Shoppers Also Asked: Generating Related Questions for E-commerce Search

Saar Kuzi, Zhiyu Chen and Shervin Malmasi

Amazon.com Inc. *Seattle, WA, USA*

Abstract

Suggesting related questions on search result pages is a popular feature of search engines which helps users satisfy their information needs. In this paper, we study the problem of related question generation in e-commerce search. We study the effectiveness of various approaches for the task that use a Large Language Model (LLM) with different inputs. In particular, by experimenting with the TREC Product Search dataset, we show that leveraging information from the top products in the result list can improve the performance of an approach that only uses the query. Our analysis also reveals that the most promising approach for the task is to use the different types of product information to generate questions and let the LLM select the best ones among them.

Keywords

Large Language Models, E-commerce Search, Question Generation

1. Introduction

Web search and product search are two key retrieval technologies for helping users find information and products, respectively. While both are search tasks, they are fundamentally different and use distinct techniques [1]. Web search relies on indexing unstructured documents, while product search uses structured data. Among other things, they also have distinct approaches to query processing, ranking, and user interfaces. While these differences have resulted in distinct services (e.g. Google vs. eBay), users on a *shopping mission* [2] often need to use both to make a purchase decision. Customers use information search to gather product knowledge, update their beliefs, and then refine their product search [3, 4]. This results in a cyclical process of switching between the two tasks. We explore how Large Language Models (LLMs) can help bridge the gap between the two experiences by presenting information-seeking questions within product search. For example, a user that issues the query "*juicer*" in a product search engine can be show the question "*What juicer features help with easy cleaning?*". Then the user can click on the question to get an answer to help them choose the right product.

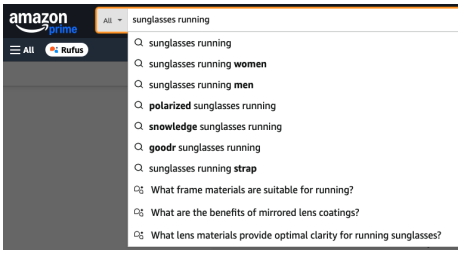

Figure 1: An example of how question suggestions for a search query can be integrated into autocomplete (the last 3 entries).

Presenting related questions to users in the Search Results Page (SERP) is a popular feature of search engines like Google, also known as People Also Asked (PAA) [5] and helps users better satisfy their information need [6]. We study the problem of Shoppers Also Asked (SAA), aiming to enrich e-commerce search results with related questions that can help users make a purchase decision [7, 8].

eCom'25: ACM SIGIR Workshop on eCommerce, July 17, 2025, Padua, Italy

skuzi@amazon.com (S. Kuzi); zhiyuche@amazon.com (Z. Chen); malmasi@amazon.com (S. Malmasi)

We envision that in the near future LLM technology will lead to integrated information discovery within e-commerce, and such questions can serve as entry points for this integration, as shown in Figure 1.

While presenting shopping questions has many benefits, such as efficient information access and making it easier for uninitiated shoppers to begin their journey, it has not been adopted at scale thus far. The reasons for this are twofold. First, unlike Web search engines, e-commerce systems do not have access to large amounts of log data from which to mine questions [9]. Second, answering these questions requires a separate document retrieval system. The recent remarkable success of LLMs has made it possible to integrate powerful generation capabilities into virtually any application, and can solve both of these problems.

This paper takes a first step in this direction by studying the task of generating related questions in e-commerce search. Using LLMs for the task has the potential of alleviating the problems that have previously impeded the adoption of the related question functionality in e-commerce search engines. In particular, the vast amount of world knowledge in LLMs facilitates the generation of high quality questions, thus removing the dependency on logs [10, 11]. While we do not investigate it, these models can also be used to answer the suggested questions with high accuracy.

The input to our system includes a query and a result list of products from a retrieval system. Using this input to represent a shopping mission, we aim to generate a set of questions that would interest many users. We study different generation approaches using different inputs, which include the search query as well as information from the top products in the result list (like product titles and reviews).

We evaluated the different question generation approaches with the public TREC Product Search dataset using automatic LLM-based annotations [12] as well as human evaluation. Our results demonstrate the effectiveness of adding top products from the result list to the input as compared to using only the query. Specifically, using several types of product information to generate questions, either by combining them all together in the same prompt or by using an LLM to select the best ones, is the best performing technique. Our empirical evaluation also provides some understanding on the sensitivity of question generation performance to different query characteristics such as the product category, specificity, and retrieval difficulty. Finally, we conclude with a case study to analyze the different characteristics of the various approaches studied.

