# Reducing the Long Tail Effect in E-Commerce through Self-Attention

Fedor Krasnov, Fedor Kurushin
Research Center of WB SK LLC,
Moscow, Russia
krasnov.fedor2@wb.ru, kurushin.fedor@wb.ru

***Abstract*—The long tail of search queries is a well-known issue that complicate the creation of efficient reverse indexes based on string-based representations of queries. Various techniques have been employed to reduce the diversity of search terms, such as proximity searching, fuzzy hashing, and collaborative filtering. Nevertheless, these approaches often struggle to handle domain-specific entities such as brand names and product characteristics, which are essential for effective product search in online marketplaces.**

**This study presents an approach that utilizes positional weighting to assess the significance of search terms based on their influence within the query context. The proposed technique takes into account domain-specific elements to more precisely determine the relevance of each term. By implementing this new C2T (context-dependent token) model, a 48% reduction in query diversity was achieved, as measured by the perplexity metric.**

## I. Introduction

From an engineering perspective, handling the long tail of search queries poses a significant challenge. The performance of an e-commerce product search system plays a crucial role, influenced by two primary factors: (1) response time to customer requests and (2) providing high-quality search results that accurately match the user's query. Research into user behavior [1] reveals that slow response times can lead to disruptive interruptions in the shopping experience and even deter customers from visiting the website. Deep learning systems for product extraction are unable to handle the full volume of requests within the required time frame. Therefore, popular queries must be indexed - that is, temporarily memorized along with the corresponding products, in order to speed up the processing. However, according to analysts, popular queries account for less than 10% of all search requests, making indexing ineffective. Given the presence of typos and variations in query formulation, translating most queries into an index is challenging. As a result, the task of converting queries into a single, standardized form through paraphrasing and word deletion is of utmost importance.

Response time also plays a crucial role in determining the bandwidth requirements of a search engine. Modern product search systems typically comprise a range of machine learning models [2]–[6], such as product retrieval [7], ranking [8], and query annotation [9]. Serving all search queries using deep learning models can be an extremely resource-intensive task for product search systems, due to limitations in latency and equipment costs [10]. Therefore, instead of using deep learning models for all queries, it is more practical to serve high-frequency queries (HFQ) directly from the query index(QI) as shown in the Fig. 1.

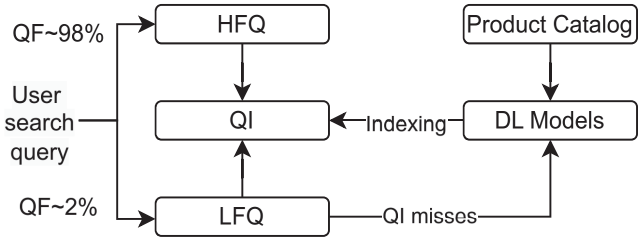


Fig. 1. User query flow

The QI is a reverse index in structure, meaning that it provides candidate products for a given search query based on the index. These products are then ranked and personalized for the user. Candidate products should meet the criteria for recall and precision, as it would be undesirable to include irrelevant products in the query index.

The number of queries indexed is also optimized, as too few queries can overload online query processing systems and too many can complicate indexing maintenance. For instance, a query such as "summer dress" might be a HFQ for a women's clothing store during the spring season, and thus it would be reasonable to include it in the index. On the other hand, a request like "summer pink zara women's dress" would be less frequent, especially during November in Novosibirsk, so it may not be worth indexing. Therefore, the QI is a dynamic and highly loaded data structure that can be sensitive to seasonal and calendar-related events.

The reformulation of search queries with the aim of enhancing their effectiveness is a widely used technique in product search. Researchers in [11], [12] identify three types of query reformulation:

1) Add – adding one or more words to the reformulated query;
2) Delete – removing one or more words from the reformulated query;
3) Replace – substituting one or more words for others in the reformulated query.

Research on the reformulation of queries has associated "Addition" with specialization, that is, refinement for a more

specific customer intention, and “Delete” with generalization for a broader information requirement [13].

Correcting typos in a query can also be considered a form of reformulation, similar to the “Replace” type. This paper describes an algorithm for rephrasing search queries using the C2T model and presents the process and outcomes of a digital experiment based on this approach.

The main innovation of this research is to develop a simple and efficient method for rephrasing search queries using a machine learning algorithm.

## II. Methods

Modern approaches to query reformulation utilize neural network techniques. By extracting textual features, vector representations of search queries are created, and the search for “nearest neighbors” in vector space is conducted utilizing RAG language models, BERT models, and ANN models [14]–[16].

