Real-time Ad retrieval via LLM-generative Commercial Intention for Sponsored Search Advertising

Anonymous ACL submission

Abstract

The integration of Large Language Models (LLMs) with retrieval systems has shown promising potential in retrieving documents (docs) or advertisements (ads) for a given query. Existing LLM-based retrieval methods generate numeric or content-based DocIDs to retrieve docs/ads. However, the one-to-few mapping between numeric IDs and docs, along with the time-consuming content extraction, leads to semantic inefficiency and limits the scalability of existing methods on large-scale corpora. In this paper, we propose the **R**eal-time **A**d **RE**trieval (RARE) framework, which leverages LLM-generated text called Commercial Intentions (CIs) as an intermediate semantic representation to directly retrieve ads for queries in real-time. These CIs are generated by a customized LLM injected with commercial knowledge, enhancing its domain relevance. Each CI corresponds to multiple ads, yielding a lightweight and scalable set of CIs. RARE has been implemented in a real-world online system, handling daily search volumes in billions. The online implementation has yielded significant benefits: a 5.04% increase in consumption, a 6.37% rise in Gross Merchandise Volume (GMV), a 1.28% enhancement in click-through rate (CTR) and a 5.29% increase in shallow conversions. Extensive offline experiments show RARE's superiority over ten competitive baselines in four major categories.

1 Introduction

An advertising system is a commercial application designed to generate revenue by presenting targeted ads to users, primarily consisting of two modules: ad retrieval and ranking. As a crucial component, ad retrieval swiftly filters relevant advertisements from vast libraries containing millions or even billions of candidates in response to user queries. Traditional ad retrieval models follow a two-stage process (Wang et al., 2024a, Ramos et al., 2003, Huang et al., 2013), first retrieving keywords from queries and then using those keywords to fetch ads. However, existing two-stage retrieval methods amplify the difference between user queries and manually chosen keywords, resulting in numerous missed retrieval issues. The query-ad single-stage approach (Gong et al., 2023, Gao et al., 2020) addresses missed recall by directly retrieving ads but still struggles with understanding deeper commercial intentions due to limited reasoning capabilities and domain knowledge.

In recent years, LLMs (Zhao et al., 2023) have garnered widespread attention and made remarkable achievements in the fields of search and recommendation (Lin et al., 2025; Shi et al., 2025; Tang et al., 2024b; Pradeep et al., 2023). Most LLM-based retrieval methods first create an index of docs by training the model to link docs with their identifiers (DocIDs). During retrieval, the model processes a query and generates the corresponding DocIDs (Li et al., 2024a). For example, DSI (Tay et al., 2022) employs numeric IDs to represent documents and establish connections between user queries and numeric IDs. LTRGR (Li et al., 2024b) extracts document content, i.e., article title and body, to represent the document and implement the retrieval from a user query to a document.

Using heavy DocIDs (Zeng et al., 2023) presents several **drawbacks**. Firstly, the inference efficiency is low due to the one-to-few mapping between DocIDs and candidates (Wang et al., 2024b), making it difficult to achieve real-time generation in large-scale scenarios. Secondly, representing docs/ads solely with heavy DocIDs fails to fully leverage the capabilities of LLMs in commercial intent mining and their advanced text generation abilities, thus hindering the effective exploration of the advertiser's intent. Thirdly, it exhibits poor generalization. When new candidates emerge, it often requires retraining the model or updating the FM-index (Ferragina and Manzini, 2000) to accommodate their DocIDs, making it difficult to quickly

update or remove candidates. Due to the requirement for real-time fetching of large sets of ads aligned with the user's commercial intent in ad retrieval, the existing semantically inefficient DocIDs are impractical and unsuitable for the task. Therefore, leveraging the powerful semantic capabilities of LLMs to design more effective semantic tokens for indexing, along with developing a more comprehensive end-to-end architecture, has become a crucial challenge.

To address this challenge, we developed a real-time LLM-generative ad retrieval framework named RARE. This framework utilizes LLM-generated commercial intentions (CIs) as an intermediate semantic representation to directly connect queries to ads, rather than relying on manually chosen keywords or heavy document identifiers. Specifically, RARE initially utilizes a knowledge-injected LLM (offline) to generate CIs for the ads in the corpus. It then selects a limited but comprehensive set of CIs and constructs a **dynamic index** that maps these CIs to their corresponding ads in a one-to-many relationship. Upon receiving a query, the RARE uses customized LLM (online) to generate CIs in real-time and retrieves the corresponding ads from the pre-built index.

A key **innovation** of RARE lies in utilizing CIs generated by a customized LLM to serve as intermediate semantic DocIDs for linking query and ads. Customized LLM is developed by knowledge injection and format fine-tuning of the base LLM. Knowledge injection involves incorporating domain-specific information to enhance expertise in the advertising domain. Format fine-tuning ensures that the LLM outputs only CIs and improves decoding efficiency. CIs are defined as aggregations of keywords, generated by the customized LLM based on relevant materials of ads. Compared to existing carefully designed DocIDs, CIs fully leverages the text generation capabilities of LLMs. The one-to-many correspondence between CIs and ads makes the decoding process highly efficient. For new ads, RARE can generates CIs with the technique of constrained beam search, without the need to retrain the mode. Keyword bidding in the traditional query-keyword-ads paradigm introduces the possibility of index manipulation. In contrast to keywords, CIs are generated by LLMs equipped with world knowledge and commercial expertise, allowing for a better exploration of the commercial intent behind ads and queries.