2. Related Work

The problem of generating related questions for a search query has already been studied in the context of Web search [6, 13]. In one work, the idea was to search for questions in the logs and select the best ones based on their similarity to the query [13, 14]. Another study investigated a multi-lingual setting to generate questions from passages using a sequence-to-sequence model. In this paper, we study the problem of generating questions for the e-commerce setting which is inherently different than Web search. Furthermore, we focus on the setting where no query logs or external information is available which makes these works inapplicable. Another line of work related to this paper is question answering on e-commerce platforms (e.g., [1, 15, 16, 17, 18]). More specifically, some works have focused on the task of question generation to either be used for QA retrieval [1] or to suggest questions to users in the context of a single product [1, 8, 19, 20, 21]. This paper focuses on a different problem of generating questions in the context of a search result page which poses novel challenges such as addressing a potentially noisy result list and generating questions that should be applicable to a user's information need rather than just a single product of interest. Finally, another line of work which is also relevant to this paper is query suggestion which was studied for both the general domain [22] and the shopping domain [23]. While query suggestions aim to help users reformulate their queries, our work focuses on suggesting questions that users can ask to assist them in making a purchase decision.

3. Problem Definition

We study the problem of generating related questions to a search query in e-commerce search. The input is a query q and a ranked list of top n products $P = \{p_1, p_2, ..., p_n\}$ returned by the e-commerce retrieval engine. Each product p is represented by different types of textual information. In this study, we focus on the title, description, and reviews of the product; we leave the study of other fields like product attributes and bullet points for future work. Given q and P, our goal is to generate a set of k related questions $RQ = \{rq_1, rq_2, ..., rq_k\}$. Since our focus is on the setting of e-commerce search engines, the questions should be generated such that they would help users to make a purchase decision. Another requirement is that we want the questions to have the right level of specificity such that they are interesting to as many users as possible; generating personalized questions is an alternative strategy to address this, but is not in the focus of this paper. Also note that we do not aim to generate a ranked list of questions. This is because the ranking should be done based on user engagement, ideally via an online algorithm. The set of questions will be then presented to users where they can click on them to get a quick and useful response. This paper focuses solely on question generation; online question ranking and answer generation are out of our scope and left for future work.

System Instructions for *Question Generation*

You are an AI assistant designed to generate questions that many users would be interested in knowing their answer after issuing a search on an e-commerce website. You will receive in the input a user query [and the top search results]. Your goal is to generate questions related to that query. [You are encouraged to use the information from the search results to come up with the most interesting questions. If any of the search results are not relevant, you can ignore them.] Generate exactly ten (10) related questions the users may be interested in asking.

Related questions requirements: (1) The questions must be relevant to the query. (2) The questions must be about broad/general topics related to the query. (3) The queries should not be specific for a single item in the result list but related to the query in general. (4) The questions must help the users in acquiring knowledge that can help them make a purchase decision. (5) If you don't understand the query or if it is inappropriate, just provide general questions that are relevant to any product.

System Instructions for *Evaluation*

You are an AI assistant designed to predict how many users would find a question useful for making a purchase decision. In an online e-commerce website, users are searching for products using a search query and are presented with a list of questions that they tap on to get an answer. You are given in the input a search query, and two product-related questions. Your task is to judge which question is more interesting and useful to users.

A good question should: (1) Ask additional information that is not asked by the user query. (2) Ask about the general knowledge regarding products instead of asking for product recommendations. (3) Help users understand a specific aspect regarding related products.
Along with your selection, provide a short explanation for your reasoning. If two questions are equally good, use a special score of 0.

Figure 2: The system prompts used for question generation and evaluation. In the question generation prompt, text in the brackets will only appear when search results are used.

4. Question Generation using LLMs

The focus of this study is on how to effectively leverage an LLM to generate related questions for e-commerce search engines. To this end, we investigate the performance of several zero-shot prompts

that use different types of input from the SERP; the system instructions can be found in the top part of Figure 2. We decided to focus on zero-shot methods since we assume the setting where no information is available for us regarding what kind of questions users might ask and we would like to provide the LLM with flexibility to generate the best questions it can. We experiment with the following approaches.

