Multi-word Term Embeddings Improve Lexical Product Retrieval

Fedor Krasnov[1], **Viktor Shcherbakov**[23]
[1] Research Center of Wildberries SK LLC based on the Skolkovo Innovation Center,
[2]University of Geneva, [3]University of Lausanne
krasnov.fedor2@wb.ru, Viktor.Shcherbakov@unil.ch

Abstract

Product search is uniquely different from search for documents, Internet resources or vacancies, therefore it requires the development of specialized search systems. The present work describes the H1 embdedding model, designed for an offline term indexing of product descriptions at e-commerce platforms. The model is compared to other state-of-the-art (SoTA) embedding models within a framework of hybrid product search system that incorporates the advantages of lexical methods for product retrieval and semantic embedding-based methods. We propose an approach to building semantically rich term vocabularies for search indexes. Compared to other production semantic models, H1 paired with the proposed approach stands out due to its ability to process multi-word product terms as one token. As an example, for search queries ”new balance shoes”, ”gloria jeans kids wear” brand entity will be represented as one token - ”new balance”, ”gloria jeans”. This results in an increased precision of the system without affecting the recall. The hybrid search system with proposed model scores mAP@12 = 56.1% and R@1k = 86.6% on the WANDS public dataset, beating other SoTA analogues.

Keywords: semantic product search, entity recognition, SentencePiece, transformers, ColBERT

1. Introduction

Product search systems are required to operate with both low latency and high recall, since they scan the whole product catalog of billions of items. Common product search methods initially used lexical search models. These models calculate the relevance metric based on heuristics that measure exact word match between the search query and textual product representations. Lexical search models such as BM25 (Robertson and Walker, 1994) have been relevant for decades, and are still widely used today. The recent alternatives, neural extraction methods, demonstrate increased search effectiveness metrics, but also possess their own flaws (Zeng et al., 2022, 2023; Pan et al., 2024; Hofstätter et al., 2020). Naturally, the research gravitates towards the hybridization of the two approaches, combining the advantages of each.

The disadvantages of lexical models are well-researched: (E1) a possible mismatch between query and document vocabularies (Furnas et al., 1987; Zhao and Callan, 2010) leads to search recall degradation; (E2) lack of semantic understanding of queries and documents (Li and Xu, 2014) decreases search precision. These described limitations result in failures to retrieve relevant documents using lexical methods for information retrieval. To resolve these issues a number of extensions to the lexical model have been introduced in the past decades, including, but not limited to: query expansion (Lavrenko and Croft, 2001; Lesk, 1969; Qiu and Frei, 1993; Xu and Croft, 2017), document expansion (Efron et al., 2012; Liu and Croft, 2004; Gao et al., 2004), term dependencies model (Metzler and Croft, 2005; Xu et al., 2010), topic modeling (Deerwester et al., 1990; Wei and Croft, 2006), machine translation models for information retrieval (Berger and Lafferty, 1999; Karimzadehgan and Zhai, 2010). Despite mentioned advances, the research in lexical models for information retrieval progresses relatively slowly, since the majority of these methods work with discrete, sparse lexical representations and inevitably inherit their limitations.

With the development of representation learning in information retrieval, semantic search models at the offline information extraction stage of the search have seen an increased research interest in recent years. During this stage the indexes are built for matching queries with the documents. The Figure 1 schematically describes an example product search system that uses indexes built during the information extraction stage for fast responses to queries.

Starting in 2013, the improvement of word embeddings (Bravo-Marquez et al., 2013; Mikolov et al., 2013; Pennington et al., 2014) has led to a number of studies using embeddings for the extraction stage (Clinchant and Perronnin, 2013; Ganguly et al., 2015; Vulić and Moens, 2015). Unlike discrete lexical representation, word embeddings offer a continuous representation that can help with the problem of query and document vocabularies mismatch to some extent. After 2016, a spike of research attention to the application of deep learning methods to the information extraction stage is seen (Boytsov et al., 2016; Henderson et al., 2017). These methods are applied either for improving document representation within

The Seventh Workshop on e-Commerce and NLP (ECNLP 7), pages 115–124
21 May, 2024.

