# VALUE: Value-Aware Large Language Model for Query Rewriting via Weighted Trie in Sponsored Search


Boyang Zuo[†,*]
zuoby22@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Xiao Zhang[†]
zx142853@alibaba-inc.com
Taobao and Tmall Group
Beijing, China

Feng Li
adam.lf@alibaba-inc.com
Taobao and Tmall Group
Beijing, China

Pengjie Wang
pengjie.wpj@alibaba-inc.com
Taobao and Tmall Group
Beijing, China

Jian Xu
xiyu.xj@alibaba-inc.com
Taobao and Tmall Group
Beijing, China

Bo Zheng
bozheng@alibaba-inc.com
Taobao and Tmall Group
Beijing, China



## Abstract

In the realm of sponsored search advertising, matching advertisements with the search intent of a user's query is crucial. Query-to-bidwords(i.e. bidding keywords) rewriting, which involves transforming user queries into keywords for bidding, is a vital technique that has garnered significant attention from both industry and academia. Recently, with the prevalence of large language models (LLMs), generative retrieval methods have proven effective in producing high-relevance rewrites. However, we have identified a significant limitation in existing approaches: While fine-tuning LLMs for specific domains enhances semantic relevance, these models have no perception of the intrinsic value of their generated outputs, such as commercial value. Therefore, after supervised fine-tuning (SFT), a reinforcement learning from human feedback (RLHF) phase is often employed to address this issue. Nevertheless, traditional preference alignment methods often face challenges in aligning fine-grained values and are susceptible to overfitting, which diminishes the effectiveness and quality of the generated results. To address these challenges, we propose **VALUE** (**V**alue-**A**ware **L**arge language model for q**U**ery rewriting via w**E**ighted trie), the first framework that ensures the generation of high-value and highly relevant bidwords. Our approach utilizes weighted trie, an innovative modification of the traditional trie data structure. By modulating the LLM's output probability distribution with value information from the trie during decoding process, we constrain the generation space and guide the trajectory of text production. Our method not only addresses fine-grained value alignment but also effectively reduces the hallucination issues often encountered with LLMs. Offline experiments demonstrate the effectiveness of our method in semantic matching and preference alignment, showing a remarkable improvement in the value attribute by more than five-fold. Online A/B tests further revealed that our Revenue Per Mille (RPM) metric increased by 1.64%. VALUE has been deployed on our advertising system since October 2024 and served the Double Eleven promotions, the biggest shopping carnival in China.


[†]These authors contributed equally to this work and should be considered co-first authors.
[*]This work was done when the first author was an intern at Alimama.


## CCS Concepts

• **Information systems → Query reformulation**; **Top-k retrieval in databases**; • **Computing methodologies → Natural language processing**.

## Keywords

Query Rewrite; Semantic Matching; Generative Retrieval; Large Language Model;


## 1 Introduction

Sponsored search is a crucial element of modern search engines, where advertisers bid on relevant bidwords to display their ads alongside original search results. Upon receiving user's query, it is essential to match the query with relevant bidwords [18], and subsequently identify suitable ads associated with those bidwords. This process is a critical component in the ad retrieval pipeline, driving the major revenue of search engines [23]. The success of sponsored search depends on the search engine's ability to accurately identify bidwords that closely align with user intent. However, users often face challenges when formulating queries, leading to suboptimal search experiences. These challenges are intensified when users do not use precise or correct terminology, employ synonyms, or mix languages in their search phrases due to varying levels of language proficiency. Additionally, search terms might be misspelled or overly general, complicating the retrieval of relevant ads. For example, a user might search for "spring fashion," which is broad and could correspond to bidwords like "2024 spring collection" or "women's spring trends". Each query reflects different intents but lacks clarity without additional context. As e-commerce platforms continue to grow in both scale and diversity, ensuring accurate and relevant ad retrieval becomes increasingly challenging, necessitating the need for advanced query rewriting and bidword matching techniques to better align user queries with advertiser bidwords.

In query rewriting, two primary paradigms exist: discriminative methods, which reformulate queries by the retrieval of similar terms and generative methods, which utilize language models for direct query-to-bidword transformation. Discriminative methods leverage sparse and dense retrieval techniques to find relevant bidwords. Sparse retrieval techniques, such as BM25 [36], employ exact lexical matching through high-dimensional term-frequency vectors. Their reliance on exact lexical matching leads to semantic

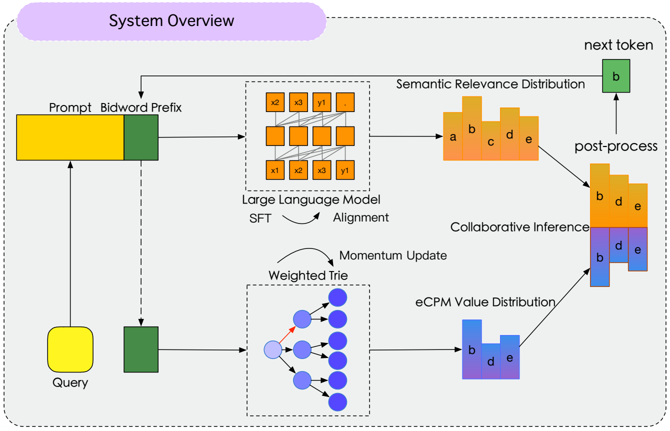


**Figure 1: System Overview of our proposed VALUE Framework includes three parts: "SFT and Alignment", "Weighted Trie Construction and Weight Momentum Update", and "Collaborative Inference". During the LLM decoding process, we derive a value distribution from a weighted trie to constrain the generation space, and subsequently merge this with another distribution to produce the final output distribution.**

brittleness. This rigidity prevents them from recognizing semantic equivalences between different expressions, resulting in significant retrieval failures when faced with semantically similar but lexically divergent queries. Dense retrieval approaches overcome the limitations of sparse methods by utilizing low-dimensional semantic embeddings generated by neural encoders like sBERT [35]. Dense retrieval, while effective in capturing semantic relationships, suffers from significant limitations, particularly the Matthew Effect and long-tail challenges. High-frequency queries dominate model training due to abundant click logs, leading to over-optimized embeddings for head queries while neglecting tail queries with sparse data. Furthermore, dense retrieval models struggle with distribution shifts and require costly strategies like meta-learning or data augmentation to address long-tail issues [32].

