A Product-Aware Query Auto-Completion Framework for E-Commerce Search via Retrieval-Augmented Generation Method

Fangzheng Sun*,†, Tianqi Zheng†, Aakash Kolekar, Rohit Patki, Hossein Khazaei, Xuan Guo, Ziheng Cai, David Liu, Ruirui Li, Yupin Huang, Dante Everaert, Hanqing Lu, Garima Patel and Monica Cheng

Amazon Search, Palo Alto, CA, USA

Abstract

Query Auto-Completion (QAC) is a fundamental component of user search experience on e-commerce websites. It assists in finding user-intended products, by automatically presenting search queries as users typing in the search bar. Traditional QAC systems build upon query popularity to suggest a list of potential completions, but they fall short for unforeseen search prefixes. A generative Large Language Model (LLM) can complete even unforeseen prefixes, but relevance to the product catalog of the generated suggestions is not guaranteed. To our best knowledge, there is no existing study using LLMs to generate product-aware search query completion suggestions.

This paper proposes a generative approach named "Product-RAG", to incorporate product metadata and adapt Retrieval Augmented Generation (RAG) in the development of QAC systems. Product-RAG contains two components: (1) a retrieval model that identifies top-K most relevant products from the product catalog given a user-input prefix, and (2) a generative model that offers suggestions based on both the given prefix and the retrieved product metadata. We evaluate this approach for its ability to match user-input prefixes to user-intended products, using the metrics of ROUGE scores, Mean Reciprocal Rank (MRR) and Hit Ratio (HR) in downstream product search. We observe that the proposed Product-RAG approach outperforms state-of-the-art generative models in auto-completing e-commerce search queries.

Keywords

Query Auto-Complete, Retrieval-Augmented Generation, E-Commerce, Product-aware

1. Introduction

Query auto-completion (QAC) [1, 2, 3, 4, 5] refers to an information retrieval system for search engines, for which, given partial context typed by the user (i.e. prefix), it offers one or multiple query suggestions to the user. In modern e-commerce, where user experience is pivotal, QAC stands as an important feature shaping the way consumers interact with search engines and plays a crucial role in smoothing all the downstream shopping experiences [6, 7, 8, 9]. By leveraging personalized signals, product-related knowledge, and advanced recommendation algorithms, QAC not only accelerates the search experience but also ensures that users receive tailored suggestions based on their unique preferences.

One major challenge of QAC tasks in e-commerce is to understand user shopping intent from an incomplete search query, and provide them relevant auto-complete suggestions. A typical production QAC works as follows: Given a prefix entered by a user, the QAC system obtains a collection of queries satisfying the prefix from query log, and adopts a selection process, often based on forecasted popularity, to select candidate queries to send to the query ranker [10, 11, 12]. This framework lacks the understanding of user shopping intent. To give more importance to users' intents and provide more personalized and relevant QAC suggestions, a number of works explore the context-aware and personalized QAC systems [13, 14, 15, 16, 17].

Another challenge for QAC in e-commerce is to attain product awareness by recognizing and predicting users' intent related to specific products, brands, or categories and providing auto-complete suggestions that align with the user's potential shopping targets. When query log falls short for unseen or rarely seen prefixes, product knowledge is particularly helpful to predict users' shopping intent and generate corresponding suggestions (e.g., in the case of Figure 1). Nevertheless, in spite of efforts attempting to understand user's shopping intent with product catalog or product attributes [18, 19], we could not find any work bridging the gap between partially complete search queries and product knowledge for e-commerce QAC systems. Herein we propose a generative approach to leverage the product knowledge in e-commerce QAC systems based on Retrieval-Augmented Generation (RAG) Framework [20], namely Product-RAG, which is capable of improving the QAC systems by providing accurate auto-completion sug-

Figure 1: A search of "guidebook for edible mus" returns an empty QAC suggestion list while multiple related products are available.

IR-RAG @ SIGIR24 workshop: The 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 18, 2024, Washington D.C., USA

*Corresponding author.

†These authors contributed equally.

