DOI: 10.17323/2587-814X.2024.2.22.34

Embedding-based retrieval: measures of threshold recall and precision to evaluate product search

Fedor V. Krasnov

E-mail: krasnov.fedor2@wb.ru

Research Center of WB SK LLC, Moscow, Russia

Abstract

Modern product retrieval systems are becoming increasingly complex due to the use of extra product representations, such as user behavior, language semantics and product images. However, adding new information and complicating machine learning models does not necessarily lead to an improvement in online and business search performance, since after retrieval the product list is ranked, which introduces its own bias. Nevertheless, the business performance of a product search will be worse from ranking an incomplete list of products than a complete one, and the relevance of search results will not improve from perfect sorting of products that do not match the search query. Therefore, the main quality indicators for the products retrieval phase remain Recall and Precision at the k threshold. This paper compares several architectures of product retrieval systems in product search for e-commerce. To do this, the concepts of threshold Recall and Precision for information retrieval are investigated and the dependence of these measures on the order of issuance is revealed. An automatic procedure has been developed for calculating $R@k$ and $P@k$, which allows us to compare the effectiveness of information retrieval systems. The proposed automatic procedure has been tested on the WANDS public dataset for several key architectures. The obtained values $R@1000 = 84\% \pm 9\%$ and $P@10 = 67\% \pm 17\%$ are at the level of SOTA models.

Keywords: embedding-based retrieval, information retrieval, threshold metrics, semantic product search

Citation: Krasnov F.V. (2024) Embedding-based retrieval: measures of threshold recall and precision to evaluate product search. *Business Informatics*, vol. 18, no. 2, pp. 22–34. DOI: 10.17323/2587-814X.2024.2.22.34

Introduction

The efficiency of product search is essential for the success of online electronic marketplaces [1]. A study [2] has shown that more than 90% of users decide to purchase products after conducting a search. An early version of Amazon's search technology contributed more than 35% to sales [3]. The modern approach to product search [4–6], based on the information retrieval paradigm, consists of two stages: document retrieval and ranking. Documents, in this context, refer to product data or product modalities. Product data retrieval is central to the search process. If a product is not found in the product catalog, it will not be displayed in the search results or ranked according to the priorities of buyers, sellers and the marketplace. Product cards are multi-modal documents, as product data can be presented in various forms, such as a product name, a list of characteristics, graphical images, video and customer reviews. To ensure maximum completeness, data retrieval must be performed from each modality. Combining product cards retrieved from different sources into a single list for subsequent ranking is a separate process that is outside the scope of this study.

Modern approaches to data retrieval based on high-dimensional vector representations using artificial neural networks with deep learning have the potential to create a unified space for combining multi-modal product cards.

There is a fundamental distinction between lexical retrieval methods [7] and embedding-based retrieval methods. Lexical retrieval techniques are based on the presence or absence of specific tokens in a document allowing for a definitive determination of whether a given document matches a search query.

For example, consider a query for the term "skirt" in a product catalog containing two product descriptions (PD1 and PD2), both of which describe a garment. Using a lexical retrieval method applied to this catalog, only PD1 would be retrieved, since it contains the exact term "skirt." In contrast, an embedding-based method would return PD1 with a matching score of 0.9 and PD2 with a score of 0.1, indicating that PD2 is less relevant to the query.

Lexical retrieval leads to sparse results, while embedding-based methods produce more dense results. In order to obtain the optimal number of relevant product data, it is necessary to set a relevance threshold using an embedding-based approach. In the event of a high cutoff threshold, the output will be shorter and easier to rank, however, there is a possibility of a decrease in the recall of the output data. In the scenario where a low cutoff threshold is used, recall will be higher, but a significant amount of computing resources will need to be allocated to rank the data. This is unacceptable, since ranking must be performed in near real-time due to the necessity of considering various factors such as the user's location, availability of products and pricing. Therefore, the challenge of finding optimal settings for the retrieval system is highly relevant.

Business metrics for product search systems can be broadly categorized into two groups: online and offline metrics. Online metrics are collected during the actual use of the product search system in realistic conditions. These metrics take into account user interactions, such as whether the user clicks on a product card that has been found. While there are numerous online metrics, they all pertain to some form of user engagement and are beyond the scope of this discussion.

Offline metrics are measured in a controlled environment prior to deploying a new version of an information retrieval system. These metrics determine whether a specific set of relevant results are returned when searching for documents using the system.

In scientific literature, two types of offline metrics are commonly distinguished: those that consider the order in which documents are retrieved and those that do not [7, 9]. Metrics that consider the order include discounted cumulative gain (DCG) and normalized DCG, as well as mean reciprocal rank (MRR). Metrics that do not consider order include recall and precision, which are the most straightforward indicators of a system's suitability for implementation.

The development of new iterations of product search engines is a lengthy and costly process that involves a range of organizational and technological measures that are crucial for business success. Evaluating the efficacy of a new iteration of the product search engine is an essential step that can be repeated. The availability of clear, informative, cost-effective and scientifically substantiated metrics increases the probability of the successful implementation of new iterations. Therefore, this research focuses specifically on metrics related to recall and precision.

This paper presents a description of the research methodology, experimental findings, and conclusions.