Recently, the rise of generative retrieval has garnered widespread attention. Wang [42] utilizes LLMs for one-step query-to-keyword rewriting, using a trie during the decoding process. Despite improvements over traditional information retrieval (IR) baselines, existing generative approaches using LLMs for query-to-bidword rewriting tasks have notable limitations. A key limitation is their inability to recognize the value of the generated results, which, in our scenario, pertains to the outcomes they can deliver. Models trained solely through supervised fine-tuning on online logs lack awareness of the value of the results. Due to the training methodology and the SFT loss function design, the model treats each query-bidword pair in the training samples equally, failing to learn any preference or value information. Although there is a great deal of work on Reinforcement Learning from Human Feedback (RLHF), such as PPO, DPO, KTO, and ORPO[17, 22, 31, 34], which can partially mitigate the value perception issues of SFT models, these methods also have significant drawbacks. Firstly, they require substantial computational resources for training. However, the effective Cost Per Mille (eCPM) value of bidwords is a statistical measure that frequently fluctuates. When these fluctuations occur, the previous alignment efforts become obsolete, and the model needs to be retrained. Secondly, these training methods are prone to overfitting [29], especially in our scenario. Unlike traditional alignment objectives for LLMs, such as harmlessness and usefulness, which can be achieved using a small dataset, the value of a bidword is completely independent of its textual content. Therefore, the model must be trained individually for each bidword. Our experiment on DPO show that models trained in this manner tend to overfitting. There is an urgent need for a new paradigm to address the challenges above.

Through our observations, we have identified an opportunity in the sponsored search advertising scenario to enhance the value of bidwords without compromising their relevance. The value of a bidword does not necessarily correlate with its literal meaning. Two bidwords might differ by only a symbol or character, yet exhibit substantial differences in value. Therefore, there is an opportunity to increase the value of the top $K$ bidwords while minimally affecting relevance by prioritizing those with the same level of relevance but higher value over others. Thereby, we present VALUE, an innovative reward-guided inference framework that integrates value-awareness into the decoding process via a novel weighted trie structure. We design algorithms for the construction and update of the value of trie node. During decoding process, VALUE obtain reward information from for the next tokens the weighted trie. This information is used to adjust the output distribution, thereby sampling the subsequent token. By this reward-guided search, the framework aims to maximize the reward of the outputs. We used a multi-task fine-tuning strategy to equip the model with essential query rewriting skills. To achieve better value alignment, we designed an optional post-training alignment phase. To evaluate the effectiveness of our proposed method, we conducted offline experiments using historical online logs. Compared to the current online service model, our method demonstrated a **9.3%** improvement in **relevance**. Additionally, online A/B tests showed increases in key profit metrics, with *Cost* rising by **1.53%** and *Revenue Per Mille (RPM)* increasing by **1.64%**. Given these significant improvements, our method has been fully deployed in the online service. Our contributions can be summarized as follows:

1. **Pioneering Query Rewriting Framework via Collaborative Inference:** We present a groundbreaking query rewriting framework that integrates LLMs with weighted trie, uniquely designed to simultaneously optimize for both high value and semantic relevance. This framework represents the first of its kind to effectively balance these two critical aspects.
2. **Renovation of Trie:** We have reinvented the trie data structure, by seamlessly integrating value information directly into its nodes. This advancement is further supported by our proposed algorithms for trie construction and momentum update of node value.
3. **Post-Training Alignment for Fine-Grained Reward Optimization:** We propose an post-training alignment algorithm that fine-tunes the model to better align with fine-grained reward attributes.

## 2 Related Works

### 2.1 Query Rewriting

The process of query rewriting, often termed query expansion or reformulation, is essential in enhancing e-commerce search technologies. It profoundly affects the shopping experience of users and the financial performance of e-commerce platforms. This approach can be generally classified into two classes: discriminative methods and generative methods.

**Discriminative methods** approach query rewriting as a retrieval task. Pseudo-relevance feedback methods [7, 39, 45, 46] enhance queries by identifying expansion terms from the top-ranked documents of an initial query, combining global corpus analysis with local feedback. Although these methods effectively address word mismatches, they are susceptible to semantic drift due to noisy or irrelevant top results. To address these challenges, researchers suggest utilizing a well-constructed thesaurus as a candidate set for query rewriting. [5, 28] However, they caution that the effectiveness of these methods is highly dependent on the thesaurus's quality. If the thesaurus is inadequate, it may lead to query semantic drift, where the intended meaning of the query is compromised. Additionally, some approaches [3, 13, 27] focus on generating candidate rewrites by leveraging search logs, using similar terms from users' search histories as extensions. However, these search logs inherently display a bias towards popular queries due to the Matthew effect. Consequently, the training data derived from this method may fall short in optimizing for less frequently searched long-tail queries.

**Generative methods**, on the other hand, focus on the direct transformation of queries into keywords [25, 30, 33, 42]. These methods involve transforming the input query directly into multiple bidwords, utilizing auto-regressive language models combined with trie-based decoding techniques to restrict the output space during inference. However, while previous methods ensure semantic relevance between input and output, they fall short in addressing value—a critical attribute in sponsored search advertising—which remains an open challenge. Our approach, as detailed in the following sections, effectively tackles this challenge by introducing innovative techniques that not only ensure semantic relevance but also optimize for value. By addressing this critical gap, our research has the potential to revolutionize sponsored search advertising, leading to more effective ad targeting and increased user satisfaction.

