Modeling shopper interest broadness with entropy-driven dialogue policy in the context of arbitrarily large product catalogs

Firas Jarboui
Gorgias
Paris, France

Issa Memari
Gorgias
Paris, France

Abstract

Conversational recommender systems promise rich interactions for e-commerce, but balancing exploration (clarifying user needs) and exploitation (making recommendations) remains challenging, especially when deploying large language models (LLMs) with vast product catalogs. We address this challenge by modeling the breadth of user interest via the entropy of retrieval score distributions. Our method uses a neural retriever to fetch relevant items for a user query and computes the entropy of the re-ranked scores to dynamically route the dialogue policy: low-entropy (specific) queries trigger direct recommendations, whereas high-entropy (ambiguous) queries prompt exploratory questions. This simple yet effective strategy allows an LLM-driven agent to remain aware of an arbitrarily large catalog in real-time without bloating its context window.

CCS Concepts

• **Computing methodologies → Multi-agent planning**.

Keywords

Recommender system, Large Language Models, e-commerce

ACM Reference Format:
Firas Jarboui and Issa Memari. 2025. Modeling shopper interest broadness with entropy-driven dialogue policy in the context of arbitrarily large product catalogs. In *Proceedings of Workshop on Generative AI for E-Commerce (GenAIECommerce 2025 @RecSys 25).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recommender systems are increasingly turning to conversational agents to create interactive, personalized shopping experiences. Instead of passively presenting a list of items, a conversational recommender can ask questions, understand user preferences in real-time, and explain or refine recommendations through dialogue[16]. The recent breakthrough of large language models (LLMs) has accelerated this trend, as LLMs can fluidly generate natural language and handle complex user utterances[4].

By leveraging LLMs, conversational recommenders (CRS) have the potential to combine the strengths of traditional recommendation algorithms (database knowledge, collaborative filtering signals) with the flexible reasoning and language capabilities of AI assistants[14]. This synergy, however, comes with new challenges. Chief among them is how to manage the dialogue policy – i.e., deciding when the system should ask clarifying questions versus when it should present actual recommendations – in a way that takes advantage of the LLM's capabilities without overwhelming it with information.

In this work, we propose an entropy-based routing strategy to address this challenge. The key idea is to quantitatively estimate the broadness or ambiguity of the user's request by looking at the distribution of retrieval scores for candidate items. Intuitively, if the retrieval results are tightly clustered around a few high-scoring items, the conversational agent should exploit this and make a recommendation. Conversely, if the scores are spread out, it signals ambiguity – the agent should explore by asking a clarifying question or preference query.

We start by setting up the orchestration framework of the developed conversational AI before focusing on the its behavior as a shopping assistant. We then detail the product search pipeline, we discuss the limitations of naive retrieval augmented generation (RAG) and how that led us to develop the entropy-driven dialogue policy. We conclude the paper with empirical evidence supporting this work and the research directions we want to take in the future. Through this work, we aim to demonstrate that modeling user interest broadness via entropy is an effective and novel way to enhance LLM-driven recommender systems, contributing to more intelligent and adaptive conversational recommenders in practice.

2 Multi–skill E-commerce AI Agent

We operate a production-grade AI agent that currently serves ∼200 **K incoming shopper messages per day** across live chat widgets and e-mails for *thousands* of e-commerce brands. Merchants expect from the agent to *both* remove purchase-blocking friction (e.g. “What's your refund policy?”) *and* actively increase basket conversion through persuasive, context-aware product recommendations[22]. Treating “support” and “sales” in isolation would miss critical dependencies (clarifying a shipping deadline often opens a sales opportunity, and vice-versa) and would force costly hand-offs. We therefore build a **multi-skill agent** that unifies these tasks inside a single control loop (similar unified assistant strategies are described in Alibaba's AliMe system for example [10]).

Figure 1 schematizes the pipeline, which is organised into **three routing nodes**.

