QUAKER: Query Understanding And top-K Embedding Retrieval

Xiangjun Fu, Xiaoyang Wan, Dong Dong
xjf@seas.upenn.edu,wan3@seas.upenn.edu,winter24@seas.upenn.edu

Abstract

Modern e-commerce platforms increasingly rely on natural language interfaces to connect users with relevant products. However, traditional keyword-based systems struggle with real user queries that are often obscure, conversational, and under-specified, and exhaustive neural retrieval over millions of items is computationally prohibitive. To study this setting, we use the Amazon-C4 test set from McAuley's Lab, which comprises ambiguous, review-derived queries designed to simulate realistic search behavior.

We propose **QUAKER (Query Understanding And top-K Embedding Retrieval)**, a dual-stage retrieval framework that first performs lightweight query understanding via category prediction, and then applies high-quality embedding-based retrieval over a reduced candidate pool.

This design explicitly balances semantic expressiveness with efficiency: by narrowing the candidate set before dense retrieval, QUAKER remains scalable while better capturing the intent of obscure queries. On the Amazon-C4 benchmark, our best configuration—using large versions of both BERT and E5—achieves a top-200 (top-1%) accuracy of 74.77%, outperforming a TF–IDF baseline by over 30 percentage points.

All code and LaTeX source are available at this repository.

1 Introduction

Language plays a pivotal role in e-commerce platforms, serving as a key modality for describing and retrieving products. Tasks such as product retrieval and recommendation are increasingly reliant on advanced language modeling techniques [1, 2]. Traditional recommendation systems have often relied on keyword-based features, which fail to capture the nuanced semantics of natural language and struggle with informal, conversational queries.

With the advent of large language models (LLMs) [3, 13], there is growing interest in leveraging their semantic capabilities to enhance recommendation systems [6]. However, integrating LLMs into large-scale retrieval scenarios that involve millions of items remains challenging due to computational constraints. Moreover, user queries in real-world applications are frequently obscure and under-specified, making it difficult for models to accurately infer intent and retrieve the intended items.

Existing methods for such tasks typically follow one of two paradigms:

- **End-to-end neural models** fine-tuned on task-specific data, which can perform well within a narrow domain but often lack generalizability across diverse tasks and query distributions [2].
- **Pre-trained language models (PLMs)** used purely as embedding generators, which are not specifically tailored for recommendation contexts and can be brittle when handling obscure or noisy queries [6].

To address these challenges, we introduce **QUAKER (Query Understanding And top-K Embedding Retrieval)**, a two-stage retrieval framework that integrates pre-trained models with lightweight query understanding. In Stage 1, a fine-tuned BERT classifier predicts high-level product categories given a query and restricts retrieval to a compact candidate pool. In Stage 2, an E5 encoder maps both the query and item metadata into a shared embedding space, and similarity search over the filtered pool ranks items by semantic relevance. This design explicitly balances semantic expressiveness with efficiency: the system "understands" the query at the category level while avoiding exhaustive dense retrieval over the entire catalog.

Our main contributions are as follows:

- **Dual-stage retrieval framework.** We propose QUAKER, a scalable two-stage architecture that combines BERT-based category prediction with E5-based embedding retrieval, achieving strong top-K performance on ambiguous queries from the Amazon-C4 dataset.
- **Ablation and scaling analysis.** We systematically evaluate the impact of the Stage 1 classifier and model size on retrieval quality, showing that removing the classifier leads to a substantial drop in accuracy and that scaling both BERT and E5 yields monotonic gains.
- **Dataset inspection and enrichment.** Through exploratory analysis, we identify issues such as extremely short or uninformative metadata and discuss simple cleaning and enrichment strategies, including using the model's top-ranked items as supplementary candidates for future dataset refinement.
- **Qualitative case studies.** We provide detailed examples of queries, top-ranked items, and their

metadata in the Appendix, illustrating how the proposed framework handles obscure real-world queries.

2 Dataset analysis

2.1 Dataset Description

In this study, we utilize two datasets to evaluate product search performance under complex contexts:

2.1.1 Amazon-C4 Dataset.

The Amazon-C4 dataset, developed by the McAuley Lab, is designed to assess a model's ability to comprehend complex language contexts and retrieve relevant items. It comprises 21,223 user reviews from the Amazon Reviews 2023 dataset, each rephrased by ChatGPT into vague, first-person queries. These queries are paired with corresponding item identifiers, facilitating the evaluation of product search tasks. The dataset is publicly available on Hugging Face at the following link: https://huggingface.co/datasets/McAuley-Lab/Amazon-C4 [7].

2.1.2 Sampled Item Metadata Dataset.

`The sampled_item_metadata_1M.jsonl` file contains around 1 million items sampled from the Amazon Reviews 2023 dataset. Each entry includes:

- **item_id**: A unique identifier corresponding to the `parent_asin` in the original dataset.
- **category**: The item's category, useful for evaluating model performance within specific domains.
- **metadata**: A concatenation of the item's title and description from the original metadata.

This sampled item pool is utilized for evaluation in the BLaIR paper, providing a comprehensive set of items for retrieval tasks. The dataset can be accessed from Hugging Face at: https://huggingface.co/datasets/McAuley-Lab/Amazon-C4 [8].

2.2 Exploratory Data Analysis

As shown in Figure 1, over 90% of metadata lengths are below 400 words, with a small number of outliers exceeding 2000 words. However, certain metadata entries are excessively short, providing insufficient information about the associated items, which could hinder effective analysis. This will be addressed later in our data processing and improvement is demonstrated in our model evaluation.

