Contrastive Learning for User Sequence Representation in Personalized Product Search

Shitong Dai
Jiongnan Liu
Zhicheng Dou
dst111dst@ruc.edu.cn
liujn@ruc.edu.cn
dou@ruc.edu.cn
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China

Haonan Wang
Lin Liu
Bo Long
wanghaonan@jd.com
liulin1@jd.com
bo.long@jd.com
JD.com, Inc.
Beijing, China

Ji-Rong Wen
jrwen@ruc.edu.cn
Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, China
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China

ABSTRACT

Providing personalization in product search has attracted increasing attention in both industry and research communities. Most existing personalized product search methods model users' individual search interests based on their historical search logs to generate personalized search results. However, the search logs may be sparse or noisy in the real scenario, which is difficult for existing methods to learn accurate and robust user representations. To address this issue, we propose a contrastive learning framework CoPPS that aims to learn high-quality user representations for personalized product search. Specifically, we design three data augmentation and contrastive learning strategies to construct self-supervision signals from the original search behaviours. The contrastive learning tasks utilize an external knowledge graph and exploit the correlations within and between user sequences, thereby facilitating the discovery of more meaningful search patterns and ultimately enhancing the quality of personalized search. Experimental results on the public Amazon datasets verify the effectiveness of our approach.

CCS CONCEPTS

• **Information systems** → **Personalization**.

KEYWORDS

Contrastive Learning, Personalized Product Search

ACM Reference Format:
Shitong Dai, Jiongnan Liu, Zhicheng Dou, Haonan Wang, Lin Liu, Bo Long, and Ji-Rong Wen. 2023. Contrastive Learning for User Sequence Representation in Personalized Product Search. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599287

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599287

1 INTRODUCTION

Product search aims to retrieve a list of products that match the users' search intent based on their submitted queries. Therefore, the quality of product search results directly impacts user satisfaction, as well as e-commerce platforms' transactions and revenue. Previous studies on product search have focused on modeling the correlation between queries and product aspects (e.g., categories and brands) [17]. However, these studies generally conducted non-personalized product searches without taking into account users' search preferences.

Personalized product search systems extensively utilize users' historical search logs, which contain abundant data on search preferences. Several personalized search methods [2, 4, 6, 29] have emerged, intending to extract user preferences from these logs for more precise results. Considering the dynamic nature of user search interests, these methods typically arrange search logs into chronological behavior sequences and employ diverse techniques to discern user search intent from these sequences. For example, Guo et al. [18] applied a hierarchical RNN to the search sequence to model the user's current search preference. Ai et al. [2] utilized query-aware zero attention mechanism to aggregate historical search sequences, thereby extracting user search preferences. Bi et al. [6] used transformer [37] to mine the correlations between products in the search sequences for more fine-grained user search interest.

The main challenge in sequential product search is how to acquire high-quality user representations from their historical search sequences. Based on these user representations, the product search model can accurately return personalized products to users. While existing personalized search methods have shown promising performance, they typically rely on large amounts of user search data to infer user search preferences. However, in the real search scenario, users' search behaviors are often highly sparse and noisy, which could easily weaken user representation learning ability of the personalization model. For example, the search data may involve rather sparse user-item interactions (e.g., short sequences) and the user-item interactions in user sequences may not reflect users' real interest preferences (e.g., impulse consumption). Inference from such sparse and noisy search data would generate inaccurate user

representations and produce less satisfying search results. Therefore, how to learn real user intentions under the limited and noisy historical search sequences is still a great challenge.

Recently, self-supervised learning techniques have been applied to the sequential recommendation to address similar data sparsity problems [7, 39]. They attempt to exploit the intrinsic correlation in the user sequences to construct augmentations for the unlabeled user data. Inspired by these approaches, this paper aims to use self-supervised learning to model user representations from search sequences to improve personalized product search. Previous studies [23, 46] have shown that learning pre-trained sequence representations adapting to personalized search can benefit the personalized ranking and improve search quality. To obtain high-quality user representations, we attempt to apply self-supervised learning to pre-train a more robust sequence encoder, and fine-tune the encoder on the ranking task. To achieve this goal, an intuitive way is to construct data augmentations on the original user search sequences, and to apply corresponding contrastive losses to enhance the encoder's discrimination ability.

To this end, we propose a contrastive learning framework CoPPS that learns high-quality user representations for personalized product search. Specifically, we first pre-train the sequence encoder based on the augmented user search sequences in the pre-training stage, and then fine-tune the encoder with ranking objectives to implement personalized product search. In the pre-training stage, we adopt a contrastive sampling approach to construct self-supervised signals from the original search sequences. The contrastive sampling approach constructs augmented sequence pairs by masking, reordering, replacing, and deleting queries, products, and query-product pairs. By fully exploiting knowledge bases and the correlation within and outside the search sequences, three data augmentation strategies are considered, including (1) a sequence-based strategy that augments sequences based on random perturbations (masking and reordering query-product pairs), (2) a rule-based strategy which uses the intrinsic structure of the behavioral data (i.e., product attributes) to replace the products in the sequences. (3) a graph-based strategy that uses the similarity of corresponding knowledge embeddings to replace and delete products and queries in the sequences. With these augmentation strategies, three corresponding contrastive learning tasks are designed to pre-train the sequence encoder. In the fine-tuning stage, we fine-tune the encoder with an MLP layer to adaptively learn individual search preferences and user representations for personalized product search. We experiment with our proposed model on the Amazon datasets [19, 28]. Experimental results show that our model outperforms existing approaches, indicating that the proposed contrastive learning framework is effective in improving search quality.

To summarize, our main contributions are as follows. (1) We propose a novel method called CoPPS, which adopts a contrastive learning framework to improve the quality of user representations for personalized product search. (2) We design three different data augmentation approaches, including sequence-based, rule-based, and graph-based augmentation, to improve user representations learned from different perspectives. (3) Extensive experiments on four Amazon datasets demonstrate the effectiveness of our model. It brings at least 5% performance improvement over existing models.

2 RELATED WORK

There are three lines of research that are directly related to our work: product search, personalized web search, and contrastive learning for information retrieval.

2.1 Product Search

Product search, as a branch of information retrieval, is to retrieve and return the products to customers with their submitted queries. Previous studies could be divided into two categories: aspect-based models and representation-based models. For the first category, Lim et al. [21] explored the product (e.g., brand, category, context) and retrieved items by simply matching queries with the product aspects. Despite their success, these aspect-based models are not applicable in the real scenario because they limited the queries to the product aspects. In addition, many representation-based models [13, 30] have been proposed, which focused on the representation similarities between queries and items and used semantic information to retrieve relevant products. The typical method LDA [44] retrieved products by matching queries and products with their latent representations. However, these methods are non-personalized, which may fail to capture the internal needs of users. With the increasing complexity of user needs, some researchers have tried to use various information (e.g., users' historical interaction sequences, the structural information of the user-product knowledge graph) to enrich the representations of the queries and the products. For example, many transformer-aware models, such as TEM [6], HEM [4], and ZAM [2], have been proposed to model users' tastes through historical purchase sequences. Moreover, some graph-based models, such as DREM [5] and CAMI [22], extracted structural information of the products and modeled the relationships between products and users. Different from the above models, in this paper, we introduce a pre-training stage with contrastive learning tasks to learn better user representations, which can improve personalized ranking and search quality.