GenAIECommerce 2025 @RecSys 25, Prague, Czech Republic

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

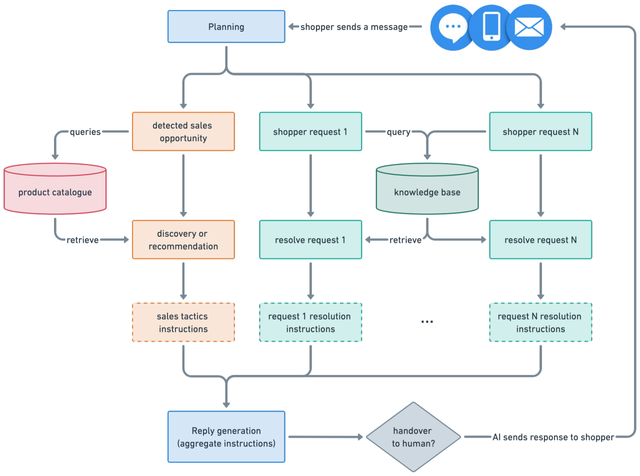
Planning
shopper sends a message
queries
detected sales opportunity
shopper request 1
query
shopper request N
product catalogue
knowledge base
retrieve
discovery or recommendation
resolve request 1
retrieve
resolve request N
sales tactics instructions
request 1 resolution instructions
...
request N resolution instructions
Reply generation (aggregate instructions)
handover to human?
AI sends response to shopper

Figure 1: High-level data flow of the agent (see §2).

(1) Planning node: Given the raw shopper message (and session metadata such as pages visited, cart contents, and prior orders), a large language model (LLM) decomposes the interaction into

- *support requests*—independent units of friction to be solved;
- a binary decision on whether a *sales opportunity* exists.

The planner outputs a structured JSON plan that names each request and, if applicable, flags the turn for sales activation. This approach aligns with multi-intent detection in dialog systems, where a single utterance may contain multiple intents that need handling[6].

(2) Resolution workflows: Each element of the plan is processed by a dedicated **workflow**, i.e. a guarded chain of LLM calls with retrieval-augmented generation (RAG) over merchant data[9].

- **Support-agent workflow** queries the merchant *knowledge base* (policies, FAQs, etc.) and generates a concise resolution instructions. This ensures accurate answers to policy or support questions by retrieving relevant documents[3].
- **Shopping-assistant workflow** (focus of the remainder of this paper) selects a *sales tactic*—*discovery* or *recommendation*—then produces tactic-specific *instructions* that will later surface to the shopper. This design follows the conversational recommendation paradigm where the system can ask clarifying questions to discover user needs or present product recommendations when confident, as shown to improve recommendation relevance in interactive e-commerce settings[22].

(3) Reply-generation node: A final LLM aggregates all instruction digests, resolves overlaps, checks coherence, and produces a single reply. If any digest signals low confidence or excessive complexity, the node triggers a *handover* to a human agent instead of auto-replying.

The remaining sections of this paper dive into the *shopping-assistant workflow* and, in particular, the detailed mechanism that handle routing between discovery and recommendation tactics.

3 AI Shopping Assistant

In every sales-oriented turn, the agent must decide between two conversational *tactics*:

(1) **Discovery** — ask clarifying questions or surface a small set of representative items to help the shopper refine their intent or explore the merchant's offer.
(2) **Recommendation** — once the intent is precise, present a concrete product (or bundle) and explain how it satisfies the expressed need.

Choosing the appropriate tactic at each turn is vital for both customer experience and conversion [8, 17]. Implementing this mixed-initiative behaviour raised two technical challenges:

Product context retrieval How can we fetch high-quality candidate products without overflowing the LLM context window? We start by answering this in subsection 3.1 by detailing our two-stage search pipeline and the distinction between our *identification* and *recommendation* retrieval operators.

Tactic selection. How can the system infer turn by turn whether the dialogue is still exploratory or ready for a recommendation? The subsequent subsections review our baseline solution, its limitations, and how those limitations led us to the entropy-based approach this paper introduces.

3.1 Product Search System

Our shopping assistant relies on a two–stage neural search pipeline whose input is an **LLM-generated query** assembled from the live context—conversation turns, pages visited, cart contents, and past orders. Each query is executed on the merchant's product catalogue in one of two modes:

- **Identification**: retrieve *matching* items to the expressed need.
- **Recommendation**: retrieve a *cross-sell* or *up-sell* items that complements the current interest.