### 2.2 Preference Alignment

Reinforcement Learning from Human Feedback (RLHF) represents a pivotal technique for aligning LLMs with human preferences and values [4, 11, 31], commonly applying the Bradley-Terry model [6] to estimate the probability of a pairwise competition between two independently evaluated instances. The conventional RLHF pipeline is structured into three distinct phases: supervised fine-tuning [12, 15, 40, 47], reward model training [9, 19, 21, 26], and policy optimization [2, 37]. Notably, Proximal Policy Optimization (PPO) [37] is extensively employed during the policy optimization phase. The RLHF framework has found broad applicability across various domains, including the mitigation of toxicity [1, 24], the assurance of safety [14], the enhancement of helpfulness [43] and the improvement of model reasoning capabilities [20]. Recent studies [8] have underscored the challenges inherent in the entire RLHF pipeline, spanning from the collection of preference data to the training of models. Furthermore, subsequent research has illuminated potential biases introduced by RLHF, such as the generation of verbose outputs [16, 38, 41].

## 3 Methodology

Traditional methods, such as standard beam search or top-k sampling, prioritize candidates with higher probabilities, often neglecting their value attributes. Although these approaches are effective in numerous contexts, they may fall short in applications where the value attributes are important. VALUE addresses this limitation by incorporating a Weighted Trie that allows for value-aware token selection. At each step of the decoding process, VALUE modifies the model's output probabilities based on the value obtained from the trie, prioritizing tokens of higher value among those already highly relevant.

### 3.1 System Overview

As can be seen from Figure 1, our entire system comprises the following key modules: "LLM SFT and Alignment," "Weighted Trie Construction and Weight Momentum Update," and "Collaborative Inference." For each generated token, we obtain the probability distribution from the LLM. Concurrently, we retrieve all possible candidates under the current prefix from the weighted trie along with their respective values. By integrating these values with the probability distribution, we derive a new probability distribution and sample the next token from it.

### 3.2 Weighted Trie

*3.2.1 Definition and Construction of Weighted Trie.* Our Weighted Trie is an renovation of the traditional trie data structure. Each node in the Weighted Trie, referred to as a **WeightedTrieNode**, contains additional attributes to store weight-related information. The primary attributes of a WeightedTrieNode are:

- **children**: A dictionary where the keys are integers representing tokens, and the values are child nodes of type *WeightedTrieNode*.
- **mean**: A floating-point number representing the average weight of the node's children.
- **max**: A floating-point number representing the maximum weight among the node's children.
- **is_word**: A boolean value indicating whether the node represents a complete word.

**Construction of Weighted Trie:** the construction of the Weighted Trie involves initializing the root node, inserting bidwords with their corresponding values, and updating the weights of the nodes. The pseudocode in Algorithm 1 demonstrates the construction process.

*3.2.2 Weighted Trie Momentum Update.* Momentum updates are essential due to the frequent changes in eCPM and the evolving bidword space. The process is delineated as follows:

*Tokenization and Node Insertion:* Each bidword undergoes tokenization, converting it into a sequence of token IDs. These IDs

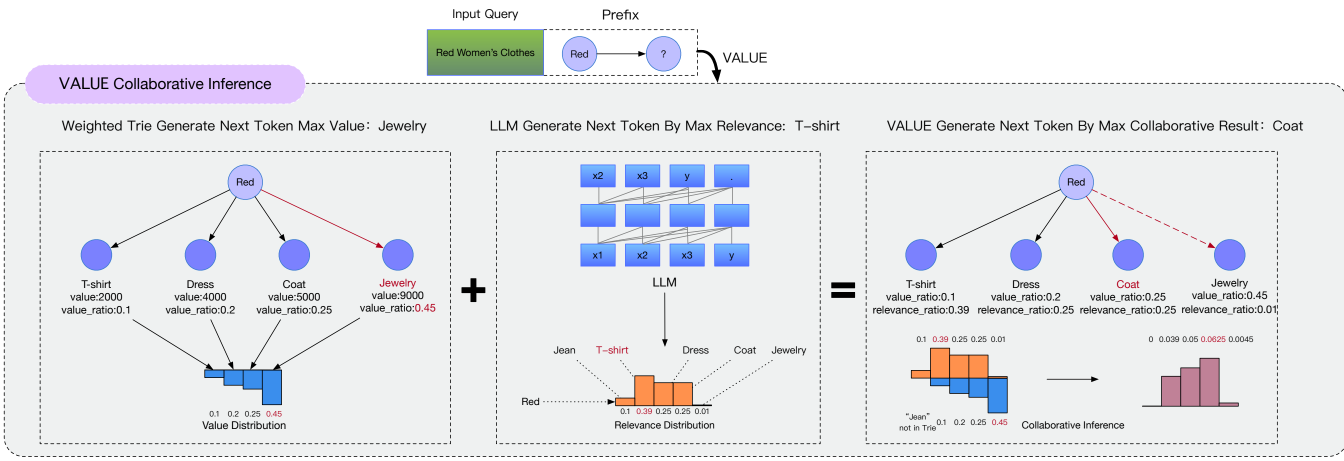


**Figure 2: Collaborative Inference of VALUE framework. When generating the next token, we obtain two output distribution from the LLM and weighted trie. We exclude tokens not present in the value distribution and adjust the probabilities of the remaining tokens. Then, we sample the next token from the modified distribution.**

---

**Algorithm 1** Construct Weighted Trie

---

**Input:** List of pairs $(bidword, ecpm)$
**Output:** Weighted Trie
Initialize weighted trie: $weighted\_trie$
**for** each $(bidword, ecpm)$ **in** list **do**
  $token\_ids \leftarrow$ tokenizer$(bidword)$
  Insert $token\_ids$ into the trie:
  **for** each $token\_id$ **in** $token\_ids$ **do**
    Traverse or create node for $token\_id$
  **end for**
  Set $leaf\_node.mean \leftarrow ecpm$
  Set $leaf\_node.max \leftarrow ecpm$
**end for**
Update weights in the weighted trie using post-order traversal:
**for** each $node$ **in** post-order **do**
  $node.mean \leftarrow \frac{1}{N}\sum_{child \in children} child.mean$
  $node.max \leftarrow \max_{child \in children} child.max$
**end for**

---

facilitate the insertion or updating of nodes within the trie. The update process for a node is governed by:

$$V_{\text{new}} = \max(eCPM_{\text{new}}, V_{\text{old}}) \tag{1}$$

$$W_{\text{new}} = \alpha \cdot eCPM_{\text{new}} \ + \ \beta \cdot W_{\text{old}} \tag{2}$$

where $\alpha$ and $\beta$ are hyperparameters that control the update rate, with the constraint $\alpha + \beta = 1$. V is the max attribute of the node and W is the mean attribute of the node. If the node is absent, it is initialized as:

$$W_{\text{new}} = eCPM_{\text{new}}, \ V_{\text{new}} = eCPM_{\text{new}} \tag{3}$$

*Bottom-up Update:* Following node insertion, a bottom-up iteration is executed to update all intermediate nodes, ensuring that the entire trie structure accurately reflects the most recent eCPM values.