Product search enables the utilization of user behavioral data during the purchasing process. Consequently, a collaborative approach is employed: the query-product correlation determines the queries that led to purchases, which are then utilized as options for rephrasing the current query [17].

The proposed approach is based on the following assumption: Short queries in the long tail are easier to process than search queries that contain more than five tokens. Users often include redundant or clarifying words in their queries in the hope that a search engine will focus on one of these words and provide the desired results. For example, in a query such as “Apple iPhone phone case,” the words “Apple” and “phone” are unnecessary and can be ignored. These words are referred to as target tokens, which can occur in any position within a search query. A naive approach assumes that the target token occurs at the end of the query. As part of an experiment, the efficacy of this approach was compared to a method that assigns weights to each token based on its relevance to the query context.

A method for identifying relevant tokens for refining a search query is proposed. The probability of finding a target-token $P_q(t_j)$ , $j \in [1, N]$ in a query $q$ consisting of $N$ tokens is expressed in the form of an equation (1), where $ctx_j$ represent all tokens in $q$ except for the target-token $t_j$. Additionally, $ctx_j$ is defined as the context surrounding the target-token $t_j$.

$$P_q(t_j) = s_\theta(\vec{t_j}, \vec{ctx_j}) \tag{1}$$

Where, $\vec{ctx_j} = \frac{1}{N-1} \sum_{i=1, i \neq j}^{N} (\vec{t_i})$ is the average vector of context $\vec{ctx_j}$, and $s_\theta(\circ, \circ)$ is the matching function between two vectors in space $\theta$, which takes values between 0 and 1.

In other words, from equation (1), if the target-token $t_j$ can be inferred from the context $ctx_j$, the chances of successfully deleting the item are high. If it is difficult to $t_j$, it would be impractical to remove this token from the search query in order to reformulate it.

To train the model for vector representations of tokens $\theta$, the author selected the architecture of the Dual Encoder [18], with the fitting approach of metric learning. In accordance with the methodology presented in the study [10], only linear (fully connected) layers of the artificial neural network and transformations using nonlinear activations based on the hyperbolic tangent function were used in the individual “towers” of the Dual Encoder.

The strategies for selecting positive and negative examples for training the model are illustrated in the Figures 2 and 3.

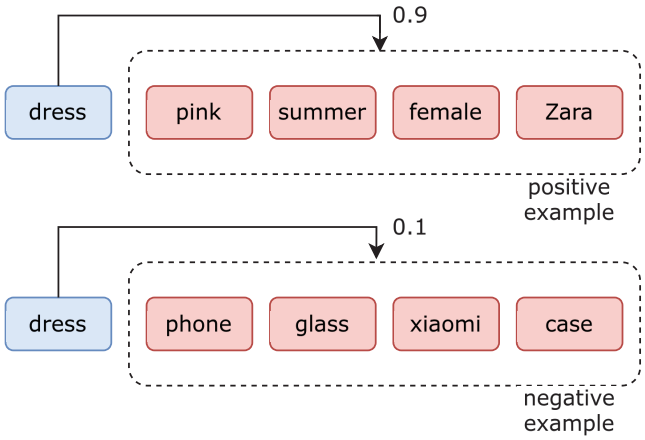


Fig. 2. A token2ctx strategy for sampling positive and negative examples

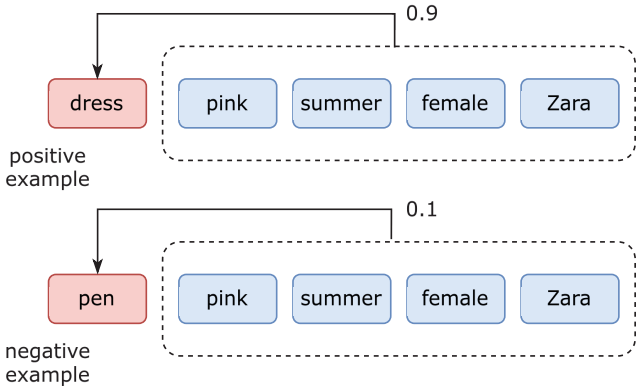


Fig. 3. A ctx2token strategy for sampling positive and negative examples

The ctx2token approach is based on the random selection of target-token that have not been previously encountered in any context. Conversely, the token2ctx approach randomly selects context tokens such that none of them matches the target-token.

The main difference between the proposed strategies (ctx2token and token2ctx) and CBOW and skip-gram strategies in study [19] is that they do not rely on a specific context window for context consideration. Transformer architecture models [20], which use the token masking technique, differ from ctx2token and token2ctx strategies in that they do not utilize negative examples for training purposes.

