Information Fusion 81 (2022) 240–249

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

A multi-representation re-ranking model for Personalized Product Search

Elias Bassani [a,b,*], Gabriella Pasi [b]

[a] Consorzio per il Trasferimento Tecnologico - C2T, Milan, Italy
[b] University of Milano-Bicocca, Milan, Italy

ARTICLE INFO

Keywords:
Product Search
Personalization
Results re-ranking

ABSTRACT

In recent years, a multitude of e-commerce websites arose. Product Search is a fundamental part of these websites, which is often managed as a traditional retrieval task. However, Product Search has the ultimate goal of satisfying specific and personal user needs, leading users to find and purchase what they are looking for, based on their preferences. To maximize users' satisfaction, Product Search should be treated as a personalized task. In this paper, we propose and evaluate a simple yet effective personalized results re-ranking approach based on the fusion of the relevance score computed by a well-known ranking model, namely BM25, with the scores deriving from multiple user/item representations. Our main contributions are: (1) we propose a score fusion-based approach for personalized re-ranking that leverages multiple user/item representations, (2) our approach accounts for both content-based features and collaborative information (i.e. features extracted from the user–item interactions graph), (3) the proposed approach is fast and scalable, can be easily added on top of any search engine and it can be extended to include additional features. The performed comparative evaluations show that our model can significantly increase the retrieval effectiveness of the underlying retrieval model and, in the great majority of cases, outperforms modern Neural Network-based personalized retrieval models for Product Search.

1. Introduction

The last 25 years have witnessed the birth of major e-commerce websites as well as a multitude of smaller ones. Online shopping is a popular activity nowadays, and it is expected to become even more popular in the next years, reaching over 2 billion people [1]. In 2020, retail e-commerce sales worldwide amounted to 4.28 trillion US dollars [2] and accounted for 18% of all retail sales [3].

Usually, users decide which items to buy after they have searched the available products through a search engine. In the context of Product Search, users' needs are highly personal, and a search engine should tailor the result lists on the user preferences, as users' diversity largely affects the notion of relevance of the retrieved products. Therefore, personalization is inherently an integral part of Product Search. Generally, e-commerce websites allow users to express their opinions and considerations on the products they have purchased. This feedback takes the form of ratings and reviews. Customers' reviews provide valuable information for modeling both users and items, as they contain clues about user preferences and product properties, which are often not specified in their descriptions. While user-generated content allows capturing the specificity and diversity of users, the analysis of users' purchasing behavior can provide complementary information to enrich both user and item representations. This information allows capturing the similarities among users as well as items' popularity.

To address personalization in Product Search many Neural Network-based retrieval models have been recently proposed. They make use of auxiliary side information to infer item properties and users' interest towards them. Recent efforts mainly focused on leveraging user reviews [4–11], brands and categories [10], and product images [5] to personalize the user search experience. Guo et al. [11] also studied the effect of the long and short-term preferences in Product Search, while Zamani et al. [9] proposed to jointly model personalization in Product Search and Product Recommendation tasks. Bi et al. [8] and Zhang et al. [6] approached the problem of personalization in Product Search in a conversational context. Inspired by [12], these approaches share the assumption that Product Search is an inherently semantic task, due to the severe vocabulary mismatch [13] between user queries and item descriptions. Because of that, the authors model the internal query matching procedure on the semantic similarity between queries and item information, by mapping them in the same latent space.

Differently from recent works where personalization is directly injected into the retrieval model, we tackle personalization in Product Search as a results re-ranking task, where the list of items retrieved by a search engine is re-ranked based on the computation of a new relevance

* Corresponding author at: University of Milano-Bicocca, Milan, Italy.
E-mail addresses: e.bassani3@campus.unimib.it (E. Bassani), gabriella.pasi@unimib.it (G. Pasi).

https://doi.org/10.1016/j.inffus.2021.11.010
Received 31 May 2021; Received in revised form 24 September 2021; Accepted 9 November 2021
Available online 1 December 2021

score obtained by fusing the relevance score assessed by the search engine with several user–item compatibility scores. More specifically, we propose a simple yet effective personalized results re-ranking model based on the fusion of the relevance score computed by the well-known ranking model BM25 [14] with a popularity-based value of the items (an important relevance signal that appears to have been overlooked in previous works) and three compatibility scores computed between latent representations of users and items built upon both content-based and collaborative information (i.e., reviews, categorical information, purchasing behaviors).