Query-based Generation The most straightforward approach that we study uses only the query as an input to the LLM. The advantage of only using the query is that the model has the greatest flexibility in generating questions since it is provided with the most minimal input. Furthermore, the quality of the generated questions is expected to be less sensitive to noise in the result list. Using only the query is also likely to have some disadvantages. In particular, the model may not have enough knowledge to understand challenging domain-specific queries without some context from the result list. For example, the model may have difficulty in knowing some brand names or specific model numbers of products. Furthermore, since we rely mostly on the LLM's world knowledge, there is the risk of asking questions that are not interesting or useful with respect to the actual products presented in the result list.

Retrieval-based Generation To mitigate the risks of relying solely on the search query, we experiment with prompts that include information about the top products in the result list. We examine the effectiveness of using different types of product information to represent it in the prompt. The first two approaches use either the **Title** or the **Description** of the product. We also experiment by representing a product with a customer **Review** which is randomly selected from the set of reviews. We note that selecting a random review may not result in an optimal performance but is sufficient for us to obtain initial insights for the effectiveness of using reviews for question generation; we leave the study on how select the best set of reviews for future work.

Combining Multiple Product Fields Finally, to understand whether the different approaches are complementary, we study the performance of combining them together. The **All Fields** approach uses the query and all types of product information in a single prompt. An alternative approach that we study is **Fusion** which pools all questions from the different fields and prompts the LLM to select the most promising ones out of them.

5. Experiments and Results

5.1. Experimental Setup

5.1.1. Dataset

We use the publicly available 2023 TREC Product Search dataset [12]. The dataset includes a collection of 1.6M products with information such as title, description, and reviews. For the query set, we used a sample of 1,600 queries from the pool of all queries (30.3K). To examine the performance of different types of queries, we performed sampling as follows. First, we are interested in studying the performance for queries with different level of specificity. To this end, we used an LLM in a few-shot manner to classify all queries into either "broad" or "specific". On a high level, we define broad queries as ones that are applicable to a product category or a large group of products. Specific queries, on the other hand, apply to only a small number of products. Furthermore, to increase the classification accuracy, we filter the queries based on length, ensuring that broad questions have between one to three tokens, while specific queries can have between six to ten. The second aspect of queries that we study is the product category. To determine the product category of a query, we select the category that is the most frequent in its relevant products. We then narrow down the product types to the ones that have at least 100 queries in both the broad and specific type. This process results in 8 product categories and a final test set of 1,600 queries; several examples of queries and their classification are provided in Table 1.

Table 1
Examples of queries from the test set.

Query	Specificity
tatoos sleeves	Broad
desktop computers	Broad
vacuum cleaner carpet	Broad
cat bed set for 10 year old	Specific
trecking gear backpack with back support system	Specific
little mermaid lego friends lego sets	Specific

5.1.2. Implementation

For the implementation of the retrieval system, we used the Pyserini library [24]. For the retrieval algorithm, we used dense retrieval which is implemented with a Faiss engine and the TCT-ColBERT-V2 embeddings for the products and the query [25]. We examined the performance of the various approaches using the top 10 results.[1] For the LLM, we used the Claude 3 Sonnet and GPT-4o models and set the hyper-parameters, such as temperature, to default values. For each query, we generate exactly 10 questions. Since we are interested in using an LLM with state-of-the-art instruction following capabilities, we do not compare to other open source models.

5.1.3. Evaluation Approach

For the evaluation, we mostly relied on using an LLM-as-a-judge approach with the Claude model; studying the effectiveness of different LLMs as evaluators is out of the scope of this paper and we leave it for future work. Using LLM-as-a-judge has two main advantages over using human annotators. First, LLMs have vast world knowledge which can be beneficial in assessing the usefulness of questions for a product as compared to humans which may not be familiar with certain product categories. Second, the usefulness of questions can be subjective, requiring potentially many annotators to reach an agreement while LLMs can be prompted to determine if a question is potentially useful for many users. Motivated by previous work which demonstrated the effectiveness of LLM-based pair-wise evaluation [26], we ask the LLM to decide between two questions which one is more useful to users in making a purchase decision (they can also be annotated as equally good) along with a short explanation for its reasoning; the system instructions can be found at the bottom part of Figure 2. We note that a more accurate way to measure the usefulness of questions for making a purchase decision would be to quantify the purchase behavior of a user after clicking the question. We leave this direct evaluation approach for future work and use the LLM-as-a-judge technique instead as a proxy to help us gain initial understanding of the effectiveness of different approaches for the task.