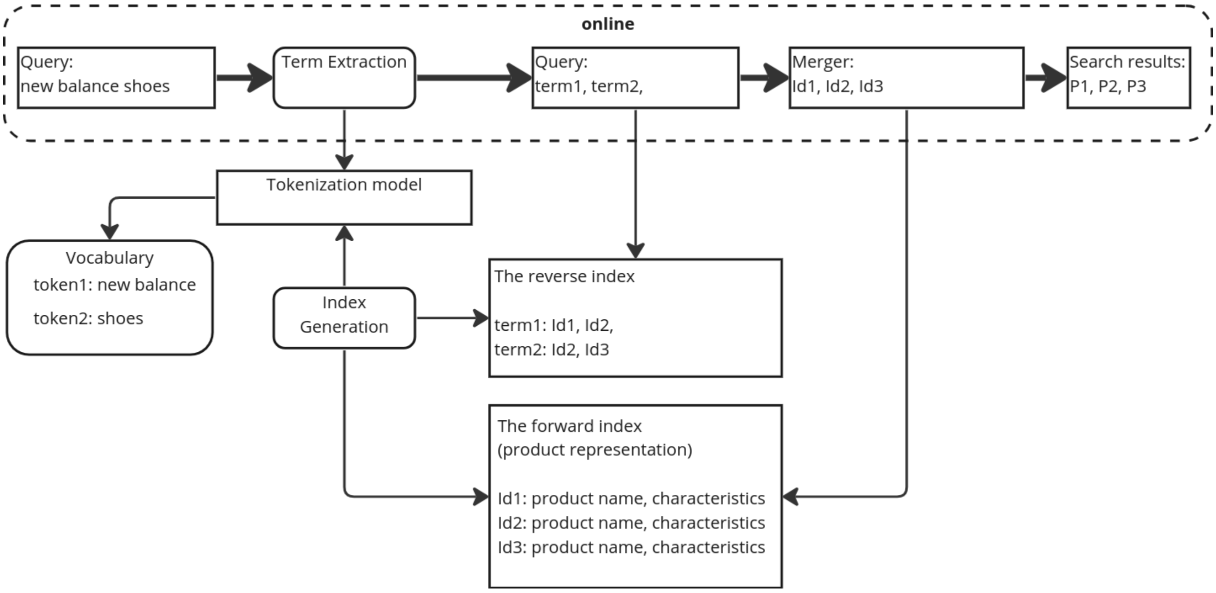

Figure 1: The indexes and a tokenization model built during offline information extraction are used in online setting to respond to queries with low latency. The quality of built index is detrimental to the performance of the search system.

the framework of the traditional paradigm of discrete lexical representation (Bai et al., 2020; Dai and Callan, 2019; Nogueira et al., 2019), or directly for forming novel semantic search models within the sparse/dense representation paradigm (Gillick et al., 2018a; Jean et al., 2015; Khattab and Zaharia, 2020; Zamani et al., 2018).

While closely related to document information retrieval, the product search problem is uniquely different in a few aspects:

- Ranking mechanisms based on weighing textual features (TF/IDF, BM25) differ in product search. For example, the token frequency in the product title does not affect the query relevancy.

- Products are multimodal. A product page includes a title, description, characteristics, images, videos, etc. The search system can take into account multiple modalities of a page.

- Search queries are motivated by an interest in purchasing a product. Customer behavior differs significantly from vacancy search or Internet resource search behavior.

- Product search effectiveness is evaluated on a modality-wise basis.

The primary research question of the present paper is to evaluate the impact of the semantic model and tokenization architectures on offline metrics of a hybrid product search system.

In the following sections, we describe in detail the research methodology, conducted experiments and conclusions.

2. Related work

2.1. Neural Information Retrieval

Similar to document information retrieval trends, the development of product search systems has transitioned from lexical retrieval methods to neural retrieval methods (Li et al., 2021; Magnani et al., 2022; Nigam et al., 2019). DSSM (Huang et al., 2013), being one of the most popular neural network architectures, is based on a Dual Encoder paradigm (Gillick et al., 2018b; Yang et al., 2019; Karpukhin et al., 2020). The two independent “towers” of encoders—one for search queries and the other for product representation—embed queries and products into a shared space of fixed dimensionality. The shared space is used for similarity search (Vanderkam et al., 2013; Johnson et al., 2017) to retrieve products that are relevant to a search query. Thus far, the most promising results have been achieved by using the BERT model in a Dual Encoder architecture (Chang et al., 2020; Xiong et al., 2021; Lu et al., 2020). The general operating principle of these models is described in Eqs. (1) to (3).