2.2 Personalized Web Search

Personalized web search, which aims to find and return web pages to users based on both their submitted queries and their search histories, is very similar to product search. In personalized web search, researchers often mine the user's search intent from sequences of user behavior, such as browsing or clicking on documents, to achieve more accurate retrieval results. In these personalized search methods, web pages are ranked and returned based on the semantic relevance between the user's search intent and the textual content of the document. For example, Dou et al. [12] proposed a click-based model to enhance personalized ranking results by counting the number of historical clicks on the same query by a single user. Recently, numerous studies have been conducted on personalized web search to improve the quality of search engines [43] [47]. Given the fundamental difference between documents and products, product search techniques can be developed not only based on textual content but also by incorporating various auxiliary information such as product attributes, user-product interaction graphs, and knowledge graphs. Therefore, our approach integrates knowledge graph auxiliary data into sequential product search methods to

enhance user sequences, resulting in more robust user representations.

2.3 Contrastive Learning for IR

Contrastive learning (CL) is a type of self-supervised learning, which has been widely used in many fields, such as computer vision [9, 35, 41], natural language processing [14–16] and information retrieval [47, 48]. The principle of CL is to train a representation learning model to automatically distinguish the constructed similar instances and dissimilar instances, where the similar instances are closer together while the dissimilar instances are further apart in the representation space. Focusing on the field of information retrieval, CL techniques are mainly developed for web search and item recommendation (e.g., news, products, music, etc.). In order to enhance the representations of documents or submitted queries in web search, some CL-based web search methods optimize the representations of queries and documents by augmenting the original data [47]. Besides, many attempts have been made in item recommendation to learn more robust user representations through CL tasks. For example, CoSeRec [24] attempted to perform data augmentation on user sequences (including query and clicked product) to learn the user representation. Specifically, they proposed the method of randomly masking terms in a sequence and reordering the user sequence at some positions. Inspired by these strategies, we propose enhanced contrastive learning tasks which not only involve random operations but also incorporate the external knowledge from graph structures. These informative contrastive learning tasks enable us to generate higher quality sequences and improve the user sequence representation.

3 METHODOLOGY

This paper focuses on modeling user representations from user search sequences for more accurate personalized product search. To achieve this, we propose a CL-based framework called CoPPS, which leverages the self-supervised learning technique to pre-train a more robust sequence encoder and fine-tunes the encoder for ranking tasks. Since our CoPPS is a general framework, we select one of the state-of-the-art sequential models, the BERT encoder [11], as our user backbone representation model. Correspondingly, our CoPPS consists of two stages: (1) in the pre-training stage, we pre-train the BERT encoder with three contrastive learning tasks to learn robust user representations, and (2) in the fine-tuning stage, we fine-tune the encoder to rank products for personalized product search. The architecture of our model CoPPS is shown in Figure 1.

3.1 Preliminaries and Notations

In this section, we first introduce the notations used in this paper and formulate the sequential product search problem. In product search scenarios, the problem we address in this paper is to return a ranked product list to the user such that these products can satisfy the current query (i.e., the user's search intent). The typical approach to achieve this is to infer the user's search preference with respect to the current query from the user's historical search sequence, and to return products by ranking the plausibility of the current user's search preference and the products. Assume that the current query is q, and the historical search sequence of a user u is $S_u = \{q_1, p_1, q_2, p_2 ..., q_m, p_m\}$, where p_i represents the product that the user u purchased after submitting a query q_i at the i-th time with a total times of m. Our proposed model could be formalized as

$$z = F(p|S_u, u, q), \tag{1}$$

where F is our product search model, z determines whether to return the product p to the user u based on the historical search sequence S_u and the current query q.

3.2 Data Augmentation in Pre-training

This section aims to introduce our data augmentation module, which explores augmentation approaches to sequential product search to obtain a powerful user representation encoder. Specifically, we design three data augmentation approaches to construct augmented sequence pairs : (1) a sequence-based strategy that augments sequences by randomly perturbing (masking and reordering) the user sequence, (2) a rule-based strategy that uses the data features (i.e., product attributes) to make replacements on the user sequences, (3) a graph-based strategy that uses the structural information of KG to make replacements and deletions on the user sequences. We will give the details of the three data augmentation strategies in the following part.

3.2.1 Sequence-based Data Augmentation. Regarding data augmentation strategies used in self-supervised learning, the most common type is to randomly generate sequences that resemble the original ones. In this context, there are many existing attempts, and we refer to two main random operators [24, 42, 45] in our work to augment original sequences.

(1) Random Mask of Query or Product (RM) .

Randomly masking a product or a query in a history sequence is a common strategy in information retrieval (IR) [45]. Inspired by this strategy, we propose the random mask of query or product (RM) strategy to randomly mask products or queries in a user sequence to generate a new sequence that is similar to the original one.

Specifically, we simplify the user historical search sequence $S_u = \{q_1, p_1, q_2, p_2, ..., q_m, p_m\}$ as $S_u = \{i_1, i_2, ..., i_N\}, N = 2m$ for convenience, then we randomly mask $l = \lceil \gamma \cdot N \rceil$ terms in the sequence S_u (γ is the hyper-parameter of mask ratio) and get the new sequence as:

$$f^{\mathrm{RM}}(S_u) = \left\{\widehat{i}_1, \cdots, \widehat{i}_N\right\},$$
$$\widehat{i}_j = \begin{cases} i_j, & j \notin T_{S_u}, \\ [\mathrm{MASK}], & j \in T_{S_u}, \end{cases} \tag{2}$$

where T_{S_u} is the indexes of the terms selected to be masked in S_u. In this augmentation strategy RM, the terms (products or queries) masked in the sequence will be replaced by the token "[MASK]".

(2) User Sequence Reorder (RO).