For calculating the evaluation metrics, we compare between every pair of questions that are at the same position, resulting in 16K comparisons in total. Based on the LLM annotations, we report the win-rate of an approach that uses information from the result list compared to only using the query. The win-rate is defined as the portion of winning cases out of all of the cases where the questions are not equally good. An alternative approach would be to perform point-wise evaluation. However, our preliminary investigation of the point-wise strategy showed that all approaches generate good quality questions in at least 85% of the cases with no major differences. We hence focus on the pair-wise evaluations to better highlight the differences between approaches. Finally, we also conduct a pair-wise human annotation experiment with three annotators on a set of 100 questions pairs.

[1] We also experimented with 5, 15, and 20 products which resulted in similar performance.

5.2. Experimental Results

5.2.1. Main Results

The results in Table 2 demonstrate the strong performance of the approaches that leverage product information from the result list. In particular, we can see that for the vast majority of cases, the win-rate is greater than 0.5, indicating improvements when using almost any type of information and LLM. Comparing the performance of broad and specific queries, we can see that in most of the cases, the win-rate is similar. Still, we can see that in some cases, questions for specific queries perform slightly better than for broad ones. One possible explanation for this can be that for specific queries, since they usually refer to a single or a very small number of products, leveraging information from the result list makes it easier to identify useful product aspects, where for broad queries it may be sufficient to just rely on the knowledge of the LLM.

Table 2
The win-rate of questions generated using product information from the result list compared to only using the query. The best result in a column is boldfaced.

	Claude 3			GPT-4		
	All Queries	Broad	Specific	All Queries	Broad	Specific
Title	**0.54**	**0.54**	**0.54**	0.55	0.55	0.55
Reviews	0.51	0.50	0.52	0.54	0.54	0.54
Description	0.48	0.48	0.48	0.59	0.59	0.59
All Fields	0.51	0.50	0.51	**0.61**	**0.61**	**0.61**
Fusion	0.52	0.51	0.53	0.56	0.55	0.57

The results also show that GPT performs better than Claude, especially for Description and All Fields. Our manual examination of the results indicated that the GPT model has better capability to filter out noise in the relatively lengthy context of those two approaches. Focusing on the results from GPT, we can see that the best approach is to use all types of product information to generate questions. Examining the individual product fields, we can see that using the description is the most promising approach, but it requires a good capability of the model to distill the useful information from it. Finally, we can see that using just the title is the most robust to model changes.

To further validate the results from the automatic evaluation, we performed human annotations. Specifically, we focused on the All Fields approach and randomly sampled 100 question pairs to compare it with the Query approach. We asked three annotators to determine the winner between every pair of questions and all annotators were assigned the same questions. The results showed that for 83% of the questions, there was a majority agreement on a winner. For this portion of questions, the win-rate was measured as 54%, attesting to the strength of the All Fields approach. Finally we measured the overall inter-annotator agreement using Fleiss Kappa [27] and achieved the value of 0.24 which indicates slight agreement. The relatively low value of agreement may be due to the subjectivity of the task which may require a larger number of annotators; we leave the extension of such human study for future work.

Table 3
The question win-rate for queries belonging to different product categories. We report the win-rate of an approach compared to only using the query and the percentage of win-rate change compared to the group of all queries.

	Title				Reviews				Description			
	Claude 3		GPT-4		Claude 3		GPT-4		Claude 3		GPT-4	
All Queries	0.54	-	0.55	-	0.51	-	0.54	-	0.48	-	0.59	-
Electronics	0.57	6%	0.61	12%	0.51	1%	0.58	7%	0.52	7%	0.64	8%
Clothing, Shoes & Jewelry	0.56	4%	0.56	2%	0.51	0%	0.58	6%	0.49	3%	0.60	2%
Home & Kitchen	0.55	2%	0.56	2%	0.53	4%	0.55	2%	0.49	2%	0.60	2%
Toys & Games	0.55	1%	0.57	3%	0.51	0%	0.56	3%	0.50	4%	0.61	3%
Sports & Outdoors	0.53	-2%	0.52	-5%	0.52	2%	0.52	-3%	0.49	1%	0.55	-6%
Beauty & Personal Care	0.52	-3%	0.54	-2%	0.53	5%	0.53	-3%	0.49	2%	0.57	-2%
Tools & Home Improvement	0.51	-6%	0.54	-1%	0.50	-2%	0.53	-3%	0.44	-8%	0.57	-3%
Health & Household	0.53	-2%	0.50	-9%	0.46	-9%	0.49	-10%	0.43	-11%	0.54	-7%