The loss functions $\mathbf{L}_{ctx2token}(c\vec{t}x, \vec{t^+}, \vec{t^-})$ and $\mathbf{L}_{token2ctx}(\vec{t}, c\vec{t}x^+, c\vec{t}x^-)$ were employed to train the vector representations of tokens for the ctx2token and token2ctx strategies, presented in the formulas (3, 5):

$$\mathbf{L}_{ctx2token}(c\vec{t}x, \vec{t^+}, \vec{t^-}) = \tag{2}$$

$$\left[\gamma - s_\theta\left(c\vec{t}x \cdot \vec{B}, \vec{t^+}\right) + s_\theta\left(c\vec{t}x \cdot \vec{B}, \vec{t^-}\right)\right]_+ \tag{3}$$

$$\mathbf{L}_{token2ctx}(\vec{t}, c\vec{t}x^+, c\vec{t}x^-) = \tag{4}$$

$$\left[\gamma - s_\theta\left(c\vec{t}x^+ \cdot \vec{B}, \vec{t}\right) + s_\theta\left(c\vec{t}x^- \cdot \vec{B}, \vec{t}\right)\right]_+ \tag{5}$$

In equations 3 and 3, $\gamma$ represents the lower threshold for the indicators of $s_\theta$, and $[\circ]_+$ sets any values less than zero to zero, as a ReLU activation function. To protect brand tokens, a $\vec{B}$ tensor containing the weights of the tokens corresponding to the brands is included in the training process.

Based on the vector representation of tokens in the $\theta$ space, a C2T (C2T stands for "context-to-token") model is developed to determine the semantic significance of a given target token. To evaluate the effectiveness of the approach proposed in the paper for a variety of search queries, the author employ the perplexity metric. Perplexity for a collection of search queries can be calculated as the reciprocal of the probability of the entire set of tokens, normalized by the total number of tokens [21] (6):

$$\mathbf{PP} = 2^{-\frac{1}{M}\sum_{i=1}^{M}\log_2 p(t_i)} \tag{6}$$

In Equation (6), $M$ represents the total number of tokens in the query set and $p(t_i)$ represents the probability of occurrence of a single token. Reducing the perplexity $\mathbf{PP}$ of a set of queries implies reducing the diversity and, consequently, improving the query indexing.

## III. Experiment

In order to test the developed methodology, the authors selected data from the search query log that led to the purchase of products. The data included queries containing more than 5 and less than 12 words, separated by spaces. In total, approximately 100 million queries were selected for analysis. We selected the *Sentencepiece* method [22], with a dictionary size of 32 thousand tokens, as the tokenizer for this experiment.

To enhance the sampling of negative instances, a matrix of token occurrences and a frequency distribution of tokens were calculated. The results are presented in Figure 4.

The findings demonstrate that the occurrence of tokens together is infrequent in large datasets. To generate negative examples, the authors utilized the token frequency distribution depicted in Figure 5.

With a wide range of data, sampling techniques play a significant role in the process. Based on the formulas (3) and (5), brand data is incorporated into the loss function.

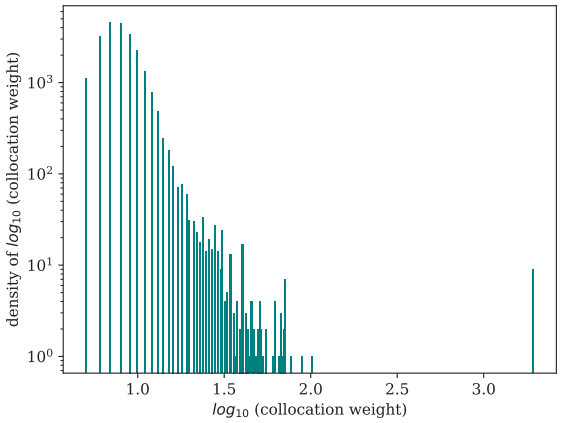


Fig. 4. Distribution of the weights of the collocations of tokens in search queries

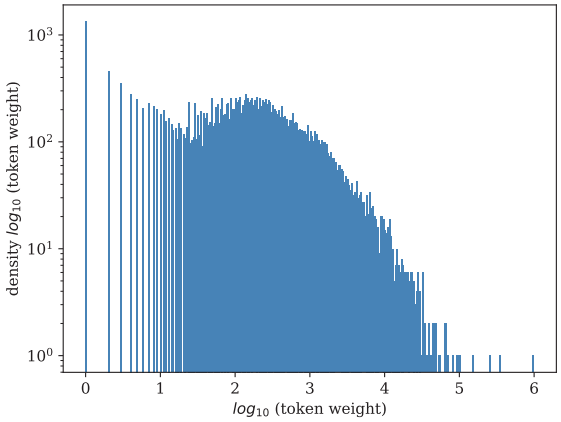


Fig. 5. Distribution of token frequencies

The weights of each brand are determined based on publicly available information regarding the popularity of the brands. Brand data is essential for the relevance of the queries, as per research findings.