The main contributions of our work are as follows: (1) We propose a novel end-to-end generative retrieval framework named RARE to achieve real-time retrieval, which is the first known work on LLM-generative architecture that displays real-time retrieving on tens-of-millions scale system. (2) We propose a method for knowledge injection and format fine-tuning to enable the base LLM to uncover the deep commercial intentions of advertisers and users, expressed as CIs. (3) We have deployed an online system based on LLMs for real-time inference and ad retrieval, which serves tens of millions of users in real-world scenarios everyday. (4) We conduct online A/B testing and offline experiments to verify the effectiveness of RARE. A/B testing has yielded a 5.04% increase in consumption, a 6.37% increase in Gross Merchandise Value (GMV), a 1.28% increase in Click-Through Rate (CTR), a 5.29% increase in shallow conversions, and a remarkable 24.77% increase in deep conversions. Simultaneously, in terms of offline evaluation metrics, RARE demonstrates superior performance in HR@500, MAP, and ACR metrics compared to 10 other competitive baselines.

2 Related Works

Ad Retrieval. Traditional ad retrieval (Zhao and Liu, 2024; Wang et al., 2024c) typically follows a query-keyword-ad architecture, where queries retrieve keywords that are then used to pull ads. This approach includes both word-based and semantic-based methods. Word-based methods (Ramos et al., 2003; Robertson et al., 2009) parse user queries to obtain keywords and use an inverted index to retrieve candidate ads. Semantic-based methods (Huang et al., 2013; Yates et al., 2021) use a dual encoder to obtain embeddings for queries and keywords in a shared semantic space, enabling retrieval based on semantic similarities. These methods rely on manually chosen keywords resulting in numerous missed retrieval issues. Nowadays, some researchers have leveraged LLMs for generative doc/ad retrieval (Sun et al., 2024; Lin et al., 2024; Tang et al., 2024a). However, existing methods still face challenges in achieving real-time retrieval from large-scale online repository.

Generative-based LLMs Retriever. Generative based retrievers utilize the generative capabilities of LLMs to construct end-to-end retrieval models. Some approaches, such as DSI (Tay et al., 2022), NCI (Wang et al., 2022), Tiger (Rajput et al., 2024), use document IDs as the generation target to im-

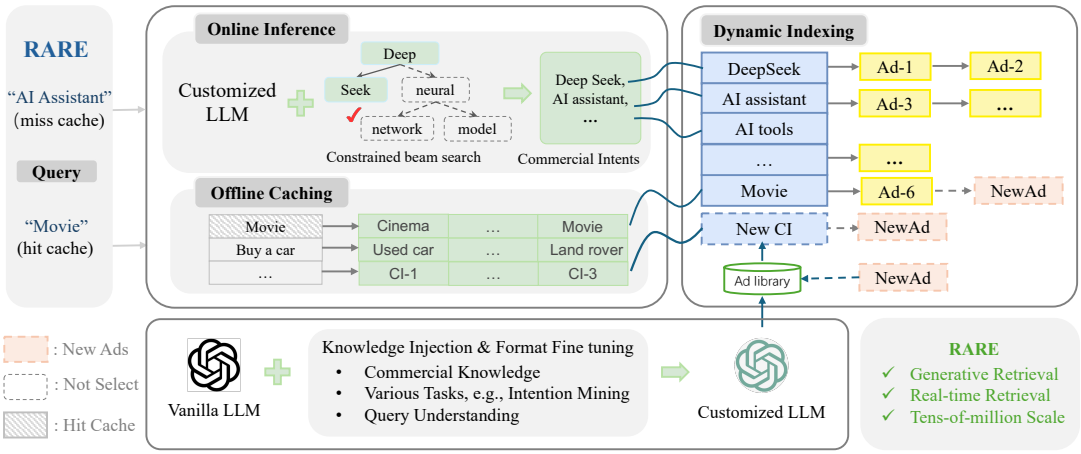

Figure 1: The Real-time LLM-Generative Ad Retrieval framework (RARE) processes user queries by generating commercial intentions (CIs) through LLM/caching, which are subsequently used to retrieve ads from the dynamic index. The customized LLM are created by injecting knowledge and learning rules based on vanilla LLM.

plement query retrieval for docs/ads. These methods leverage LLMs to learn the correspondence between docs/ads and their IDs, directly generating the ID of the relevant docs/ads for query retrieval. Other approaches, such as SEAL (Bevilacqua et al., 2022)and LTRGR (Li et al., 2024b), use document content as an intermediary to achieve document retrieval. They employ FM-Index to generate fragments that appear in the document, facilitating query-to-document retrieval. MINDER (Li et al., 2023) employs pseudo-queries and document content for retrieval, but this significantly increases indexing volume, making it unsuitable for scenarios with large candidate sets.

Semantic DocIDs. LLM-generative retrieval typically employs DocIDs to perform query-to-document retrieval tasks. Existing DocIDs mainly include numeric IDs and document content. For instance, the numeric IDs in Tiger is represented as a tuple of discrete semantic tokens. In LTRGR, document content consists of predefined sequences that appear within the document. However, the semantic tokens used in these approaches are ID-like features, which suffer from low decoding efficiency since each DocID corresponds to few candidates. For new candidate docs or ads, it is necessary to retrain the model or rebuild the FM-Index to obtain their DocIDs, making it challenging to fast update or delete ads.

3 Method

In this paper, we introduce a novel end-to-end generative retrieval architecture designed for online retrieval, named **R**eal-time **A**d **re**trieval (RARE). RARE effectively shortens the link structure, which allows advertisements to overcome the limitations of keyword bidding and helps advertisers acquire more accurate traffic, as illustrated in Figure 2.

3.1 An End-to-end Generative Architecture

Upon receiving a user query, RARE first analyzes it to generate corresponding Commercial Intents (CIs)—text with specific linguistic meaning—and then utilizes these CIs to retrieve the final ads. In the following, we detail the indexing of CIs to ads and explain the retrieval process.