This method enables the trie to dynamically adjust to variations in the bidword space and eCPM values over time, preserving the framework's efficacy without necessitating frequent retraining, maintaining stability and performance in a dynamic e-commerce environment.

### 3.3 Collaborative Inference

As shown in Figure 2, when generating the $t$-th token, we insert the query into the prompt template and concatenate it with the bidword prefix to form the input of LLM to get the output probabilities. We use generated bidword prefix to retrieve all child nodes from the trie. The value of these child nodes is computed using the following formula:

$$\mathcal{V}_k = \alpha \cdot \text{mean}_k + \beta \cdot \text{max}_k \tag{4}$$

Here, $\alpha$ and $\beta$ are tunable hyperparameters, both set to 0.5, allowing us to prioritize paths with higher average values or to explore bidwords with the maximum potential value, depending on our specific needs.

We then apply the softmax function to these values to obtain the normalized value $\hat{\mathcal{V}}_k$. Next, we mask out all probabilities in the LLM-generated distribution that are not children of the current node. For the remaining probabilities, we adjust them by weighting with the normalized value $\hat{\mathcal{V}}_k$ and a depth-dependent factor $\theta$. The adjusted probability for each token $k$ is given by:

$$p(k \mid x_{<t}) = p_{\text{LLM}}(k \mid x_{<t}) \cdot (1 + \mathcal{V}_k \cdot \theta) \tag{5}$$

where $\theta$ is designed to vary with the depth of the trie, playing a crucial role in balancing relevance and value in our model's output. At shallower levels of the trie, $\theta$ is set to smaller values to minimize the influence of value on the output, thereby preventing the model from favoring high-value but less relevant options. As we delve deeper into the trie, $\theta$ increases, allowing us to prioritize higher-value candidates from a pool of already highly relevant choices.

Consequently, we derive a modified probability distribution $p(k \mid x_{<t})$, from which the next token is sampled as usual.

## 3.4 Multi-task Fine-Tuning

As no open-source LLMs are trained to perform query rewriting for the e-commerce scenario, they lack an awareness of the bidword space. Consequently, they are unable to effectively generate bidwords. To address this, we need to perform supervised fine-tuning (SFT) on these models to inject domain-specific knowledge into the model and enhancing their capability to understand and effectively rewrite e-commerce queries for improved user experience and search accuracy.

**Query Rewriting Dataset:** To construct our training dataset, we initially obtain query-bidword pairs from online logs. Specifically, when a user searches for a query $x$ on our e-commerce platform, the system logs the corresponding bidwords $Y = \{y^1, y^2, \ldots, y^n\}$ that are associated with the advertisements the user interacts with. From this list, we select the top-ranked bidword $y^1$ as the gold standard candidate to construct our initial dataset $D$ with $N$ samples:

$$D = \{(x_i, y_i^1)\}_{i=1}^{N} \quad \text{where} \quad x_i \sim p(x) \tag{6}$$

where $p(x)$ denotes the query distribution in our search engine.

We apply a relevance filter to $D$ using a representation model fine-tuned from an open-source embedding model. The relevance score $\text{rel}(x, y)$ is computed as the cosine similarity between the embeddings of $x$ and $y$:

$$\text{rel}(x, y) = \frac{\text{emb}(x) \cdot \text{emb}(y)}{\|\text{emb}(x)\|\|\text{emb}(y)\|} \tag{7}$$

We then filter the dataset $D$ to obtain $D_r$ ensuring that the relevance score exceeds a threshold $\tau_{\text{rel}}$:

$$D_r = \{(x_i, y_i) \mid (x_i, y_i) \in D, \ \text{rel}(x_i, y_i) > \tau_{\text{rel}}\} \tag{8}$$

For head queries, the number of corresponding bidwords can be in the tens of thousands. To ensure class balance, we truncate the bidwords for each query based on their value.

**Fine-tuning Tasks:** We employ two types of supervised fine-tuning (SFT) tasks to enhance the model's understanding and performance in query rewriting for e-commerce.

**Task 1: `<prompt + query + bidword>`**
This task aims to enhance the model's spatial awareness of the bidword space and the association between queries and bidwords. Subsequently, the model will develop fundamental rewriting skills.

**Task 2: `<prompt + query + bidword_list>`**
In this task, the model is trained with lists of bidwords sorted in descending order based on their eCPM values. The bidword_list for each query is structured as follows:

$$\text{bidword\_list} = [\text{bidword}_1, \text{bidword}_2, \ldots, \text{bidword}_k]$$

## 3.5 LLM Post-training Alignment with Weighted Trie

Although conventional RLHF approaches can effectively integrate with our framework, we propose a more tailored solution. Our Weighted DPO (WDPO) approach directly incorporates the economic value of bidwords into the alignment process, leveraging the eCPM values for bidwords. It is important to note that WDPO is **optional** as the performance improvements it offers are not as significant relative to computational cost when compared to SFT and the varying nature of bidwords. Therefore, WDPO can be considered when resources permit, providing an additional layer of refinement to the training process.