In the product search scenarios, the order of user-product interactions might be flexible as the users may purchase the same products at different positions under some situations. Considering this case as well as inspired by the reordering operation in NLP field [8, 32], we propose an augmentation strategy named user behavior reorder (RO) to build the variants of user sequence. For brevity and convenience, we treat each query q_k and its corresponding product p_k as a behavior sub-sequence u_k and denote the user sequence as $S_u = \{u_1, u_2, \ldots, u_m | u_k = (q_k, p_k), k \in [1, m]\}$. In this RO strategy,

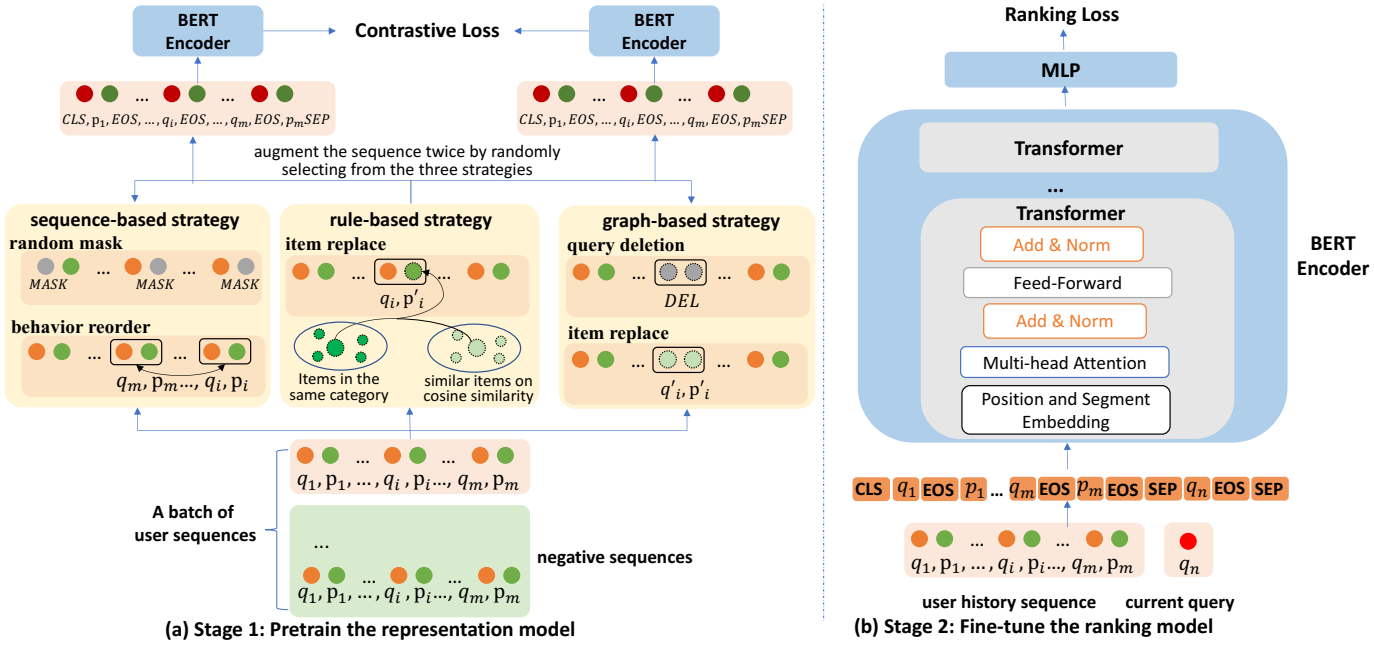


Figure 1: The structure of CoPPS. (1) At the pre-train stage, we generate two sequences for the query-product sequence S_1 under two contrastive learning tasks, and they're regarded as a positive pair. Other contrastive learning tasks are also used in the same batch. (2) The product ranking stage shares the same encoder with the contrastive learning pre-train stage.

we randomly select two behavior sub-sequences and exchange their positions, the operation is conducted $l_r = \lceil \eta \cdot m \rceil$ times, where η is the hyper-parameter of reordering ratio. Assuming the selected i-th pairwise positions as (a_i, b_i), we switch u_{a_i} and u_{b_i}, and the RO strategy is formulated as f^{RO} on S_n :

$$f^{\text{RO}}(S_u) = \{\widehat{u}_1, \cdots, \widehat{u}_m\},$$
$$\widehat{u}_j = \begin{cases} u_j, & j \neq a_i \text{ and } j \neq b_i, \\ u_{b_i}, & j = a_i, \\ u_{a_i}, & j = b_i. \end{cases} \tag{3}$$

3.2.2 Rule-based Data Augmentation. Besides the sequence-level augmentation described in the section 3.2.1, we additionally consider a rule-based strategy to make more fine-grained and convincible augmentation on the user sequence, since random strategies may introduce too much noise into augmented data. In the real business scenario of personalized product search, the attribute information (e.g., category and brand) is valuable to make high-quality augmented sequences. The reason is that products with the same attributes may be similar and replacing items with highly similar items will introduce less corruption to the original sequence, which then yields more confident positive pairs. Specifically, the **rule-based replacement (RE)** strategy is to replace the items with the out-of-sequence items that have the same category. This is because the category is the main feature in the real search scenario, which can work better than other feature-based replacements. Formally, for a user behavior sequence $S_u = \{u_1, u_2, \ldots, u_m\}$, we select randomly a sub-sequence $u_j = (q_j, p_j)$ and replace p_j with the product p'_j belonging to the same category. The replaced sub-sequence is denoted by $u'_j = (q_j, p'_j)$ and the operation is conducted $l_c = \lceil a \cdot m \rceil$ times (a is the hyper-parameter of replacement ratio in RE). Thus, the strategy is formulated as a function f^{RE} on $S(u)$ and defined as:

$$f^{\text{RE}}(S_u) = \{\widehat{u}_1, \cdots, \widehat{u}_m\},$$
$$\widehat{u}_j = \begin{cases} u_j, & j \notin I_{S_u}, \\ u'_j, & j \in I_{S_u}, \end{cases} \tag{4}$$

where I_{S_u} is the index of the sub-sequence selected to be replaced in S_u.

3.2.3 Graph-based Data Augmentation. However, rule-based data augmentation cannot cover all circumstances where items or queries are similar. An optimal way is to use embedding-based methods to find similar queries and products to augment sequences. It has shown that the user-query-product KG can integrate the information of items' attributions, user-item interactions and item-item correlations, which could be leveraged to robust the user encoder [3]. Inspired by this, we calculate the KG-based embeddings of users, products, and queries by the DREM [5] model, and then use these embeddings in our graph-based data augmentation strategy. Specifically, we propose two novel graph-based contrastive learning tasks, which aim at reducing the possible issues caused by augmentations in the former two augmentation strategies. Before we introduce these methods, it's necessary to design an effective pattern for calculating the similarity between items (or queries), as this strategy needs the correlations among queries or items to augment sequences. In the vector space learned by DREM, similar entities tend to have similar attributes. Based on this, we propose a straightforward method to infer the correlations between items (or

between queries). It measures the cosine-similarity of their embeddings from the graph model DREM. Given the representations of items or queries i and j as e_i and e_i, the item correlation score is defined as:

$$\mathrm{Cor}_e(i,j) = \mathrm{Sim}(e_i, e_j), \tag{5}$$

where the function $\mathrm{Sim}(\cdot)$ is implemented by cosine similarity in this work, and it could be replaced by inner product as well. Having introduced the correlation calculation, we will introduce the augmentation strategies as follows:

(1) Query Deletion (QD)

When considering the search scenario in reality, there may be queries related to the current query in the user history sequence. If we delete these historical queries, we will not be able to fully understand the user's current interests. Contrarily, queries that are not related to the current query in the history sequence might be noise. Deleting such irrelevant queries might not affect modeling user's intent or even make the user's current preference clearer. Following this thought, we propose to remove the queries irrelevant to the current queries to conduct data augmentation. In specific, we first sort the query and its corresponding product chronologically and then take the latest query as the current query issued by the user. We then search for the most irrelevant queries and products according to Eq. (5) and delete the unrelated query-item pairs.