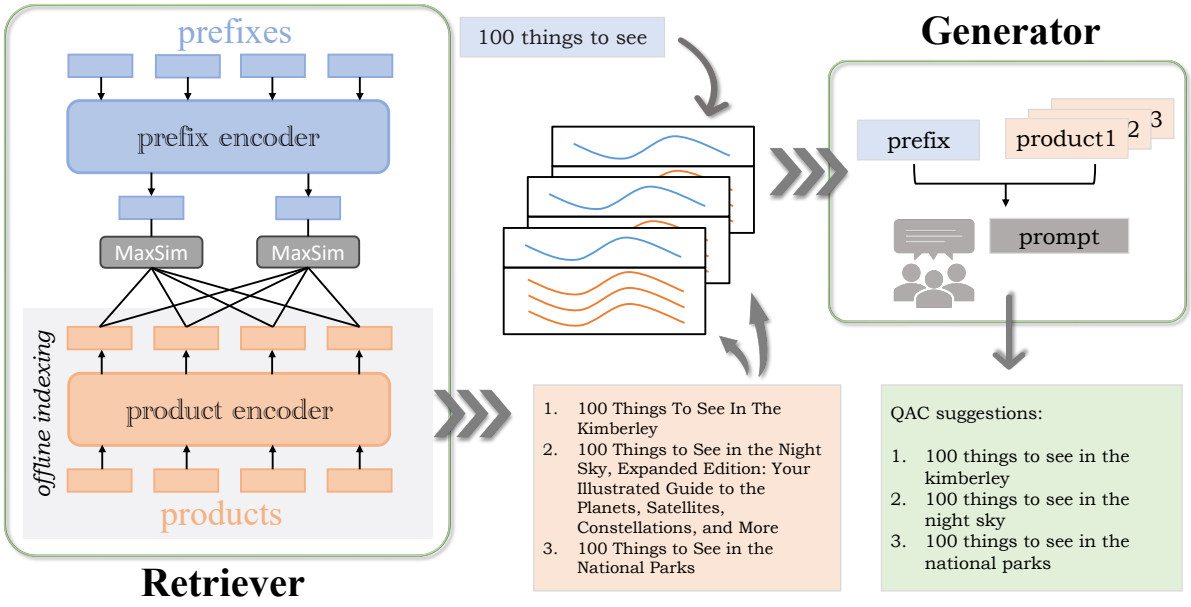

Figure 2: Schematic architecture of Product-RAG model for retrieving top-K products relevant to a prefix and generating K product-aware query auto-completion suggestions from them.

gestions based on product knowledge relevant to search prefixes. The RAG framework exhibits outstanding efficacy in domain-specific sequence generation tasks through exploiting the domain knowledge as additional context when generating the target sequences. Under the RAG scheme, generative large language models (LLMs) are supervised by retrieved relevant information from heterogeneous and predetermined knowledge sources, and consequently, present accurate and controlled text output. This is a compelling benefit for e-commerce QAC systems where relevancy of QAC suggestions is essential for user trust. Additionally, the RAG framework brings QAC systems great versatility against frequently-updated information sources. This attribute allows the utilization of the vast universal product knowledge base into e-commerce QAC systems without undergoing the high costs of re-training the generative model on the giant product catalog. The superiority of the proposed Product-RAG model is empirically demonstrated by human-annotated e-commerce QAC tasks. We illustrate that the proposed method could outperform state-of-the-art generative Large Language Models (LLMs) by generating QAC suggestions that 1) are more similar with the ground truths and in terms of ROUGE scores, and 2) lead to more relevant product search results in terms of Mean Reciprocal Rank (MRR) and Hit Ratio (HR).

2. Related Work

Most QAC systems work on a sourcing-and-ranking basis - they first source candidates from a large pool to limit the scope, and then rank on the sourced candidates. This paper focuses on the sourcing part of the QAC system, ensuring that the sourced candidates are product-aware, before sending them to the query ranker.

Established approaches for sourcing candidate query completions often rely on most popular candidate (MPC) approach, which could not incorporate important semantic information of prefix and session context. Newer approaches incorporate semantic information using neural network representations learned with query prefix, candidate query completions, along with session context [21, 22]. Context-aware QAC considers users' past queries instead of solely relying on popularity [13, 14, 15, 16, 17]. Meanwhile, recent advances in natural language processing [23] have inspired enormous work on semantically understanding users' search context and intent. They employ a high-dimensional space to measure the semantic similarities between search queries in contextual representations. These methods either offer users more contextually relevant suggestions from query log [24, 25, 26, 27] or generate new keywords [28, 29, 30]. Candidate sourcing approaches usually fall into one of the three categories: retriever-only approaches, pure generative approaches, and retrieval-augmented generation.