5.2.2. Product Category Analysis

We analyze the question generation performance for queries from different categories in Table 3. Overall, we can see that most result list-based approaches provide consistent improvements across different product categories with most win-rates being higher than 0.5. It is also interesting to see that some categories are uniformly good/bad performing across different approaches and models. For example, while queries in the Electronics category have performance that is at least 7% higher than all queries, for Health & Household there is performance degradation of around 10%. One possible reason for the difference across categories is that some of them may have queries that are easy for the LLM to just leverage its internal knowledge to generate useful questions while for others, there can be domain specific details that the model is lacking understanding of. This can happen for certain electronic products for niche areas, for instance. We note that using the product category can be a useful tool that is easy to implement in a production system to control/block the generation for certain use cases.

5.2.3. Query Difficulty Analysis

In Table 4, we report the performance of different groups of queries with different levels of retrieval performance (as measured by $ndcg@10$). The results indicate mostly positive correlation between the retrieval effectiveness and the question generation performance. This finding aligns with the reasoning that relying on irrelevant products has the potential of generating irrelevant questions to the query. The results also show the sensitivity to the retrieval accuracy which is to a lesser extent in the case of GPT compared to Claude. This finding aligns with our previous observation regarding GPT's better ability to handle noise. Finally, we can see that even though there is some sensitivity with respect to the retrieval accuracy, all approaches, for most levels of accuracy, outperform the query approach which indicates stable performance when using product information from the result list.

Table 4
Question performance for queries with different levels of retrieval accuracy as measured by $ndcg@10$.

	Claude			GPT		
$ndcg@10$	$[0.0, 0.1)$	$[0.1, 0.5)$	$[0.5, 1]$	$[0.0, 0.1)$	$[0.1, 0.5)$	$[0.5, 1]$
Title	0.53	0.54	0.56	0.55	0.56	0.54
Reviews	0.49	0.51	0.53	0.55	0.53	0.55
Description	0.48	0.48	0.49	0.58	0.59	0.59
All Fields	0.49	0.51	0.52	0.59	0.61	0.62
Fusion	0.50	0.52	0.55	0.56	0.55	0.57

5.2.4. Question Position Analysis

We analyze the win-rate of questions with respect to their position in the output of the LLM in Figure 3. The graphs reveal a very interesting trend of question performance. According to the results, a very high performance is observed for the first question which then drops and followed by a steady performance increase that peaks at the last question. A possible explanation for this can be the better likelihood of the model to come up with the most interesting question at the first position as no other questions were generated previously. The performance increase afterwards, on the other hand, may be attributed to the presence of prior tokens that better condition the generation of the subsequent new questions. Finally, we can see that regardless of the observed trend, the performance of questions in different positions is mostly consistently better than when only using the query.

5.2.5. Case Study

Several examples of questions generated by either using only the query or by using a single field of the product are presented in Table 5. The first observation from the table is the relatively high diversity of questions across all approaches which can be attributed to each one relying on different types of

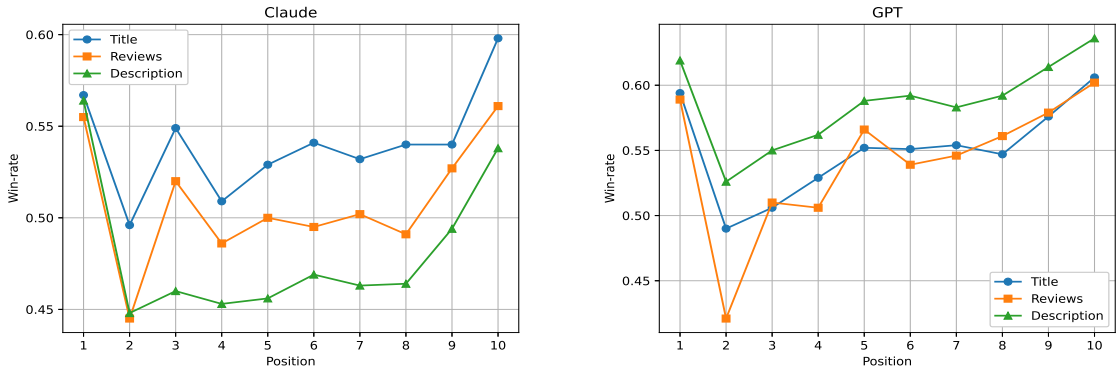

Figure 3: Win-rate of the questions generated in different positions by the LLM.