Despite the preeminent adoption of semantic matching-based models by recent works in the literature, we opt for a classic lexical matching retrieval model. This choice is driven by the assumption that in Product Search, user queries usually contain "a producer's name, a brand or a set of terms that describe the category of the product" [15] and this kind of information is usually present as-is in product-related information.

Finally, our approach is fast and scalable, it can be added on the top of any search engine, and it is easily extendable to accommodate additional relevance/compatibility scores.

To verify the effectiveness of the proposed approach we have performed several experiments. In particular, we have comparatively evaluated its effectiveness with respect to recently proposed Neural Network-based approaches specifically designed for Product Search [4,10,12], on a variety of datasets from Amazon,[1] which have been previously employed in the literature. Our model consistently increases the retrieval effectiveness of the underlying retrieval model, BM25, and, in the great majority of cases, considerably outperforms modern Neural Network-based baselines.

The main contributions of this work are threefold:

- we propose a score fusion-based approach for personalized re-ranking in Product Search that leverages multiple user/item representations;
- the proposed model makes use of both content-based information (i.e. reviews, categorical information) and collaborative information (i.e. representations extracted from the user–item interaction graph);
- the proposed approach is fast and scalable, it can be added on top of any search engine and it is easily extendable to include additional user/item representations.

The article is structured as follows: after reviewing the related works in Section 2, we present the proposed Personalized Re-Ranking model in Section 3. In Section 4, we introduce the experimental setup of the performed evaluation, and in Section 5, we present and discuss the evaluation results, conducting an in-depth performance analysis.

2. Related works

With the widespread of online shopping in the past few years, the task of Product Search has received increasing attention from the research community. Early works in this area focused on modeling the interaction between users and products-related information stored in relational databases through faceted search [16–19]. Later, to reduce the gap between Web search, the kind of search to which users are most familiar, and search on products' structured data, which usually requires the formulation of structured queries, some works [20–22] investigated the application of language modeling [23] approaches to Product Search. In the meantime, other studies tackled the problem of results diversity [24,25], paving the way for the more recent efforts on personalization in this area. More recently, learning-to-rank strategies were also studied in Product Search [26–28].

The great majority of the works recently published in the context of Product Search rely on the application of Neural Network-based models to tackle both problems of vocabulary mismatch between queries and item descriptions, and personalization. Gysel et al. [12] introduced a latent vector space model for Product Search to address the problem of vocabulary mismatch. The proposed model maps queries and items in the same latent space where their semantic similarity can be directly computed. Ai et al. [4] enhanced the approach proposed in [12] by adding personalization. This effect was obtained by mapping users in the same latent space of queries and items. Ai et al. [10] refined their first model by adding side information, such as item brands and item categories, to user and item representations. Guo et al. [11] modeled user preferences from both long-term and short-term perspectives. Guo et al. [5] studied the effect of the visual modality on user preferences by leveraging item images in the personalization process. Their approach resulted particularly effective on fashion-related domains.

Other directions in the context of Product Search have also been explored. For example a recent study [7] addressed the problem of when and how to rely on personalization to enhance product retrieval. Zamani et al. [9] investigated the possibility of jointly modeling and optimizing product retrieval and recommendation tasks, due to their complementary nature. They proposed a general framework to simultaneously learn a retrieval model and a recommendation model by optimizing a joint loss function. Bi et al. [8] and Zhang et al. [6] tackled Product Search from a conversational perspective. Lin et al. [29] proposed an unsupervised method relying on implicit user's feedback from clicks to collect a large amount of query classification data and highlighted some shortcomings of neural approaches in learning useful representations for queries, due to the fact that queries are composed of a few words. Sondhi et al. [30] recently focused on understanding user search behavior in E-Commerce search applications and how it relates to user query generation, by proposing a query taxonomy for Product Search. Other studies focused on query intent for query refinement [31] and term weighting [32], and perceived satisfaction [33] in Product Search.