Retriever-only approaches. This class of approaches are based on *retrieving* the candidates from a pool of candidates that is usually built from scraping historical logs of the QAC system. The candidates are often selected based on popularity-based approaches such as MPC and neural matching [31, 32, 33]. In MPC-based approaches, candidates with higher forecasted popularity are selected; in lexical matching approaches, candidates with higher similarities are selected; neural matching approaches for candidate retrieval allow selecting queries in semantic representations [34]. Session context [21, 17] or personal profile signals [35, 36] could be considered as auxiliary inputs. However, since they are limited to the existing candidates, offering suggestions is not guaranteed. In addition, the achieved product-awareness is biased toward popular candidates in the past.

Generative approaches. The generative approaches use language models to generate the candidate based on inputs like prefix, session context, and personalization signals[37, 38, 39]. Generative approaches can provide candidates for prefixes fist appear. Two major challenges facing pure generative approaches in e-commerce QAC system

development are: (1) They may suffer from hallucinations, generating plausible queries but without a reference to product information. (2) In a dynamic environment of e-commerce where products changes continuously, they lack the mechanism to automatically incorporate new information post-training. This necessitates periodic fine-tuning to maintain the model's relevance and accuracy, which can be prohibitively costly and impractical.

Retrieval Augmented Generation (RAG). RAG approaches [20, 40, 41], extend the capabilities of language models by integrating external knowledge sources as auxiliary inputs to enhance the performance of the overall system. To our best knowledge, our work is the first study that adapts RAG framework for QAC systems. Prior to this, RAG has been applied to various tasks and application domains such as question answering [42], text summarization [20]. Since RAG uses product information as auxiliary inputs for the language model during generation, and that the product retrieval updates in response to underlying database updates, the language model does not need to be fine-tuned to capture new products.

3. Product-RAG

Existing studies show that the RAG framework is effective and efficient in extending the already powerful capabilities of LLMs to specific domains without the need to retrain the model with a heterogeneous database. The proposed Product-RAG model is based upon the architecture of the RAG-Sequence Model [20] to generate one suggestion for each relevant product. The schematic architecture of the Product-RAG model for generating product-aware query auto-completion suggestions from relevant products is depicted in Figure 2. Our framework leverages two components:

1. A retrieval model η that retrieves the best-matching product titles or catalog from product pool.
2. A generative LLM θ that outputs auto-complete suggestions for a given prefix and retrieved products from the retriever.

Task formulation. We denote a search prefix as x and the target QAC suggestions as y. The retriever $p_\eta(\mathbf{Z}|x)$ consumes a product knowledge base $\mathbb{P}$ and returns top-K relevant products $\mathbf{Z} = \{z_1, z_2, ..., z_K\}, z_i \in \mathbb{P}$ given the prefix x. For each $z \in \mathbf{Z}$, the generative LLM $p_\theta(y|x, z)$ generates a QAC suggestion for x with context from the retrieved product, rendering the top-K suggestions $\mathbf{Y} = \{y_1, y_2, ..., y_K\}$. The Product-RAG can be parameterized as

$$p_{Product-RAG}(\mathbf{Y}|x) \doteq \sum_{z_i \in top-K(p(\cdot|x))} p_\eta(z_i|x) p_\theta(y_i|x, z_i) \quad (1)$$

Multi-vector retrieval model. State-of-the-art methods typically fine-tune deep pre-trained language models, such as BERT [43], to generate dense vector representations for both input queries and documents. The top-K documents with the highest similarity scores are then retrieved. Inspired by recent advances in multi-vector representations [34, 44], we adopt a retrieval model in the proposed approach, as depicted in Figure 2, where we fine-tune the prefix and product encoders with e-commerce data.

Precisely, given the representation of a prefix x and a product z, the relevance score of z to x, denoted as $S_{x,z}$ is defined as the sum of maximum cosine similarity between each vector $\mathbf{E}_i^x$ in prefix embedding and the vectors in a a product bag $\mathbf{E}^T$:

$$S_{x,z} := \sum_{i=1}^{x} \max_{j \in T} \mathbf{E}_i^x \cdot \mathbf{E}_j^T, z \in T \quad (2)$$

Offline product knowledge indexing. We pre-compute all product embeddings and offline index these vector representations to support the efficient lookup of relevant products. The index includes 1) centriods representing centers partitioning product embeddings into bags, 2) residuals storing embedding for a product and comparing to its nearest centroid, and 3) index inversion representing an inverted map from a centroid to products to support the fast nearest neighbor search. They are encoded offline and loaded into the memory of QAC service. Given a prefix, the prefix encoder vectorizes it and the retrieval model looks for the top-K most relevant products through operating MaxSim between the prefix embedding and already-loaded product indexing in memory.