Retrieval architecture: In a similar fashion to what prior research direction found to be relevant [5, 11, 12, 18], all the LLM generated queries and product descriptors are embedded with a **multilingual encoder** fine-tuned using a triplet loss (**E5** in our case[20]). We start by fetching candidates via an HNSW approximate-nearest-neighbour index[15]. Then we rerank the top-k (here k=50) with a transformer-based[19] reranker model trained under *binary cross-entropy* (BCE) on click-through labels, yielding a calibrated relevance score $s(q, p)$ used later for entropy computation (we used **RoBERTa** in our case[13]).

Training data and negatives: We exploit two complementary data sources to mitigate the exploration vs exploitation tradeoff of online recommender systems:

(1) **Out of distribution signals** collected organically without the AI agent intervention:
  - Identification: storefront search → landing-page pairs
  - Recommendation: cart co-occurrence tuples
(2) **In distribution signals** success events collected from the live agent conversation: LLM-generated queries → identification or recommendation pipeline → click through events

Negative examples are sampled uniformly from *(i)* other merchants' catalogues and *(ii)* disjoint collections within the same merchant, ensuring hard negatives without semantic overlap.

Cold-start robustness: Because the encoder is trained on *cross-merchant* data, it generalises to unseen shops; newly onboarded products are embedded on ingest and immediately indexed without additional fine-tuning—eliminating cold-start latency.

3.2 Baseline: LLM-driven dialogue policy

At each sales focused round of interaction with the shopper, we define the shopping assistant dialogue policy as the decision to be made between:

- Asking clarifying questions or showing representative items to help refine the shopper interest (we will refer to this as the Discovery behavior)
- Being assertive and pushing the shopper toward purshasing a specific item (shopping assistant Recommendation mode)

Our first generation shopping assistant relied on an LLM router to decide if we should go with either strategies. To achieve this, we combined **retrieval augmented generation (RAG)** with an LLM classifier. The workflow comprised three sequential phases:

(1) **Setup:** Two parallel LLM calls are issued:
   - *Interest classifier*—predicts the *customer interest stage* as either Discovery (intent unclear) or Interested (intent precise);
   - *Query generator*—produces one or more text queries conditioned on the full context (dialogue history, pages viewed, cart, etc.). These queries are targetting either the identification or the recommendation operators of our search engine.

(2) **Candidate retrieval:** Each query is executed by the two-stage search pipeline described in Section 3.1.

(3) **Tactic prompting** Depending on the predicted interest stage, the retrieved product snippets are injected into specialised prompts (either Discovery or Recommendation). The prompt output is an *instruction digest* forwarded to the reply-generation node.

This baseline is very straight forward; nevertheless, it revealed important shortcomings.

3.3 Observed Limitations

Context-dependent granularity: The LLM based interest stage classifier performed well when the shopper's goal was *orthogonal* to any specific product ("Can you ship before Friday?"). However, when the user explicitly mentioned a product or category, the classifier often misjudged whether the request was *vague* or *precise relative to the merchant's catalogue.*

This problem is analogous to query quality performance in retrieval systems [1, 7]. For instance, the query *"nails"* is broad in a specialised nail-supply store (hundreds of SKUs) but highly specific in a general beauty store with only a few nail items. Because the classifier does not have access to the full catalog breadth, it could not make this distinction.

A seemingly natural fix (providing the entire catalog as context for the LLM) is infeasible in practice as injecting thousands of products prohibitively increased hallucination risk. We could also add another intermediate prompt (between the **Candidate retrieval** and the **Tactic prompting** steps) but that would increase the first response time of AI agent, which is not desirable in practice as it increases the shopper drop off.

Shopping assistant discovery is shopper driven: Now that we understand the limitations of the LLM-driven dialogue policy (deciding to go with a Discovery or a Recommendation behavior in the next round), we needed to quantify the missed opportunity addressable market among organic shopper interactions. For this, we evaluated the shopper intentions of their first message.

