et al. 2020. Hubble: An industrial system for audience expansion in mobile marketing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2455–2463.

## A Design of SELL

We provide a detailed explanation of the design and the usage of the SELL. The SELL is mainly composed of four elements:

**Keys** are a series of tags that describe the features of the user, such as *Gender*, *Marital Status*, *Resident City*, and so on. In practical marketing scenarios, the database contains tens of thousands of tags that describe user profiles. Selecting the appropriate keys from this vast array of tags is a challenging task.

**Values** are the fillings of the corresponding keys. They can be generally divided into three types: numerical type, string type, and boolean type. For instance, the value corresponding to the key *Monthly Income* is of numerical type, because they are continuous and non-enumerable. The value of *Gender* is of string type, with limited options: *Male* or *Female*. Boolean type values have only two states, i.e., *True* or *False*, such as *Marital Status*.

**Operators** represent the relationship between keys and values. For numerical values, seven types of operators can be used to connect them with corresponding keys: *Equal To*, *Greater Than*, *Less Than*, *Not Equal To*, *Not Greater Than*, *Not Less Than*, and *Between*. For values of string type and boolean type, there are two types of operators: *Belongs To* and *Not Belongs To*. Keys, operators, and values can form a basic **conditional expression** in SELL, formatted as “(key#operator#value)”, such as “(Marital Status#Belongs To#True)”. A conditional expression represents a cluster of target users, for example, in the above example, the target users are married.

**Intersection and Union symbols** are used to take the intersection or union of multiple conditional expressions. In practical marketing scenarios, the features of the target users are often complex and difficult to describe using a single conditional expression. Therefore, it is necessary to combine the target users through intersection or union operations. In SELL, we use the symbols *AND* and *OR* to denote the intersection and union operations, respectively. For example, “((Resident City#Belongs To#City A) OR (Resident City#Belongs To#City B)) AND (Pet Owning#Belongs To#True)” describes the pet owners who live in City A or City B.

Some demands and their corresponding SELL expressions are shown in Table 7. More data can be found at <https://github.com/alipay/Analogic-Reasoning-Augmented-Large-Language-Model>.

## B Human-Written Reasoning Steps

As mentioned in the main paper, we write reasoning steps for 10% of data in the reasoning library. Table 8 shows an example.

You are a robot skilled in reasoning. Please learn the following examples and complete the final reasoning steps:

```
###
Demand: Mothers who pay attention to baby education
Tag list: [User Has Child/Days of Listening to Audiobooks/Homepage Visits/Red Packet Cover Collection/Current Insured Products/Balance of Points/Alipay User in Gourmet Scene/User Age Group/Number of Days Since User's First Account Opening/User Child Age/User Gender/User Marital Status/Preferences/.....]
Answer: (Preference#Belongs To# Baby Education) AND (User Gender#Belongs To# Female) AND ((User Has Child#Belongs To#True) OR (User Child Age#Between#0,4))
Reasoning:
(1)Extract keywords: Mothers, baby education
(2) Select tags: User Has Child, User Child Age, User Gender, User Marital Status, Preferences
(3) Form conditional expressions:
(User Has Child#Belongs To#True)
(User Gender#Belongs To#Female)
(User Child Age#Between#0,4)
(Preference#Belongs To#Baby Education)
(4) Combine:(Preference# Belongs To# Baby Education) AND (User Gender#Belongs To#Female) AND ((User Has Child#Belongs To#True) OR (User Child Age#Between#0,4))
###
.....
###
Demand: {Input Demand}
Tag list: {Input Tag list}
Answer: {Input Answer}
Reasoning:
```

Figure 5: Reasoning steps generation prompt.

You are a robot skilled in reasoning and story writing, giving you a logical expression that describes a certain group of people, and your task is to transform them into smooth natural language for expression. When generating natural language, please do not copy the expressions in logical expressions, but rather convert them into some synonymous expressions. The following are some examples. Please understand and complete the conversion of the final logical expression.

```
###
Answer: (Reference#Belongs To#Game) AND (User Age Group#Between#15,25) AND ((Career#Equals To#Student) OR (Is College student#Belongs To#True))
Demand: A group of students aged 15-25 who love games.
###
.....
###
Answer: {Input Answer}
Demand:
```

Figure 6: Demand generation prompt.

## C Reasoning Steps Generation Prompt

As mentioned in the main paper, we use GPT-3.5 to help us complete the reasoning steps of remained data. Figure 8 shows the prompt we use in this step.

## D Demand generation prompt

The prompt framework for demand generation given SELL expressions is shown in Figure 6.

## E Logic Template

As mentioned in the main paper, we first construct a series of logical templates of SELL to artificially synthesize answers. Table 9 shows all the logical templates we used. The *c* in the template

**Table 7: Some examples of marketers' demands and corresponding SELL.**

Middle-aged women	(User Age Group#Between#35,55) AND (Gender#Belongs To#Female)
Young people in City A who enjoy swimming or female white-collar workers in City B who enjoy reading	((Resident City#Belongs To#City A) AND (Preference#Belongs To#Swimming) AND (User Age Group#Between#18,35)) OR ((Resident City#Belongs To#City B) AND (Gender#Belongs To#Female) AND (Career#Belongs To#White-collar) AND ((Preference#Belongs To#Reading) OR (Days of Listening To Audiobooks#Greater Than#1)))
Users who have a certain understanding or interest in financial products, insurance, etc.	(Preference#Belongs To#Wealth Management Products) OR (Preference#Belongs To#Insurance) OR (Eligible group for Wealth Infinity Card#Belongs To#True) OR (Has actively invested in major financial products#Belongs To#True)