Formally, for a user behavior sequence $S_u = \{q_1, p_1, \cdots, q_m, p_m\}$, also denoted by $S_u = \{u_1, u_2, \ldots, u_m\}$, we delete a proportion μ of sub-sequences $R_{S_u} = \{r_1, \cdots, r_L\}$ according to the relevance scores, where $L = \lfloor \mu \cdot m \rfloor$, and r_i is the index of the sub-sequence to be deleted ($\mu \in [0,1]$ is the deletion ratio, and $\lfloor$ denotes rounding down). In implementation, we set the last query q_m as the current query issued by the user, then we select the L queries which are least similar to q_m in the previous $m-1$ queries, then we drop them and the corresponding products from the user sequence. Once a query is deleted, the whole sub-sequence is replaced by a special token "[DEL]". This augmentation strategy is formulated as a function f^{QD} on S_u and defined as:

$$f^{\mathrm{QD}}(S_u) = \{\widehat{u}_1, \cdots, \widehat{u_m}\},$$
$$\widehat{u}_j = \begin{cases} u_j, & j \notin R_{S_\mu}, \\ [\mathrm{DEL}], & j \in R_{S_\mu}. \end{cases} \tag{6}$$

(2) Item Replacement (IR)

In the rule-based augmentation, we make the item replacement according to the similarities based on categories. In this strategy, we change the substitution rule to use the KG embeddings to measure similarity. Formally, for a user behavior sequence $S_u = \{q_1, p_1, q_2, p_2, \ldots, q_m, p_m\}$, also denoted by $S_u = \{u_1, u_1, \ldots, u_m\}$, where $u_j = \{q_j, p_j\}$. Then we randomly select k different indices $I_{S_u} = \{\mathrm{idx}_1, \mathrm{idx}_2, \ldots \mathrm{idx}_k\}$ in the sequence S_u, where $k = \lceil \alpha \cdot m \rceil$ and $\mathrm{idx}_i \in [1, 2, \ldots, m]$ ($\alpha \in [0,1]$ is the substitution ratio).

For each $\mathrm{idx}_j \in I_{S_u}$, assuming the corresponding product of idx_j is p_x, then we select the product p_y from product set which is the most similar to p_x based on Eq. (5) and does not appear in the current sequence. Then we replace $u_x = \{p_x, q_x\}$ with $u'_x = \{p_y, q_y\}$ in the user sequence. This augmentation strategy is formulated as a function f^{IR} on S_u and defined as:

$$f^{\mathrm{IR}}(S_u) = \{\tilde{u}_1, \cdots, \tilde{u}_n\},$$
$$\tilde{u}_j = \begin{cases} u_j, & j \notin I_{S_\mu}, \\ u'_j, & j \in I_{S_\mu}. \end{cases} \tag{7}$$

3.3 User Representation with Contrastive Tasks

3.3.1 User Representation Encoder. We train a BERT encoder based on the augmented sequences to obtain meaningful user representations. BERT has strong capabilities in sequence representation task, which has been widely studied in various fields, such as recommendation [20, 38] and NLP [26]. In our work, in terms of sequence representation, we also adopt BERT and use it to encode the augmented sequences. A user sequence defined in this paper follows the design schema of the vanilla BERT, i.e., we add special tokens "CLS" to the head and "SEP" to the tail. In addition, we also need to add a special token "EOS" at the end of each query and product. Formally, a user sequence S is represented as:

$$S = [\mathrm{CLS}]q_1[\mathrm{EOS}]p_1[\mathrm{EOS}]\cdots q_n[\mathrm{EOS}]p_n[\mathrm{EOS}][\mathrm{SEP}]. \tag{8}$$

And for each token, the token embedding, positional embedding and segment embedding are added together and fed into BERT to obtain the contextual representation of a sequence, we also use the representation of "CLS" token as the sequence representation h:

$$V = \mathrm{BERT}(S)_{[\mathrm{CLS}]}, \text{ and } h = g_1(V), \tag{9}$$

where $V \in \mathbb{R}^{768}$, and $g_1(\cdot)$ is a linear projection function.

3.3.2 Contrastive Tasks. At the pretraining stage, we employ a contrastive learning objective to learn a generalized user sequence representation. The contrastive learning loss is based on the contrastive prediction task. In this work, we augment each sequence twice in a batch to construct the training set $\{\mathcal{S}\}$, i.e., we select two strategies randomly from sequence-based, rule-based and graph-based strategy. Assume there are N sequences in a batch, we could obtain the set $\{\mathcal{S}\}$ with size $2N$. The augmented two sequences form the positive pair and all the other sequences from the same batch would be treated as negative samples. Then following the previous work in NLP[9, 14, 15, 41], we could construct the contrastive learning loss for a positive pair as :

$$L(i,j) = -\log \frac{\exp\left(\mathrm{sim}\left(h_i, h_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbf{1}_{k \neq i} \exp\left(\mathrm{sim}\left(h_i, h_k\right)/\tau\right)}, \tag{10}$$

where h_i, h_j are from one positive pair of sequences, and $\mathbf{1}_{k \neq i}$ is the indicator function to judge whether $k \neq i$, and τ is a hyper-parameter representing temperature. Formally, the contrastive learning loss could be defined as all positive pairs' losses in a batch:

$$\mathcal{L}_{\mathrm{CL}} = \sum_{i=1}^{2N} \sum_{j=1}^{2N} I(i,j) L(i,j), \tag{11}$$

where $I(i,j) = 1$ when $\left(S_i, S_j\right)$ is a positive pair, and $I(i,j) = 0$ otherwise. The contrastive loss is to automatically differentiate between the constructed instances that are similar and dissimilar. In this context, the loss function encourages similar instances to be positioned closer together in the representation space, while ensuring that dissimilar instances are located further apart.