Offline indexing of pre-computed product embeddings also brings convenience to refreshing the product knowledge pool frequently with low cost and requires no effort from re-training models to adapt to newly added products.

QAC Suggestion Generation. For the generative component of the Product-RAG, we use a generative LLM where the input is a prompt containing both prefix x and top-K retrieved products $z \in \mathbf{Z}$ and the outputs are K product-aware QAC suggestions $\mathbf{Y}$ for the prefix. As we train the retrieval model and the generative model separately, we can use any state-of-the-art generative LLMs such as Mistral-7B [45], PaLM [46], GPT-4 [47], as long as they can perform text summarization and QAC or equivalent tasks. In our proposed Product-RAG we empirically choose Mistral-7B based on offline evaluations on the performance and latency of different generative LLMs in QAC tasks. Moreover, we are able to fine-tune Mistral-7B with e-commerce search and product data.

4. Experiments

We now evaluate Product-RAG on e-commerce QAC tasks, testing its ability to generate QAC suggestions for a given prefix in the e-commerce domain. We define a baseline LLM by fine-tuning the Mistral-7B model on the e-commerce QAC database without the help of the RAG framework. In the Product-RAG framework, we employ the multi-vector retrieval model (denoted as MultiVec), fine-tuned as outlined in the previous section, as the primary retrieval component. To demonstrate the effectiveness of our proposed retrieval method, we establish a baseline retrieval model, BM25 [48], within the RAG framework for comparison. The generative component for both the Product-RAG-MultiVec and Product-RAG-BM25 frameworks is a fine-tuned Mistral-7B.

Experimental dataset. We perform an experiment on 1,500 search queries corresponding to book products with the help of human expert annotation: given a search prefix, a human expert manually annotates an auto-completion

Table 1
Evaluation scores of generated QAC suggestions. Each model generates 3 suggestions and we obtain the maximum evaluation scores out of these suggestions as the evaluation score of the data point.

Model	ROUGE-1	ROUGE-2	ROUGE-L	MRR@10	HR@10
Mistral-7B	77.2	66.7	76.6	0.65	0.76
Product-RAG-BM25	75.3	64.9	74.6	0.62	0.74
Product-RAG-MultiVec	**82.2**	**74.1**	**81.5**	**0.75**	**0.87**

keyword as the ground truth QAC suggestion and, through the product search page, find an available book product, which we use as the ground truth targeting product. Thus, each evaluation data point is composed by a triple <prefix, QAC suggestion, product>. For each data point, we use proposed models to generate top-3 suggestions based on the prefix. For the Product-RAG models, we employ 7 million book products as product knowledge.

Evaluation metrics. We evaluate the generated suggestions in both query and product levels. We compute

1. the similarity between generated suggestions with annotated ground truth QAC suggestions in terms of ROUGE-1/2/L F1 scores. We use the maximum score out of the 3 suggestions for each data point.
2. for each target product, the Mean Reciprocal Rank in the top 10 results (MRR@10) and the Hit Ratio (target product is included) in the top 10 results (HR@10) on the product search page triggered by the generated suggestions. We use the maximum MRR@10 and HR@10 out of the 3 suggestions generated as well.

Evaluation results. We report the ROUGE-1, ROUGE-2, ROUGE-L F-1 scores, MRR@10, and HR@10 for the 3 experimented models in Table. 1. We observe that the proposed Product-RAG-MultiVec outperforms both baseline generative Mistral-7B and the counterpart based on the BM25 retrieval model, in terms of all ROUGE scores and the 2 product-level metrics. These findings demonstrate the supremacy of the Product-RAG-MultiVec model in generating high-quality QAC suggestions for e-commerce systems.

Discussions. In the experiment above, we notice a negative impact in these metrics led by the BM25 retriever compared with baseline Mistral-7B. Inspecting the top-3 retrieved products from BM25 and those from MultiVec model, we notice a non-trivial gap between the performance of two retrievers: in our experiment, MultiVec model is able to successfully retrieve the targeting product or its equivalents in 74.7% cases (e.g., *Harry Potter and the Chamber of Secrets: Gryffindor Edition Red* or its equivalent *Harry Potter and the Chamber of Secrets*), whereas BM25 retriever only succeeds in 37.1% cases. The accuracy of two retrievers explains the performance gap between the two Product-RAG models. This leads to a conclusion that when the retriever provides highly-relevant products, the proposed Product-RAG framework is capable of improving upon the state-of-the-art generative approaches in e-commerce QAC tasks. And we believe that improving the precision of the retriever model is one future direction of refining the proposed Product-RAG framework.