In this paper, we focus on the task of personalization in Product Search. In particular, instead of injecting personalization in the retrieval process, we propose a novel model for re-ranking the results produced by an underlying search engine; our model aims at fusing a number of different personalization scores, which are obtained by considering both content-based and collaborative features, related to both users and items. As the underlying search engine, we employ a classic probabilistic retrieval model, i.e. BM25, and we perform comparative evaluations finalized at the assessment of the effectiveness of our approach with respect to state-of-the-art approaches in personalized Product Search. We rely on the re-ranking process to both personalize the ranking of the initially retrieved relevant items and smooth the lexical matching score of BM25 with scores computed from latent representations of users and items. Finally, our approach is much less resource-intensive both in training and online with respect to the recent Neural Network-based approaches.

3. The proposed re-ranking approach

In this section we introduce an extendable approach for personalized re-ranking of search results in Product Search. As previously outlined, our approach relies on a novel multi-faceted personalized re-ranking model applied to the ranked list of items (products) computed by a traditional search engine in response to a user query. The proposed re-ranking algorithm makes use of various information (relevance signals) related to users/items, which may indicate the possible relevance of the products to the user's preferences.

The considered relevance signals are carried by various kinds of information shared by users and items (e.g., reviews and user-system interactions) and by specific properties of the items, such as items' popularity. Each of the above kinds of information is formally represented,

[1] https://www.amazon.com.

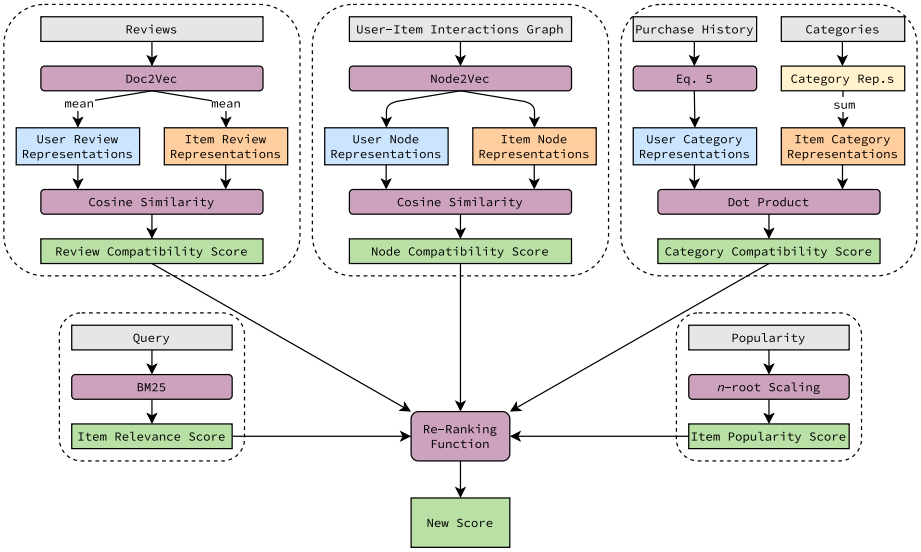

Fig. 1. Overview of the re-ranking model.

and a score associated with a given relevance aspect is computed either as a compatibility score between the related representation of users and items or as a score assessing the items' properties.

The various relevance signals concur to an informed definition of the re-ranking process to make it effective, as they provide complementary information regarding the compatibility between users and items, which we express through the computation of distinct relevance scores, as previously commented. The re-ranking function employed by the proposed model is based on the fusion of these multiple scores, which provide evidence of the possible relevance of an item to a user from different perspectives. The approach we propose can be easily extended to accommodate additional representations to enrich the model.