data sources. Furthermore, most questions seem relevant and useful which makes it reasonable to use approaches that leverage several sources of information like All Fields or Fusion. It is also interesting to see that all approaches generate questions in a decreasing order of broadness. For example, for all approaches the first question asks about the features or benefits of a specific product. Comparing the Query approaches to the others, we can see that the result list-based approaches are generally longer and more specific by mentioning non-trivial aspects of products that are likely extracted from the result list. For example, mentioning the "stabilization technology" of a GoPro. Finally, we can see that the different result list based approaches can differ in style and topic which resonates with their respective data source. For instance, using reviews, we can see questions that are likely to be addressed by reviews like quality and potential problems. Title questions, on the other hand, mention broader technical product aspects that can be found in titles such "weatherproof".

Table 5
Examples of questions generated for the query "go pro" with GPT.

Query
What are the key features to look for in a GoPro camera?
How does the battery life of different GoPro models compare?
What are the available accessories for GoPro cameras?
Title
What features should be considered when choosing an action camera like a GoPro?
How does image stabilization impact the quality of the videos in action cameras?
What are the benefits of having a weatherproof and shockproof monopod for GoPro cameras?
Reviews
What are the most important features to look for in a GoPro camera for action shots?
How does video quality of recent GoPro models compare to other action cameras?
Are there common issues with GoPro cameras stopping mid-recording, and how can they be resolved?
Description
What are the key features to look for when choosing a GoPro camera for action photography?
How does GoPro's stabilization technology compare to other action cameras in the market?
What are the benefits of having built-in WiFi and Bluetooth in a GoPro camera?

6. Conclusion

This paper studied the problem of generating related questions to a search query in e-commerce search engines, in line with the increasing integration of Generative AI experiences in e-commerce [28]. Our empirical analysis demonstrated the benefit of using information from products in the result list as input to the LLM as compared to prompting it only with the search query. Furthermore, we showed that leveraging various types of product information is the most promising approach. For future work, we plan to conduct an online study to more accurately measure the usefulness of questions to real users.

Declaration on Generative AI

This work studies the effectiveness of generative AI approaches for the task of generating related questions for e-commerce search. Generative AI tools were not employed in any way to assist with the writing of this paper.

References

[1] Z. Chen, J. Choi, B. Fetahu, O. Rokhlenko, S. Malmasi, Generate-then-retrieve: Intent-aware FAQ retrieval in product search, in: S. Sitaram, B. Beigman Klebanov, J. D. Williams (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 763–771. URL: https://aclanthology.org/2023.acl-industry.73. doi:10.18653/v1/2023.acl-industry.73.

[2] P. Sondhi, M. Sharma, P. Kolari, C. Zhai, A taxonomy of queries for e-commerce search, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 1245–1248.

[3] J. Rowley, Product search in e-shopping: a review and research propositions, Journal of consumer marketing 17 (2000) 20–35.

[4] F. Branco, M. Sun, J. M. Villas-Boas, Optimal search for product information, Management Science 58 (2012) 2037–2056.

[5] Y. Matias, D. Keysar, G. Chechik, Z. Bar-Yossef, T. Shmiel, Generating related questions for search queries, 2017. US Patent 9,679,027.

[6] R. Mitra, M. Gupta, S. Dandapat, Transformer models for recommending related questions in web search, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2153–2156.

[7] S. Kuzi, S. Malmasi, Bridging the Gap Between Information Seeking and Product Search Systems: Q&A Recommendation for E-commerce, in: ACM SIGIR Forum, volume 58, 2024, pp. 1–10.

[8] L. K. Senel, B. Fetahu, D. Yoshida, Z. Chen, G. Castellucci, N. Vedula, J. I. Choi, S. Malmasi, Generative explore-exploit: Training-free optimization of generative recommender systems using LLM optimizers, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 5396–5420. URL: https://aclanthology.org/2024.acl-long.295/. doi:10.18653/v1/2024.acl-long.295.