1. Methodology

Search quality assessment indicators require precise definitions in order to accurately interpret research findings. The presence of a mathematical formula for an indicator within a paper can be ambiguous without a detailed explanation. For instance, while books [10] and studies [11] present a formula for calculating the accuracy index for a singular search query, it is evident that the accuracy index will vary for different queries. The reduction of dependencies between recall and precision metrics in studies [5, 12] was conducted without specifying a threshold, significantly complicating the interpretation of results. It is essential to justify the selection of indicators used to evaluate product search quality. It is uncommon to find justifications for the utilization of specific indicators in scholarly articles. For instance, in [12], various accuracy metrics were chosen for two datasets: MS MARCO Dev [13], *MRR@10*; and TREC2019 DL [14], *MAP@10*. The relevance algorithm's significance in formulas for *AP@k* in work [15] has been overlooked. Additionally, the accuracy metric for high threshold values $k > 1000$ has been considered in article [16], which necessitates separate justification because the accuracy metric is significant for the "top page" search results. Study [17] only provides the recall metric for thresholds 10, 50, and 100 without analyzing the precision metric. Note that the formulas for recall and precision metrics in statistics differ from those for information retrieval, so we will provide a textual explanation of the algorithms used to calculate *R@k* and *P@k* values for information retrieval purposes.

Definition 1: The threshold recall *R@k* is the average value $Q = \{q_i\}$ across all search queries q_i. For each search query q_i, we calculate the intersection of the set of product cards that match the query $C^g_{q_i}$ – the true positive set, with the top k product cards in the sorted list of retrieved cards results $C^r_{q_i}@k : \left|C^g_{q_i} \cap C^r_{q_i}@k\right|$, divided by $\left|C^g_{q_i}\right|$ – the total number of product cards matching the query (1):

$$R@k = \frac{1}{|Q|}\sum_{q_i \in Q} \frac{\left|C^g_{q_i} \cap C^r_{q_i}@k\right|}{\left|C^g_{q_i}\right|}, \quad (1)$$

where

$|Q|$ – the number of search queries being considered;

k – the threshold for cutting off search results;

q_i – the search query $q_i \in Q$;

$C^g_{q_i}$ – the set of all products matching the search query q_i;

$C^r_{q_i}$ – search results, the set of all products found by the search query q_i.

By analogy with *R@k*, the formula for the precision indicator *P@k* is given by the following expression (2):

$$P@k = \frac{1}{|Q|}\sum_{q_i \in Q} \frac{\left|C_{q_i}^{g} \cap C_{q_i}^{r}@k\right|}{k}. \quad (2)$$

The difference between the $R@k$ and $P@k$ formulas lies in the calculation method. In the denominator of the $R@k$ formula, the number of product cards relevant to the search query is used, while in the $P@k$ formula, only the number of relevant product cards up to a certain threshold is considered.

Based on formulas (1) and (2), we can analyze the behavior of the $R@k$ and $P@k$ indicators depending on the threshold value k. The limit values for the completeness indicator $R@k$ are presented in formula (3):

$$\lim_{k\to 0} R@k = 0,$$
$$\lim_{k\to \infty} R@k = 1. \quad (3)$$

For the precision indicator $P@k$, the limit values are provided in formulas (4):

$$\lim_{k\to 0} P@k = 1,$$
$$\lim_{k\to \infty} P@k = 0. \quad (4)$$

To demonstrate the dependence of the recall and precision threshold indicators as defined by formulas (1) and (2), on the order of search results, we propose the following lemma:

Lemma 1: The threshold values for recall and precision are dependent on the order in which search results are presented.

The relationship between recall and precision metrics and the ordering of search results is examined in more detail. *Figure 1* illustrates an example of how these metrics can be calculated.

It can be inferred from *Fig. 2* that the order in which search results are produced affects the values of recall and precision metrics when these metrics are calculated. For example, if an item with *Id4* is closer than a threshold of 3 to the beginning of search results, then the precision for *@3* will be 2/3 and the recall for *@3* will be 2/7. However, within the threshold ($k = 3$), the position of item with *Id4* will not influence the values of these metrics for this particular threshold (see *Fig. 2*). Therefore, the threshold-specific indicators of recall and precision are dependent on the output order. *Lemma 1* can therefore be considered analytically proven.

Given that the recall and precision thresholds depend on the ordering of items in the output, it is necessary to evaluate the retrieval system in an integrated manner, rather than at a single point K, at which the behavior of the metric may be biased. Therefore, it is recommended to evaluate the system using the sum of

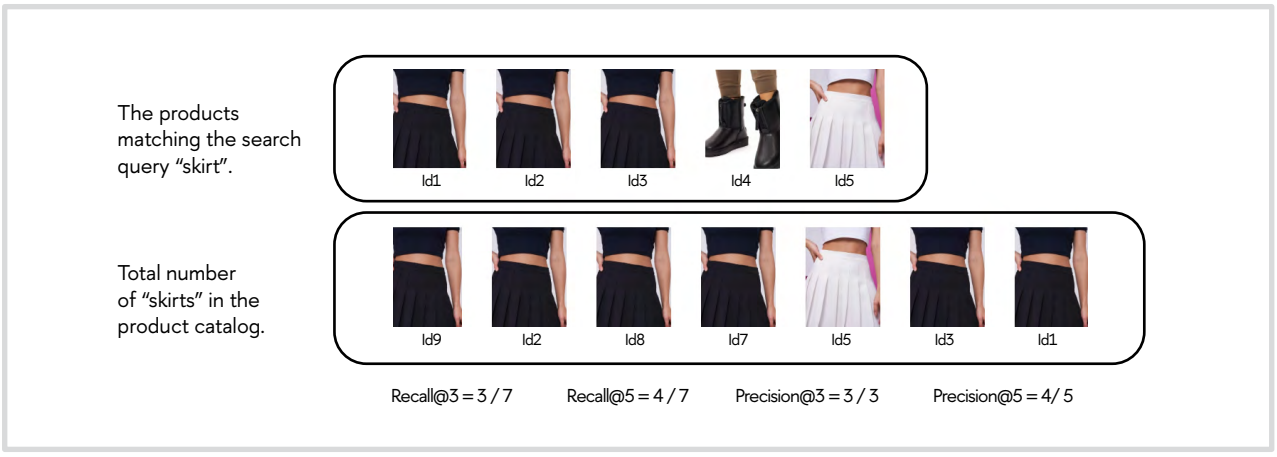

Fig. 1. Recall and Precision of search results.