3.4 Ranking Module

As we obtain the optimized BERT encoder under contrastive learning stage, we next employ this BERT encoder to learn the product ranking task. Related researches have applied BERT at ranking stage as a task of sequence pair classification [10, 29, 33, 49]. In this paper, the ranking stage aims at measuring the relationship among the user history behaviour sequence, the new query and the candidate products, denoted by $S_{n-1} = \{q_1, p_1, \cdots, q_{n-1}, p_{n-1}\}$, q_n and $p_{n,i}$ respectively. Then $S_{n-1} \cup q_n$ is considered as one sequence and $p_{n,i}$ as another sequence, so the input sequence S is represented as:

$$S = [\text{CLS}]q_1[\text{EOS}]p_1[\text{EOS}] \cdots q_n[\text{EOS}][\text{SEP}]p_{n,i}[\text{EOS}][\text{SEP}].$$

Similarly, the positional embedding and the segment embedding are added together and then input into BERT. While S contains two sequences, we need to segment them and use 0 and 1 to distinguish them. We also use the output representation of "[CLS]" to represent the sequence, and the ranking score could be formulated as:

$$V = \text{BERT}(S)_{[\text{CLS}]}, \quad k = g_2(V), \tag{12}$$

where $V \in \mathbb{R}^{768}$, and $g_2(\cdot)$ is a linear projection to map the representation into a score.

Learning Objective. Inspired by the previous studies [1, 33], we derive the cross-entropy loss to optimize this model:

$$\mathcal{L}_{\text{rank}} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log k_i + (1 - y_i) \log (1 - k_i), \tag{13}$$

where N is the number of samples in the training set.

4 EXPERIMENT

4.1 Experimental setup

4.1.1 Datasets. We use the public Amazon product dataset[1] as our dataset. This dataset contains product metadata and reviews from the Amazon website covering millions of reviews ranging from May, 1996 to July, 2014. It contains 24 product categories. In this work, we adopted the 5-core dataset provided by McAuley et al. [27] , in which the users and products have at least 5 reviews. We then selected 4 categories including *Cell Phones & Accessories, CDs & Vinyl, Clothing, Shoes & Jewelry, Musical instruments* in our experiment. For a fair comparison, we adopt the same strategy in the baseline methods [2, 6] to generate the train, valid and test data. The basic statistics of the datasets are shown in Table 2.

Query Extraction. In e-commerce, a user usually uses a producer's name, a brand or a series of product attributes to search [34]. Inspired by the query extraction strategies in baselines [4, 36], we extract the query of each user transaction based on the product categories in the transaction. Specifically, we extract the categories of each product from its metadata, then we concatenate the words from a single category to formulate a raw query for this product. Then we remove the stop words, duplicated words and punctuation from the raw query as the extracted query. Importantly, to retain more information about the products, we treat categories and subcategories differently. For instance, in Musical Instruments dataset, for Musical Instruments->Electric Guitar, the extracted query can be "musical instrument electric guitar". Additionally, if there are multiple queries generated for one product, then we randomly selected one query for this product.

Table 2: Data Statistics.

Item	Cell	CDs	Cloth	Music
#reviews	194,439	1,097,591	278,677	10,261
#users	27,879	75,258	56,770	8,216
#items	10,429	64,443	19,913	3,200
#brands	955	1,414	1,245	117
#categories	206	770	428	94
Train				
#queries	134	534	396	42
#(u,q) pairs	114,177	1,287,214	244,595	1,374
Test				
#queries	31	160	92	14
#(u,q) pairs	655	45,490	2,704	203

4.1.2 Baselines. We compared our model CoPPS with several typical product search models.

- LSE [36] is a classical product search model based on latent vector representations. It learns the product representation in a latent vector space from its related reviews.
- HEM [4] is developed with personalization modeling based on LSE [36]. It exploits the representation learning framework that learns the representations of users, products and queries for ranking.
- ZAM [2] considers how much personalization affects the search quality and designs zero attention to user sequence for ranking.
- TEM [6] utilizes the transformer structure to replace zero attention in ZAM [2], which can learn different weights of behaviors in user sequences for ranking.
- DREM [5] and DREM-HGN [3] are two graph-based models that model the relationship between users, queries and products by leveraging the structural information from the knowledge graph.

4.1.3 Evaluation. In this paper, we used three standard evaluation metrics to measure the performance of our model and the baselines: Mean Average Precision(MAP), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG). The three metrics are commonly used in information retrieval research. MAP is the mean of the average precision scores for each query, MRR is the average of the reciprocal ranks of the recalled products, and NDCG@10 measures the accuracy of the ranking list. In the testing phase, we retrieve only 100 items to generate the ranking list for each user-query pair.

4.1.4 Implementation Details . We use PyTorch [31] and Transformers [40] to implement our model, where the code of Transformers is provided by Huggingface[2]. In our model, the maximum number of tokens in the two stages is set to 128, which means that sequences with more than 128 tokens are truncated by dropping query-product pairs from the head. And we use AdamW[25]

[1] http://jmcauley.ucsd.edu/data/amazon/

[2] https://huggingface.co/bert-base-uncased

Table 1: Performance comparisons of different methods. The best score is bold in each column. Symbols $*$ and † denote the statistical significance with two-sided t-test of $p < 0.05$ and $p < 0.01$, respectively, compared with the best baseline

	Cell			*Cloth*			*Music*			*CD*		
Method	MRR	NDCG	MAP	MRR	NDCG	MAP	MRR	NDCG	MAP	MRR	NDCG	MAP
HEM	0.078	0.093	0.078	0.098	0.113	0.098	0.172	0.201	0.172	0.063	0.072	0.063
ZAM	0.083	0.095	0.083	0.015	0.016	0.015	0.039	0.047	0.039	0.084	0.096	0.084
TEM	0.085	0.098	0.085	0.017	0.018	0.017	0.041	0.049	0.041	0.106	0.122	0.106
DREM	0.094	0.110	0.094	0.053	0.062	0.053	0.334	0.390	0.334	0.122	0.140	0.122
DREM-HGN	0.185	0.208	0.185	0.099	0.115	0.099	0.318	0.384	0.318	0.159	0.185	0.159
CoPPS (our)	**0.291***	**0.249**†	**0.291***	**0.122**	**0.143**†	**0.122**	**0.333**†	**0.391***	**0.333**†	**0.176***	**0.190***	**0.176***