The alignment process proceeds as follows:

(1) Identify relevant bidwords from online logs and calculate their corresponding eCPM values for each query in our dataset.
(2) Randomly sample two bidwords $(y_w, y_l)$ with differing eCPM values, ensuring:
$$|eCPM(y_w) - eCPM(y_l)| > \tau \tag{9}$$
where $\tau$ is a predefined threshold, $y_w$ represents the bidword with higher eCPM, and $y_l$ the lower.
(3) Process the query and selected bidwords through both policy and reference models to obtain probabilities: $\pi_\theta(y_w|x)$, $\pi_\theta(y_l|x)$, $\pi_{ref}(y_w|x)$, and $\pi_{ref}(y_l|x)$.
(4) Apply the modified DPO loss function:
$$\mathcal{L}_{WDPO}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}\right)\cdot w\right] \tag{10}$$
where $w$ is a weight function based on KL divergence:
$$w = \exp(-KL([\pi_\theta(y_w|x), \pi_\theta(y_l|x)] \parallel [P(y_w), P(y_l)])) \tag{11}$$
and $P(y_i)$ represents normalized eCPM values:
$$P(y_i) = \frac{eCPM(y_i)}{\sum_{j\in\{w,l\}} eCPM(y_j)} \tag{12}$$

## 3.6 Online Depolyment

In the context of deploying LLMs for online serving, the balance between computational efficiency and latency is paramount. Our approach leverages the Qwen2-7B model for offline inference, targeting head queries that constitute 50% of the page views (PV) in our system. This offline process generates the top 500 results per query, which are subsequently cached for rapid access. For mid-tail and long-tail queries that do not benefit from the cached results, we employ the Qwen2-1.5B model in an online serving capacity. This model is optimized to deliver a latency of 50 milliseconds, efficiently producing the top 50 results per query. In the future, we plan to integrate user behavior sequence data into our online serving framework to enhance personalized recommendation systems and gradually remove the offline part.

# 4 Experiments

## 4.1 Datasets

**Training Dataset:** For multi-task fine-tuning, we extracted 150 billion records from 30 days of online logs, followed by several rounds of data cleaning. Initially, we employed manual observation and regular expressions to filter out the majority of noisy queries. In the subsequent screening phase, we calculated the click-through rate (CTR) based on page views (PV) and clicks, retaining only those $\langle$`query, bidword`$\rangle$ pairs associated with purchase behavior

that meet the click-through rate requirements. We then applied a relevance model for further filtering. This relevance model is a fine-tuned version of BAAI/bge-large-zh-v1.5[44], optimized for e-commerce scenarios. We truncated the number of bidwords for each query, retaining a maximum of 50 bidwords per query. After third round of filtering, we were left with 110 million records.
**Test Dataset:** Our test dataset was constructed by randomly sampling 30,000 queries from online logs. These queries were stratified into three categories: 40% head queries, 40% torso queries, and 20% tail queries.

## 4.2 Evaluation Metrics

**Offline Metrics:** To comprehensively evaluate our query rewrite approach, we utilize several offline metrics:

(1) **Hit Rate:** A precision-oriented metric measuring the alignment between generated rewrites and user-clicked bidwords. For a query set $Q$, it is defined as:

$$\text{hitrate@}K = \frac{\sum_{q\in Q} |\mathcal{R}_q^K \cap C_q|}{\sum_{q\in Q} |C_q|} \tag{13}$$

where $\mathcal{R}_q^K$ represents the top-$K$ rewrite candidates for query $q$, and $C_q$ denotes the set of clicked bidwords for $q$. The numerator counts total hits across all queries, while the denominator normalizes by the total clickable bidwords.

(2) **Relevance Score:** A semantic alignment measure between rewritten queries and the original query intent, typically quantified via pretrained language model similarity (e.g., BERTScore).

(3) **Spearman Rank Correlation ($\rho$):** A non-parametric measure of rank consistency between model-generated and ground-truth bidword rankings. For $n$ observations:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{14}$$

where $d_i$ is the rank difference for the $i$-th bidword pair. The coefficient $\rho \in [-1, 1]$ reflects the monotonicity of the predicted rankings.

**Online Metrics:** To evaluate the model's performance, we use three key metrics: **cost**, which is the total advertiser expenditure for clicks within a specific traffic segment, **revenue per mille (RPM)**, which measures the revenue generated per thousand ad impressions and **Page View Relevance (PV rele)**,

## 4.3 Implementation Details

**Multi-task Fine-tuning:** We fine-tuned the Qwen2-7B model using a learning rate of 1e-5 and a cosine learning rate scheduler. The process was optimized with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a weight decay of 0.001. Gradient accumulation was set to 4. We used a batch size of 128 and applied the DeepSpeed ZeRO Stage 2 parallelism strategy with bfloat16 precision. The training was performed on 32 NVIDIA H20 GPUs for a total of 12 hours.
**Post-training Alignment:** For the post-training alignment phase, we use our fine-tuned Qwen2-7B model. The alignment phrase was carried out with a learning rate of 1e-6 and a batch size of 64. We set the $\beta$ parameter to 0.1. The training was executed on 16 NVIDIA H20 GPUs for a duration of 3 hours.

## 4.4 Offline Experiments

In this section, we present the results of offline experiments conducted to evaluate the performance of our proposed models against several baseline models. The comparative performance analysis is summarized in Table 1. We compared our models with the following baselines: first, the **Online Model**, which represents the current state-of-the-art multi-channel recall system deployed online; second, the **DeepNB** model [10], which employs a vector-based approach focusing solely on value; and finally, the **BEQUE** model [32], which utilizes a LLM fine-tuned with multi-task SFT and PRO alignment, representing the latest query rewriting approach. The performance metrics used for evaluation include hitrate, eCPM, Spearman's $\rho$, Relevance, and OOVR(Out-Of-Vocabulary Rate. To a certain extent, it is also equivalent to the hallucination rate of LLMs.). These metrics provide a comprehensive assessment of the models' effectiveness in terms of query rewriting, relevance, and OOVR.

*4.4.1 Comprehensive Evaluation of the VALUE Framework.* The VALUE framework we propose is designed to optimize online RPM revenue by utilizing offline eCPM values as a reference for learning effectiveness, rather than as direct revenue indicators. Our subsequent online experiments have demonstrated the validity and rationality of this modeling approach.The results presented in Table 1 clearly illustrate the significant effectiveness of our VALUE framework. Notably, the eCPM metric shows a remarkable improvement, serving as a strong indicator of offline learning success. Compared to the model trained with Supervised Fine-Tuning (SFT) on the same dataset, our VALUE model achieves a 722.6% increase in eCPM. Furthermore, when compared to the current online service model, our framework exhibits a 589.3% enhancement in eCPM. Importantly, the relevance metric for our VALUE model does not decline when compared to the SFT model; in fact, it demonstrates an improvement of 6.4 percentage points over the online service model. This observation substantiates our earlier hypothesis that relevance and value are decoupled in this domain, allowing us to reorder the generation sequence of bidwords to enhance both metrics simultaneously without trade-offs.