5. Conclusions

In this work, we introduce Product-RAG, an RAG framework that advances e-commerce QAC systems by identifying relevant products to search prefix and informing product-aware suggestions. This framework generates suggestions close to user search intention, and it highlights product relevance at an early stage of the shopping journey, before downstream product searches. Through empirical experiments on auto-completing search queries, we compare the proposed framework with baseline LLM and we test various retrieval models. In particular, we find that the Product-RAG-MultiVec remarkably outperforms its counterparts in terms of query similarity and product relevance. This work sheds light on bridging the semantic gap between partial search queries and product knowledge in the scenario of e-commerce QAC systems.

References

[1] M. Jakobsson, Autocompletion in full text transaction entry: a method for humanized input, ACM SIGCHI Bulletin 17 (1986) 327–332.

[2] H. Bast, I. Weber, Type less, find more: fast autocompletion search with a succinct index, in: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006, pp. 364–371.

[3] H. Bast, D. Majumdar, I. Weber, Efficient interactive query expansion with complete search, in: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, 2007, pp. 857–860.

[4] F. Cai, M. De Rijke, et al., A survey of query auto completion in information retrieval, Foundations and Trends® in Information Retrieval 10 (2016) 273–363.

[5] C. Xiao, J. Qin, W. Wang, Y. Ishikawa, K. Tsuda, K. Sadakane, Efficient error-tolerant query autocompletion, Proceedings of the VLDB Endowment 6 (2013) 373–384.

[6] M. A. Hasan, N. Parikh, G. Singh, N. Sundaresan, Query suggestion for e-commerce sites, in: Proceedings of the fourth ACM international conference on Web Search and Data Mining, 2011, pp. 765–774.

[7] S. K. Karmaker Santu, P. Sondhi, C. Zhai, On application of learning to rank for e-commerce search, in: Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval, 2017, pp. 475–484.

[8] L. Wu, D. Hu, L. Hong, H. Liu, Turning clicks into purchases: Revenue optimization for product search in e-commerce, in: The 41st International ACM SIGIR

Conference on Research & Development in Information Retrieval, 2018, pp. 365–374.

[9] A. Block, R. Kidambi, D. N. Hill, T. Joachims, I. S. Dhillon, Counterfactual learning to rank for utility-maximizing query autocompletion, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 791–802.

[10] G. Di Santo, R. McCreadie, C. Macdonald, I. Ounis, Comparing approaches for query autocompletion, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 775–778.

[11] M. Shokouhi, K. Radinsky, Time-sensitive query autocompletion, in: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, 2012, pp. 601–610.

[12] A. Strizhevskaya, A. Baytin, I. Galinskaya, P. Serdyukov, Actualization of query suggestions using query logs, in: Proceedings of the 21st International Conference on World Wide Web, 2012, pp. 611–612.

[13] Z. Bar-Yossef, N. Kraus, Context-sensitive query autocompletion, in: Proceedings of the 20th international conference on World wide web, 2011, pp. 107–116.

[14] F. Cai, S. Liang, M. De Rijke, Time-sensitive personalized query auto-completion, in: Proceedings of the 23rd ACM international conference on conference on information and knowledge management, 2014, pp. 1599–1608.

[15] M. Shokouhi, Learning to personalize query autocompletion, in: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval, 2013, pp. 103–112.

[16] S. Whiting, J. M. Jose, Recent and robust query autocompletion, in: Proceedings of the 23rd international conference on World wide web, 2014, pp. 971–982.

[17] N. Yadav, R. Sen, D. N. Hill, A. Mazumdar, I. S. Dhillon, Session-aware query auto-completion using extreme multi-label ranking, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 3835–3844.

[18] J. Zhao, H. Chen, D. Yin, A dynamic product-aware learning model for e-commerce query intent understanding, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 1843–1852.

[19] C. Luo, R. Goutam, H. Zhang, C. Zhang, Y. Song, B. Yin, Implicit query parsing at amazon product search, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 3380–3384.

[20] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[21] B. Mitra, Exploring session context using distributed representations of queries and reformulations, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 3–12. URL: https://doi.org/10.1145/2766462.2767702. doi:10.1145/2766462.2767702.

[22] S. Wang, W. Guo, H. Gao, B. Long, Efficient neural query auto completion, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 2797–2804. URL: https://doi.org/10.1145/3340531.3412701. doi:10.1145/3340531.3412701.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[24] B. Mitra, N. Craswell, Query auto-completion for rare prefixes, in: Proceedings of the 24th ACM international on conference on information and knowledge management, 2015, pp. 1755–1758.