In the system implemented and evaluated in this paper (Fig. 1), we consider three relevance signals that are supported by three distinct formal representations of users and items: (1) a *review-based representation* built by employing the neural text embedding model PV-DBOW [34] on the product reviews written by the users, (2) an *interaction-based representation* built by leveraging the node embedding model Node2Vec [35] on the user–item interactions graph, which is built upon user–item purchasing relations, and (3) a *category-based representation* that captures the user's categorical interests towards the items' categories. For the items only, we also use a *popularity-based score*, which provides a valuable relevance indicator.

In the following, we first introduce the underlying retrieval function (the base search engine) employed by our model, BM25 [14]. Then, we accurately describe the proposed user and item representations and the related compatibility scores. Finally, we outline the re-ranking function used to compute the personalized ranking scores for the items retrieved by BM25 in response to a user query.

3.1. BM25

BM25 is a bag-of-words-based probabilistic relevance model [36] proposed by Robertson et al. [14]. It assumes binary relevance of documents w.r.t. the user information needs and its scoring function assesses the probability of a document to be relevant w.r.t. a given query. It also assumes statistical independence between term occurrences so as to provide a simple and tractable scoring function. Given a query q and a document d, BM25 computes a relevance score for d w.r.t. q as:

$$r_{q,d} = \sum_{i=1}^{|q|} IDF(q_i, C) \cdot \frac{tf(d_i, d) \cdot (k_1 + 1)}{tf(d_i, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{avg(|d|)})} \quad (1)$$

where q_i is the ith query term, $IDF(q_i, C)$ is the inverse document frequency of q_i in the corpus C, $tf(q_i, d)$ is the term frequency of q_i in d, $|d|$ is the length of d, and $avg(|d|)$ is — with a slight abuse of notation — the average length of documents in the corpus. The coefficients k_1 and b are the hyper-parameters of the model. The former is the term frequency saturation coefficient — it regulates the contribution of each term so that it cannot exceed a saturation point. The latter controls the normalization effect of document length. k_1 and b need to be tuned according to both the queries and the corpus.

3.2. Review-based representations

A first information that may carry a relevance signal is constituted by the textual reviews that the users write to express their opinions towards the products they purchased. In fact, reviews are short textual descriptions that contain information regarding user preferences and product characteristics. We first collect the reviews written by each user and associated with each item. Then, we define a vector representation for each of the reviews by employing the PV-DBOW model [34], which we will describe later in this section, and compute the *review-based representations* for both users and items as the arithmetic mean of the vector representations of the related reviews, i.e., the reviews written by the users for representing the users and the reviews associated with the items for representing the items. Finally, when a user queries the system, we compute a compatibility score for every item in the top-k results retrieved by BM25 as the *cosine similarity* between the user's and the items' *review-based representations*. The re-ranking function uses those scores to compute the personalized relevance scores used for re-ranking, as described later.

We now describe the neural text embedding model employed to create the review representations. PV-DBOW [34] is a text embedding model that maps documents — in our case reviews — into a low-dimensional latent vector space where semantically similar documents are close to each other. Following the *distributional hypothesis* [37–39] and similarly to the Word2Vec Skip-Gram model [34], where the latent representation of each word is learned by predicting nearby words, the PV-DBOW model is trained to predict for each document the words it contains. PV-DBOW operates under the bag-of-words assumption, i.e., it assumes independence between words, and models the generative probability of a word w in document d (a *review* in our case) through the *softmax* function over the vocabulary as follows:

$$P(w|d) = \frac{exp(\vec{w} \cdot \vec{d})}{\sum_{w' \in V} exp(\vec{w'} \cdot \vec{d})} \quad (2)$$

where $\vec{w}$ and $\vec{d}$ are the vector representations of w and d, and V is the vocabulary of the training corpus. The softmax function outputs a probability distribution over the input. As the softmax function becomes prohibitively heavy to compute for large vocabularies, *Negative Sampling* [34] is employed to approximate its computations. *Negative Sampling* is a procedure that samples a certain number of words from the vocabulary, which are then used in the softmax instead of the whole corpus vocabulary, drastically increasing the training speed and avoiding unnecessary numerical computations.