Figure 2 demonstrates that all query lengths are under 220 words, with the majority falling well below this threshold. Despite this, short metadata lengths remain a challenge as they can obscure a clear understanding of the items.

Figure 3 illustrates the number of items available across various categories. The "Home" category dominates with the highest number of items, exceeding 3,000,

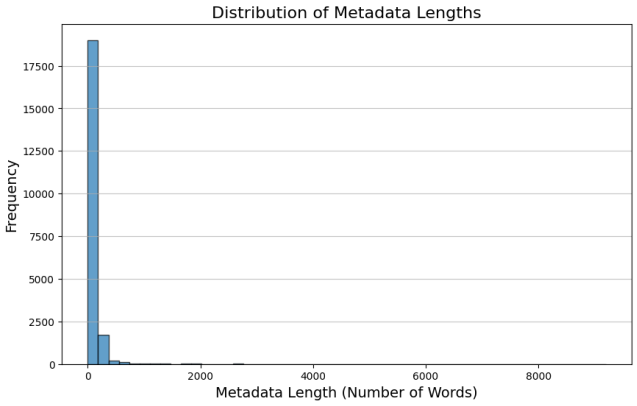

Figure 1. Distribution of Metadata Lengths.

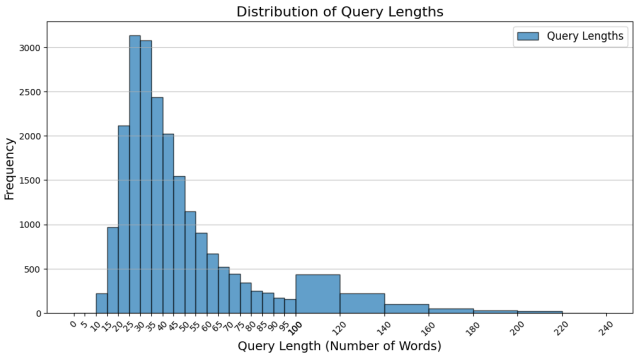

Figure 2. Distribution of Query Lengths.

while categories like "Beauty" and "Gift" have significantly fewer items, showing a clear imbalance in category representation. Such distribution indicates that certain categories, such as "Home" and "Electronics," might play a larger role in the dataset, potentially influencing retrieval tasks.

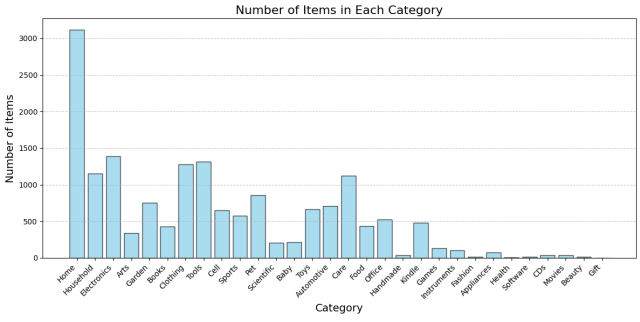

Figure 3. number of items in different categories.

2.3 Dataset Merging and Motivation

The `Amazon-C4` Dataset and the `sampled_item_metadata_1M` dataset serve complementary roles in facilitating our

analysis. While the `Amazon-C4` Dataset provides user queries and their corresponding ground-truth `item_id`s, it lacks detailed descriptions of the items themselves, such as their attributes or categories. This information is crucial for understanding the characteristics of the items that users are interested in and for designing systems capable of accurately matching queries to relevant items.

The `sampled_item_metadata_1M` dataset addresses this gap by offering rich metadata and categorical information for approximately 1 million items.

To fully leverage the strengths of both datasets, we merge them by matching the `item_id`s in the `Amazon-C4` Dataset with those in the `sampled_item_metadata_1M` dataset. For instance, consider the following entry from the `Amazon-C4` Dataset:

```
{
    "qid": 0,
    "query": "I need filters that effectively...",
    "item_id": "B0C5QYYHTJ",
    "user_id": "AGREO2G3GTRNYOJK4CIQV2DTZLSQ",
    "ori_rating": 5,
    "ori_review": "These filters work..."
}
```

The ground-truth `item_id` in this entry corresponds to the following entry in the `sampled_item_metadata_1M` dataset:

```
{
    "item_id": "B0C5QYYHTJ",
    "category": "Home",
    "metadata": "Flintar Core 300 True HEPA..."
}
```

By combining the `query`, `item_id`, `category`, and `metadata`, we construct a unified entry, such as:

```
{
    "query": "I need filters that effectively...",
    "item_id": "B0C5QYYHTJ",
    "category": "Home",
    "metadata": "Flintar Core 300 True HEPA..."
}
```

This merging process allows us to enrich the query data with additional contextual information about the items, enabling a more comprehensive evaluation of query-to-item relevance. It also facilitates downstream tasks, such as identifying item features most relevant to user queries or categorizing user preferences.

3 Problem Definition

In this work, we study *query-to-item retrieval* in an e-commerce setting with obscure, under-specified user queries. Let q denote a user query (free-form text). Let $\mathcal{I} = \{i_1, \ldots, i_M\}$ be the catalog of items (approximately $M \approx 21{,}000$ in our dataset), where each item i is associated with textual metadata m_i (title, description, etc.) and a category label c_i. For each query q the dataset provides a ground-truth item $i^* \in \mathcal{I}$.