$$\overrightarrow{q} = AvgPool\left[BERT_{\theta}^{l}(q)\right] \tag{1}$$

$$\overrightarrow{p} = AvgPool\left[BERT_{\theta}^{r}(p)\right] \tag{2}$$

$$s_{BERT}(\overrightarrow{q}, \overrightarrow{p}) = \overrightarrow{q}^{T} \cdot \overrightarrow{p} \tag{3}$$

Where $BERT^t_\theta$ and $BERT^r_\theta$ are the “left” and “right” encoders, respectively, transforming texts q and p into a shared space θ. The similarity function $s_{BERT}(\cdot, \cdot)$ is implemented with a scalar product of $\overrightarrow{q}$ and $\overrightarrow{p}$. The bottleneck in this architecture lies in the averaging of the token vectors.

The ColBERT (Khattab and Zaharia, 2020) model represents a particular variant of the Dual Encoder architecture, termed a Single Encoder. Models based on this architecture use the same encoder for both queries and products. However, the novelty of ColBERT lies in computing the similarity scores token-wise, instead of comparing the mean vectors. Given a search query q comprising m tokens and a product p comprising n tokens, the similarity function $s_{ColBERT}(\cdot, \cdot)$ is:

$$s_{ColBERT}(q_{1:m}, p_{1:n}) = \sum_{1}^{m} \max_{1..n} \left(\overrightarrow{q}^{T}_{1:m} \cdot \overrightarrow{p}_{1:n} \right) \tag{4}$$

The sum over maximum similarity scores for each token of a query in Eq. (4) implies that $n \cdot m$ scalar products need to be calculated, compared to one scalar product in $s_{BERT}(\cdot, \cdot)$.

2.2. Hybridization

It is accepted to understand hybridization as mixing the lexical and neural methods of information retrieval within one product search system. Hybridization can be applied at different stages of the search. For instance, the authors of the study Nigam et al. (2019) combined the search results of several distinct models based on lexical, behavioral, and semantic methods. Another hybridization principle was applied in the study Gao et al. (2021)—the lexical method was the primary retrieval mechanism, while a semantic model was trained to correct the mistakes of the lexical model.

2.3. Tokenization

The progress in tokenization methods has led to significant improvements in the offline metrics of natural language processing models (Kudo and Richardson, 2018; Sennrich et al., 2016). The BPE (Byte-Pair-Encoding) tokenization method was originally introduced as a data compression method (Gage, 1994). In constructing the BPE tokenizer, the initial vocabulary is sequentially extended until the preset limit is reached. The primary goal of applying BPE to natural text is to split words into commonly occurring subwords. Usually, little care is given to the actual semantics of the final tokens. However, unlike most applications, where semantic information can be represented by the combination of tokens, information retrieval often requires semantically rich tokens in order to use them as terms to construct effective search indexes, see Fig 1.

The later proposed alternative, the unigram tokenization method (Kudo, 2018a), demonstrates the opposite approach—the vocabulary size is sequentially pruned by removing rare tokens that can be replaced by common tokens. The unigram method was primarily introduced to provide multiple possible tokenizations for a given text with the use of a unigram language model. During vocabulary construction, both methods aim to minimize the length of text encoded in tokens and, in practice, produce similar tokenizations.

3. Methodology

3.1. H1

The H1 semantic model draws significant inspiration from ColBERT but architecturally simplified. It processes both queries and documents by tokenizing them and then passing them through a BERT-based Dual Encoder. The resulting embeddings are evaluated using the $s_{ColBERT}(\cdot, \cdot)$ similarity function. We explore the impact of different tokenization techniques in Ablation Study Section 4.2. A distinctive aspect of H1 is its approach to token-level lexical hybridization, where we enhance the tokenizer’s vocabulary with semantically rich terms to improve the semantic independence of standalone terms. The Experiments Section 4 provides a comprehensive analysis of the H1 system’s application in a product retrieval task. For this task specifically, we augmented the tokenizer’s vocabulary with a carefully selected list of brand names.