3.3. Interaction-based representations

The analysis of the user behavior can unveil meaningful information regarding the user's interests. In our case, the user behavior coincides with the actions performed by him/her on an e-commerce platform. Typically, those actions correspond to user–item interactions, such as visiting the page of an item (*view*), assigning a numerical evaluation to an item (*rate*), expressing an opinion towards an item through text (*review*), purchasing an item (*buy*), and so on. In our work, to define an interaction-based representation of both users and items, we consider only the interactions corresponding to purchasing actions.

First of all, we build the user–item interaction graph (Fig. 2a). The nodes of this graph represent users and items while the edges represent the user–item interactions. Specifically, if a user purchased an item an edge is drawn between the node representing the user and the node representing the item. Secondly, we create vector representations of the user and item nodes by employing the Node2Vec model [35], which we will describe later in this section. Finally, similar to the review-based representations, when a user queries the system, we compute a compatibility score for every item in the top-k results retrieved by BM25 as the *cosine similarity* between the user's and the items' *interaction-based representations*. Again, the re-ranking function employs those scores to compute the personalized relevance scores used for re-ranking.

We now review the node embedding model employed to create the node representations. Node2Vec [35] is a neural model that takes a graph as input and maps each node of the graph into a low-dimensional vector space where the nodes that share similar neighbors are close to each other.

Firstly, Node2Vec generates training samples by employing a sampling strategy based on a random walk procedure. While the model explores the graph following the random walk procedure, it stores the sequences of nodes visited during each random walk (Fig. 2b). These sequences are *ordered lists* of nodes' unique identifiers (e.g., $walk = [n_1, n_5, n_2]$). Secondly, the model converts the node sequences to strings (e.g., $[n_1, n_5, n_2] \rightarrow \}\}n_1\ n_5\ n_2\varepsilon$) so that they can be treated as sequences of words, i.e., *sentences*. Finally, Node2Vec feeds the string version of the node sequences to the Word2Vec Skip-Gram model [40] that, by treating the node sequences as word sentences, learns vector representations of the nodes (Fig. 2c). As previously described, Word2Vec learns representations of words by predicting nearby words, i.e., the words that co-occur in the same sentences. As Node2Vec feeds Word2Vec with sequences of nodes, the model embeds the neighborhood information carried by the node sequences in the same fashion as learning representations of words by predicting their nearby words. By construction of the user–item interactions graph, the neighborhood of a user node comprises (1) the items he/she purchased, (2) the users who bought the same items purchased by the user, and (3) the other items they bought. Therefore, Node2Vec will map users with similar purchasing behaviors and the items they purchased in the same region of the latent space. Consequently, the similarity between the vector representations of user and item nodes is suitable to evaluate the likelihood of a user purchasing an item. As the vector representations embed behavioral neighborhood information, this approach is foundationally similar to the collaborative filtering approaches used in Recommender Systems.

3.4. Category-based representations

E-commerce websites organize items into product categories (e.g., smartphones, sports clothing, home products, etc.), which are often structured into hierarchies. This information can be leveraged to evaluate the user's interest in different product categories. In particular, by combining the customer's purchase history with the product categories of the items he/she purchased, we can infer his/her categorical interest towards unseen products. To do so, we rely on the item category tree structure found in many e-commerce platforms and the users' purchase history to define a *category-based representation* for both users and items, as described later in this section. Again, when a user queries the system, we compute a category interest-related score of the user towards every item in the top-k results retrieved by BM25 as the *dot product* between their *category-based representations*. The re-ranking function uses those scores to compute the personalized relevance scores used for re-ranking, as described later in this section.