Figure 6 illustrates the architecture of the Dual Encoder model that was developed for training vectors within the $\theta$ vector space.

To apply the C2T model, a validation dataset of 1 million search queries was assembled from user search logs. When training the C2T model, the following hyperparameters were selected:

- The number of training cycles: 60 epochs;
- Graphics accelerators: 4 NVIDIA A800;
- Training time: 42 hours.

An example of using the C2T model to index search queries for token2ctx and ctx2token strategies can be seen in Tables I and II.

In Tables I and II, the target-tokens for deletion that have the least importance for the query text have been highlighted

TABLE I. SEARCH QUERIES WITH C2T WEIGHTS AND TARGET-TOKENS (CTX2TOKEN)

| № | Search query tokens |
|---|---|
| 1 | (0.47, summer), (0.16, pink), (0.13, dress), **(0.09, female)**, (0.15, Zara) |
| 2 | **(0.04, comfy)**, (0.49, sneakers), (0.15, female), (0.13, white), (0.19, nike) |
| 3 | (0.17, notebook), **(0.15, with)**,(0.16, narrow),(0.17, lines),(0.15, for),(0.20, school) |

TABLE II. SEARCH QUERIES WITH C2T WEIGHTS AND TARGET-TOKENS (TOKEN2CTX)

| № | Search query tokens |
|---|---|
| 1 | (0.60, summer), (0.09, pink), (0.11, dress), **(0.08, female)**, (0.12, Zara) |
| 2 | (0.61, comfy), (0.11, sneakers), (0.10, female), **(0.08, white)**, (0.10, nike) |
| 3 | (0.11, notebook), **(0.09, with)**, (0.11, narrow), (0.12, lines), (0.41, for), (0.16, school) |

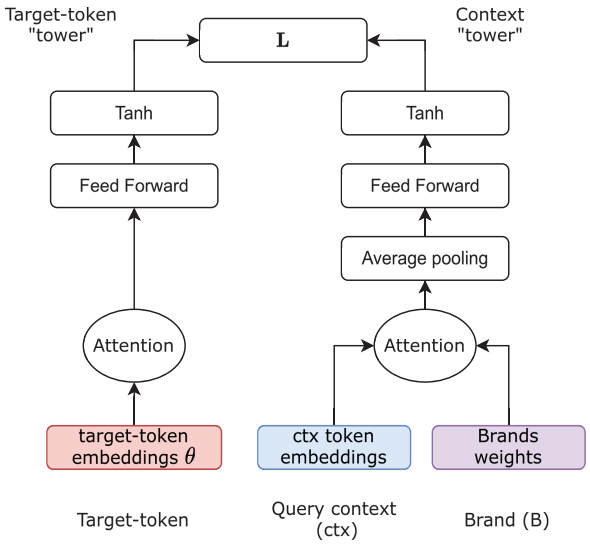


Fig. 6. The C2T model

in bold. These are the tokens that have the lowest values of the semantic significance index. The tables show the values of the perplexity metric for two variants of the C2T model (ctx2token and token2ctx) on a validation dataset. The original value of the perplexity was $\mathbf{PP}_{orig} = 115.4$.

Table III shows the values of perplexity after deleting the first and second tokens with the lowest semantic significance.

TABLE III. PERPLEXITY VALUES

| Model / strategy | first target-token | second target-token |
|---|---|---|
| C2T (ctx2token) | **68.2** | **12.3** |
| C2T (token2ctx) | 84.1 | 15.0 |
| "a naive strategy" | 95.3 | 29.8 |

Based on the Table III, the perplexity decreases to a greater extent when removing tokens using the ctx2token strategy. In contrast, when a naive strategy is employed to delete the final token, the diversity of search queries reduces to a lesser extent compared to using the C2T model.

## IV. CONCLUSION

The study presents a method for accelerating the process of searching for products in e-commerce by enhancing the quality of the Query Index for string representations of queries. This method involves rewriting the search query in order to create a more condensed Query Index. The search query is shortened by weighting the tokens in the query based on their significance. First, those tokens that have the least influence on the search result for the given query are eliminated. In the experiment, this resulted in a reduction of 48% in the size of the Query Index. This study will continue as part of efforts to introduce more information about product characteristics and to measure the impact of the C2T model on product retrieval.