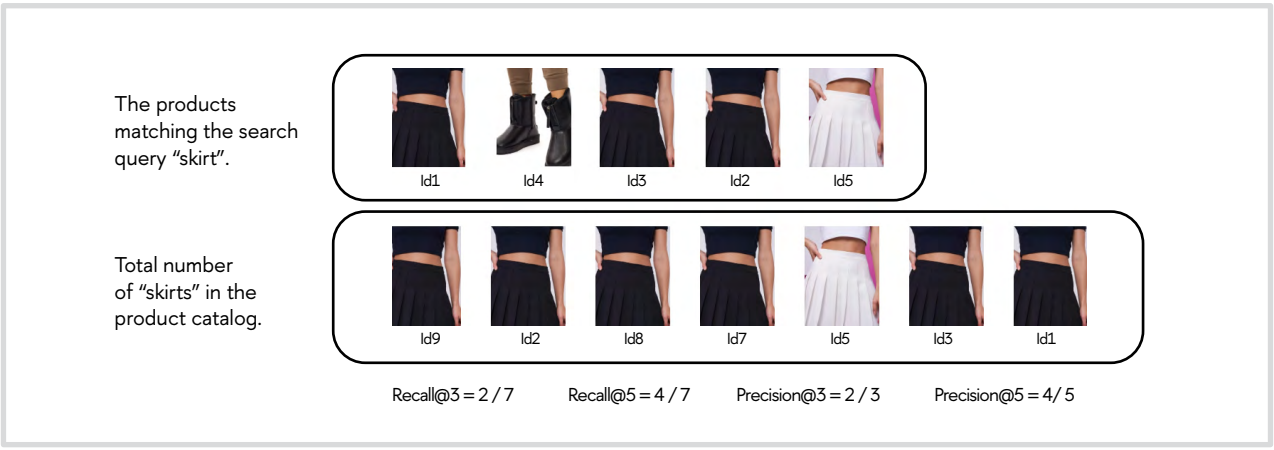

Fig. 2. Threshold recall and precision of search results.

discrete precision metrics for threshold values between 1 and K, where K is a hyperparameter of the system. In the example shown in *Fig. 1*, the value of the integrated precision metric for the threshold:

$$AP@K = \frac{1}{K}\sum_{k=1}^{K} P@k.$$

Next, we will discuss how the intersection of two sets of products is achieved. *Figure 1* illustrates the products with Id_i, than $C^g_{q_i} = [Id9, Id2, Id8, Id7, Id5, Id3, Id1]$ and $C^r_{q_i}@k = [Id1, Id2, Id3, Id4, Id5]$ for threshold k equal to 5. To calculate the component $\left|C^g_{q_i} \cap C^r_{q_i}@k\right|$, an intersection operation is performed. In this operation, each element from $C^g_{q_i}$ set is compared with each element from $C^r_{q_i}@k$ set. In the case under consideration, the elements are products that have several modalities. Therefore, they can be compared using different matching algorithms.

Figure 1 shows two product modalities: a digital identifier (*Id*) and an image. Additionally, other product modalities based on textual representations, such as the name of a product on eBay [16] or the characteristics of a product in an Amazon study [5], are also considered in the literature.

In general, the problem of determining the identity of two products can be solved by combining the similarity functions of different modalities. For example, this approach is used in the study by [18].

The general form of the product identification function is given by $S_V(\cdot,\cdot)$, where V represents the vector space in which products will be compared and a $\cdot$ represents the modality of the product. Next, the following vector spaces will be analyzed: N, which is the space of natural numbers representing digital identifiers for products; T, which is a space of strings representing product names; and I, which is a space of raster image representations of products. It can then be written that $V \in \{N, T, I\}$. Each of these vector spaces $\{N, T, I\}$ can be described in terms of sparsity and density.The space N postulates that two cards are identical only if their digital identifiers match. The digital identifier refers to a unique product code associated with each card. Each card can only have one identical counterpart with the same product code. The space N is considered sparse, as the ratio of pairs of identical products to all possible product pairs is close to zero. In other words, when considering a matrix of product identities, identical pairs would be located on the main diagonal.

- The space T postulates that products with identical names are considered duplicates (i.e., identical). This is a more lenient condition for the identification of products compared to the N space, as all products bearing the name "blue nail polish" would be considered identical. Given the significantly larger number of such products in a catalog, this is a practical approach. Within the T space, additional

subspaces can be defined to allow for even more intuitive comparisons between product cards. One such subspace could take into account the presence of specific tokens (e.g., Bag of Words) without considering their order. This subspace would consider products with names like "blue nail polish" and "blue polish" to be identical. Furthermore, the T space is sparse, which means it represents a more efficient representation of products in terms of storage and processing.