We defined 52 intent categories and we refined a prompt to be aligned with a manually labeled balanced dataset of 50 examples of each category. We then used this prompt to infer the intent of 1M interactions sampled randomly over the last 6 months (January - June 2025). We report the distribution of the intents in Figure 2. The top 3 categories are Product:Details, Product:Usage and Product:Availability. They represent almost **70**% of the full dataset (**54**%, **10**%, and **5.4**% respectively).

These categories are basically indicating that almost always shoppers engage the AI with a clear product in mind. They are either asking about general information regarding the merchant catalog, specific details regarding the usage of a given product or its availability.

Depending on the broadness of these queries (i.e. how precisely can we pin point a specific product based on the shopper context), the shopping assistant dialogue policy should behave very differently. Intuitively, high quality queries and precise context with respect to the merchant catalog should lead to the Recommendation behavior, whereas vague queries should lead to the Discovery mode.

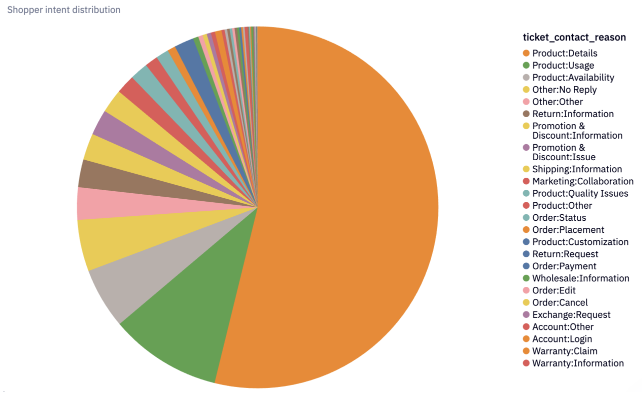

Figure 2: Distribution of shopper intents in e-commerce.

Looking at the scientific literature, we noticed that prior work investigated the correlation between query quality and the entropy of the similarity distribution[21]. This motivated our second-generation shopping assistant with **entropy-driven dialogue policy**, which leverages the *distribution shape* of retrieved scores rather than shopper context to decide between Discovery and Recommendation.

3.4 Modeling Query Breadth via Entropy

Let the merchant catalogue contain N items $\{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$. For an LLM-generated query $\mathbf{q}$ we obtain a calibrated re-ranker score $s(\mathbf{q}, \mathbf{p}_i) \in [0, 1]$ for each candidate product (Section 3.1). Over the top-k candidates we form a probability mass function

$$\mathbf{P}_i^q = \frac{s(\mathbf{q}, \mathbf{p}_i)}{\sum_{j=1}^{k} s(\mathbf{q}, \mathbf{p}_j)}, \qquad i = 1, \ldots, k.$$

Let $\mathcal{D}_q$ be the distribution over the top-k retrieved items for query $\mathbf{q}$, where the probability of item i is $\mathbf{P}_i^q$

Shoppers arriving with very precise queries will produce a highly skewed distribution $\mathcal{D}_q$, with one or a few items taking most of the probability mass (very high similarity for a small set of products, and very low for the rest). In contrast, a shopper with a vague or broad-interest query yields a much flatter, uniform-like distribution of scores (many items have similarly low scores). Prior work has in fact quantified query ambiguity or specificity by measuring the entropy of the distribution of retrieval results or user clicks[2].

In our context, this motivates using the entropy of $\mathcal{D}_q$ as a proxy for the breadth of the customer's intent. We define the **broadness score** as the normalized entropy of $\mathcal{D}_q$

$$B_k^N(\mathbf{q}) = \frac{-\sum_{i=1}^{k} P_i \log P_i}{\log k} \in [0, 1].$$

- $B_k^N(\mathbf{q}) = 0$: *minimum entropy*—all probability mass is concentrated on a single product, indicating **precise** intent.
- $B_k^N(\mathbf{q}) = 1$: *maximum entropy*—scores are nearly uniform, signaling a **vague** query.

Entropy estimator: The broadness (or normalized entropy) of any given query $\mathbf{q}$, contextualized to the whole product catalog is $B_N^N(\mathbf{q})$. In practice and for computational concerns, we only have access to the broadness contextualized to the top-k nearest retrieved items $B_k^N(\mathbf{q})$.