The rationale behind incorporating brand names into the vocabulary is rooted in understanding user search behavior, particularly when it comes to specific brands. For instance, when a customer searches for ”new balance shoes”, their intent is not to explore products related to the terms ”new” and ”balance” independently. Instead, they are looking for items specifically associated with the ”New Balance” brand. However, these customers may still be open to considering various types of ”shoes”.

H1 model is optimized on positive and negative product-query pairs using the following loss function:

$$L_{H1} = \left[\gamma - s_\theta(q_{1:m}, p^{+}_{1:n}) + s_\theta(q_{1:m}, p^{-}_{1:n}) \right]_{+0} \tag{5}$$

Where γ is a threshold and s_θ is a similarity relation parametrized by θ, applied to an m-token query q with a positive p^+ and a negative p^- product description example. Negative examples are sampled by selecting a random product from the current batch. The square brackets around the

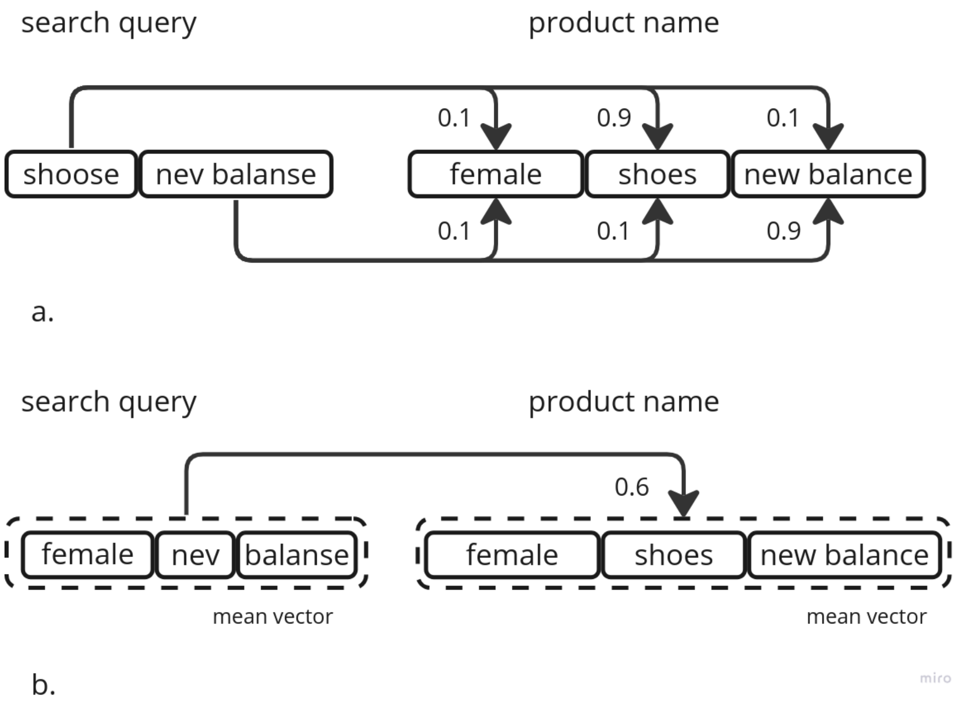

Figure 2: Token handling principle of H1 (a), compared to that of a FastText (b). H1 attributes scores to each pair of query and document tokens, while the FastText-based system compares the mean vector representations.

equation, $[]_{+0}$, denote that negative values are set to 0.

3.2. Evaluation

Neural retrieval methods, given their computational intensity, are impractical for online product searches within catalogs containing billions of items. Instead, their utility shines in building indexes for product descriptions, as schematically outlined in the example in Figure 1. The actual neural encoder is never utilized to generate the embeddings for user queries. Our evaluation methodology mirrors these practical limitations, ensuring that our approaches are both realistic and aligned with the constraints of large-scale product retrieval systems.

For the query encoder E^q_θ, the product encoder E^p_θ, the similarity measure s, and the tokenization method T, the evaluation procedure employed in the experiments (Section 4) is as follows:

1. The vocabulary of query tokens V_q is collected using T.

2. For every token t_i from the vocabulary V_q, its embedding $e^q_i = E^q_\theta[t_i]$ is produced.

3. The embeddings for the tokens in every product description $p^j_{1:n}$ are computed as:

$$(e_{j,k})^n_{k=1} = E^p_\theta[T(p^j_{1:n})]$$

4. An index that maps every query term to relevant products is built using query tokens as terms:

$$I(t_i) = \{p^j_{1:n} \mid s(e^q_i, e^p_{j,1:n}) > \gamma\}$$

where γ is a relevancy threshold.