We now describe the method proposed for modeling *the category-based representations* of both users and items. First of all, we leverage the hierarchical structure of the categories to assign to each of them a weight equal to the inverse of its position in the hierarchy it belongs to, so that *root* categories weights more than *intermediate* and *leaf* categories (Fig. 3a). The rationale behind this process is that generic categories better capture the long-term user interests (e.g., a user that recently bought a new laptop will probably be more interested in acquiring computer accessories than another laptop). In case a category tree structure is missing, we can assign the same weight to each category. User and item *category-based representations* are then computed as follows. Firstly, categories are represented as *weighted* one-hot vectors, with the non-zero entries set to their associated weights (Fig. 3a). Then, items are represented as the sum of the representations of the categories they belong to (Fig. 3b). To map users in the category space, we first initialize their representations to a vector with all components set to 1:

$$user\_init_u = \vec{1} \tag{3}$$

In this way, we set a *minimum interest level* for all the categories regardless of the user purchase history. This initialization allows avoiding penalizing item categories for which the user interest is not known when we compute the compatibility scores (*dot product*) between users' and items' *category-based representations*. Then, we compute a vector representation of the user purchase history from a *category-based perspective* through a diminishing return formula based on exponential decay as follows:

$$purchase\_history_u = \vec{1} - exp(-\lambda \vec{p_u}) \tag{4}$$

where $\vec{1}$ is a vector with all components set to 1, $exp(\cdot)$ is the element-wise exponential function, λ is the decay constant and $\vec{p_u}$ is a vector representing the actual purchases of the user u. The components of vector $\vec{p_u}$ correspond to the item categories and its entries are equal to the number of items purchased by u in the corresponding categories. Eq. (4) has a smoothing effect that balances the representation of the user's purchase history and avoids excessively skewing it towards already purchased items' categories. Finally, we sum the user's initialization vector with the vector representation of her/his purchase history:

$$user_u = user\_init_u + purchase\_history_u \tag{5}$$

If a user did not purchase items from a specific category yet, then the value of the corresponding component of the user representation will be equal to 1 as the purchase history vector's entry for that category will be zero. This mechanism allows not penalizing the products belonging to the categories for which we do not know the user's interest while increasing the importance of the other items according to the user's interest towards the related categories.

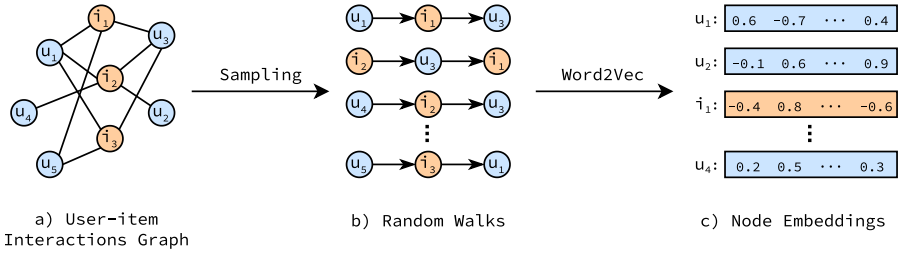

Fig. 2. Node2Vec.