- The space I determines the identity of products by comparing their visual representations. *Figure 1* illustrates an example of a product, "skirt", which has different digital identifiers. However, in the I-space, such products would be considered identical. Image comparison is achieved through an algorithmic process that reduces the dimensionality of the image space (embedding) while maintaining the essential features of the original image. This process calculates the cosine similarity between images, which is used as a metric for determining whether two images are similar. The approach involves setting a threshold value for the cosine proximity between product images. If the similarity between two images exceeds this threshold, they are considered identical in the I-space. It should be noted that the I-space is dense, meaning that all elements within it are similar to each other to some degree.

Therefore, it is necessary to define vector spaces $\{N, T, I\}$ for the product identity function $S_V(\cdot,\cdot)$.

It has been analyzed which hyperparameters control the semantic search for products. Possible spaces for the identity function $S_V(\cdot,\cdot)$ have been considered above, but not for the information retrieval system for products itself. Modern information retrieval systems for products based on embeddings are based on a combination of modeling the semantics of language, product images and user behavior (*Fig. 3*).

The compositional approach to information retrieval is discussed in the Amazon research [5], which demonstrates that various types of user interactions can lead to substantial improvements in performance when integrating outputs. In a study of the information retrieval system for product searches on the Taobao online marketplace [6], the authors present a model called Multi-Grained Deep Semantic Product Retrieval (MGDSPR), which simultaneously models query semantics and historical user behavior data to produce a more comprehensive output of relevant products. Similarly, the Walmart online retail platform also utilizes a combination of data sources for its information retrieval system [19]. The semantic model architecture consists of two "towers", each of which is an artificial neural network based on deep learning, creating an embedding representation for a search query and product, respectively. Evaluation of the "query – product" pair is performed using a loss function that is based on cosine similarity. A study by Tencent researchers on a retrieval system in sparse space [20] demonstrates an improvement in efficiency in terms of completeness while reducing disk space

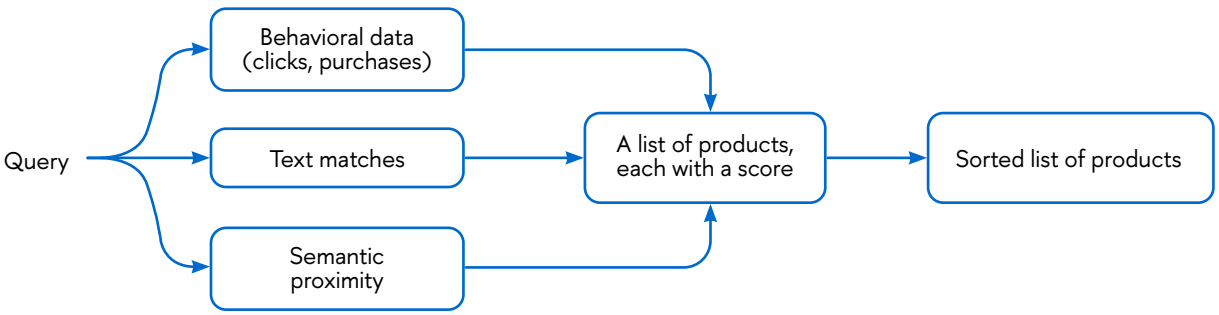

Fig. 3. Composition of search results and ranking.

consumption. According to this study [21], Etsy also utilizes multimodality in its UPPER product search model, albeit to a slightly broader extent than typical, as personalization is achieved through training a "two-tower" model based on user behavior data.

Despite the fact that various threshold indicators of recall and precision have been used as performance metrics in previous studies [5, 6, 19], there has been a lack of attention given to the ranking of the combined output. A common feature among these studies is the use of threshold values as proxies for model performance, rather than as part of a loss function for finding optimal retrieval system parameters. Additionally, it is worth noting that there is a need for separate analysis of the formulas for threshold recall and precision indicators, which are not provided in some studies [6, 19]. Lastly, the aforementioned articles [5, 6, 20, 21] do not account for an error that can occur when calculating threshold indicators for recall and precision on a set of search queries. *Table 1* displays the recall thresholds *R@1000* from these studies.

Table 1.

Indicators of industry research

Model	Indicator	Value
UPPER [21]	*R@1000*	0.85
MGDSPR [6]	*R@1000*	0.85
SPS [5]	*R@1000*	0.79
SPS [5]	*MAP@10*	0.74

It is not practical to consider the recall indicator for industrial retrieval systems with values k less than 100, because the denominator in formula (1) would have too large values. However, the threshold precision indicator for $k = 10$, conversely, reflects well the quality of the retrieval system, as it corresponds to the most relevant products. Therefore, it is surprising that the threshold precision indicator has not been measured in previous studies [5, 6, 20, 21].

Based on the established lemma and the findings of recent research by industry experts, the primary objective of this study is to conduct a pilot testing of an automated process for comparing various versions of product information retrieval systems for product searches using autonomous indicators of threshold recall and precision.

2. Experiment

To address the research objective, a digital experiment was conducted. The following steps were taken:

1. A manually annotated dataset was selected D_G.
2. Three models of the retrieval system were trained: DE (a "two-tower" model with one modality), DE2 (a "two-tower" model with two modalities) and a model using a single encoder with one modality (SE).
3. The performance of the retrieval systems as compared on the selected dataset D_G based on industry-standard metrics for recall and precision benchmarks.

WANDS [22] has been selected as the dataset D_G for this study, which provides a comprehensive and objective benchmark for evaluating search engine performance based on an e-commerce dataset. The key features of this dataset include:

- 42 994 unique product candidates;
- 480 search queries;
- 233 448 ratings for relevance (query, product) pairs.