5. For a query $q_{1:m}$ with tokens

$$T(q) = (t'_1, \dots, t'_m)$$

a list of all relevant products according to the index I,

$$R = I(t'_1)| \dots |I(t'_m)$$

is collected, and the metric is computed on R, sorted with respect to the similarity of relevant products to the query.

The described evaluation approach mimics the product search implemented with a simple term index-based hybrid search system. This system combines the efficiency of fast lexical term lookup in an index for high precision, with the computation of similarity scores on only a subset of all product descriptions, ensuring low latency responses. The performance of the system is entirely dependent on the similarity measure s within embedding space defined by semantic model of choice.

The offline metrics for product search differ from those of document information retrieval. The objective of product search is to identify several, or ideally, all products relevant to the query, including identical items. This requirement stems from the customer's need to compare prices for identical products. Hence, the formulas for recall and precision are adapted to include an equivalence relation M. Precision metrics for product search are defined as follows, with recall metrics being similarly formulated.

$$P@k = \frac{1}{|Q|}\sum_{q\in Q}\frac{\left|M(p_q^r@k, p_q^g)\right|}{k} \tag{6}$$

$$mAP@k = \frac{1}{k}\sum_{i=1}^{k} P@i \tag{7}$$

Q – the set of all search queries.

p_q^g – all ground truth products for query q.

p_q^r – the retrieved products for query q at rank k.

$M(A, B)$ – the set of products in A that are equivalent to any of the products in B.

4. Experiments

We evaluated the proposed H1 model against several existing information retrieval models, specifically TCT-ColBERT (Lin et al., 2020), Single Encoder (SE) (Nigam et al., 2019), and Dual Encoder (DE) (Huang et al., 2013). Additionally, we experimented with three tokenization methods: Byte Pair Encoding (BPE), unigram, and word tokenizations. For each tokenization method, we proposed two variations: one enriched with a predefined set of brand names as special tokens (referred to as multi-token or *mt* variations), and a standard version without added brand names (non-multi-token or *non-mt* variations).

We employed the SentencePiece library for all tokenization tasks, configuring it with the *split_by_whitespace=False* option to ensure multi-word brand names could be incorporated as special tokens.

Following the evaluation methodology outlined in Section 3.2, we calculated the metrics mean Average Precision at 12 items ($mAP@12$) and Recall at 1000 items ($R@1k$) for H1, SE, DE models combined with every tokenization method described earlier. Two products are considered to be equivalent if they share the same title.

We compare the performance of the best combination of the model type and tokenization method against ColBERT implemented by Terrier (Macdonald et al., 2021) and trained with Tight Coupling Teachers method (Lin et al., 2020).

4.1. Dataset

Our data source is the publicly available WANDS dataset, chosen for its suitability in objectively benchmarking retrieval systems in the context of e-commerce. The dataset's key characteristics are as follows:

- 42,994 product candidates,
- 480 queries,
- 233,448 relevancy scores for query-product pairings.

The relevancy of query-product pairs in the WANDS dataset is annotated with three levels: fully relevant (Exact), partially relevant (Partial), and Irrelevant. For the purposes of training our models, we utilized only two labels: Exact (labeled as 1) and Irrelevant (labeled as -1), with class balancing implemented prior to training.

4.2. Ablation study

First, we ablate over the tokenization method and model hyperparameter (embedding dimensions) for each of the model types: H1, SE, DE. For ColBERT model, the pretrained version was used, so it was not included in the ablation study. The results of the experiment are shown in Fig 3. The best results, $R@1k = 86.6\%$ and $mAP@12 = 56.1\%$, were achieved by the combination of H1 model with 768 embedding dimensions and BPE tokenization with brand names added.

We note that for both BPE and unigram tokenizations, the variation with brand names added (mt) produces consistently better results for any model with any embedding dimensionality.