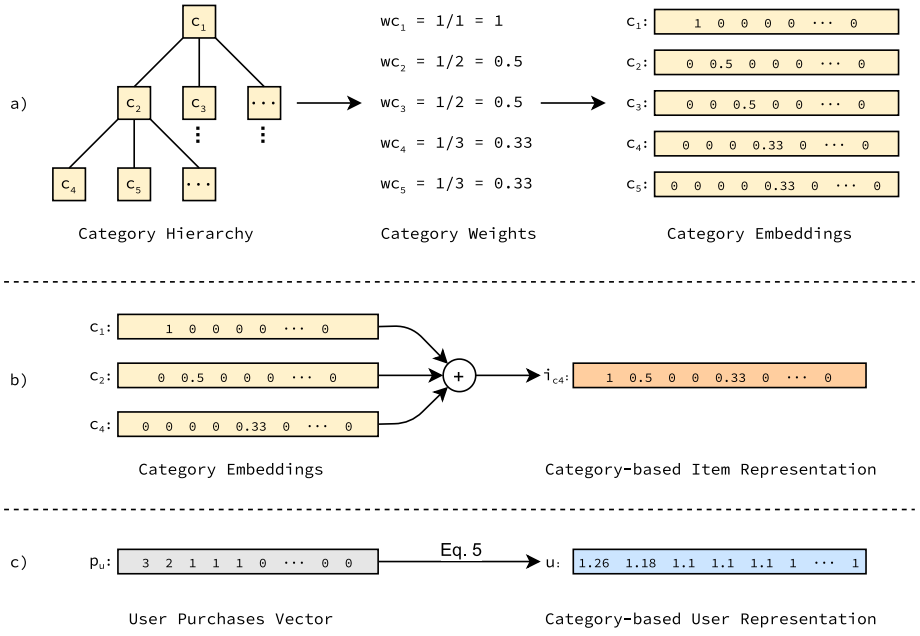

Fig. 3. Category-based representations.

3.5. Item popularity

Item popularity is inherently valuable in the context of Product Search as users often buy the most purchased, most rated, or most reviewed items corresponding to their needs. Therefore, we employ an item-popularity-based score during re-ranking to account for this relevance indicator and promote items usually purchased by the users. The popularity score of a given item is computed as the n-root of the total number of times the item has been purchased. The n-root scaling allows us to avoid penalizing low popular items and to smooth the popularity gap between low and high popular items. Although this score could skew the re-ranked results list towards popular items, the employed scaling mechanism, and the other scores — that are all personalized — counterbalance this effect.

3.6. Re-ranking function

In Information Retrieval, the fusion of distinct relevance scores of a document with respect to a given query is usually computed via their linear combination, as follows:

$$new\_score = (1 - \lambda) \cdot a + \lambda \cdot b \tag{6}$$

where, a and b are the relevance scores computed by two hypothetical ranking methods.

As our approach accounts for multiple relevance scores besides the relevance score computed by the underlying search engine (BM25), we extend Eq. (6) to fuse those scores, in order to compute the new rank score for each retrieved item i with respect to the user u as follows:

$$new\_score_{u,q,i} = \left(1 - \sum_{k=1}^{n} \frac{w_k}{n}\right) \cdot r_{q,i} + \sum_{k=1}^{n} \frac{w_k}{n} \cdot s_k \tag{7}$$

where n is the number of the considered user/item representations; $r_{q,i}$ is the relevance score computed by the employed search engine — based on BM25 in our case — for the item i and a given query q. s_k and w_k are the values of the score related to the user/item representation k and the weight associated with it, respectively. In our case the scores are the cosine similarity between user and item review-based representations, the cosine similarity between user and item interaction-based representations, the dot product between user and item category-based representations and the item popularity score. The value $\sum_{k=1}^{n} \frac{w_k}{n}$ in the linear combination depends on the weights associated with the considered representations. Both the representation weights and their sum range in the interval $[0, 1]$. As reported later (see Section 4.3), the representation weights choice has been automatically conducted as an hyper-parameter search problem using the Python package Optuna [41]. The search result lists are re-ranked according to the ranks computed by the linear combination in Eq. (7).

Depending on the items for sale in e-commerce sites, users may not want to buy the same product twice. This is not the case of grocery products, but if we think to products such as books or CDs a user may not like to see among the top ranked results the items he/she already purchased. To avoid this situation, we suggest to employ a simple

mechanism to alter the ranking position of the already purchased items regardless of their relevance score. In the experiments presented in this paper, we have pushed to the bottom of the re-ranked list the products already bought by the user as this is consistent with the datasets we have used for the evaluation of the proposed approach (see Section 4.1). However, other strategies can be employed for this purpose. For example, we could insert the already purchased items at a fixed position in the results list so that the user would be able to find them easily. We think there is no correct choice in general, as it depends on specific real-world applications, and it is strictly related to the products an e-commerce platform sells and to the behavior of its typical user.

4. Experimental setup

In this section we describe the experimental settings and introduce the baseline methods used in our experiments to perform comparative evaluations. We also present and discuss our model settings and hyper-parameters.