Our goal is to learn a scoring function

$$s_\theta q, i \in \mathbb{R}$$

that measures the semantic relevance between a query q and an item i, such that the induced ranking over items places the ground-truth item near the top:

$$\text{rank}_{s_\theta} i^* \mid q \text{ is small.}$$

Practically, we focus on top-K retrieval, with $K = 200$ (approximately the top 1% of the catalog). For a given query q, let $\text{TopK}_{s_\theta} q$ be the set of K items with the highest scores under s_θ. Our main evaluation metric is top-K accuracy:

$$\text{Acc}_K = \frac{1}{|\mathcal{Q}|} \mathbb{I}\big[i^* \in \text{TopK}_{s_\theta} q \big],$$

where $\mathcal{Q}$ is the set of test queries.

4 Baseline Model: TF-IDF with Cosine Sim.

As a baseline, we implement a simple yet effective model using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization combined with cosine similarity for passage ranking. This approach is designed to provide a point of comparison for our deep learning-based methods.

4.1 TF-IDF Vectorization

TF-IDF is a widely used technique for representing text data in information retrieval tasks. It computes a weighted representation of terms, emphasizing terms that are frequent in a specific document but rare across the corpus. Formally, the TF-IDF score for a term t in a document d is defined as:

$$\text{TF-IDF}\left(t, d\right) = \text{TF}\left(t, d\right) \times \log\left(\frac{N}{\text{DF}\left(t\right)}\right)$$

where:

- $\text{TF}\left(t, d\right)$: Term frequency of t in d.
- $\text{DF}\left(t\right)$: Document frequency of t (number of documents containing t).
- N: Total number of documents.

We use the `TfidfVectorizer` from the Scikit-learn library to compute TF-IDF vectors for both queries and passages.

4.2 Cosine Similarity for Ranking

To rank passages for a given query, we compute the cosine similarity between the TF-IDF vector of the query and each passage. Cosine similarity is defined as:

$$\text{Sim}\mathbf{q}, \mathbf{p} = \frac{\mathbf{q} \cdot \mathbf{p}}{\|\mathbf{q}\|\|\mathbf{p}\|},$$

where $\mathbf{q}$ and $\mathbf{p}$ are the TF-IDF vectors of the query and passage, respectively. Since TF-IDF vectors are L2-normalized by default, the dot product directly yields cosine similarity.

4.3 Performance Evaluation

For each query, the top 200 passages with the highest cosine similarity scores are retrieved. The accuracy of the baseline is measured by checking if the ground truth `item_id` appears among the top-200 ranked passages. While simple, this approach achieves a top-200 accuracy of 42.43%, serving as a benchmark for evaluating the effectiveness of advanced models.

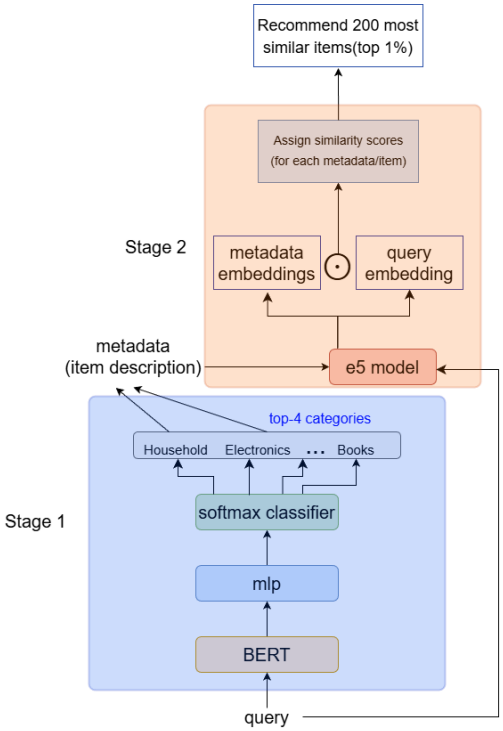

Figure 4. Overview of the model structure, including category prediction with BERT (Stage 1) and similarity matching with E5 (Stage 2).

5 Model Structure and evaluations

Because the catalog is large and queries are often obscure, directly computing $s_\theta q, i$ for all $i \in \mathcal{I}$ is both computationally expensive and error-prone (as we show in our ablation results in Table 1). Our framework therefore decomposes retrieval into two stages:

1. **Category prediction (Stage 1).** Given a query q, a BERT classifier predicts a small set of likely categories $C_{\text{top}}q$ (top-4 in our implementation) and restricts candidates to items whose category lies in this set:
$$\mathcal{I}q = \{i \in \mathcal{I} : c_i \in C_{\text{top}}q\}.$$
2. **Semantic ranking (Stage 2).** Within $\mathcal{I}q$, an E5 embedding model maps both q and each item's metadata m_i into a shared vector space and computes a similarity-based score $s_\theta q, i$. We then rank candidates by $s_\theta q, i$ and derive $\text{TopK}_{s_\theta}q$.

This decomposition captures both aspects of our objective: accurately matching each query to its ground-truth item while limiting expensive semantic retrieval to a small, query-dependent candidate set.

Figure 4 provides a high-level overview of the two-stage pipeline.

5.1 Data Splitting and Preprocessing

The cleaned dataset is first shuffled randomly and then split into training and test sets with a 9:1 ratio. This ensures that the model is evaluated on unseen data, maintaining the integrity of the evaluation process.