This means that the product catalogs items that will not be taken into account when computing the broadness are necessarily of lower similarity. Subsequently, the top-k broadness $B_k^N(\mathbf{q})$ will always over-estimate the full product catalog broadness score $B_N^N(\mathbf{q})$ as we will not see the long tail of irrelevant items.

This begs the question: *How far off is the top-k broadness score compared to the full product catalog contextualized one.* While we do not derive convergence bounds of this estimator in this paper, we investigated empirically the error over the organic search dataset (storefront search → landing-page pairs) averaged across all of Gorgias 15**K** supported stores. We report this in Figure 3.

What we noticed is that average errors converges to 0 quite rapidly, and that with 50 neighbors we already have a reliable signal for our practical needs.

3.5 Entropy-Driven Dialogue Policy

Using the broadness score, we design a simple entropy-sensitive dialogue policy for the shopping assistant. At each dialogue turn, a query generator LLM defines two types of search queries based on the user's utterance and context:

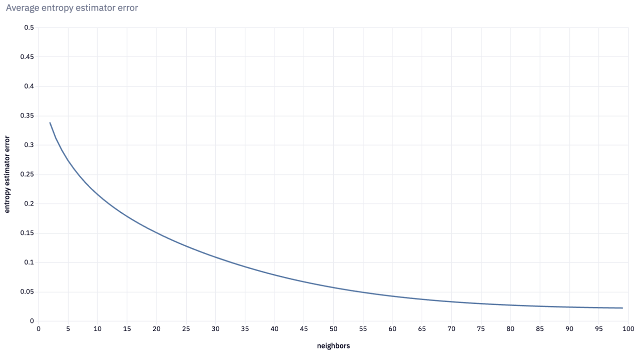

Figure 3: Convergence of the entropy estimator.

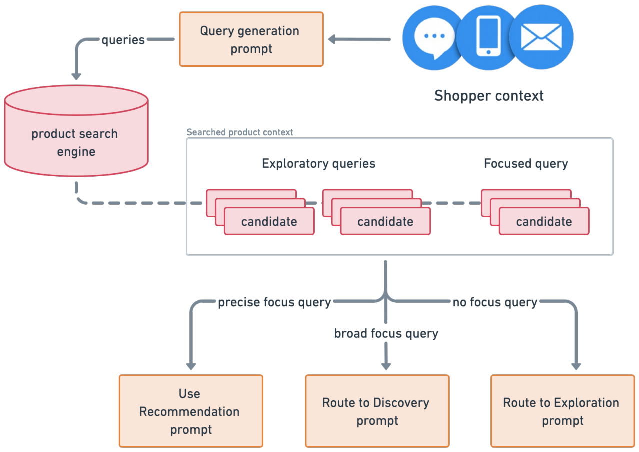

Figure 4: Entropy controlled sales tactic (see §3.5).

Exploratory one or more catalog-level identification or recommendation searches that reflect the user's general context (pages viewed, cart contents, dialogue history).

Focused a single identification query distilled from the most recent utterance. This is our best guess at the shopper's immediate target.

We execute all candidate queries through the product search engine, and then compute the broadness score for the focused query B_{focused}. The dialogue policy then branches based on the presence of a confident focused query and its entropy value:

- **No focus query** If the LLM cannot produce a plausible focused search, we route the conversation to an Exploration phase. The system aims to pique the shopper's interest by showcasing a diverse set of contextually relevant products or asking a broad question. The assistant might show popular categories or related items to help the user articulate their needs.
- **Focus query present**:

- $B_{\text{focus}} < \tau_{\text{merchant}}$: This indicates low entropy (highly concentrated relevance on a few items) – the user's intent appears precise. The assistant will confidently suggest the top result or a very narrow set of products that are most likely relevant to the shopper's need. This corresponds to a high-confidence scenario where exploitation is favored over exploration.
- $B_{\text{focus}} \geq \tau_{\text{merchant}}$ This indicates high entropy (a broad or ambiguous query) – the user's intent is still vague or underspecified. The assistant will ask one or two clarifying questions or request additional details from the shopper. The goal is to reduce the entropy in the next turn by encouraging the user to narrow down their requirements.