4.3. Best models comparison

Model	Threshold	Precision	Recall
ColBERT	12	**41%**	26%
	128	21%	61%
	512	9%	78%
	1024	5%	**84%**

Table 1: The ColBERT results on WANDS dataset.

The Table 1 presents the results of the evaluation of the ColBERT model on the WANDS dataset with varying thresholds. The H1 model demonstrates better results, especially for Precision at 12 items.

To further demonstrate the superiority of the H1 model, we compare H1, SE, DE, and ColBERT models with the best hyperparameters seen in the Ablation Study Section 4.2 on a single query with multiple thresholds k.

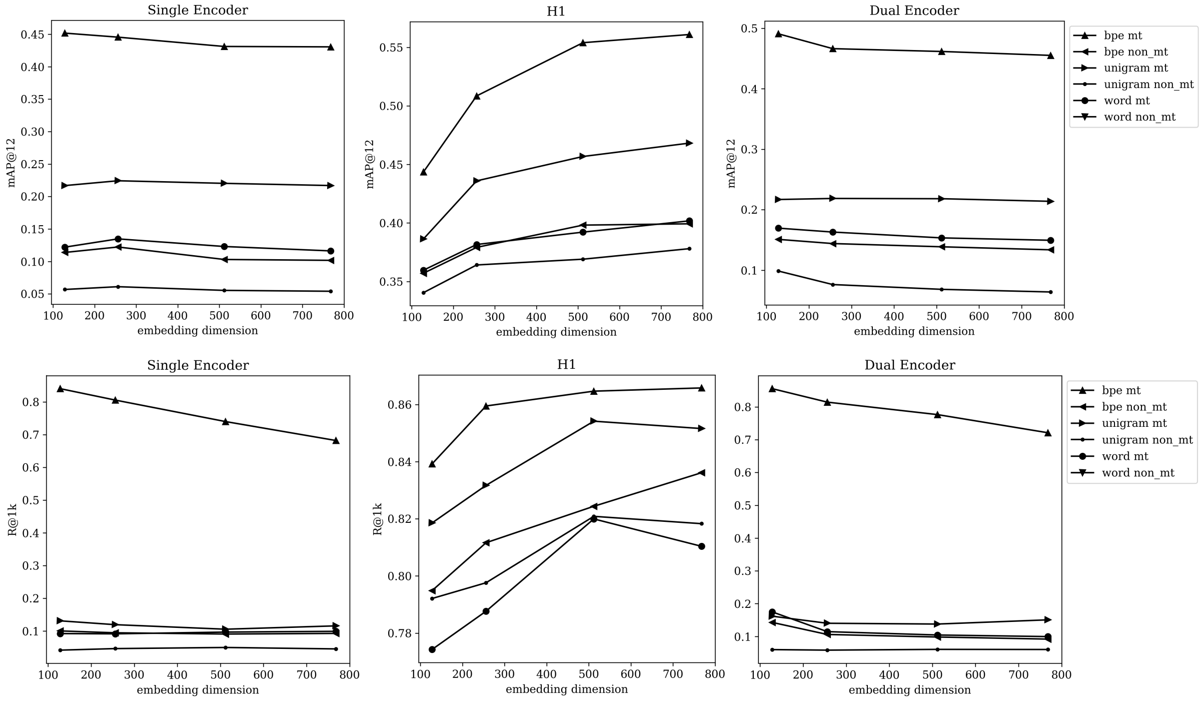

Figure 3: Ablation study results over tokenization methods and model architectures.

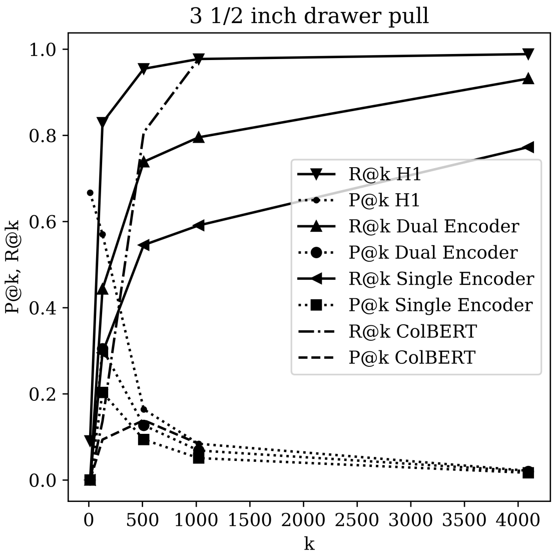

Figure 4: An illustrative one-query example of how Precision decreases and Recall increases for different semantic retrieval models with respect to cut-off threshold k.