5.2 Part 1: Category Prediction with BERT

The first stage of the model is a classifier that predicts the category of the desired item based on the user's query. We employ a pre-trained `BERT` family model, which is developed by *Devlin et al.* [4]. BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based language model that captures bidirectional context, making it highly effective for understanding complex language semantics.

In our setup, the BERT model is fine-tuned, and a softmax classifier is added on top to predict the category. The model learns to associate semantic patterns in the query text with specific categories. After a training of 3 epochs as shown in Figure 5 the classifier achieves a probability of 90.06% on the test set for correctly identifying the true category within the top-4 predicted categories.

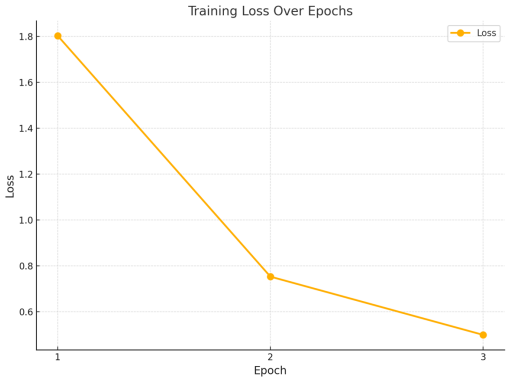

Figure 5. training classifier

5.3 Part 2: Similarity Matching with E5

The second part of the model uses the E5 embedding model to compute the similarity between the user's query and the metadata of items within the top-4 predicted categories (among 30 total categories). E5, as proposed in *Text Embeddings by Weakly-Supervised Contrastive Pre-training* [16], is a general-purpose embedding model designed to generate high-quality semantic embeddings for various tasks, including information retrieval and recommendation systems. By projecting both queries and metadata into the same semantic space, E5 enables effective similarity matching.

For each query in the test set, we compute the embeddings for the query and all items in the top-4 categories. The similarity scores are calculated between the query embedding and each item's metadata embedding. Items are then ranked by their similarity scores. One example of a random query is shown in Figure 6

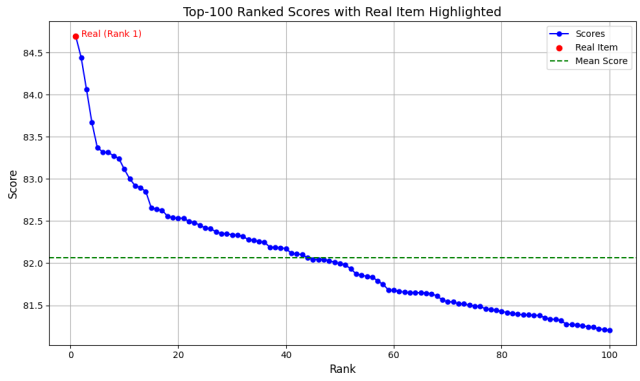

Figure 6. Top-100 ranked score of metadata with real item highlighted for a random query.

5.4 Evaluation

5.4.1 Top-200 Evaluation. We evaluate the model's performance by checking whether the ground truth `item_id` appears in the top-200 ranked items (approximately the top 1% of the catalog). This hit rate is equivalent to a Recall@200 metric and directly reflects the model's ability to surface a relevant item near the top of the ranked list.

Unless otherwise noted, all models are trained and evaluated on the preprocessed dataset, where items with empty or extremely short (< 10 tokens) `metadata` are removed in our data-processing pipeline.

Table 1 summarizes the Top-200 accuracy for all configurations we consider. The upper block reports results for our two-stage pipeline under different choices of BERT (stage-1 classifier) and E5 (stage-2 retriever) model sizes. The lower block reports several baselines and ablations that highlight the effect of the classifier.

Table 1. Top-200 accuracy (%) of different model configurations. The upper block reports our two-stage pipeline under different BERT and E5 sizes; the lower block reports TF-IDF baseline and the ablation case in not using stage 1.

Configuration	Top-200 Accuracy (%)		
	E5-small	E5-base	E5-large
BERT base	69.58	71.16	73.88
BERT large	70.52	71.90	**74.77**
TF–IDF baseline		42.43	
No classifier (E5-base)		65.21	

The best configuration is obtained when using the large versions of both BERT and E5, achieving a test accuracy of 74.77%. Compared to a simple TF–IDF baseline (42.43%), our approach improves accuracy by more than 30 percentage points. Removing the classifier and relying solely on retrieval severely hurts performance (65.21%), showing that the category-aware filtering in Stage 1 is crucial for both accuracy and efficiency.

5.4.2 Effect of Model Scaling. To better understand how model capacity affects performance, we vary the sizes of BERT and E5 individually while keeping the other component fixed. Table 1 summarizes these results, and Figure 7 visualizes them: the left panel shows the effect of scaling E5 while holding BERT fixed, and the right panel shows the effect of scaling BERT while holding E5 fixed. In both cases we observe an approximately monotonically increasing trend in Top-1% (Top-200) accuracy as model size grows, consistent with the quantitative results in Table 1. This indicates that both the classifier and the retriever benefit from increased parameter capacity.

We believe the positive correlation between embedding model size and retrieval performance arises because larger models learn more expressive semantic representations. With greater capacity, the encoder can better compress queries and item metadata into a latent space that preserves finer-grained distinctions and captures more diverse contextual cues, making it easier to perform both category classification and within-category retrieval.