Indexing. RARE first generates CIs for the entire ad corpus and determines the commercial intention set, then the inverted index of CIs-Ads are built. For subsequent new ads, we perform constrained inference based on the current commercial intention set to ensure that each new candidate can be accurately updated in the index. Notably, CIs are texts with specific linguistic meanings generated by customized LLM to mine the commercial intention of ads. Further details on the implementation are discussed in Section 3.3.

Retrieval. The real-time generation of CIs for queries is based on a combination of offline caching strategies and online inference. The inferred CIs of high-frequency queries are stored in the cache. When a query arrives, RARE first checks whether the current query matches an entry in the cache. If a match is found, RARE directly retrieves the corresponding CIs to fetch ads. Otherwise, RARE uses the customized LLM with constrained beam search for real-time inference. The detailed implementations are introduced in 3.4.

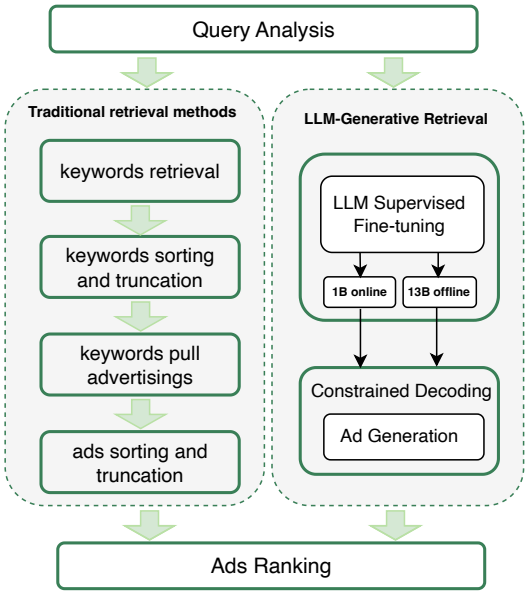

Figure 2: Comparison of RARE and Traditional Retrieval Methods. The Direct Generation of Candidate Ads from User Queries Shortens Link Structure.

The traditional query-keyword-ads architecture utilizes manually purchased keywords as search targets, subsequently retrieving ads based on a **fixed/predetermined index** linking keywords to ads. In contrast, our RARE framework uses advertisements themselves as retrieval targets and utilizes CIs as a **dynamic bridge** to index these ads, enhancing the system's flexibility and accuracy. CIs, generated by LLMs using comprehensive information from ads/queries, facilitate the generation of high-quality ad candidates and deeper user intent modeling.

3.2 Customized LLM

To enhance the LLM's understanding of commercial and advertising knowledge and to generate more accurate CIs, we performed knowledge injection into the base LLM. To achieve real-time inference, where the model directly outputs CIs based on the query without intermediate reasoning process, we conducted format fine-tuning on the LLM. Details on the customization of LLM and the data organization are present in Appendix A.

Stage 1: Knowledge Injection. This stage primarily involves injecting commercial and advertising knowledge into the base LLM. We collected knowledge from advertising systems and produced synthesized data, which were then injected into base LLM-1B and base LLM-13B models for online and offline scenarios, respectively. For the detailed information of knowledge data, please see the Table 4 of Appendix A. The knowledge injection process can be formalized as follows:

$$\theta^{'} = h(\theta, K), \tag{1}$$

where the comprehensive function h takes the LLM model parameters θ, and the advertising knowledge data K as inputs, and outputs the updated model parameters $\theta^{'}$. Subsequently, the new parameters $\theta^{'}$ are utilized to generate predictions, i.e.,

$$y = P(y|x; \theta^{'}). \tag{2}$$

Stage 2: Format Fine-Tuning. Building on the LLM enhanced with commercial knowledge, this stage focuses on refining the format of the generated CIs and increasing their diversity. The training data for format fine-tuning is obtained from real-world online data after making necessary format adjustments. For the detailed information of fine-tuning data, please see the Table 4 of Appendix A. The generation loss of format fine-tuning is shown as follows:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T_i} logp(y_{i,t}|y_{i<t}, x_i; \theta), \tag{3}$$

where fine-tuning data set is $D = (x_i, y_i)_{i=1}^{N}$, x_i is the input sequence and y_i is the target output sequence. The probability $p(\cdot)$ is the probability predicted by the model with parameters θ base on x_i and the previously generated words $y_{i<t}$.

The customization of LLM significantly enhances its ability to understand and extract the intentions behind ads and user queries. The customized LLM compresses and summarizes ads into a commercial intention space, clustering similar ads.This process enhances diversity by reducing homogeneous retrieval, leading to improved retrieval performance both online and offline.

3.3 Indexing

We use the customized LLM to generate the Commercial Intentions (CIs) of ads and then construct the inverted index of CIs-Ads.

Commercial Intentions (CIs). CIs are short texts generated by the customized LLM that describe the commercial intentions of users or ads. Given a prompt containing ad information (such as

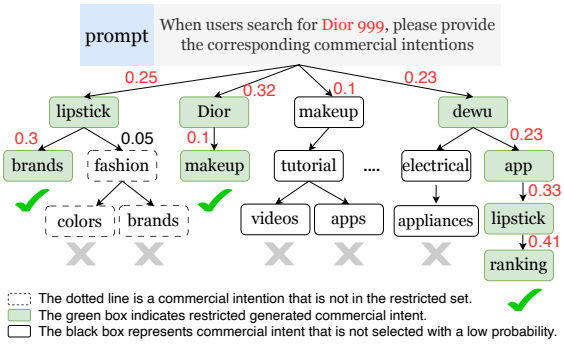

Figure 3: Constrained Beam Search Decoding Process.

ad title, landing page) or a user query, the generation of CIs is formalized as:

$$CIs = arg \max_{y^{<1>}...y^{<b>}} \sum_{t=1}^{T} logP(y_t^{<1>}, y_t^{<2>}... y_t^{<b>} | x, y_{t-1}^{<1>}, y_{t-1}^{<2>}...y_{t-1}^{<b>}), \tag{4}$$

where $y_t^{<i>}$ is the output of the top-i commercial intention at time t, b is the beam size, and T is the maximum length of the CIs.