The WANDS dataset has a three-level classification of "request – product" pairs: "exactly matching" (Exact), "partially matching" (Partial) and "not matching" (Irrelevant). For training models of the retrieval system, only these two values were used to create the loss function: exact matches were assigned a value of 1 and irrelevant matches a value of −1. The classes are balanced in the training data.

Figure 4 illustrates the relationship between the threshold for recall and precision, based on the classification, which was created using formulas (1) and (2), respectively.

Depending on the threshold precision value (P), *Fig. 4* demonstrates significant deviations from the

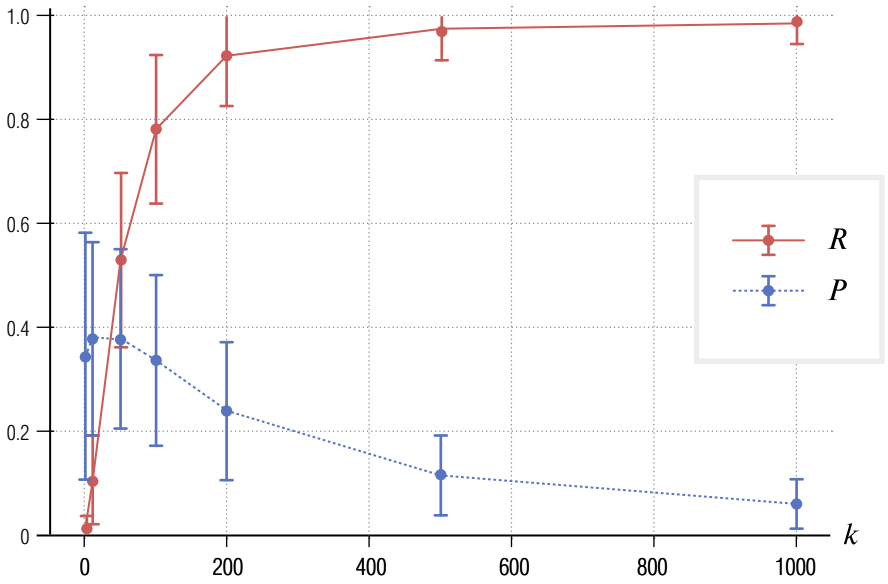

Fig. 4. The dependence of recall (R) and precision (P) on the threshold k for a dataset with markup D_G without a retrieval system.

expected model behavior. At threshold values of $k = 10$ and 50, the corresponding threshold precision values are $P@10 = 0.37 \pm 0.17$ and $P@50 = 0.38 \pm 0.18$, respectively. To investigate the factors influencing the allowable range of values, *Fig. 5* presents the dependencies of recall and precision for individual queries.

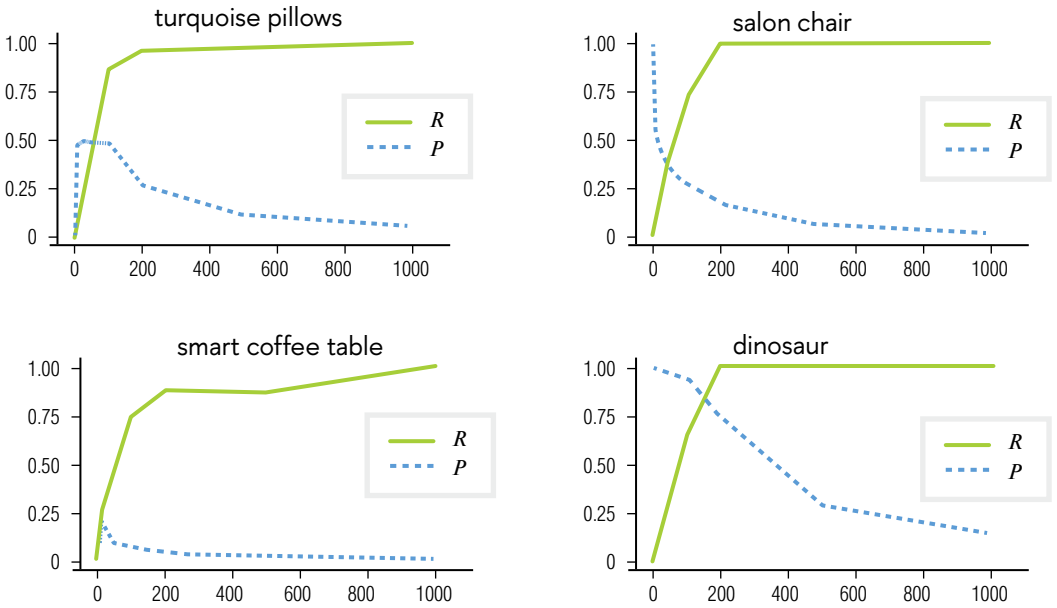

Fig. 5. Recall and precision for individual queries without a retrieval system.

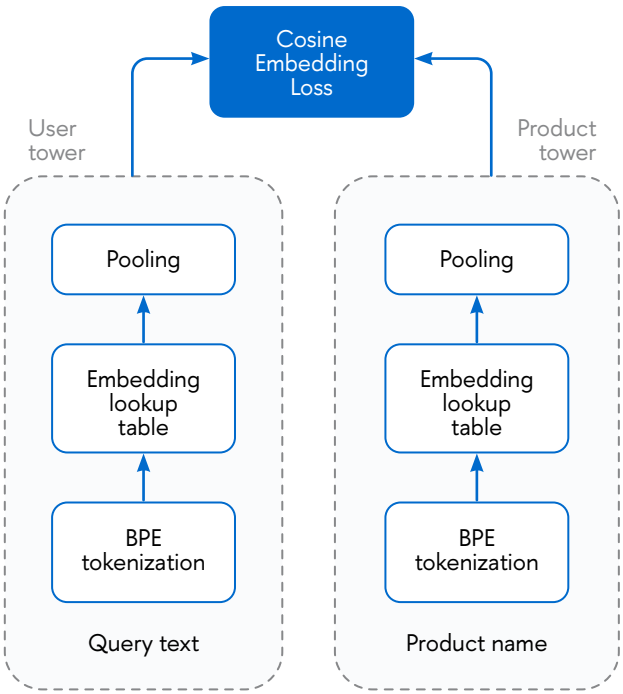

*Fig. 6.*DE retrieval model.