Compared to the DeepNB model, which focuses on efficiency-oriented rewriting using click data to maximize value, our VALUE framework demonstrates superior performance across multiple dimensions. While the DeepNB model effectively maximizes value, it suffers from poor relevance metrics. In contrast, our VALUE framework not only maintains high relevance but also significantly outperforms DeepNB in terms of value attributes. Specifically, the eCPM metric for our VALUE model is substantially higher than that of DeepNB. When compared to the BEQUE model, our framework achieves comparable relevance metrics but excels in generating high-value bidwords. This is evidenced by the eCPM metric, where VALUE achieves a value of 59803, far surpassing BEQUE's 7582. Additionally, our VALUE framework demonstrates a superior hit rate, with hitrate@500 reaching 91.52%, compared to BEQUE's 62.85%. This significant improvement in hit rate underscores the effectiveness of our approach in generating bidwords that are not only relevant but also highly likely to be clicked by users. It is

**Table 1: Comparative Performance Analysis of Various Models. The eCPM metric is used as an approximate estimation of reward, serving as a reference for the effectiveness of weighted trie collaborative inference.**

| Model | hitrate@50 | hitrate@500 | Spearman's $\rho$ | Relevance | OOVR | eCPM |
|---|---|---|---|---|---|---|
| OnlineModel | 59.93% | 85.41% | 0.02 | 68.15% | 0% | 8676 |
| DeepNB | 44.16% | 79.82% | 0.13 | 46.26% | 0% | 12852 |
| BEQUE | 39.26% | 62.85% | 0.08 | 74.37% | 38.41% | 7582 |
| raw Qwen2-7B | 12.88% | 18.37% | -0.03 | 59.81% | 68.63% | 2937 |
| SFT Qwen2-7B | 38.18% | 63.71% | 0.02 | 76.33% | 34.85% | 7439 |
| SFT+Trie | 59.11% | 86.38% | -0.01 | 75.31% | 0% | 7270 |
| SFT+Trie+DPO | 38.61% | 62.19% | 0.05 | 69.48% | 0% | 28391 |
| **SFT+WeightedTrie(*VALUE*)** | 60.37% | 91.52% | 0.46 | 74.55% | 0% | 59803 |
| SFT+WeightedTrie+DPO | 50.35% | 77.73% | 0.23 | 70.18% | 0% | 38301 |
| **SFT+WeightedTrie+WDPO(*VALUE+WDPO*)** | 62.67% | 91.91% | 0.56 | 73.49% | 0% | 63775 |

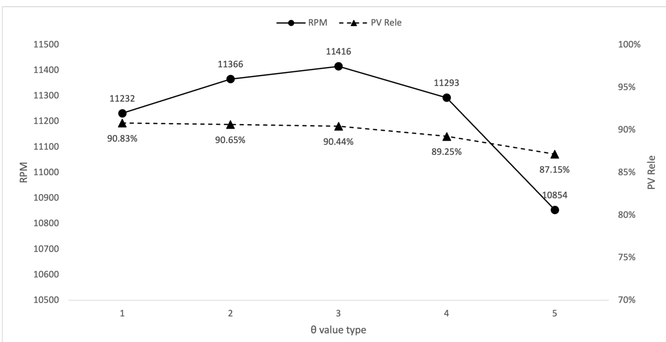


**Figure 3: Trade-off between RPM and PV rele through $\theta$.**

**Table 2: Comparative Performance Analysis of Hyperparameter.**

| Hyperparameter($\theta$) | hitrate@500 | Relevance |
|---|---|---|
| $\theta_1$ | 86.38% | 75.31% |
| $\theta_2$ | 89.65% | 75.04% |
| $\theta_3$ | 91.52% | 74.55% |
| $\theta_4$ | 88.63% | 69.85% |
| $\theta_5$ | 73.71% | 60.53% |

noteworthy that the BEQUE method suffers from a high Out-Of-Vocabulary Rate (OOVR) of 38.41%, which adversely affects its hit rate.

Moreover, our VALUE framework outperforms the online service model across all evaluated metrics, further validating the efficacy of our approach. The comprehensive improvements in eCPM, relevance, and other metrics underscore the robustness and effectiveness of our method in generating high-value and contextually relevant bidwords for e-commerce query rewriting tasks.

*4.4.2 Enhanced Performance in Fine-Grained Value Alignment.* Compared to traditional alignment methods such as DPO and Preference Ranking Optimization (PRO) used in the BEQUE model, our VALUE framework demonstrates a significant lead in the Spearman rank correlation metric. This indicates that conventional value alignment methods struggle with fine-grained value alignment tasks. Specifically, the Spearman's $\rho$ for our VALUE model is 0.46, which is substantially higher than the 0.05 achieved by DPO model and 0.08 by BEQUE.

Additionally, our framework seamlessly integrates with conventional model alignment methods. When we combined our proposed WDPO with the VALUE framework, we achieved the highest Spearman's $\rho$ of 0.56.

DPO suffers from significant overfitting issues, which severely impact its performance in terms of hit rate and relevance. In our experiments, the SFT + DPO + Trie model exhibited a hitrate@500 of only 62.19% and a relevance score of 69.48%, both of which are considerably lower than those achieved by our **VALUE** framework. DPO faces inherent challenges when dealing with positive and negative samples that are lexically very similar. The overfitting problem in DPO arises because the method tends to excessively optimize for the specific preference pairs seen during training, leading to a decline in generalization capability. DPO enhances the model's belief in specific events. Therefore, when we use DPO in conjunction with our VALUE framework, the improvement in eCPM is not significant. Specifically, the eCPM for the VALUE + DPO model is 38301, which is only marginally higher than the 28391 achieved by the SFT + DPO + WeightedTrie model.