Example. When the ad pertains to "flowers", RARE not only generates multiple business intents related to flowers—such as buying flowers online, finding a local flower shop, comparing flower prices, ordering flower delivery, and arranging flowers—but also includes intents for occasions like "Mother's Day" and "Valentine's Day". The CIs proposed in RARE can more accurately align with the traffic advertisers want to reach.

Ad Indexing Building. Initially, We use the customized LLM to generate CIs for all ads in the library, based on information such as the ads' titles, landing pages, and delivery materials. Subsequently, we perform operations such as deleting irrelevant CIs and clustering to refine the generated results, resulting in a refined set of approximately 2 million CIs. For new ads, we employ a constrained beam search technique to generate an average of 30 CIs per ad. This approach ensures that each new ad can be effectively indexed. In addition, we update the CIs set monthly to introduce new products and refine commercial intents.

Utilizing generated CIs to index ads offers numerous advatages, including accurate extraction of both ad content and user intent, as well as high efficiency and robust generalization capabilities. For new ads, our approach performs only simple inference rather than retraining the model.

3.4 Efficient inference

Efficient inference is essential for real-time retrieval from millions of candidate sets, as sponsored search advertising has strict requirements on retrieval time. In this section, we mainly introduce efficient decoding methods including constrained decoding and caching technology.

Constrained Beam Search. In this work, we employ a constrained beam search algorithm for generating commercial intentions(CIs), ensuring that the model's outputs are confined to a predefined CIs. We have developed a CUDA-based implementation of the constrained beam search and integrated it with the LLM inference process to enable parallel generation of beam-size CIs, thereby enhancing decoding efficiency. Furthermore, we introduced a truncation function within the constrained beam search framework, which allows for the discarding of individual tokens with lower scores to improve the accuracy of the model's output. The specific restriction process is illustrated in Figure 3.

Caching Technology. The search system exhibits a pronounced long-tail effect, where 5% of the queries account for 60% of the total query requests. To enhance inference efficiency, we perform offline inference and storage for these high-frequency queries. When a user submits a query, the system first checks the offline cache. If a match is found, the result is returned immediately. If no match is found, the inference service processes the request.

Offline processing is less time-sensitive, allowing us to utilize a large model—a 13B LLM—to handle these queries. Online inference has stringent time constraints, typically requiring completion within milliseconds, so a small size 1B model is used. By caching millions of head queries offline, we can reduce online machine consumption by 70%, which not only decreases the time required for inference but also enhances the quality of CIs for head queries.

4 Experiments

In this section, we primarily introduce our experimental settings, discuss the effect of RARE on offline metrics and online systems, and present the results of ablation studies.

Method		HR@50	HR@100	HR@500	MAP	ACR
Word-based	BM25	0.0870	0.1336	0.3807	0.1232	76.01%
Semantic-based	Bert-small	0.0995	0.1518	0.4311	0.1719	78.65%
	Bert-base	**0.1038**	0.1511	0.4714	0.1739	80.50%
	SimBert-v2-R	0.0978	0.1428	0.3419	0.1797	81.07%
Generative Retrieval	SimBert-v2-G	0.0572	0.0792	0.1026	0.1405	43.27%
	T5	0.0265	0.0447	0.1130	0.1036	83.31%
LLM-based Generative Retrieval	Qwen-1.8B	0.0527	0.0986	0.4099	0.1168	96.13%
	Hunyuan-2B	0.0491	0.0937	0.3904	0.1038	96.09%
	DSI	0.0258	0.0480	0.1764	0.0745	**96.15%**
	Substr	0.0225	0.0341	0.0744	0.1042	96.15%
Ours	**RARE**	0.0985	**0.1541**	**0.5134**	**0.1845**	95.05%

Table 1: Comparison of RARE and Baseline Models in Offline Scenarios.

Real-Time Online Scenarios	Consumption	GMV	CTR	Shallow Conversions	Deep Conversions
WTS(anonymous)	+5.04%	+6.37%	+1.28%	+5.29%	+24.77%
DSP(anonymous)	+7.18%	+5.03%	-	+6.85%	+5.93%
QBS(anonymous)	+4.50%	+5.02%	-0.74%	+17.07%	+7.86%

Table 2: Application of RARE to Real-World Search Systems. Results of Online A/B Testing.

4.1 Experimental Settings

Training Dataset. To facilitate knowledge injection into the vanilla LLM, we utilized commercial knowledge and synthetic data. The raw data was derived from real online logs and was processed to generate the final synthetic data by having open-source LLMs perform tasks such as query intent mining and ad intent mining. Format fine-tuning primarily involves the CIs of queries and advertisements. These data are sourced from real online interactions and are combined according to fixed rules. For details, please refer to Table 4.

Evaluation Dataset. To evaluate the model's effectiveness, we collected pairs of head queries and corresponding clicked ads online over the course of one day in the real-world scenario. After cleaning the data, we obtained 5,000 queries and 150,000 ads to serve as the ground truth, with each query having a maximum of 1,000 ad candidates.