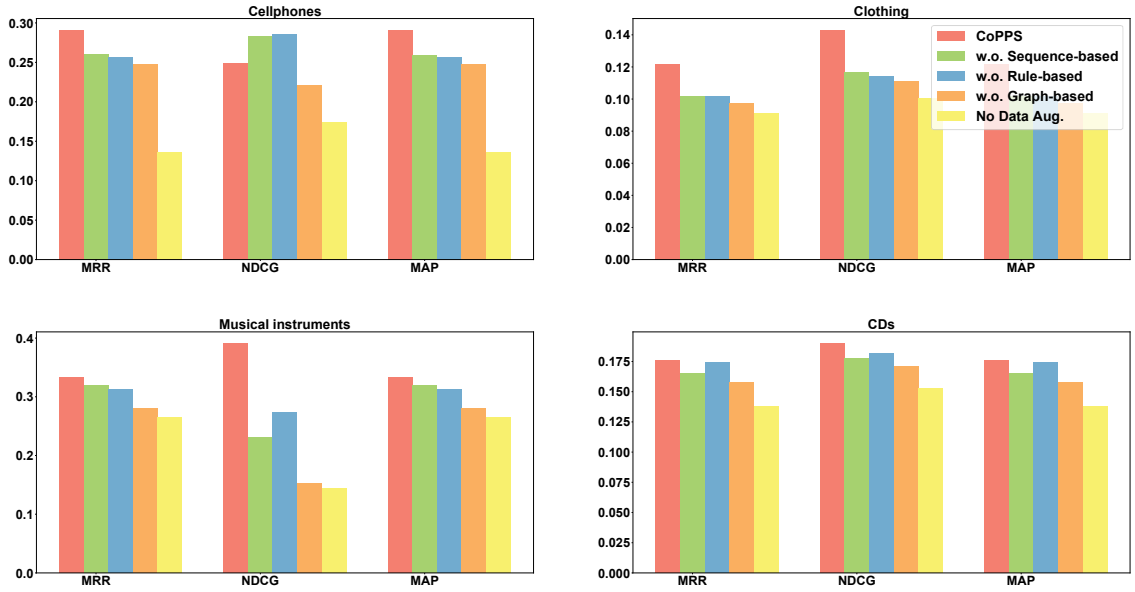


Figure 2: Leave-one-out comparison on four datasets.

optimizer in both two stages. In the pre-training stage, for the sequence-based strategy, we refer to the previous work [24] and tuned on the datasets to find the optimal mask ratio as 0.3 and reordering ratio as 0.1 respectively.

Another thing to be noted is that the reordering strategy will not be applied if the user history sequence contains less than two queries. We also set the batch size to 128, the training epoch to 4, the temperature to 0.1 and the learning rate to 1e-5. We further discuss the two hyper-parameters are in Section 4.4 to explore their effectiveness for our model. In the fine-tuning stage, we use the dropout in the MLP layer with the ratio of 0.1 to train the ranking module. We also set the learning rate to 1e-5 and the training epoch to 4. In both stages, optimal hyperparameters are determined based on the performance of the validation set.

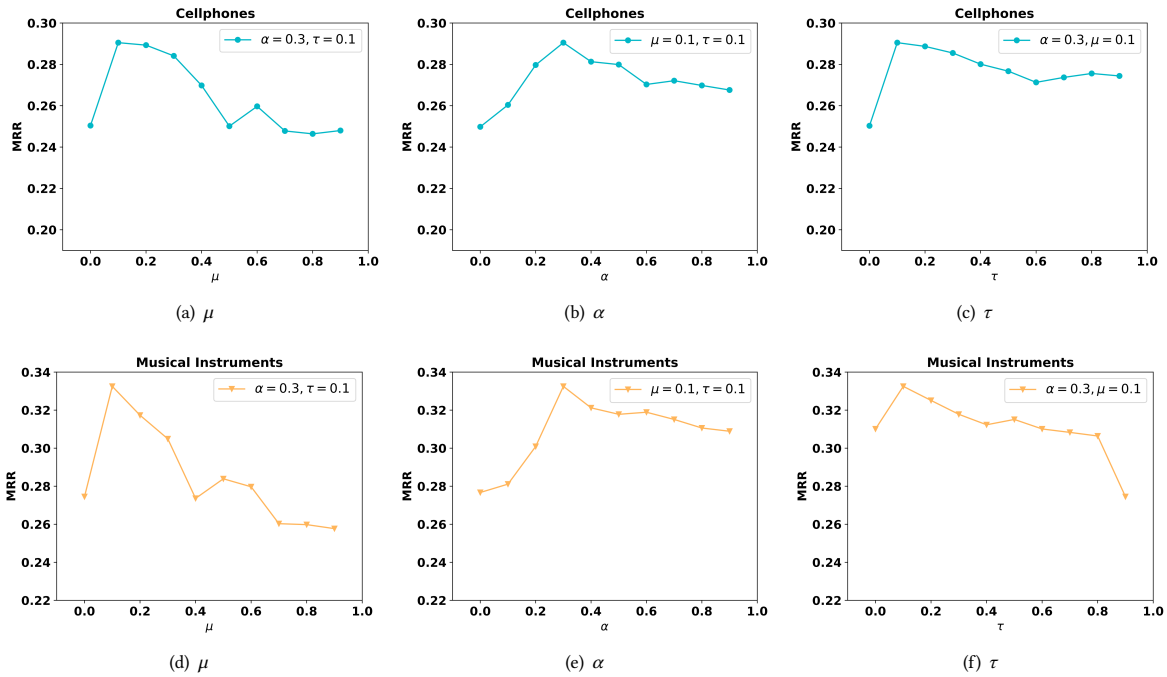
4.2 Experimental Results

Table 1 demonstrates the performance of our model compared with the baselines, we have the following observations from the results:

(1) Across the four datasets, our model CoPPS significantly outperforms all baselines. This result reveals that the contrastive learning tasks can help to obtain a stronger encoder for better detecting users' search intentions in personalized ranking. Our model improves MRR over the best baseline model DREM-HGN by 7.23%, 23.32%, 5%, and 11% on the Cellphones, Clothing, Music instrument, and CDs datasets respectively. In the term of NDCG metrics, there are at least 7% higher performance than baseline models.

(2) Most personalized product search models outperform non-personalized models, indicating the importance of mining personalized interests. Among the personalized model baselines, the graph-based approach achieves better performance than the traditional sequence-based PPS models on most datasets, suggesting that graph knowledge containing rich structural information of users and products, can provide an effective complement to the user's purchase log history.

(3) Compared with the sequential personalized models, our CoPPS model achieves better performance. Although the most competitive sequential model, TEM, also uses a encoder based on transformer



(a) μ (b) α (c) τ

(d) μ (e) α (f) τ

Figure 3: Performance comparison (in MRR) w.r.t. different α, μ and τ on Cellphones and Musical Instruments datasets.