Furthermore, the improvement in the Spearman rank correlation metric with DPO is not significant. The SFT + DPO + Trie model achieved a Spearman's $\rho$ of only 0.05, which is marginally better than the baseline but far from satisfactory. This limited improvement underscores the inherent limitations of DPO in handling fine-grained value alignment tasks.

## 4.5 Exploration of Hyperparameter Configurations

In this section, we provide a concise analysis of the impact of the hyperparameter $\theta$ on system performance. As shown in Table 2, an increase in $\theta$ results in a decrease in relevance. Figure 3 visually represents the dynamics of this trade-off. Initially, optimizing for value leads to an increase in RPM; however, when relevance becomes significantly low, RPM ultimately decreases. This underscores the

**Table 3: Comparative Performance Analysis of Base Model Size.**

| Model Size | hitrate@500 | Relevance |
|---|---|---|
| 0.5B | 68.32% | 61.85% |
| 1.5B | 87.49% | 71.29% |
| 7B | 91.52% | 74.55% |
| 14B | 92.69% | 76.68% |
| 65B | 93.98% | 77.99% |

**Table 4: Online A/B Test Results**

| Query Type | Cost | RPM | PV rele |
|---|---|---|---|
| VALUE all queries | +1.53% | +1.64% | +0.32pt |
| VALUE head queries | +0.97% | +1.03% | +0.10pt |
| VALUE torso queries | +1.68% | +1.74% | +0.33pt |
| VALUE tail queries | +2.35% | +2.66% | +0.75pt |

critical importance of meticulous hyperparameter tuning to ensure alignment with specific application objectives. For a detailed exploration of the hyperparameter $\theta$, please refer to Appendix A.

## 4.6 Scaling Laws in Model Performance

In our investigation of model performance across varying scales, we conducted experiments using models ranging from 0.5 billion to 65 billion parameters. Our findings consistently demonstrate that larger models exhibit superior performance across all evaluated metrics. Specifically, as model size increases, there is a notable improvement in hitrate, indicating enhanced capability in generating high-value and relevant bidwords. This trend aligns with the theoretical expectations of scaling laws, which suggest that larger models can capture more complex patterns and nuances in data, thereby improving overall effectiveness.

## 4.7 Online Experiments

To further validate the effectiveness of our proposed VALUE framework, we conducted 14-day online A/B tests on our system, focusing on cost and RPM. Results are summarized in Table 4 and Table 5.

The online A/B tests corroborate the offline findings, consistently demonstrating improvements in both cost and RPM across all query types. Our framework exhibits particularly notable enhancements for torso and tail queries, which are typically characterized by greater variability and complexity. This underscores the framework's robust generalization capabilities. As illustrated in Table 5, our model excels in discerning the intent behind very long and complex queries. The observed improvements in RPM and relevance for medium to super long queries underscore the model's proficiency in managing intricate and challenging query structures.

## 4.8 Case Study

In this section, as illustrated in Figure 4, we examine a specific case involving the user query "fatty's drink." Previously, without the use of LLMs, our system could only match the query at the keyword level, specifically "drink." With the integration of LLMs, we gained a

**Table 5: Comparative Performance Analysis of Query Length Layer.**

| Query Length | Cost | RPM | Relevance |
|---|---|---|---|
| Short Query | +0.41% | +0.56% | +0.02pt |
| Medium Query | +0.88% | +0.98% | +0.30pt |
| Long Query | +1.26% | +1.56% | +0.79pt |
| Super Long Query | +1.17% | +1.55% | +0.54pt |

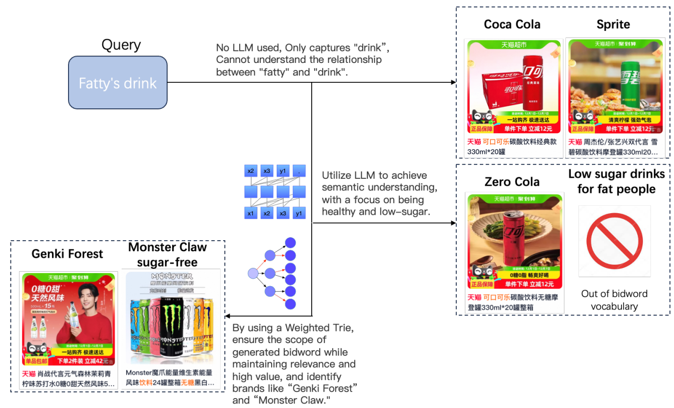


**Figure 4: Case Study of VALUE Framework**

deeper understanding of the user's intent, identifying a preference for low-sugar, low-calorie beverages. Additionally, by employing a weighted trie, we ensured that the advertisements presented were not only closely aligned with the user's intent but also of high value.

## 5 Conclusion

In this study, we introduced **VALUE**, an innovative framework designed to enhance query rewriting in sponsored search advertising by integrating value-awareness into the generation process. Our approach addresses the significant challenge of producing rewrites that are not only semantically relevant but also economically valuable—a gap that existing methods often fail to bridge. By embedding value information directly into the LLM's decoding process through a novel weighted trie structure, we effectively steer generation towards high-value bidwords without substantially compromising relevance.

Comprehensive experiments underscore the effectiveness of **VALUE** in enhancing the economic attributes of bidword generation. Offline evaluations demonstrate a significant increase in eCPM, outperforming traditional methods while preserving high relevance scores. The improvement in Spearman rank correlation confirms the superior ranking quality achieved through our fine-grained value alignment. Furthermore, online A/B testing validates the real-world efficacy of **VALUE**, highlighting notable improvements in key performance indicators such as cost and RPM. Our framework's versatility also suggests potential applications beyond sponsored search advertising, wherever the generation space is fixed and reward signals are present.

## References


[1] Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571* (2024).

[2] Thomas Anthony, Zheng Tian, and David Barber. 2017. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems* 30 (2017).