## REFERENCES

[1] *Nah, Fiona Fui-Hoon.* "A study on tolerable waiting time: how long are web users willing to wait?." *Behaviour & Information Technology* 23.3 (2004): 153-163.

[2] *Ahuja A., Rao N., Katariya S., Subbian K., Reddy C. K.* "Language-agnostic representation learning for product search on e-commerce platforms" *Proceedings of the 13th International Conference on Web Search and Data Mining.* (2020): 7-15.

[3] *Liang T., Zeng G., Zhong Q., Chi J., Feng J. Ao X., Tang J.* "Credit risk and limits forecasting in e-commerce consumer lending service via multi-view-aware mixture-of-experts nets" *Proceedings of the 14th ACM international conference on web search and data mining* (2021): 229-237.

[4] *Shi J., Yao H., Wu X., Li T., Lin Z., Wang T., Zhao B.* "Relation-aware meta-learning for e-commerce market segment demand prediction with limited records" *Proceedings of the 14th ACM International Conference on Web Search and Data Mining* (2021): 220-228.

[5] *Xu D., Ruan C., Korpeoglu E., Kumar S., Achan K.* "Product knowledge graph embedding for e-commerce" *Proceedings of the 13th international conference on web search and data mining* (2020): 672-680.

[6] *Xu D., Ruan C., Korpeoglu E., Kumar S., Achan K.* "Theoretical understandings of product embedding for e-commerce machine learning" *Proceedings of the 14th ACM international conference on web search and data mining* (2021): 256-264.

[7] *Nigam P., Song Y., Mohan V., Lakshman V., Ding W., Shingavi A., Teo C. H., Gu H., Yin B.* "Semantic product search" *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2019): 2876-2885.

[8] *Xiao R., Ji J., Cui B., Tang H., Ou W., Xiao Y., Tan J., Ju X.* "Weakly supervised co-training of query rewriting and semantic matching for e-commerce" *Proceedings of the twelfth ACM international conference on web search and data mining* (2019): 402-410.

[9] *Bi K., Teo C. H., Dattatreya Y., Mohan V., Croft W. B.* "A study of context dependencies in multi-page product search" *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019): 2333-2336.

[10] *Lin H., Xiong P., Zhang D. D., Yang F., Kato R., Kumar M., Headden W., Yin B.* "Light feed-forward networks for shard selection in large-scale product search. (2020)

[11] *Huang J., Efthimiadis E. N.* "Analyzing and evaluating query reformulation strategies in web search logs *Proceedings of the 18th ACM conference on Information and knowledge management* (2009): 77-86.

[12] *Jansen B. J., Booth D. L., Spink A.* "Patterns of query reformulation during web searching" *Journal of the american society for information science and technology* (2009): vol. 60. num. 7. 1358-1371.

[13] *Jiang J. Y., Ke Y. Y., Chien P. Y., Cheng P. J.* "Learning user reformulation behavior for query auto-completion" *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (2014): 445-454.

[14] *Ma X., Gong Y., He P., Zhao H., Duan N.* "Query Rewriting for Retrieval-Augmented Large Language Models" *arXiv preprint arXiv:2305.14283* (2023)

[15] *Chen Z., Fan X., Ling Y.* "Pre-training for query rewriting in a spoken language understanding system" *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE*, (2020): 7969-7973.

[16] *Douze M., Guzhva A., Deng C., Johnson J., Szilvasy G., Mazare, P., Lomeli, M., Hosseini L., Jegou H.* "The FAISS library" *arXiv preprint arXiv:2401.08281* (2024).

[17] *Bhandari M., Wang M., Poliannikov O., Shimizu K.* "RecQR: Using Recommendation Systems for Query Reformulation to correct unseen errors in spoken dialog systems" *Proceedings of the 17th ACM Conference on Recommender Systems* (2023): 1019-1022.

[18] *Huang P. S., He X., Gao J., Deng L., Acero A., Heck L.* "Learning deep structured semantic models for web search using clickthrough data" *Proceedings of the 22nd ACM international conference on Information & Knowledge Management* (2013): 2333-2338.

[19] *Mikolov T., Chen K., Corrado G., Dean J.* "Efficient estimation of word representations in vector space" *arXiv preprint arXiv:1301.3781* (2013).

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

[21] *Meister C., Cotterell R.* "Language model evaluation beyond perplexity" *arXiv preprint arXiv:2106.00085* (2021).

[22] *Sennrich R., Haddow B., Birch A.* "Neural machine translation of rare words with subword units" *arXiv preprint arXiv:1508.07909* (2015).