Baselines. We compare RARE with 10 competitive baselines across 4 major categories, including word-based BM25, semantic-based BERT, generative-based T5 and LLM-based Qwen, etc.

BM25 segment the query to be calculated into w_1, w_2, ..., w_n, and then calculate the relevance score of each w_i and the keyword. Finally, these scores are accumulated to finally get the text similarity calculation result.

BERT-small employs a 4-layer transformer network with a hidden layer size of 768 and the number of parameters is approximately 52.14M. BERT-base utilizes a 12-layer transformer network with a hidden layer size of 768 and 12 heads. The number of parameters is about 110M. We used online clicked data as positive examples, and randomly sampled within the batch as negative examples. We trained the BERT using contrastive learning techniques. The trained BERT was used to obtain the embeddings of the query and keywords and then use HNSW (Malkov and Yashunin, 2018) to retrieve candidate keywords for the query. SimbBert-v2-R (Su, 2021) is a model that integrates both generation and retrieval capabilities. It serves as a robust baseline for sentence vectors and can also be utilized for automatic text generation. In our work, we reproduced the Simbert-v2-base[1] model and trained it on millions of online click query-keyword pairs. This resulted in two specialized

[1]https://github.com/ZhuiyiTechnology/roformer-sim

versions: Simbert-v2-G, designed for keyword generation, and Simbert-v2-R, intended for calculating keyword sentence vectors. We also reproduced T5-base[2], a strong baseline for generative recall, and fine-tuned it on a large number of online click queries and keywords.

DSI is a typical method that employs semantic ID-based retrieval. Initially, we fine-tune the HunYuan-1B model to learn the correspondence between ads and their respective IDs. Subsequently, we feed the query along with the IDs corresponding to the clicked ads into the HunYuan model for further training. Qwen-1.8B and Hunyuan-2B are models with the same scale of parameters as RARE. We incorporated format fine-tuning into the Qwen 1.8B and Hunyuan-2B models to ensure that they generate outputs exclusively focused on CIs, without including any additional information. A LLM without constrained decoding may generate CI without corresponding advertisements. To address this issue, we employ HNSW retrieval to find the most similar CI within the library of CIs, using it as the final result for the CI generated by the LLM.

Evaluation Metrics. We use ACR (Fan et al., 2019), Hit Ratio (HR@K) (Alsini et al., 2020) and Mean Average Precision (MAP) (Cormack and Lynam, 2006) to evaluate the effectiveness of RARE.

Ad Coverage Rate (ACR) in ad retrieval means coverage, which is the proportion of requests with ad recall. As shown in Formula 5, Ad Pave View (AdPV) is the number of requests with ad recall, and Pave View (PV) is the number of requests.

$$ACR = AdPV/PV \tag{5}$$

Hit Ratio (HR@K) is shown in Formula 6, where Ground Truth (GT) represents set of candidate ads, and Hits@K represents the number of relevant ads within the top-K retrieved candidates that belong to the ground truth set.

$$HR@K = \frac{Hits@K}{|GT|} \tag{6}$$

Mean Average Precision(MAP) is the average of Average Precision (AP) of all queries (Q), as shown in formula 7.

$$MAP = \frac{\sum_{q \in Q} AP_q}{Q} \tag{7}$$

[2]https://github.com/bojone/t5_in_bert4keras

Average Precision (AP) is shown in formula 8, where Ω_q represents the ground-truth results, p_{qj} represents the position of ad_j in the generated list, and $p_{qj} < p_{qi}$ means that ad_j ranks before ad_i in the generated list.

$$AP_q = \frac{1}{\Omega_q} \sum_{i \in \Omega_q} \frac{\sum_{j \in \Omega_q} h(p_{qj} < p_{qi}) + 1}{p_{qi}} \tag{8}$$

Implementation Details. We utilize an anonymous base LLM as the backbone, with parameter sizes of 1B and 13B. For the offline cache, we employ a 13B model with a beam size of 256, a temperature of 0.8, and a maximum output length of 6. For online inference, we use a 1B model with a beam size of 50, a temperature of 0.7, and a maximum output length of 4 to ensure that inference latency remains within 60 milliseconds. RARE will assign appropriate CIs to newly added advertisements and products within the existing CI set, and will update the CIs-Ads index on an hourly basis. The entire CIs set is updated monthly, allowing new products to receive more fine-grained and accurate CIs. Additionally, we periodically inject new commercial information into the LLM, such as new brand names and product details, to ensure its knowledge remains up to date.

4.2 Experimental Results

Offline Evaluation. We compared RARE with 10 retrieval methods in 4 categories on the industrial evaluation dataset. Results are shown in Table 1. The RARE model excels in HR@500 and MAP while maintaining a high ACR, demonstrating its ability to understand user search intent and optimize ad delivery. Notably, it achieves a ACR exceeding 90%, and its high HR@500 metric confirms its strong capacity to retrieve commercially valuable ads. This synergy indicates the model's success in balancing user intent comprehension with commercial value-driven ad retrieval.

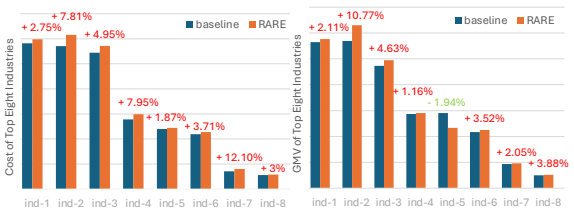

Figure 4: RARE Outperforms Online Benchmark Models Across Major Real-World Industries.