[3] Ioannis Antonellis, Hector Garcia-Molina, and Chi-Chao Chang. 2008. Simrank++ query rewriting through link analysis of the clickgraph (poster). In *Proceedings of the 17th international conference on World Wide Web*. 1177–1178.

[4] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* (2022).

[5] Jagdev Bhogal, Andrew MacFarlane, and Peter Smith. 2007. A review of ontology based query expansion. *Information processing & management* 43, 4 (2007), 866–886.

[6] Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* 39, 3/4 (1952), 324–345.

[7] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 243–250.

[8] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217* (2023).

[9] Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319* (2024).

[10] Xiuying Chen, Daorui Xiao, Shen Gao, Guojun Liu, Wei Lin, Bo Zheng, Dongyan Zhao, and Rui Yan. 2020. RPM-Oriented Query Rewriting Framework for E-commerce Keyword-Based Sponsored Search (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 13769–13770.

[11] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).

[12] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks* (2023).

[13] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web*. 325–332.

[14] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773* (2023).

[15] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233* (2023).

[16] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475* (2024).

[17] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306* (2024).

[18] Daniel C Fain and Jan O Pedersen. 2006. Sponsored search: A brief history. *Bulletin-American Society For Information Science And Technology* 32, 2 (2006), 12.

[19] Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*. PMLR, 10835–10866.

[20] Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642* (2024).

[21] Alex Havrilla, Sharath Raparthy, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Railneau. 2024. Glore: When, where, and how to improve llm reasoning via global and local refinements. *arXiv preprint arXiv:2402.10963* (2024).

[22] Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691* 2, 4 (2024), 5.

[23] Bernard J Jansen and Tracy Mullen. 2008. Sponsored search: an overview of the concept, history, and technology. *International Journal of Electronic Business* 6, 2 (2008), 114–131.

[24] Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*. PMLR, 17506–17533.

[25] Mu-Chu Lee, Bin Gao, and Ruofei Zhang. 2018. Rare query expansion through generative adversarial networks in search advertising. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*. 500–508.

[26] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871* (2018).

[27] Sen Li, Fuyu Lv, Taiwei Jin, Guiyang Li, Yukun Zheng, Tao Zhuang, Qingwen Liu, Xiaoyi Zeng, James Kwok, and Qianli Ma. 2022. Query Rewriting in TaoBao Search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3262–3271.

[28] Aritra Mandal, Ishita K Khan, and Prathyusha Senthil Kumar. 2019. Query Rewriting using Automatic Synonym Extraction for E-commerce Search.. In *eCOM@ SIGIR*.

[29] Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734* (2024).

[30] Akash Kumar Mohankumar, Bhargav Dodla, Gururaj K, and Amit Singh. 2023. Unified Generative & Dense Retrieval for Query Rewriting in Sponsored Search. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4745–4751.

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[32] Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Tongxu, and Enhong Chen. 2023. Large Language Model based Long-tail Query Rewriting in Taobao Search. *Companion Proceedings of the ACM on Web Conference 2024* (2023). https://api.semanticscholar.org/CorpusID:265042961

[33] Weizhen Qi, Yeyun Gong, Yu Yan, Jian Jiao, Bo Shao, Ruofei Zhang, Houqiang Li, Nan Duan, and Ming Zhou. 2020. Prophetnet-ads: A looking ahead strategy for generative retrieval models in sponsored search engine. In *Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, October 14–18, 2020, Proceedings, Part II 9*. Springer, 305–317.

[34] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).

[35] N Reimers. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint arXiv:1908.10084* (2019).

[36] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[37] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[38] Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716* (2023).

[39] Tao Tao and ChengXiang Zhai. 2006. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 162–169.

[40] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: an instruction-following llama model (2023). *URL https://github. com/tatsu-lab/stanford_alpaca* 1, 9 (2023).

[41] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2023. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems* 36 (2023), 74764–74786.

[42] Yang Wang, Zheyi Sha, Kunhai Lin, Chaobing Feng, Kunhong Zhu, Lipeng Wang, Xuewu Jiao, Fei Huang, Chao Ye, Dengwu He, Zhi Guo, Shuanglong Li, and Lin Liu. 2024. One-step Reach: LLM-based Keyword Generation for Sponsored Search Advertising. In *Companion Proceedings of the ACM Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 1604–1608. https://doi.org/10.1145/3589335.3651943

[43] Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, et al. 2024. Long-form factuality in large language models. *arXiv preprint arXiv:2403.18802* (2024).

[44] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 641–649.

[45] Jinxi Xu and W Bruce Croft. 2017. Quary expansion using local and global document analysis. In *Acm sigir forum*, Vol. 51. ACM New York, NY, USA, 168–175.
[46] Jinxi Xu and W Bruce Croft. 2017. Quary expansion using local and global document analysis. In *Acm sigir forum*, Vol. 51. ACM New York, NY, USA, 168–175.
[47] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems* 36 (2024).

# A Hyperparameter Details

In this appendix, we delve into the specifics of the hyperparameter $\theta_d$, which is designed to vary with the depth of the trie. The selection of $\theta_d$ is critical in modulating the influence of value relative to relevance as the trie depth changes. The configurations explored in our research are as follows:

- $\theta_1$: $[0, 0, 0, 0, 0, 0, 0, \dots]$, indicating that the weighted trie is ineffective and the original LLM reasoning result is used.
- $\theta_2$: $[1, 1, 1, 1, 1, 1, 1, \dots]$, implying equal weighting of relevance and value at each depth level.
- $\theta_3$: $[1, 2, 3, 4, 5, 6, 7, \dots]$, where the weight increases linearly with depth, emphasizing value as the trie deepens.
- $\theta_4$: $[1, 2, 4, 8, 16, 32, 64, \dots]$, representing an exponential increase in weight with depth, further prioritizing value.
- $\theta_5$: $[2, 4, 8, 16, 32, 64, 128, \dots]$, offering a more pronounced exponential tendency compared to $\theta_4$.

These configurations allow for nuanced control over the relevance-value trade-off, enabling the system to adapt its focus based on the hierarchical structure of the weighted trie. The choice of $\theta_d$ thus plays a pivotal role in optimizing the overall performance of the inference system.