5.4.3 Error Attribution and Performance Analysis. A key advantage of our modular two-stage design, separating the classifier and retriever, is that it enables clear attribution of errors to individual components. Unlike end-to-end approaches where failure modes are often entangled, our architecture allows us to quantify how much each stage contributes to the final system performance

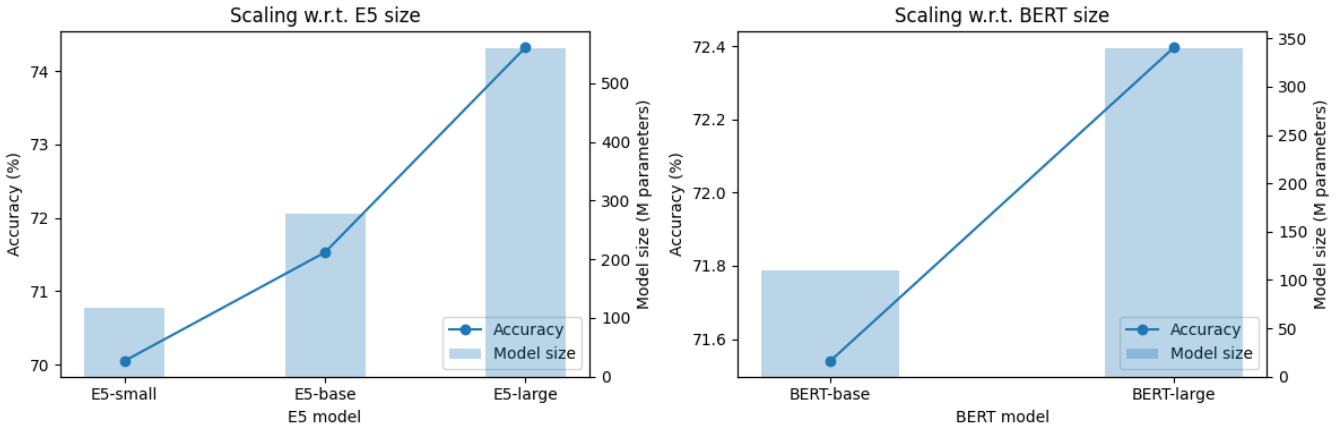

Figure 7. Scaling behavior of the two-stage pipeline with respect to E5 size (left) and BERT size (right). Accuracy is plotted on the left y-axis, and model size (in millions of parameters) is plotted on the right y-axis.

Table 2. Retrieval success rates with a BERT-large classifier. "No classifier" reports end-to-end Top-200 accuracy when retrieval is performed over the full catalog. The other rows report *conditional* success rates when the query's category is correctly predicted.

Retriever	Top-200 Success (%)
E5-base (no classifier, full catalog)	65.21
E5-small (within true category)	77.87
E5-base (within true category)	79.39
E5-large (within true category)	82.56

and to reason explicitly about where additional capacity is most valuable.

Error decomposition. With the BERT-large classifier achieving 90.56% accuracy and the E5-large retriever achieving a conditional Top-200 success rate of 82.56% *given* a correct category prediction (Table 2), the overall Top-200 accuracy of 74.77% can be written as

$$\text{Acc}_{\text{overall}} = \text{Acc}_{\text{cls}} \times \text{Acc}_{\text{ret|cls}} = 0.9056 \times 0.8256 \approx 0.7477.$$

The complementary error rate (25.23%) can be decomposed into:

- **Classification errors**: 9.44 percentage points, coming from the 9.44% of queries whose category is misclassified. For these queries, retrieval in the wrong category cannot recover the ground-truth item.
- **Retrieval errors within the correct category**: among the 90.56% of correctly classified queries, E5-large fails 17.44% of the time (100% − 82.56%), which contributes $0.9056 \times 0.1744 \approx 15.79$ percentage points to the overall error.

Thus, out of the total 25.23% error, 9.44 points (37.4%) are due to classification and 15.79 points (62.6%) are due to retrieval within the correct category. Retrieval failures dominate, but both stages contribute substantially.

Impact of classification on retrieval. To understand the value added by classification, we compare retrieval performance with and without the classifier. As shown in Table 2, direct retrieval over the full catalog without classification ("No classifier" baseline) achieves only 65.21% Top-200 accuracy with E5-base. In contrast, when we restrict retrieval to the ground-truth category (i.e., conditional on the classifier being correct), E5-base attains a 79.39% success rate—a relative improvement of approximately 21.7%.

For E5-large, the conditional success rate increases further to 82.56%. Combining this with the classifier's 90.56% accuracy yields the end-to-end Top-200 accuracy of 74.77%. In other words, the classifier's category-aware filtering substantially amplifies the effectiveness of the retriever by reducing the candidate set to items that are already plausible for the query.

This gain can be attributed to two factors: (1) the classifier reduces the effective search space by filtering out irrelevant categories, thereby decreasing the number of hard negatives that the retriever must distinguish, and (2) within each category, the remaining candidates are more semantically homogeneous, allowing the embedding model to focus on fine-grained distinctions rather than coarse cross-category separation.

System-level implications. Despite a non-trivial 9.44% classification error rate, the two-stage approach achieves 74.77% accuracy compared to 65.21% for direct retrieval, demonstrating that the benefits of reduced search complexity substantially outweigh the risk of early-stage

classification errors. This finding empirically validates our design choice to separate classification and retrieval.

The error attribution analysis suggests two primary directions for future improvement: (1) improving fine-tuned classifier accuracy beyond 90.56%, which would directly reduce unrecoverable errors, and (2) enhancing retrieval robustness within categories, targeting the 15.79 percentage points of error arising from failures in correctly classified queries. Because retrieval errors currently account for the majority of the overall failure rate, finetuning the retriever or switching to a better pre-trained model may offer the largest gains, while classifier improvements provide complementary benefits.