**Table 8: Examples of human-written reasoning steps.**

Demand	Mothers who pay attention to baby education.
Tag List	User Has Child/Days of Listening to Audiobooks/Homepage Visits/Red Packet Cover Collection/Current Insured Products/Balance of Points/Alipay User in Gourmet Scene/User Age Group/Number of Days Since User's First Account Opening/User Child Age/User Gender/User Marital Status/Preferences/.....
Reasoning Steps	<p><b>(1) Extract keywords:</b> Mothers, baby education</p> <p><b>(2) Select tags:</b> User Has Child, User Child Age, User Gender, User Marital Status, Preferences</p> <p><b>(3) Form conditional expressions:</b>            (User Has Child#Belongs To#True)            (User Gender#Belongs To#Female)            (User Child Age#Between#0,4)            (Preference#Belongs To#Baby Education)</p> <p><b>(4) Combine:</b>            (Preference#Belongs To#Baby Education) AND (User Gender#Belongs To#Female) AND ((User Has Child#Belongs To#True) OR (User Child Age#Between#0,4))</p>
Answers	(Preference#Belongs To#Baby Education) AND (User Gender#Belongs To#Female) AND ((User Has Child#Belongs To#True) OR (User Child Age#Between#0,4))

You are a professional grading expert. Given a predicted logic expression and a ground truth logical expression, your task is to assess the accuracy of the predicted logic expression compared to the standard logical expression. Score the accuracy of the predicted logic expression between 0 and 10. 10 if the predicted answer and the ground truth answer are identical, and 0 if they are entirely different. If only part of them is identical, please rate it between 0 and 10 based on your judgment. The closer the score is to 10, the predicted answer is more accurate. Here are some examples:

####  
 Predicted answer: (Game#Belongs To#Genshin Impact) AND (Age#Greater Than#High) AND (Enthusiast#Equal To#Game) AND (Gender#Equal To#Male)  
 Ground Truth Answer: (Preference#Belongs To#Genshin Impact) AND (User Age Group#Between# 18,35) AND (Gender#Belongs To#Male)  
 Accuracy: 3  
 ####  
 Predicted answer: (User Age Group#Between#16,45) AND (Gender#Belongs To#Female) AND (Pet Owning#Belongs To#True)  
 Ground Truth answer: (User Age Group #Between#16,45) AND (Gender#Belongs To#Female) AND (Pet Owning#Belongs To#Have Pets)  
 Accuracy: 8  
 .....  
 Now given a predicted answer and a ground truth answer, please think carefully and output the accuracy score (don't output other contents):  
 Predicted answer: {Input Predicted answer}  
 Ground truth answer: {Input Ground truth answer}  
 Accuracy:

**Figure 7: GPT4 Evaluation prompt.**

You are a robot skilled in reasoning. Please learn the following examples and complete the final reasoning steps:

###  
 Demand: Mothers who pay attention to baby education  
 Tag list: [User Has Child/Days of Listening to Audiobooks/Homepage Visits/Red Packet Cover Collection/Current Insured Products/Balance of Points/Alipay User in Gourmet Scene/User Age Group/Number of Days Since User's First Account Opening/User Child Age/User Gender/User Marital Status/Preferences/.....]  
 Answer: (Preference#Belongs To# Baby Education) AND (User Gender#Belongs To# Female) AND ((User Has Child#Belongs To#True) OR (User Child Age#Between#0,4))  
 Reasoning:  
 (1) Extract keywords: Mothers, baby education  
 (2) Select tags: User Has Child, User Child Age, User Gender, User Marital Status, Preferences  
 (3) Form conditional expressions:  
 (User Has Child#Belongs To#True)  
 (User Gender#Belongs To#Female)  
 (User Child Age#Between#0,4)  
 (Preference#Belongs To#Baby Education)  
 (4) Combine:(Preference#Belongs To# Baby Education) AND (User Gender#Belongs To#Female) AND ((User Has Child#Belongs To#True) OR (User Child Age#Between#0,4))  
 ####  
 .....  
 ####  
 Demand: {Input Demand}  
 Tag list: {Input Tag list}  
 Answer: {Input Answer}  
 Reasoning:

**Figure 8: Reasoning steps generation prompt.**

represents the basic conditional expression in SELL. To ensure that the synthesized answers are logically reasonable, we limit the number of conditional expressions in the logical template, and abandon overly complex intersection and union operations.

## F GPT4 Evaluation Prompt

The prompt framework for guiding GPT4 to evaluate the quality of answer SELL is shown in Figure 7.

**Table 9: Logic template for answer generation.**

Template
$c$
$c \text{ AND } c$
$c \text{ AND } c \text{ AND } c$
$c \text{ AND } c \text{ AND } c \text{ AND } c$
$c \text{ OR } c$
$c \text{ OR } c \text{ OR } c$
$c \text{ OR } c \text{ OR } c \text{ OR } c$
$(c \text{ AND } c) \text{ OR } c$
$(c \text{ OR } c) \text{ AND } c$
$(c \text{ AND } c) \text{ OR } (c \text{ AND } c)$
$(c \text{ OR } c) \text{ AND } (c \text{ OR } c)$
$c \text{ AND } (c \text{ OR } c \text{ OR } c)$
$c \text{ OR } (c \text{ AND } c \text{ AND } c)$
$(c \text{ AND } c) \text{ OR } (c \text{ OR } c)$
$(c \text{ AND } c) \text{ AND } (c \text{ OR } c)$
$c \text{ AND } ((c \text{ AND } c) \text{ OR } c)$
$c \text{ AND } ((c \text{ OR } c) \text{ AND } c)$
$c \text{ OR } ((c \text{ OR } c) \text{ AND } c)$
$c \text{ OR } ((c \text{ AND } c) \text{ OR } c)$