An important advantage of this entropy-driven policy is that it remains continuously catalog-aware without expensive context injection. The decision logic depends only on the distribution of retrieval scores, which means any newly indexed product or change in the catalog can immediately influence the outcome in real-time.

Threshold selection: The cutoff level at which we switch between strategies controls a trade-off between how many clarifying questions the shopping assistant would ask during discovery and how likely are we to identify the actual product of interest for the shopper when we move into the recommendation mode.

The threshold τ_{merchant} is bucketed into three aggressiveness presets (*educational*, *balanced*, *pushy*), allowing merchants to trade off depth of discovery against conversion pressure. These presets are experimental and can be refined, but they illustrate how entropy-based routing can flexibly support different sales tactics.

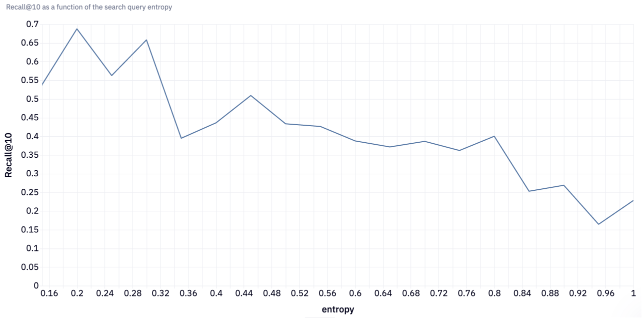

Figure 5: Recall at 10 plateau for different values of entropy

In order to define a the levels of each bucket, we used the organically collected user search queries from the shoppers session data which were mapped into the next landing page after the search. For each query, we measure the recall at 10 our retrieval system (binary, 0 or 1) and we measure the entropy of the search.

We then measured the average normalized entropy of search queries of same entropy buckets, and we report the average recall at 10 as a function of the entropy bins if Figure 5.

Looking at the data we noticed that recall is showing plateaus of similar recall at 10 before dropping around the values (0.3 and 0.8):

- **lower than 0.3:** we have a probability of around 0.6 of retrieving the correct product item among the top 10 retrieved ones
- **between 0.3 and 0.8:** the probability is stable around 0.4
- **beyond 0.8:** the success here drops to 0.2 for identifying the next landing product page for queries with such broadness score

We used these levels as an educated guess for the threshold τ_{merchant} of the three aggressiveness presents mentionned above.

4 Impact on engagement and future research

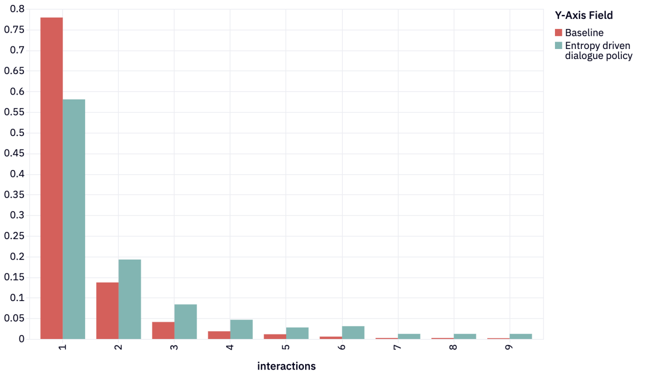

Figure 6: The shopping assistant powered with the entropy driven dialogue policy had longer conversations with shoppers than our baseline

The goal of this direction of work was to increase the shopper engagement with the shopping assistant as a success metric. We define engagement as the average number of rounds of interactions a shopper has with the AI during the same session.

We ran an AB test on real customers to evaluate how different dialogue policies affect the conversation length, the distribution of conversation length is reported in Figure 6.

We can clearly see that the entropy driven policy led to more engaging conversations with the shopper as the shopping assistant was not pushing items for sale but trying to collect more context from the user to give them a better recommendation.

We measured during the AB test whether this affected the conversion rate of conversations, but while we noticed a slight increase, it remained statistically insignificant to be reported in these results. We believe that there are way more factors involved in affecting the conversion rate for the dialogue policy to have a direct effect on it. This work merely covers one aspect of the complex dynamic involved with the conversational recommender system that interacts with live shoppers.