[9] A. Papenmeier, D. Kern, D. Hienert, A. Sliwa, A. Aker, N. Fuhr, Dataset of natural language queries for e-commerce, in: Proceedings of the 2021 Conference on Human Information Interaction and Retrieval, 2021, pp. 307–311.

[10] P. Hämäläinen, M. Tavast, A. Kunnari, Evaluating large language models in generating synthetic hci research data: a case study, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, 2023, pp. 1–19.

[11] T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, E. Kamar, Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection, arXiv preprint arXiv:2203.09509 (2022).

[12] D. Campos, S. Kallumadi, C. Rosset, C. X. Zhai, A. Magnani, Overview of the trec 2023 product product search track, arXiv preprint arXiv:2311.07861 (2023).

[13] R. Mitra, R. Jain, A. S. Veerubhotla, M. Gupta, Zero-shot multi-lingual interrogative question generation for" people also ask" at bing, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 3414–3422.

[14] Z. Chen, J. I. Choi, B. Fetahu, S. Malmasi, Identifying high consideration E-commerce search queries, in: F. Dernoncourt, D. Preoţiuc-Pietro, A. Shimorina (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, Association

for Computational Linguistics, Miami, Florida, US, 2024, pp. 563–572. URL: https://aclanthology.org/2024.emnlp-industry.42/. doi:10.18653/v1/2024.emnlp-industry.42.

[15] S. Gao, Z. Ren, Y. Zhao, D. Zhao, D. Yin, R. Yan, Product-aware answer generation in e-commerce question-answering, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 429–437.

[16] Y. Wang, K. Song, L. Bing, X. Liu, Harvest shopping advice: Neural question generation from multiple information sources in e-commerce, Neurocomputing 433 (2021) 252–262.

[17] O. Rozen, D. Carmel, A. Mejer, V. Mirkis, Y. Ziser, Answering product-questions by utilizing questions from other contextually similar products, arXiv preprint arXiv:2105.08956 (2021).

[18] X. Li, Z. Chen, J. I. Choi, N. Vedula, B. Fetahu, O. Rokhlenko, S. Malmasi, Wizard of shopping: Target-oriented e-commerce dialogue generation with decision tree branching, arXiv preprint arXiv:2502.00969 (2025).

[19] H. Roitman, U. Singer, Y. Eshel, A. Nus, E. Kiperwasser, Learning to diversify for product question generation, arXiv preprint arXiv:2207.02534 (2022).

[20] Z. Zhang, K. Zhu, Diverse and specific clarification question generation with keywords, in: Proceedings of the web conference 2021, 2021, pp. 3501–3511.

[21] N. Vedula, O. Rokhlenko, S. Malmasi, Question suggestion for conversational shopping assistants using product metadata, arXiv preprint arXiv:2405.01738 (2024).

[22] Q. Mei, D. Zhou, K. Church, Query suggestion using hitting time, in: Proceedings of the 17th ACM conference on Information and knowledge management, 2008, pp. 469–478.

[23] Y.-C. Lien, R. Zhang, F. M. Harper, V. Murdock, C.-J. Lee, Leveraging customer reviews for e-commerce query generation, in: European Conference on Information Retrieval, Springer, 2022, pp. 190–198.

[24] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, R. Nogueira, Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations, in: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), 2021, pp. 2356–2362.

[25] S.-C. Lin, J.-H. Yang, J. Lin, In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval, in: Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), 2021, pp. 163–173.

[26] Y. Liu, H. Zhou, Z. Guo, E. Shareghi, I. Vulic, A. Korhonen, N. Collier, Aligning with human judgement: The role of pairwise preference in large language model evaluators, arXiv preprint arXiv:2403.16950 (2024).

[27] J. L. Fleiss, Measuring nominal scale agreement among many raters., Psychological bulletin 76 (1971) 378.

[28] K. Dhole, N. Vedula, S. Kuzi, G. Castellucci, E. Agichtein, S. Malmasi, Generative product recommendations for implicit superlative queries, in: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop), Association for Computational Linguistics, Albuquerque, USA, 2025, pp. 77–91. URL: https://aclanthology.org/2025.naacl-srw.8/. doi:10.18653/v1/2025.naacl-srw.8.