Three retrieval system architectures were selected for the experiment – DE (*Fig. 6*), DE2 (*Fig. 7*), SE (*Fig. 8*).

To train models for retrieval systems, a dataset D_G was used with the following parameters: the AdamW optimizer, a learning rate that cyclically varied between 0.01 and 0.1 over 500 epochs with early stopping. The BPE (Byte-Pair Encoding) method was used for tokenization, with a dictionary size of 16 000 tokens for products and 512 for queries for DE and DE2 models, and 16 000 tokens for SE. Among the hyperparameters for the retrieval model, attention was focused on the dimensionality of the token vector, which affects both prediction speed and model size in memory.

As part of the experiment, we observed that when switching from a dimension of 256 to 32, the validation error increased by 3.5% while the size of the token table decreased by a factor of 8. This resulted in a more than threefold increase in forecasting speed and learning. All dependencies between validation errors and token vector dimensions are illustrated in *Fig. 9*.

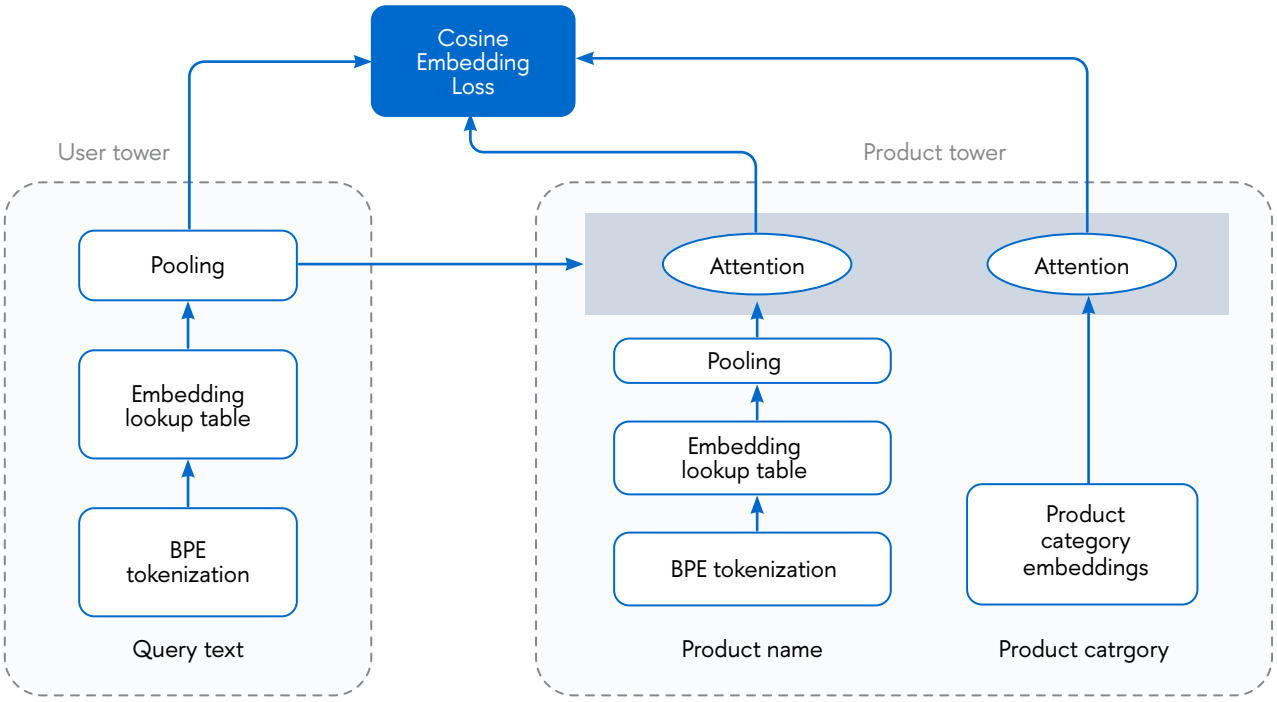

Fig 7. DE2 retrieval model.

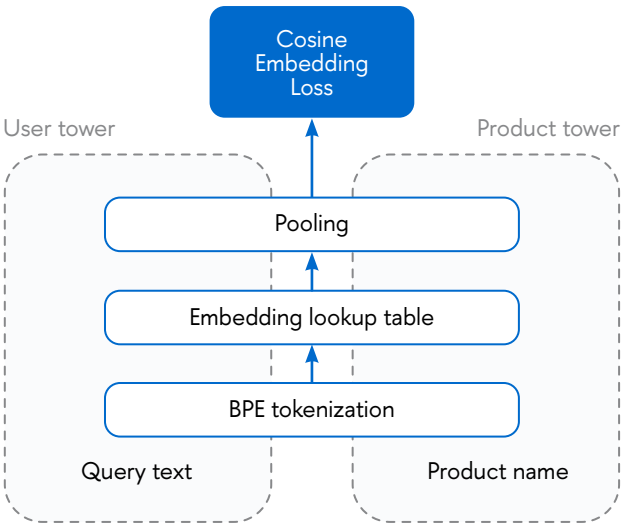

Fig 8. SE retrieval model.