A future research direction that we are actively looking into is how to make the shopping assistant proactive while minimizing the risk of distrcting the end user (instead of waiting for the customer to engage the AI, devise a policy that would review the shopper browsing data and context to decide if the shopping assistant should offer help organically).

Acknowledgments

We thank **Gorgias** for supporting this research and enabling its deployment at scale. We also wish to acknowledge the engineering and analytics contributors behind the implementation of the shopping assistant, whose work streamlined the measurement of conversational strategies in production and made it possible to rigorously evaluate entropy-based dialogue policies in live shopper interactions.

References

[1] Steve Cronen-Townsend, Yun Zhou, and W Bruce Croft. 2002. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.* 299–306.

[2] Huizhong Duan, Emre Kiciman, and ChengXiang Zhai. 2012. Click patterns: An empirical representation of complex query intents. In *Proceedings of the 21st ACM international conference on Information and knowledge management.* 1035–1044.

[3] Song Feng, Hui Wan, Chulaka Gunasekara, Siva Sankalp Patel, Sachindra Joshi, and Luis A Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. *arXiv preprint arXiv:2011.06623* (2020).

[4] Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. 2023. A large language model enhanced conversational recommender system. *arXiv preprint arXiv:2308.06212* (2023).

[5] Niklas Freymuth, Dong Liu, Thomas Ricatte, and Saab Mansour. 2025. Hierarchical Multi-field Representations for Two-Stage E-commerce Retrieval. *arXiv preprint arXiv:2501.18707* (2025).

[6] Rashmi Gangadharaiah and Balakrishnan Narayanaswamy. 2019. Joint Multiple Intent Detection and Slot Labeling for Goal-Oriented Dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 564–569. doi:10.18653/v1/N19-1055

[7] Claudia Hauff, Djoerd Hiemstra, and Franciska de Jong. 2008. A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM conference on Information and knowledge management.* 1419–1420.

[8] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–36.

[9] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[10] Feng-Lin Li, Minghui Qiu, Haiqing Chen, Xiongwei Wang, Xing Gao, Jun Huang, Juwei Ren, Zhongzhou Zhao, Weipeng Zhao, Lei Wang, et al. 2017. Alime assist: An intelligent assistant for creating an innovative e-commerce experience. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* 2495–2498.

[11] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based product retrieval in taobao search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining.* 3181–3189.

[12] Juexin Lin, Sachin Yadav, Feng Liu, Nicholas Rossi, Praveen R Suram, Satya Chembolu, Prijith Chandran, Hrushikesh Mohapatra, Tony Lee, Alessandro Magnani, et al. 2024. Enhancing Relevance of Embedding-based Retrieval at Walmart. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management.* 4694–4701.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[14] Yuanxing Liu, Weinan Zhang, Yifan Chen, Yuchi Zhang, Haopeng Bai, Fan Feng, Hengbin Cui, Yongbin Li, and Wanxiang Che. 2023. Conversational Recommender System and Large Language Model Are Made for Each Other in E-commerce Pre-sales Dialogue. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 9587–9605. doi:10.18653/v1/2023.findings-emnlp.643

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

[16] Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Zi Huang, and Kai Zheng. 2021. Learning to ask appropriate questions in conversational recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval.* 808–817.

[17] Zhaochun Ren, Xiangnan He, Dawei Yin, Maarten de Rijke, et al. 2024. Information Discovery in E-commerce. *Foundations and Trends® in Information Retrieval* 18, 4-5 (2024), 417–690.

[18] Nicholas Rossi, Juexin Lin, Feng Liu, Zhen Yang, Tony Lee, Alessandro Magnani, and Ciya Liao. 2024. Relevance filtering for embedding-based retrieval. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management.* 4828–4835.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

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

[21] Oleg Zendel, Binsheng Liu, J Shane Culpepper, and Falk Scholer. 2023. Entropy-based query performance prediction for neural information retrieval systems. (2023).

[22] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management.* 177–186.

Received 10 July 2025; revised -; accepted 30 August 2025