The dynamics of Precision and Recall metrics for the H1 model with respect to the threshold k are illustrated in Fig 4, clearly separating the H1 model from the rest. The Recall of the search results is higher with lower values of the threshold k, and Precision declines more slowly as k increases, compared to other models.

5. Conclusions and Future Work

This study introduced the H1 embedding model, a cutting-edge approach designed to refine the landscape of e-commerce search systems by leveraging multi-word term embeddings. Our extensive evaluations demonstrate that H1, through its innovative use of semantically rich tokens and hybrid search methodologies, notably enhances the accuracy and efficiency of product retrieval. By achieving mAP@12 = 56.1% and R@1k = 86.6% on the WANDS dataset, H1 has set a new benchmark, surpassing other state-of-the-art models in terms of precision and recall.

Our research underscores the criticality of integrating semantic understanding with traditional lexical search techniques to address the inherent limitations of each approach. The H1 model's unique ability to treat multi-word terms as singular entities not only improves the search relevance but also aligns with the natural language processing of user queries, thereby significantly enhancing the user experience in e-commerce platforms.

Future efforts will be dedicated to establishing a definitive benchmark for semantic models operating within the framework of hybrid search systems. By exploring a broader range of system architectures, the aim of our future work is to provide a comprehensive and objective evaluation framework that will not only assess the efficacy of current models but also inspire the development of more advanced and effective search solutions.

6. References

Eneko Agirre and Arantxa Otegi. 2010. Document expansion based on wordnet for robust ir. volume 2, pages 9–17.

Yang Bai, Xiaoguang Li, Gang Wang, Chaoliang Zhang, Lifeng Shang, Jun Xu, Zhaowei Wang, Fangshan Wang, and Qun Liu. 2020. Sparterm: Learning term-based sparse representation for fast text retrieval. *CoRR*, abs/2010.00768.

Adam Berger and John Lafferty. 1999. Information retrieval as statistical translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 222–229, New York, NY, USA. Association for Computing Machinery.

Leonid Boytsov, David Novak, Yury Malkov, and Eric Nyberg. 2016. Off the beaten path: Let's replace term-based retrieval with k-nn search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM'16. ACM.

Felipe Bravo-Marquez, Gaston L'Huillier, Sebastián Ríos, and Juan Velasquez. 2013. Hypergeometric language model and zipf-like scoring function for web document similarity retrieval. pages 303–308.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *CoRR*, abs/2002.03932.

Yan Chen, Shujian Liu, Zheng Liu, Weiyi Sun, Linas Baltrunas, and Benjamin Schroeder. 2022. Wands: Dataset for product search relevance assessment. In *Advances in Information Retrieval*, pages 128–141, Cham. Springer International Publishing.

Stéphane Clinchant and Florent Perronnin. 2013. Aggregating continuous word embeddings for information retrieval. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 100–109, Sofia, Bulgaria. Association for Computational Linguistics.

Zhuyun Dai and Jamie Callan. 2019. Context-aware sentence/passage term importance estimation for first stage retrieval.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Miles Efron, Peter Organisciak, and Katrina Fenlon. 2012. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 911–920, New York, NY, USA. Association for Computing Machinery.

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. 1987. The vocabulary problem in human-system communication. *Commun. ACM*, 30(11):964–971.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth J.F. Jones. 2015. Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 795–798, New York, NY, USA. Association for Computing Machinery.

Jianfeng Gao, Jian-Yun Nie, Guangyuan Wu, and Guihong Cao. 2004. Dependence language model for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 170–177, New York, NY, USA. Association for Computing Machinery.

Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. 2021. Complementing lexical retrieval with semantic residual embedding.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018a. End-to-end retrieval in continuous space.

Daniel Gillick, Alessandro Presta, and Gaurav Singh Tomar. 2018b. End-to-end retrieval in continuous space.

Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving efficient neural ranking models with cross-architecture knowledge distillation. *CoRR*, abs/2010.02666.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, page 2333–2338, New York, NY, USA. Association for Computing Machinery.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1681–1691, Beijing, China. Association for Computational Linguistics.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models.