Method	HR@500	MAP	ACR	Avg CIs	Accuracy
w/o. KI	0.1706	0.1540	59.51%	22.78	90.4%
w/o. CBS	0.1868	0.1687	67.12%	4.84	95.2%
w/o. CBS & KI	0.1562	0.1592	48.28%	9.09	94.5%
RARE	**0.5134**	**0.1845**	**95.05%**	**74.49**	**96.5%**

Table 3: Ablation Studies on RARE.

Online A/B Testing. We apply RARE to three different online retrieval scenarios (with billions of daily requests): WTS, DSP and QBS. During a one-month A/B testing experiment with a 20% user sample, we observed significant benefits across multiple scenarios, including increased system revenue, enhanced user experience, and boosted advertiser conversions. Take WTS as an example, we achieved a 5.04% increase in consumption (cost), a 6.37% increase in GMV, a 1.28% increase in CTR and a 5.29% increase in shallow conversions. Significant improvements of CTR and conversions demonstrate that RARE can effectively understand user intent and deliver high-quality ads. The evaluation of RARE across eight popular real-world industries, as shown in Figure 4, further demonstrates its effectiveness in various scenarios.

Online Inference Support. We analyzed tens of thousands of data points to determine the average time consumption for various output lengths during the real-time online inference process, and the results are shown in Figure 5. Our CIs have an average token count of 3, ensuring that online real-time inference meets safety thresholds. To facilitate online inference, we developed a specialized GPU cluster with hundreds of L40, achieving effective load balancing and peak GPU utilization rates up to 90%. We quantized the well-trained model to FP8 precision, enabling each L40 to handle about 30 Queries Per Second. Efficient caching techniques increased the cache hit rate to approximately 65% for head queries. These supports enhanced generation quality while reducing computational costs notably.

Ablation Study. We conducted two types of ablation studies to investigate the contribution of each component. First, table 3 displays the results of RARE on ad retrieval under various settings. w/o. KI refers to RARE without knowledge injection. Its recall rate is only 59.51%, significantly lower than RARE's 95.05%. This demonstrates that without knowledge injection, the LLM struggles to understand intents of user queries and ads. w/o. CBS refers to RARE without constrained beam search. Its average number of CIs is only 4.84, significantly lower than RARE's 74.49. This indicates that constrained beam search can substantially increase the diversity of commercial intents generated by the LLM. w/o.CBS & KI refers to RARE without both constrained beam search and knowledge injection. It is evident that its HR@500, MAP, and ACR metrics are the lowest among the compared methods. Second, table 5 in Appendix C presents a qualitative analysis of each component's contribution to RARE through a case study.

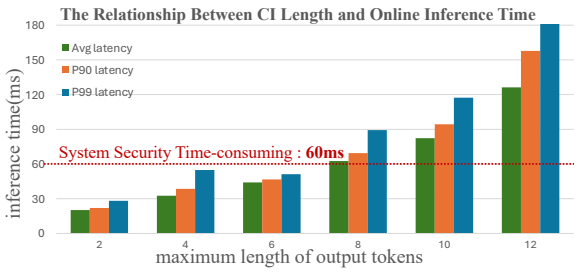

Figure 5: Time Consumption for Different Lengths.

5 Conclusion

In this paper, we propose a LLM-generative **R**eal-time **A**d **RE**trieval called RARE. This framework utilizes commercial intentions (CIs) as semantic representation that retrieve ads directly for querys. To mine deeper intentions of ads and users, we inject commercial knowledge and conduct format fine-tuning on vanilla LLM to obtain the customized LLM. Besides, we employs constrained decoding, which allows the model to generate CIs from a fixed set in parallel. The proposed architecture enables real-time generation and retrieval from a library containing tens of millions of ads. Evaluations on offline data and online A/B testing indicate that our architecture achieves state-of-the-art (SOTA) advertising retrieval performance, while substantially improving search system revenue, user experience and advertiser conversion.

Limitations

We briefly outline limitations of our work. The end-to-end generation architecture proposed in this paper primarily facilitates the generation process from query/ad to commercial intention, while the correlation between query and ad is managed by downstream processes. In future work, we aim to integrate correlation assessment into LLMs, thereby empowering the model to evaluate the pertinence between prompts and commercial intents concurrently with the generation phase. We anticipate that this integration of generative and discriminative capabilities will significantly augment the efficacy of the generation process.

References

Areej Alsini, Du Q Huynh, and Amitava Datta. 2020. Hit ratio: An evaluation metric for hashtag recommendation. *arXiv preprint arXiv:2010.01258*.

Michele Bevilacqua, Giuseppe Ottaviano, Patrick Lewis, Scott Yih, Sebastian Riedel, and Fabio Petroni. 2022. Autoregressive search engines: Generating substrings as document identifiers. *Advances in Neural Information Processing Systems*, 35:31668–31683.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Gordon V Cormack and Thomas R Lynam. 2006. Statistical precision of information retrieval evaluation. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 533–540.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Miao Fan, Jiacheng Guo, Shuai Zhu, Shuo Miao, Mingming Sun, and Ping Li. 2019. Mobius: towards the next generation of query-ad matching in baidu's sponsored search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2509–2517.

Paolo Ferragina and Giovanni Manzini. 2000. Opportunistic data structures with applications. In *Proceedings 41st annual symposium on foundations of computer science*, pages 390–398. IEEE.

Weihao Gao, Xiangjun Fan, Jiankai Sun, Kai Jia, Wenzhi Xiao, Chong Wang, and Xiaobing Liu. 2020. Deep retrieval: An end-to-end learnable structure model for large-scale recommendations. *arXiv preprint arXiv:2007.07203*.