5.4.4 Further Evaluation. In addition to the quantitative results above, we conduct a qualitative analysis by randomly selecting several queries and examining the top-5 most relevant items ranked by the model. For these top-5 items, we inspect their `metadata` (see Appendix for detailed examples). We observe that the metadata of these items is often highly relevant to the query requirements, even when the ground truth `item_id` is not included in the Top-200 results.

We believe this phenomenon is closely related to the construction of the dataset itself. Since the queries were generated by modifying reviews, they may not fully capture the precise requirements of the ground-truth items. As a result, the measured accuracy saturates even though the model is capable of retrieving semantically appropriate items that align well with the query intent.

To address this limitation, we propose leveraging our model's predictions to refine the dataset: for each query, we can take the top-5 most relevant items identified by our approach and use them as supplementary candidates or alternative labels. This would enrich the dataset and better align the ground-truth labels with actual query intent. Detailed examples of queries, their top-5 relevant items, and the associated metadata are included in the Appendix to further illustrate this point.

6 Related Work

Our work touches upon several aspects of e-commerce product retrieval. We review related research from four perspectives: dense retrieval techniques, multi-stage retrieval architectures, domain-adaptive models for e-commerce, and evaluation benchmarks, to clarify QUAKER's positioning and contributions within the existing research landscape.

6.1 Dense Retrieval and Neural Ranking Models

The field of information retrieval is undergoing a paradigm shift from traditional sparse representations (e.g., TF-IDF and BM25) to deep learning-based dense representations. This shift provides a stronger semantic foundation for understanding complex and ambiguous natural language queries, forming the technical background for QUAKER's second stage (semantic ranking).

Traditional keyword matching methods struggle to capture semantic relationships beyond the lexical level, performing poorly when dealing with synonyms, polysemy, and ambiguous queries. Pre-trained language models represented by BERT [4] have revolutionized information retrieval through bidirectional contextual encoding, enabling the generation of high-quality text embeddings and thus achieving semantic-based matching rather than merely lexical matching.

The pioneering work of **Dense Passage Retrieval (DPR)** [9] employs a dual-encoder architecture that independently generates dense vectors for queries and documents, performing large-scale retrieval through efficient vector similarity search (e.g., FAISS). This paradigm of "independent encoding and vector retrieval" has become the mainstream approach in dense retrieval and is also the core idea adopted by QUAKER's second stage. DPR significantly outperforms BM25 baselines on multiple open-domain QA datasets, demonstrating the practical feasibility of dense retrieval.

Furthermore, **ColBERT** [10] introduces a "late interaction" mechanism that maintains efficient retrieval while allowing fine-grained interactions between query and document token vectors, achieving better ranking accuracy than standard dual-encoders. By delaying yet retaining fine-grained interaction, ColBERT can pre-compute document representations, significantly accelerating query processing. These works demonstrate different trade-offs between efficiency and effectiveness, and QUAKER's two-stage design embodies precisely such a trade-off.

6.2 Multi-Stage Retrieval Architectures

To balance the efficiency of large-scale retrieval with the effectiveness of complex ranking models, multi-stage (or cascade) architectures have become standard practice in both industry and academia [17]. QUAKER's two-stage design is a concrete application of this mature architectural philosophy in e-commerce search scenarios.

Running complex neural ranking models directly on millions or even billions of documents is computationally infeasible. Therefore, a common solution is to adopt a "retrieve-and-rerank" or cascade architecture: the first stage uses lightweight, high-recall models to quickly filter a smaller candidate set from massive data; the second stage applies computationally intensive but more accurate models for fine-grained re-ranking on this small candidate set.

RankFlow [14] proposes a joint optimization framework for multi-stage cascade ranking systems. The framework emphasizes that each stage should adapt to its specific data distribution, generating training data from

preceding stages and learning supervision signals from subsequent stages. RankFlow's core idea is to first estimate the selection bias of subsequent stages, then learn a ranking model adapted to the downstream modules' selection bias. Similarly, **Full Stage Learning to Rank** [18] proposes the Generalized Probability Ranking Principle (GPRP), explicitly modeling ranking preferences from both users and subsequent stages, emphasizing the selection bias problem in multi-stage systems.

These studies demonstrate that multi-stage architectures are well-validated effective strategies. QUAKER's design—first performing "coarse filtering" through category prediction, then "fine ranking" through E5 embeddings—perfectly aligns with this philosophy, with the innovation of designing the first stage as a "query understanding" task specifically for e-commerce scenarios.

6.3 Query Understanding and Domain-Adaptive Models for E-Commerce

In e-commerce domains, due to the special characteristics of user queries (vague, conversational, diverse intents), general-purpose language models often underperform. Therefore, query understanding and domain-adaptive models specifically for e-commerce are crucial research directions, providing direct motivation for QUAKER's first stage (category prediction) and overall framework.

"Query Understanding" plays a central role in modern search engines, including tasks such as intent classification, entity recognition, and query rewriting. In e-commerce scenarios, accurately predicting the product category users want to purchase is a key step in understanding their intent. Recent domain-adaptive language models specifically designed for e-commerce and recommendation systems have achieved significant progress.