Three selected retrieval techniques were applied to the dataset and candidate products were generated. For these products, thresholds of recall and precision were established. As a result of evaluating the performance of the various retrieval methods, the following findings were obtained (*Table 2*).

Table 2.

Values of threshold indicators of various retrieval systems

Model	*R@1000*		*P@10*	
	mean	std	mean	std
DE	0.75	0.10	0.68	0.16
DE2	0.73	0.11	0.66	0.17
SE	0.84	0.09	0.67	0.17
Dataset D_G	0.99	0.03	0.37	0.17

In *Table 2*, the dataset D_G row shows the *R@1000* and *P@10* values without retrieval systems. Without the use of retrieval systems, the precision at $k = 10$ is the lowest for all considered models, at 0.37. This indicates that no special attention was paid to the order of examples when marking them up.

With the help of retrieval models, it was possible to sort examples in descending order based on precision, so that the threshold value for DE increased to 0.68. It is noteworthy that completeness at $k = 1000$ was the highest for all models, which may be a self-check for the calculation process. Errors (std) across all models are similar for precision (0.10, 0.11, and 0.09) and recall (0.16, 0.17, and 0.17). Therefore, models make errors on different queries for the same type of data. DE2 did not perform as well as expected, despite including an additional modality in training. The SE model with the lowest number of training parameters demonstrated the highest level of recall of 0.84 ± 0.09.

Conclusion

The evaluation of product search systems is critical for making informed business decisions on online electronic trading platforms. Large technology companies often achieve success through a well-designed product search. Measuring the effectiveness of a product search is a continuous process that is linked to the ongoing improvement of data and scientific advancements in the field of machine learning.

Key indicators of recall and precision are easily interpretable, in contrast to other metrics for evaluating product search effectiveness and can function as objective measures of both data tagging and the performance of product information retrieval systems.

Using the WANDS public dataset, we have demonstrated that relatively straightforward model architectures for retrieval systems can attain values for metrics similar to those produced by industry leaders, despite having significantly fewer model parameters. As part of the research, an automated process has been developed to determine thresholds for a set of search terms. As a consequence of this research, an automated method for measuring the effectiveness of product information retrieval systems (first stage retrieval) has been created and experimentally validated.

The aim of the study is to conduct experiments on a greater number of product cards, measuring the impact of pre-trained models compared with training from scratch and retraining models to retrieve information about products. ■

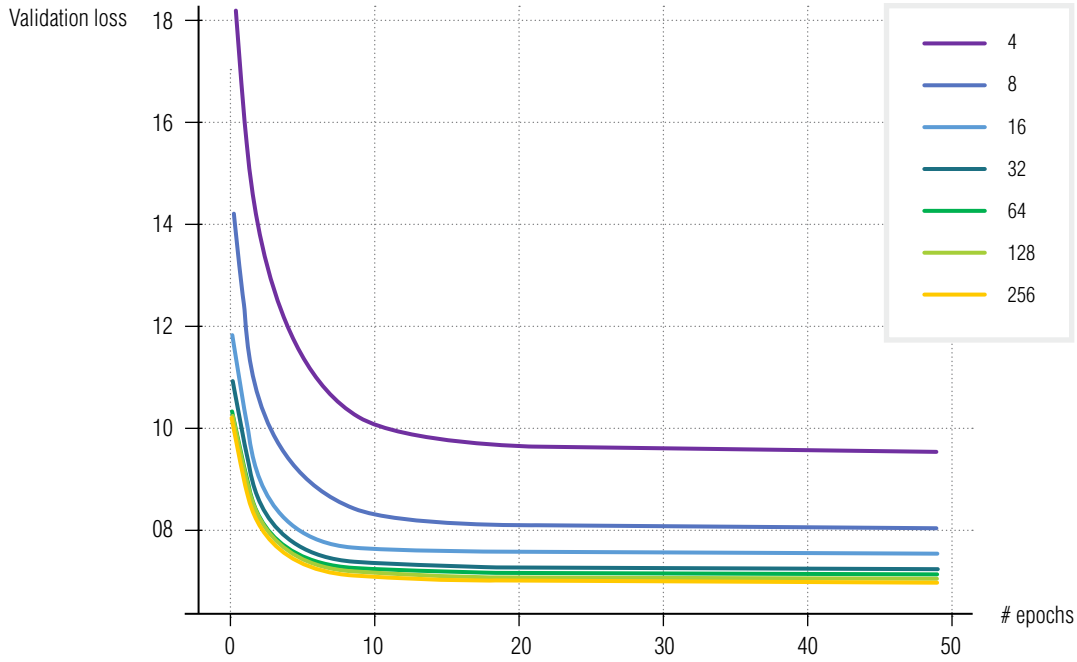

Fig 9. Dependencies of validation errors on the dimension of token vectors.