[25] F. Cai, M. de Rijke, Learning from homologous queries and semantically related terms for query auto completion, Information Processing & Management 52 (2016) 628–643.

[26] T. Shao, H. Chen, W. Chen, Query auto-completion based on word2vec semantic similarity, in: Journal of Physics: Conference Series, volume 1004, IOP Publishing, 2018, p. 012018.

[27] K. Arkoudas, M. Yahya, Semantically driven autocompletion, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2693–2701.

[28] S. Wang, W. Guo, H. Gao, B. Long, Efficient neural query auto completion, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2797–2804.

[29] Y. M. Kang, W. Liu, Y. Zhou, Queryblazer: efficient query autocompletion framework, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 1020–1028.

[30] D. Maxwell, P. Bailey, D. Hawking, Large-scale generative query autocompletion, in: Proceedings of the 22nd Australasian Document Computing Symposium, 2017, pp. 1–8.

[31] S. Whiting, J. M. Jose, Recent and robust query autocompletion, in: Proceedings of the 23rd International Conference on World Wide Web, WWW '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 971–982. URL: https://doi.org/10.1145/2566486.2568009. doi:10.1145/2566486.2568009.

[32] F. Cai, S. Liang, M. de Rijke, Time-sensitive personalized query auto-completion, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 1599–1608. URL: https://doi.org/10.1145/2661829.2661921. doi:10.1145/2661829.2661921.

[33] M. Shokouhi, K. Radinsky, Time-sensitive query autocompletion, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, Association for Computing Machinery, New York, NY, USA, 2012, p. 601–610. URL: https://doi.org/10.1145/2348283.2348364. doi:10.1145/2348283.2348364.

[34] O. Khattab, M. 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, 2020, pp. 39–48.

[35] G. Aslanyan, A. Mandal, P. Senthil Kumar, A. Jaiswal,

M. Rangasamy Kannadasan, Personalized ranking in ecommerce search, in: Companion Proceedings of the Web Conference 2020, WWW '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 96–97. URL: https://doi.org/10.1145/3366424.3382715. doi:10.1145/3366424.3382715.

[36] A. Jaech, M. Ostendorf, Personalized language model for query auto-completion, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 700–705. URL: https://aclanthology.org/P18-2111. doi:10.18653/v1/P18-2111.

[37] D. Yin, J. Tan, Z. Zhang, H. Deng, S. Huang, J. Chen, Learning to generate personalized query autocompletions via a multi-view multi-task attentive approach, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 2998–3007. URL: https://doi.org/10.1145/3394486.3403350. doi:10.1145/3394486.3403350.

[38] D. Maxwell, P. Bailey, D. Hawking, Large-scale generative query autocompletion, in: Proceedings of the 22nd Australasian Document Computing Symposium, ADCS '17, Association for Computing Machinery, New York, NY, USA, 2017. URL: https://doi.org/10.1145/3166072.3166083. doi:10.1145/3166072.3166083.

[39] A. Sordoni, Y. Bengio, H. Vahabi, C. Lioma, J. Grue Simonsen, J.-Y. Nie, A hierarchical recurrent encoder-decoder for generative context-aware query suggestion, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 553–562. URL: https://doi.org/10.1145/2806416.2806493. doi:10.1145/2806416.2806493.

[40] P. Xu, W. Ping, X. Wu, L. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi, B. Catanzaro, Retrieval meets long context large language models, arXiv preprint arXiv:2310.03025 (2023).

[41] N. Kandpal, H. Deng, A. Roberts, E. Wallace, C. Raffel, Large language models struggle to learn long-tail knowledge, in: Proceedings of the 40th International Conference on Machine Learning, ICML'23, JMLR.org, 2023.

[42] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, W. Chen, Generation-augmented retrieval for open-domain question answering, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 4089–4100. URL: https://aclanthology.org/2021.acl-long.316. doi:10.18653/v1/2021.acl-long.316.

[43] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[44] K. Santhanam, O. Khattab, J. Saad-Falcon, C. Potts, M. Zaharia, Colbertv2: Effective and efficient retrieval via lightweight late interaction, arXiv preprint arXiv:2112.01488 (2021).

[45] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7b, arXiv preprint arXiv:2310.06825 (2023).

[46] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., Palm: Scaling language modeling with pathways, Journal of Machine Learning Research 24 (2023) 1–113.

[47] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[48] S. Robertson, H. Zaragoza, et al., The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389.