BLaIR [6] is a series of sentence embedding models pre-trained specifically for recommendation scenarios. BLaIR continues pre-training on the Amazon Reviews 2023 dataset, aiming to "bridge language and items," which is highly aligned with QUAKER's objectives. The model effectively captures correlations between item metadata and natural language contexts through contrastive learning that aligns embeddings of user reviews and item metadata. **TransRec** [11] attempts to create multi-facet identifiers for items (containing IDs, titles, and attributes) and align them with large language models, achieving a balance between distinctiveness and semantic richness. **e-Llama** [5] is a large language model adapted for the e-commerce domain, obtained by continuing pre-training the Llama 3.1 base model on 1 trillion domain-specific data tokens.

These works all demonstrate a common trend in academia and industry: general-purpose models are insufficient for solving domain-specific problems, requiring specialized design and training. QUAKER executes query understanding through an explicit category prediction stage, an approach that is more direct, interpretable, and proven efficient in experiments compared to relying entirely on implicit understanding from end-to-end models.

6.4 Evaluation Benchmarks and Datasets

High-quality datasets closely aligned with real-world scenarios are crucial for advancing product search research. The Amazon-C4 dataset used by QUAKER shares the common goal with mainstream evaluation benchmarks of addressing ambiguous and challenging search queries.

Amazon ESCI (Shopping Queries Dataset) [15] is an important benchmark in the product search domain. The dataset contains approximately 130,000 queries and 2.6 million manually annotated relevance judgments, covering English, Japanese, and Spanish. The dataset provides detailed relevance grades: Exact (perfect match), Substitute (functional alternative), Complement (complementary product), and Irrelevant (irrelevant). The query sampling strategy focuses on difficult queries, selecting queries challenging for multiple production baseline models through various methods (e.g., behavioral statistics, negations, price patterns).

Compared to ESCI, the **Amazon-C4** dataset we use creates a unique and challenging evaluation scenario by generating queries from user reviews. As shown in Table 3, Amazon-C4's average query length is significantly longer than ESCI (approximately 230 characters vs. 22 characters), better simulating complex, conversational search behaviors of users in the real world. This demonstrates that our dataset choice is reasonable and aligned with research trends in the field.

Additionally, **JDsearch** [12] is another important personalized product search dataset, containing real user queries and complete interaction information (clicks, add-to-cart, follows, and purchases), emphasizing personalization and real user behavior, providing valuable evaluation resources for this field.

Table 3. Query characteristics comparison between Amazon-C4 and ESCI datasets

Dataset	#Queries	Avg. Query Length (chars)
ESCI	27,643	22.46
Amazon-C4	21,223	229.89

In summary, building upon existing research achievements, QUAKER proposes an innovative, complete, and efficient solution targeting specific pain points in e-commerce search (ambiguous queries, massive product catalogs, computational efficiency). By combining category prediction with dense retrieval, QUAKER achieves

a balance between semantic expressiveness and computational efficiency, demonstrating significantly superior performance over baselines on the Amazon-C4 benchmark.

7 Result and Conclusion

Our work presents a novel two-step retrieval framework to address challenges in generalizing the obscure query retrieval framework, leveraging fine-tuned BERT and E5 embedding models to achieve a balance between computational efficiency and semantic richness. Our results demonstrate the effectiveness of this approach, achieving a 74.77% accuracy in retrieving ground-truth items within the top-200 results. Beyond quantitative evaluation, our qualitative analysis highlights the relevance of retrieved items even when ground truth is absent, suggesting a need for improved dataset design.

Through exploratory data analysis, we identified critical limitations in the Amazon-C4 dataset, particularly the impact of short metadata on retrieval accuracy. Addressing these limitations by excluding inadequate entries led to measurable performance improvements.

By bridging semantic gaps in product search and proposing actionable enhancements to dataset construction, this work provides both practical methodologies and a foundation for advancing retrieval systems. Future work could explore replacing BERT and E5 with stronger encoder backbones and extending on a greater proportion of the search space or even the entire 1 million sampled item dataset.

References

[1] Qingyao Ai, Yongfeng Zhang, et al. Learning a hierarchical embedding model for personalized product search. In *Proceedings of SIGIR*, 2017.

[2] Keping Bi, Qingyao Ai, and W. Bruce Croft. A transformer-based embedding model for personalized product search. In *Proceedings of SIGIR*, 2020.

[3] Aakanksha Chowdhery et al. Palm: Scaling language modeling with pathways. *JMLR*, 2023.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Christian Herold, Michael Kozielski, Tala Bazazo, Pavel Petrushkov, Patrycja Cieplicka, Dominika Basaj, Yannick Versley, Seysed Hadi Hashemi, and Shahram Khadivi. Domain adaptation of foundation llms for e-commerce. *arXiv preprint arXiv:2501.09706*, 2025.

[6] Yupeng Hou et al. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2303.00345*, 2024.

[7] McAuley Lab Hugging Face. Amazon-c4 dataset. https://huggingface.co/datasets/McAuley-Lab/Amazon-C4.

[8] McAuley Lab Hugging Face. Sampled item metadata dataset. https://huggingface.co/datasets/McAuley-Lab/Amazon-C4.

[9] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.

[10] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48, 2020.

[11] Xinyu Lin, Wenjie Wang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Bridging items and language: A transition paradigm for large language model-based recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1721–1731, 2024.

[12] Jiongnan Liu, Zhicheng Dou, Guoyu Tang, and Sulong Xu. Jd-search: A personalized product search dataset with real queries and full interactions. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2989–2998, 2023.