References

1. Matveev M.G., Aleynikova N.A. Titova M.D. (2023) Decision support technology for a seller on a marketplace in a competitive environment. *Business Informatics*, vol. 17, no. 2, pp. 41–54. https://doi.org/10.17323/2587-814X.2023.2.41.54

2. Luo C., Goutam R., Zhang H., Zhang C., Song Y., Yin B. (2023) Implicit query parsing at Amazon product search. Proceedings of the *46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. https://doi.org/10.1145/3539618.3591858

3. Linden G., Smith B., York J. (2003) Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80.

4. Huang P., He X., Gao J., Deng L., Acero A., Heck L. (2013) Learning deep structured semantic models for web search using clickthrough data. Proceedings of the *22nd ACM international conference on Information Knowledge Management*, pp. 2333–2338. https://doi.org/10.1145/2505515.2505665

5. Nigam P., Song Y., Mohan V., Lakshman V., Ding W., Shingavi A., Teo C.H., Gu H., Yin B. (2019) Semantic Product Search. Proceedings of the *25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2876–2885. https://doi.org/10.1145/3292500.3330759

6. Li S., Lv F., Jin T., Lin G., Yang K., Zeng X., Wu X., Ma Q. (2021) Embedding-based product retrieval in Taobao search. Proceedings of the *27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3181–3189. https://doi.org/10.1145/3447548.3467101

7. Krasnov F.V., Smaznevich I.S., Baskakova E.N. (2021) The problem of loss of solutions in the task of searching similar documents: Applying terminology in the construction of a corpus vector model. *Business Informatics*, vol. 15, no. 2, pp. 60–74. https://doi.org/10.17323/2587-814X.2021.2.60.74

8. Mitra B., Craswell N. (2017) Neural models for information retrieval. *arXiv*: 1705.01509. https://doi.org/10.48550/arXiv.1705.01509

9. Gudivada V.N., Rao D., Gudivada A.R. (2018) Information retrieval: concepts, models, and systems. *Handbook of Statistics*, vol. 38, pp. 331–401. https://doi.org//10.1016/bs.host.2018.07.009

10. Büttcher S., Clarke C.L.A., Cormack G.V. (2010) *Information retrieval: Implementing and evaluating search engines*. The MIT Press: Cambridge, Massachusetts, London, England.

11. Leonhardt J. (2023) *Efficient and explainable neural ranking*. PhD thesis. Hannover: Gottfried Wilhelm Leibniz Universität. https://doi.org/10.15488/15769

12. Campos D.F., Nguyen T., Rosenberg M., Song X., Gao J., Tiwary S., Majumder R., Deng L., Mitra B. (2016) MS MARCO: A human generated MAchine Reading COmprehension dataset. *arXiv*: 1611.09268. https://doi.org/10.48550/arXiv.1611.09268

13. Craswell N., Mitra B., Yilmaz E., Campos D., Voorhees E.M. (2020) Overview of the TREC 2019 deep learning track. *arXiv*: 2003.07820. https://doi.org/10.48550/arXiv.2003.07820

14. Leonhardt J., Müller H., Rudra K., Khosla M., Anand A., Anand A. (2023) Efficient neural ranking using forward indexes and lightweight encoders. *ACM Transactions on Information Systems*. https://doi.org/10.1145/3631939

15. Gao L., Dai Z., Chen T., Fan Z., Durme B.V., Callan J. (2021) Complement lexical retrieval model with semantic residual embeddings. *Advances in Information Retrieval. ECIR 2021. Lecture Notes in Computer Science*, vol. 12656, pp. 146–160. https://doi.org/10.1007/978-3-030-72113-8_10

16. Trotman A., Degenhardt J., Kallumadi S. (2017) The architecture of eBay search. *SIGIR Workshop on eCommerce. eCOM@ SIGIR. Tokyo, Japan, August 2017.*

17. Chang W., Jiang D., Yu H., Teo C.H., Zhang J., Zhong K., Kolluri K., Hu Q., Shandilya N., Ievgrafov V., Singh J., Dhillon I.S. (2021) Extreme multi-label learning for semantic matching in product search. Proceedings of the *27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining,* pp. 2643–2651. https://doi.org/10.1145/3447548.3467092

18. Krasnov F. (2023) Estimation of time complexity for the task of retrieval for identical products for an electronic trading platform based on the decomposition of machine learning models. *International Journal of Open Information Technologies*, vol. 11, no. 2, pp. 72–76.

19. Magnani A., Liu F., Chaidaroon S., Yadav S., Suram P.R., Puthenputhussery A., Chen S., Xie M., Kashi A., Lee T., Liao C. (2022) Semantic retrieval at Walmart. Proceedings of the *28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington DC, USA, August 14–18, 2022*, pp. 3495–3503. https://doi.org/10.1145/3534678.3539164

20. Gan Y., Ge Y., Zhou C., Su S., Xu Z., Xu X., Hui Q., Chen X., Wang Y., Shan Y. (2023) Binary embedding-based retrieval at Tencent. Proceedings of the *29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach CA, USA, August 6–10, 2023*, pp. 4056–4067. https://doi.org/10.1145/3580305.3599782

21. Jha R., Subramaniyam S., Benjamin E., Taula T. (2023) Unified embedding based personalized retrieval in Esty search. *arXiv*: 2306.04833. https://doi.org/10.48550/arXiv.2306.04833

22. Chen Y., Liu S., Liu Z., Sun W., Baltrunas L., Schroeder B. (2022) WANDS: Dataset for product search relevance assessment. *Advances in Information Retrieval. ECIR 2022. Lecture Notes in Computer Science*, vol. 13185, pp. 128–141. https://doi.org/10.1007/978-3-030-99736-6_9

About the author

Fedor V. Krasnov

Cand. Sci. (Tech.);

Specialist in information retrieval and recommendation systems in e-commerce, Researcher of the Research Center of WB SK LLC based on the Skolkovo Innovation Center, 5, St. Nobel, Moscow, Russia;

E-mail: krasnov.fedor2@wb.ru

ORCID: 0000-0002-9881-7371