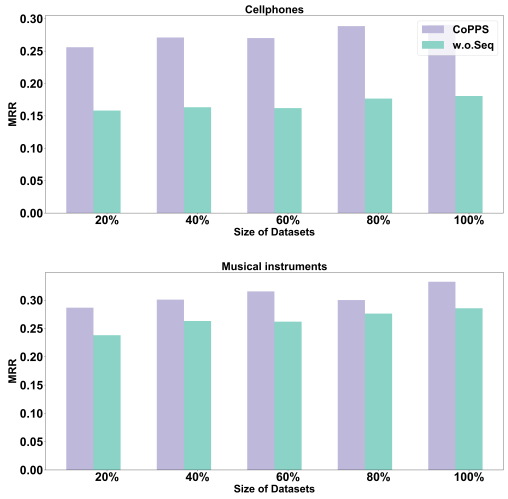


Figure 4: The performance with different sizes of training datasets ("w.o.Seq" denotes the cases without the sequence-based strategies).

architecture to capture user search preferences, our CoPPS still improves MAP and MRR by at least 5% across four datasets. The primary reason for this is the application of contrastive learning tasks in pre-training the sequence encoder within our CoPPS model. This approach enhances the representations of users, ultimately improving personalized ranking and contributing to its superior performance.

(4) Compared with graph-based personalized models, our CoPPS has higher search performance. In contrast to the graph-based approach of directly modeling the knowledge graph (KG), our CoPPS model constructs query-specific dynamic user profiles. This enables us to capture user intentions with greater precision when compared to the static query-independent user profiles used in graph-based personalized models.

4.3 Impacts of Contrastive Learning

To learn more about the impacts of contrastive learning, we test the performance on four datasets with different data augmentation strategies. The result is listed in Figure 2, where (1) "No Data Aug." means we do not pre-train the BERT encoder and use it directly in the ranking module, (2) w/o Sequence-based means we do not use sequence-based methods in the pre-train stage, (3) w/o Rule-based means the rule-based method is not used in the pre-train stage, (4) w/o Graph-based means the graph-based method is not used in the pre-train stage, (5) "all" means that we use all the three types of data augmentation strategies in the pre-training stage.

For the experimental result, we could find that:

(1) Our model CoPPS achieves the highest personalized search performance among all variants with all datasets. It is suggested that each data augmentation strategy is necessary in the pre-training stage to learn better user representations. For example, by utilizing rule-based augmentation, it becomes possible to identify items belonging to the same category. This auxiliary information provides valuable insights into the original sequences.

(2) Compared with the "No Data Aug." variant, our model has higher performance. This suggests that pre-training the sequence encoder is effective in enhancing the search quality. This is because

that the augmented user sequences using the three augmented strategies can provide more information of user preference on making more accurate personalized ranking for the users.

(3) The variant w/o sequence-based has lower search performance than our CoPPS model, indicating that incorporating intra-sequence information through sequence-based data augmentation effectively enhances the robustness of the user encoder. The performance difference between the w/o sequence-based variant and our model CoPPS is relatively smaller in the Musical dataset, possibly due to its smaller size.

(4) The variant w/o rule-based augmentation exhibits inferior performance compared with CoPPS. This suggests that incorporating the item category to perform augmentation is beneficial for modeling user representations. Note that the performance of the variant on the CD dataset is quite close to that of CoPPS. This may be due to the fact that there are more items belonging to the same category, and such substitutions will produce a similar sequence of high quality with a lower probability.

(5) The superior performance of our CoPPS model, compared to the variant w/o graph-based, demonstrates the effectiveness of using knowledge embeddings learned from the knowledge graph. These embeddings facilitate the computation of item similarities, enabling data augmentation and further enhancing the model's capabilities.

4.4 Impacts of Hyper-parameters

According to previous research [9, 15], the temperature hyper-parameter τ is important for contrastive learning. Moreover, the deletion ratio μ and replacement ratio α in graph-based data augmentation also have effects on the model performance. To further study their influence, we've trained our model under different settings. The results on dataset Cellphones and Musical instruments are shown in Figure 3. We find that: (1) Performance initially increases under the ratios, peaking at $\mu = 0.1$ and $\alpha = 0.3$, and then begins to decline. This may be due to the fact that random strategies alone lead to worse performance than when knowledge is incorporated. (2) A higher temperature would cause a higher loss, which is the definition of contrastive loss. After tuning at the ranking stage, we find that a lower loss usually leads to better performance at retrieval. And we set the temperature to 0.1, since it achieves the best results.

4.5 Impacts of the size of Training Datasets

Previous studies found that the amount of data would impact the performance of contrastive learning [9, 15]. Thus, we conduct a study to show the effects of different size of training data and the results are shown in Figure 4.

We initially reduce the amount of training data in both datasets and observe a corresponding decrease in performance during the ranking stage. Additionally, we note that contrastive learning benefits from larger dataset sizes. We then conduct experiments with different epochs and found that contrastive learning typically requires more epochs for optimal results; however, once a certain threshold is reached, further increases do not lead to significant improvements.

In addition, we find that the performance changes of the model with and without the sequence augmentation setting are different as the dataset size increases. For the *Cell Phones & Accessories* dataset, the performance of the model with and without the sequence-based strategy improves slowly with increasing data size, and the improvement is similar. For the *Musical instruments* dataset, the performance of the model changes relatively more with and without the sequence-based strategy. This may be due to the fact that the *Musical instruments* dataset itself is much smaller than the *Cell Phones & Accessories* dataset, and the sequence-based data augmentation approach may have a greater effect on short sequences, so changing the dataset size has a more dramatic effect on the *Musical instruments* dataset.

5 CONCLUSION AND FUTURE WORK

In this paper, we investigate contrastive learning tasks in personalized product search with the goal of acquiring high-quality user representations to enhance search quality. To achieve this objective, we propose a novel learning framework called CoPPS that utilizes self-supervised learning to pre-train a more robust sequence encoder and fine-tune it for ranking tasks.

During the pre-training stage, we designed three data augmentation approaches - sequence-based, rule-based, and graph-based strategies - to generate contrastive pairs with high equality. These augmented sequences were then utilized for pre-training the sequence encoder. During the fine-tuning stage, we further refine the encoder to perform ranking tasks. By incorporating these augmentation techniques into our model design, we are able to extract more meaningful search patterns for user modeling and ultimately achieve effective personalized search. We conducted extensive experiments on four benchmark datasets. The experimental results showed the effectiveness of CoPPS, which achieved remarkable search performance. Additionally, we conducted ablation experiments to prove the effectiveness of three data augmentation approaches

As part of our future research, we aim to explore additional approaches for contrastive learning tasks within the context of personalized product search. Our current work represents our initial attempt at incorporating contrastive learning in this field, and we are committed to further investigating and refining these techniques to improve the performance and effectiveness of our models.