[13] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[14] Jiarui Qin, Jiachen Zhu, Bo Chen, Zhirong Liu, Weiwen Liu, Ruiming Tang, Rui Zhang, Yong Yu, and Weinan Zhang. Rankflow: Joint optimization of multi-stage cascade ranking systems as flows. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 814–824, 2022.

[15] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. Shopping queries dataset: A large-scale esci benchmark for improving product search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3479–3488, 2022.

[16] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

[17] Lidan Wang, Jimmy Lin, and Donald Metzler. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 105–114, 2011.

[18] Kai Zheng, Haijun Zhao, Rui Huang, Na Mou, Yanan Niu, Yang Song, Hongning Wang, and Kun Gai. Full stage learning to rank: A unified framework for multi-stage systems. *arXiv preprint arXiv:2405.04844*, 2024.

A Example Queries and Top-5 Results

A.1 Query 1: Shoes for Work and After-Work Wear

Query: I am looking for shoes that are super comfy, fit wonderfully, and can be paired with professional attire for work as well as after-work wear. I want them to be the perfect height, have a neutral color, and fit well. What more can a girl ask for in a shoe?

Top-5 Ranked Metadata:

1. **Rank 1:** Ryka Women's Devotion Plus 2 Walking Shoe. Get the comfort and performance you

need every time you exercise in this light and comfortable walking sneaker with exceptional cushion, shock absorption, and our powerful Made for Women fit. BEST FOR: High-performance fitness walking. PERFORMANCE TECH: RE-ZORB® responsive cushioning for shock absorption + impact protection. MADE FOR WOMEN FIT: Designed for a woman's unique foot shape, muscle movement, and build with a narrower heel, roomier toe, and softer foot cushioning. MATERIALS: Breathable engineered mesh + soft Lycra-lined tongue and collar with built-in cushion. CLOSURE: Lace-up front for a secure fit. INSOLE: Anatomical insole with extra arch + heel support. MIDSOLE: Lightweight EVA for soft cushioning. OUTSOLE: Eight-piece rubber sole for increased traction + durability. WEIGHT: 224 g/7.9 oz per shoe. HEEL-TO-TOE DROP: 11 mm.

2. **Rank 2:** quescu 2Pcs Valentine Gnomes Plush, Valentines Day Gnomes Decor Ornaments, Sweet Valentines Day Gifts for Him Her, Tiered Tray Party Decor Home Table Decorations (2pcs).
3. **Rank 3:** Bico Christmas Gnomes Ceramic Spoon Rest, House Warming Gift, Dishwasher Safe.
4. **Rank 4:** LEVKIDS Christmas Stocking, Swedish Gnome and Snowman Pattern Xmas Stocking, Holiday Party Decorations Fireplace Hanging Ornaments, Pack of 2.
5. **Rank 5:** D-FantiX Gnome Christmas Tree Topper, 27.5 Inch Large Swedish Tomte Gnome Christmas Ornaments Santa Gnomes Plush Scandinavian Christmas Decorations Holiday Home Décor with Plaid Hat.

A.2 Query 2: Cute Decorations with Gnomes

Query: I'm looking for cute and well-made decorations that can add instant adorable-ness to any space. I want something with floppy hats and soft beards, like gnomes. They should have their own unique styles and be decorative and cheery. I plan to add them to my growing collection of decorations. It would be great if they arrive promptly and are well packaged. I'm also looking for a good price point. Highly recommend!

Top-5 Ranked Metadata:

1. **Rank 1:** 3 Pack Christmas Gnomes Decorations Handmade Santa Gnomes Plush Swedish Tomte Elf Ornaments Scandinavian Christmas Decorations Indoor Home Decor for Shelf Table Fireplace Christmas Tree Xmas Gift.
2. **Rank 2:** Hey Dude Women's Wendy Lace-Up Loafers Comfortable & Lightweight Ladies Shoes Multiple Sizes & Colors.
3. **Rank 3:** konhill Women's Casual Walking Shoes Breathable Mesh Work Slip-on Sneakers.
4. **Rank 4:** Shoe Stretcher Women, 4-way Shoe Widener Expander Shoe Tree Shape for Wide Feet.
5. **Rank 5:** somiliss Chunky Sneakers for Women High Top Lace Up Shoes for Women Sneakers Nice Women's Shoes Chunky Trainers Female Sneakers.

A.3 Query 3: Mini Filter for Betta Fish

Query: I am looking for the best mini filter for my Betta fish. It should have adjustable flow since Betta fish don't require a lot of flow.

Top-5 Ranked Metadata:

1. **Rank 1:** AQUANEAT Mini Sponge Filter, Aquarium Sponge Filter for Betta Fish Tank with Airline Tubing and Control Valve, up to 3Gal.
2. **Rank 2:** Kucbraly Fish Tank Filter Cartridge for Aqueon Filter Cartridges.
3. **Rank 3:** FS-TFC 6-Stage Portable Water Filter 0.01 Micron UF and CTO Improving Tastes Water Purifier Survival Gear 1.5L/Min Fast Flow for Hiking, Camping, Travel, and Emergency Preparedness.
4. **Rank 4:** Ameliade Aquarium Decorations Fish Tank Artificial Plastic Plants & Cave Rock Decor Set, Goldfish Betta Fish Tank Accessories Small & Large Fish Bowl Decorations (8PCS).
5. **Rank 5:** CousDUoBe 2 Pack Betta Fish Leaf Pad Improves Betta's Health by Simulating The Natural Habitat - Natural, Organic, Comfortable Rest Area for Fish Aquarium.