Zhen Gong, Xin Wu, Lei Chen, Zhenzhe Zheng, Shengjie Wang, Anran Xu, Chong Wang, and Fan Wu. 2023. Full index deep retrieval: End-to-end user and item structures for cold-start and long-tail item recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 47–57.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2024a. From matching to generation: A survey on generative information retrieval. *arXiv preprint arXiv:2404.14851*.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2023. Multiview identifiers enhanced generative retrieval. *arXiv preprint arXiv:2305.16675*.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024b. Learning to rank in generative retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8716–8723.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, and 1 others. 2025. How can recommender systems benefit from large language models: A survey. *ACM Transactions on Information Systems*, 43(2):1–47.

Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 365–374.

Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, and 1 others. 2022. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, and 1 others. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Ronak Pradeep, Kai Hui, Jai Gupta, Adam D Lelkes, Honglei Zhuang, Jimmy Lin, Donald Metzler, and Vinh Q Tran. 2023. How does generative retrieval

scale to millions of passages? In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2024. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36.

Juan Ramos and 1 others. 2003. *Using tf-idf to determine word relevance in document queries*, volume 242.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Xiang Shi, Jiawei Liu, Yinpeng Liu, Qikai Cheng, and Wei Lu. 2025. Know where to go: Make llm a relevant, responsible, and trustworthy searchers. *Decision Support Systems*, 188:114354.

Jinlin Su. 2021. Simbertv2 is here! roformer-sim model that combines retrieval and generation. Accessed: 2021-06-11.

Weiwei Sun, Lingyong Yan, Zheng Chen, Shuaiqiang Wang, Haichao Zhu, Pengjie Ren, Zhumin Chen, Dawei Yin, Maarten Rijke, and Zhaochun Ren. 2024. Learning to tokenize for generative retrieval. *Advances in Neural Information Processing Systems*, 36.

Qiaoyu Tang, Jiawei Chen, Bowen Yu, Yaojie Lu, Cheng Fu, Haiyang Yu, Hongyu Lin, Fei Huang, Ben He, Xianpei Han, and 1 others. 2024a. Self-retrieval: Building an information retrieval system with one large language model. *arXiv preprint arXiv:2403.00801*.

Yubao Tang, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, and Xueqi Cheng. 2024b. List-wise generative retrieval models via a sequential learning process. *ACM Transactions on Information Systems*, 42(5):1–31.

Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, and 1 others. 2022. Transformer memory as a differentiable search index. *Advances in Neural Information Processing Systems*, 35:21831–21843.

Yang Wang, Zheyi Sha, Kunhai Lin, Chaobing Feng, Kunhong Zhu, Lipeng Wang, Xuewu Jiao, Fei Huang, Chao Ye, Dengwu He, and 1 others. 2024a. One-step reach: Llm-based keyword generation for sponsored search advertising. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1604–1608.

Yidan Wang, Zhaochun Ren, Weiwei Sun, Jiyuan Yang, Zhixiang Liang, Xin Chen, Ruobing Xie, Su Yan, Xu Zhang, Pengjie Ren, and 1 others. 2024b. Enhanced generative recommendation via content and collaboration integration. *arXiv preprint arXiv:2403.18480*.

Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, and 1 others. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems*, 35:25600–25614.

Yunli Wang, Zixuan Yang, Zhen Zhang, Zhiqiang Wang, Jian Yang, Shiyang Wen, Peng Jiang, and Kun Gai. 2024c. Scaling laws for online advertisement retrieval. *arXiv preprint arXiv:2411.13322*.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on web search and data mining*, pages 1154–1156.

Hansi Zeng, Chen Luo, Bowen Jin, Sheikh Muhammad Sarwar, Tianxin Wei, and Hamed Zamani. 2023. Scalable and effective generative information retrieval. corr abs/2311.09134 (2023).

Chao Zhang, Shiwei Wu, Haoxin Zhang, Tong Xu, Yan Gao, Yao Hu, and Enhong Chen. 2024. Notellm: A retrievable large language model for note recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 170–179.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yu Zhao and Fang Liu. 2024. A survey of retrieval algorithms in ad and content recommendation systems. *arXiv preprint arXiv:2407.01712*.

A Fine-tuning Data

In this section, we mainly introduce the details of fine-tuning data. Fine-tuning of customized LLM mainly includes two stages, namely knowledge injection and format fine-tuning. The fine-tuning data of knowledge injection mainly includes query intent mining, advertising intent mining and advertising words buying. We input prompts containing advertisement and user information into open-source LLMs (e.g., ChatGPT) to obtain outputs that include rich reasoning processes and guidance information, which are then injected into the base LLM as knowledge data. Injecting a large amount of data during the fine-tuning phase can cause LLMs to lose their general knowledge and reasoning capabilities. Therefore, in this phase of fine-tuning, we selected only 2,000 instances for each task. Table 4 shows the fine-tuning data of these two stages in detail.

B Related Work

Beam Search. As a decoding strategy for heuristic search, beam search has been widely used in many works. For example, DSI uses beam search to generate a sorted list of candidate documents, and Tiger uses beam search to generate multiple candidate product IDs at once. As early as a few years ago, the combination of seq2seq and constrained Beam Search has achieved a win-win effect and efficiency in entity linking and document retrieval. For example, GENRE (De Cao et al., 2020) applied constrained Beam Search to document retrieval tasks and achieved SOTA.

Query-kwds-ads Architecture. Traditional query-kwds-ads approaches suffer from two critical drawbacks: (1) Keywords are manually selected by advertisers, resulting in varying quality and potential issues of either being too broad or too narrow, leading to inefficient traffic matching. (2) Advertisers often purchase a large number of keywords, which hampers the efficiency of ad retrieval after keyword inversion, imposing a significant burden on the system. In contrast, CIs are generated by a domain knowledge-injected LLM, enabling them to better represent the intentions of advertisers and achieve more precise matching with relevant traffic. This not only brings economic benefits but also ensures the long-term healthy operation of the system.