Maryam Karimzadehgan and ChengXiang Zhai. 2010. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 323–330, New York, NY, USA. Association for Computing Machinery.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *CoRR*, abs/2004.04906.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2741–2749. AAAI Press.

Taku Kudo. 2018a. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo. 2018b. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, page 120–127, New York, NY, USA. Association for Computing Machinery.

M. E. Lesk. 1969. Word-word associations in document retrieval systems. *American Documentation*, 20(1):27–38.

Hang Li and Jun Xu. 2014. Semantic matching in search. *Foundations and Trends® in Information Retrieval*, 7(5):343–469.

Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based product retrieval in taobao search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 3181–3189, New York, NY, USA. Association for Computing Machinery.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling dense representations for ranking using tightly-coupled teachers.

Xiaoyong Liu and W. Bruce Croft. 2004. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 186–193, New York, NY, USA. Association for Computing Machinery.

Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. Twinbert: Distilling knowledge to twin-structured compressed bert models for large-scale retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2645–2652, New York, NY, USA. Association for Computing Machinery.

Craig Macdonald, Nicola Tonellotto, Sean MacAvaney, and Iadh Ounis. 2021. Pyterrier: Declarative experimentation in python from bm25 to dense retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 4526–4533, New York, NY, USA. Association for Computing Machinery.

Alessandro Magnani, Feng Liu, Suthee Chaidaroon, Sachin Yadav, Praveen Reddy Suram, Ajit Puthenputhussery, Sijie Chen, Min Xie, Anirudh Kashi, Tony Lee, and Ciya Liao. 2022. Semantic retrieval at walmart. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3495–3503, New York, NY, USA. Association for Computing Machinery.

Donald Metzler and W. Bruce Croft. 2005. A markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 472–479, New York, NY, USA. Association for Computing Machinery.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian (Allen) Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic product search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2876–2885, New York, NY, USA. Association for Computing Machinery.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction.

Zaifeng Pan, Zhen Zheng, Feng Zhang, Ruofan Wu, Hao Liang, Dalin Wang, Xiafei Qiu, Junjie Bai, Wei Lin, and Xiaoyong Du. 2024. Recom: A compiler approach to accelerating recommendation model inference with massive embedding columns. pages 268–286.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Yonggang Qiu and Hans-Peter Frei. 1993. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, page 160–169, New York, NY, USA. Association for Computing Machinery.

Stephen Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. pages 232–241.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Dan Vanderkam, Rob Schonberger, Henry Rowley, and Sanjiv Kumar. 2013. Nearest neighbor search in google correlate. Technical report, Google.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Ivan Vulić and Marie-Francine Moens. 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 363–372, New York, NY, USA. Association for Computing Machinery.

Xing Wei and W. Croft. 2006. Lda-based document models for ad-hoc retrieval. pages 178–185.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Jinxi Xu and W. Bruce Croft. 2017. Quary expansion using local and global document analysis. *SIGIR Forum*, 51(2):168–175.

Jun Xu, Hang Li, and Chaoliang Zhong. 2010. Relevance ranking using kernels. In *Information Retrieval Technology*, pages 1–12, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Multilingual universal sentence encoder for semantic retrieval.

Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 497–506, New York, NY, USA. Association for Computing Machinery.

Chaoliang Zeng, Layong Luo, Qingsong Ning, Yaodong Han, Yuhang Jiang, Ding Tang, Zilong Wang, Kai Chen, and Chuanxiong Guo. 2022. FAERY: An FPGA-accelerated embedding-based retrieval system. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 841–856, Carlsbad, CA. USENIX Association.

Shulin Zeng, Zhenhua Zhu, Jun Liu, Haoyu Zhang, Guohao Dai, Zixuan Zhou, Shuangchen Li, Xuefei Ning, Yuan Xie, Huazhong Yang, and Yu Wang. 2023. Df-gas: a distributed fpga-as-a-service architecture towards billion-scale graph-based approximate nearest neighbor search. In *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '23, page 283–296, New York, NY, USA. Association for Computing Machinery.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Le Zhao and Jamie Callan. 2010. Term necessity prediction. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 259–268, New York, NY, USA. Association for Computing Machinery.