ACKNOWLEDGMENTS

Zhicheng Dou is the corresponding author. This work was supported by National Key R&D Program of China No. 2022ZD0120103, National Natural Science Foundation of China No. 62272467 and No. 61832017, Beijing Outstanding Young Scientist Program No. BJJWZYJH012019100020098, Public Computing Cloud, Renmin University of China, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China. The work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods, and Key Laboratory of Data Engineering and Knowledge Engineering, MOE.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd International ACM SIGIR Conf. on Research and Development in Information Retrieval*. 385–394.

[2] Qingyao Ai, Daniel N Hill, SVN Vishwanathan, and W Bruce Croft. 2019. A zero attention model for personalized product search. In *CIKM*. 379–388.

[3] Qingyao Ai and Lakshmi Narayanan Ramasamy. 2021. Model-agnostic vs. Model-intrinsic Interpretability for Explainable Product Search. In *CIKM '21: The 30th ACM International Conf. on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 5–15.

[4] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conf. on Research and Development in Information Retrieval*. 645–654.

[5] Qingyao Ai, Yongfeng Zhang, Keping Bi, and W Bruce Croft. 2019. Explainable product search with a dynamic relation embedding model. *ACM Transactions on Information Systems (TOIS)* 1 (2019), 1–29.

[6] Keping Bi, Qingyao Ai, and W Bruce Croft. 2020. A transformer-based embedding model for personalized product search. In *Proceedings of the 43rd International ACM SIGIR Conf. on Research and Development in Information Retrieval*. 1521–1524.

[7] Philippe Boileau, Nima S Hejazi, and Sandrine Dudoit. 2020. Exploring high-dimensional biological data with sparse contrastive principal component analysis. *Bioinformatics* 11 (2020), 3422–3430.

[8] John S Breese, David Heckerman, and Carl Kadie. 2013. Empirical analysis of predictive algorithms for collaborative filtering. *arXiv preprint arXiv:1301.7363* (2013).

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International Conf. on machine learning*. PMLR, 1597–1607.

[10] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *SIGIR*. 985–988.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. 2007. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international Conf. on WWW*. 581–590.

[13] Huizhong Duan, ChengXiang Zhai, Jinxing Cheng, and Abhishek Gattani. 2013. Supporting keyword search in product database: a probabilistic approach. *Proceedings of the VLDB Endowment* 14 (2013), 1786–1797.

[14] Hongchao Fang, Sicheng Wang, Meng Zhou, Jiayuan Ding, and Pengtao Xie. 2020. Cert: Contrastive self-supervised learning for language understanding. *arXiv preprint arXiv:2005.12766* (2020).

[15] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[16] Beliz Gunel, Jingfei Du, Alexis Conneau, and Ves Stoyanov. 2020. Supervised contrastive learning for pre-trained language model fine-tuning. *arXiv preprint arXiv:2011.01403* (2020).

[17] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *CIKM*. 55–64.

[18] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2018. Attentive Long Short-Term Preference Modeling for Personalized Product Search. arXiv:1811.10155 [cs.IR]

[19] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th international Conf. on WWW*. 507–517.

[20] Chanwoo Jeong, Sion Jang, Eunjeong Park, and Sungchul Choi. 2020. A context-aware citation recommendation model with BERT and graph convolutional networks. *Scientometrics* (2020), 1907–1922.

[21] Soon Chong Johnson Lim, Ying Liu, and Wing Bun Lee. 2010. Multi-facet product information search and retrieval using semantically annotated product family ontology. *Information Processing & Management* 4 (2010), 479–493.

[22] Jiongnan Liu, Zhicheng Dou, Qiannan Zhu, and Ji-Rong Wen. 2022. A Category-aware Multi-interest Model for Personalized Product Search. In *Proceedings of the ACM Web Conf. 2022*. 360–368.

[23] Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model for web-scale retrieval in baidu search. In *Proceedings of the 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining*. 3365–3375.

[24] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[25] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[26] Mihai Masala, Stefan Ruseti, and Mihai Dascalu. 2020. Robert–a romanian bert model. In *Proceedings of the 28th International Conf. on Computational Linguistics*. 6626–6637.

[27] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *SIGKDD*. 785–794.

[28] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR Conf. on research and development in information retrieval*. 43–52.

[29] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[30] Petteri Nurmi, Eemil Lagerspetz, Wray Buntine, Patrik Floréen, and Joonas Kukkonen. 2008. Product retrieval for grocery stores. In *Proceedings of the 31st annual international ACM SIGIR Conf. on Research and development in information retrieval*. 781–782.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* (2019).

[32] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conf. on Knowledge Discovery & Data Mining*. 1150–1160.

[33] Chen Qu, Chenyan Xiong, Yizhe Zhang, Corby Rosset, W Bruce Croft, and Paul Bennett. 2020. Contextual re-ranking with behavior aware transformers. In *Proceedings of the 43rd International ACM SIGIR Conf. on Research and Development in Information Retrieval*. 1589–1592.

[34] Jennifer Rowley. 2000. Product search in e-shopping: a review and research propositions. *Journal of consumer marketing* (2000).

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *European Conf. on computer vision*. Springer, 776–794.

[36] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM international on Conf. on information and knowledge management*. 165–174.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* (2017).

[38] Tian Wang and Yuyangzi Fu. 2020. Item-based collaborative filtering with BERT. In *Proceedings of The 3rd Workshop on e-Commerce and NLP*. 54–58.

[39] Zixin Wen and Yuanzhi Li. 2021. Toward understanding the feature learning process of self-supervised contrastive learning. In *International Conf. on Machine Learning*. PMLR, 11112–11122.

[40] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).

[41] Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466* (2020).

[42] Xu Xie, Fei Sun, Zhaoyang Liu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive pre-training for sequential recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[43] Jing Yao, Zhicheng Dou, and Ji-Rong Wen. 2021. Clarifying ambiguous keywords with personal word embeddings for personalized search. *ACM Transactions on Information Systems (TOIS)* 3 (2021), 1–29.

[44] Jun Yu, Sunil Mohan, Duangmanee Putthividhya, and Weng-Keen Wong. 2014. Latent dirichlet allocation based diversified retrieval for e-commerce search. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 463–472.

[45] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*. 1893–1902.

[46] Yujia Zhou, Zhicheng Dou, and Ji-Rong Wen. 2020. Encoding history with context-aware representation learning for personalized search. In *Proceedings of the 43rd international ACM SIGIR Conf. on research and development in information retrieval*. 1111–1120.

[47] Yujia Zhou, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2021. PSSL: Self-supervised Learning for Personalized Search with Contrastive Sampling. In *CIKM*. 2749–2758.

[48] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking. In *CIKM '21: The 30th ACM International Conf. on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 2780–2791. https://doi.org/10.1145/3459637.3482243

[49] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive learning of user behavior sequence for context-aware document ranking. In *Proceedings of the 30th ACM International Conf. on Information & Knowledge Management*. 2780–2791.