Encoder-based LLMs Retriever Encoder-based retrievers leverage the semantic capabilities of LLMs to obtain text embedding (Hou et al., 2024). For instance, cpt-text (Neelakantan et al., 2022) uses contrastive learning to train GPT-3 (Brown, 2020) from scratch, generating high-quality embedding of text. GTR (Ni et al., 2021) utilizes the T5 (Raffel et al., 2020) model, fine-tuning it to derive text vector representations. NoteLLM (Zhang et al., 2024) achieves text-image embeddings by incorporating additional training tasks and modifying the LLM model structure.

C Qualitative Analysis

Table 5 provides an intuitive example to analyze the role of each component in RARE. we can observe the following: (1) Zero-shot LLM lacks a reasoning process for prompts, relying mainly on the surface-level understanding of queries, which results in numerous poor cases. (2) Knowledge injection stage teaches LLM how to reason, enabling it to analyze the query in the first step, provide commercial intentions in the second step, and judge relevance in the third step. However, this approach lacks diversity. (3) Format fine-tuning stage instructs the model to follow rules while generating more diverse results. Since most of the training data in this stage comes from online data, the commercial intentions are relatively rich. However, due to the free generation process, the number of commercial intentions is limited. (4) Incorporating constrained decoding can increase the number of commercial intentions while ensuring relevance and diversity.

Task Name	Prompt	Output	Data Size
Knowledge Injection Task1. query intent mining	As a search advertising system, given the user query = xxx, identify the commercial intentions based on the user's search characteristics. Provide ten commercial intentions for each category from different intents, and finally, compile all the results into a single line separated by commas, step by step.	**Output from open source LLM models (e.g. ChatGPT):**Sure, let's break down the process step by step to identify commercial intentions based on the user query "xxx". *Step 1: Identify Categories... Step 2: Identify Commercial Intentions for Each Category... Step 3:Compile All Results into a Single Line...*	2000
Knowledge Injection Task2. ad intent mining	As an advertiser, given the ad title = xxx, identify the commercial intentions for ad based on search characteristics. Provide ten commercial intentions for each category from different intents, and finally, compile all the results into a single line separated by commas, step by step.	**Output from open source LLM models (e.g. ChatGPT):** Sure, let's break down the process step by step to identify commercial intentions based on the ad title "xxx" *Step 1: Identify Categories... Step 2: Identify Commercial Intentions for Each Category... Step 3: Compile All Results into a Single Line....*	2000
Knowledge Injection Task3. ad generation	As a search advertising system, your task is to recommend suitable ads for user queries. To meet the user's potential intent and ensure the relevance between the query and the ad title, you should provide a diverse range of ads. Given the query = xxx, please generate ten appropriate ad titles for each intent based on search characteristics, step by step.	**Output from open source LLM models (e.g. ChatGPT):** Sure, let's break down the process step by step to generate ten appropriate ad titles for each intent based on the user query "xxx". *Step 1: Identify Intents... Step 2: Generate Ad Titles for Each Intent...*	2000
Knowledge Injection Task4. query understanding	As a search advertising system, your task is to rewrite and expand the user's query to ensure it meets the diverse needs of users. The rewritten terms should be as varied as possible. For instance, if the query is "SF Express," the expanded terms could include "cross-border logistics", "express shipping", "express delivery", "express service" and "logistics warehousing". Following this example, when the query is = xxx, provide the expanded terms that meet these requirements, step by step.	**Output from open source LLM models (e.g. ChatGPT):** Sure, let's break down the process step by step to rewrite and expand the user's query "xxx" to ensure it meets the diverse needs of users. The expanded terms should cover a wide range of related topics and intents. *Step 1: Identify Core Concepts... Step 2: Identify Related Categories and Intents... Step 3: Generate Expanded Terms for Each Category... Step 4: Compile All Expanded Terms...*	2000
Format Fine-tuning Task1. query generates CIs	As a search advertising system, given a user query of xxx, please provide a broader range of commercial intentions that still align with the user's potential intentions.	**From real world online data:** commercial intention 1;commercial intention 2;commercial intention 3.....	2000
Format Fine-tuning Task2. ad generates CIs	As a search advertiser, given the ad title = xxx, please identify the commercial intentions of the ad based on the characteristics of the search and various user intentions.	**From real world online data:** commercial intention 1;commercial intention 2;commercial intention 3.....	2000

Table 4: Details of fine-tuning data for customized LLM.

prompt: When users search for **shantytown housing**, please provide the corresponding commercial intentions.
vanilla llm: (A total of 26 commercial intentions, but some of them are poorly relevant) commercial intentions: shantytown, demolition, housing supply, price, user feedback, case sharing..., etc.
w/o. CBS&FFT (A total of 9 commercial intentions, with good correlation and insufficient diversity.) Step 1: Mining user query intent Shantytown reconstruction is a government-led urban renewal project involving demolition.... Step 2: Generate commercial intentions Shantytown reconstruction policy, demolition compensation standards..... Step3: Ensure commercial intentions are relevant to the query commercial intentions such as shantytown reconstruction policy, Demolition compensation standards... Final result: shantytown reconstruction policy, demolition compensation standards.....
w/o. CBS (A total of 12 commercial intentions, with good correlation and diversity, but a small number) Anjuke rents a house, second-hand housing, house hunting...
RARE (A total of with 142 commercial intentions, with with good correlation, good diversity and large number) Anjuke house hunting, demolition compensation, new house decoration, renovation of old houses, public housing application.....

Table 5: Commercial intention generation effects based on different fine-tuning methods.