4.1. Datasets

Due to the lack of publicly available datasets for Product Search, the Amazon Review 5-Core dataset [42] has been recently adopted as a benchmark dataset for Product Search along with synthetically generated queries [4–12], as it does not contain query logs. The Amazon Review 5-Core dataset has been widely used for the evaluation of Recommender Systems as it contains millions of users and items as well as user-generated reviews and ratings, item descriptions, and item categories. Also, each user and each item have at least 5 associated reviews, which correspond to actual purchases.

To the aim of generating synthetic queries to be used with the Amazon Review 5-Core dataset as an appropriate benchmark for Product Search evaluation, previous research works [4–12] have followed the query extraction method proposed by Gysel et al. [12]. This approach is based on the assumption that users search for "a producer's name, a brand or a set of terms which describe the category of the product" [15] and it works as follows: for each category c generate a query q by (1) concatenating the terms in the category hierarchy of c, (2) removing the stop-words, and (3) removing duplicated words in reverse order (e.g., *Camera & Photo → Digital Camera* would generate "*photo digital camera*"). Each item belonging to the category c is considered as relevant to the query q.

To comparatively evaluate the proposed approach with the considered baselines, which are detailed in the next section, we relied on the datasets proposed and shared[2] by Ai et al. [4]. These datasets contain data from four subsets of the Amazon Review 5-Core dataset — *Electronics*, *Kindle Store*, *CDs & Vinyl*, and *Cell Phones & Accessories* — along with queries generated following Gysel et al. [12]. The datasets proposed by Ai et al. [4] are suitable for studying personalization in Product Search as the authors have built user–query pairs by linking user–item pairs with item-related queries. In this setting, only the items purchased by a user and related to a query are considered as relevant for that user–query pair. Each dataset comes already partitioned into training and test sets. Partitioning was done so that every query and user–query–item triplet in the test set is new and unobserved. Reviews related to user–query–item triplets in the test sets were removed from the training sets. Table 1 reports some statistics about the datasets; the reader can refer to [4] for additional information. Note that in our work and in all the previous efforts the four datasets were considered separately even if they come from the same e-commerce website and no information was shared among them.

[2] https://github.com/QingyaoAi/Amazon-Product-Search-Datasets.

Table 1
Statistics of the benchmark datasets.

	Electronics	*Kindle Store*	*CDs & Vinyl*	*CP & A*
# users	192 403	68 233	75 258	27 879
# items	63 001	61 934	64 663	10 429
# reviews	1 689 188	982 618	1 097 591	194 439
# queries	989	4 603	694	165

4.2. Baselines

We have compared the proposed model with four different retrieval approaches. The first one is a traditional domain-agnostic retrieval model based on bag-of-words representations, BM25 [14]. The other three are recent Neural Network-based retrieval models specifically designed for Product Search, namely Latent Semantic Entity [12], the Hierarchical Embedding Model [4] and the Dynamic Relation Embedding Model [10].

BM25. The first baseline is the well-known classic probabilistic retrieval model BM25 [14], which is also used in the basic search engine employed by our proposed re-ranking approach. We consider BM25 as a baseline for the specific purpose of assessing if our re-ranking approach is able to enhance the retrieval effectiveness of the underlying retrieval model. BM25 scores are computed on product titles, descriptions, and reviews as a whole, by first removing stop-words and applying the Krovetz stemmer [43].

Latent Semantic Entity (LSE). LSE [12] is a Product Search model based on a Neural Network trained to project both items and queries in the same latent space where their semantic representations can be directly compared to evaluate the relevance of an item with respect to a given query. LSE relies on a tunable parameter to control the embedding dimension. We considered LSE as a baseline mainly to compare the effectiveness of lexical matching approaches (BM25) with respect to semantic matching ones in Product Search.

Hierarchical Embedding Model (HEM). HEM [4] is a personalized Product Search model based on Neural Networks. Through HEM, the users, items, and queries are projected in the same latent space where they can be directly compared. The distributed representations of users, items, and queries are learned in a generative fashion, by maximizing the likelihood of the observed user–query-item triplets. Similarly to LSE, HEM is based on a purely semantic retrieval model. HEM makes use of tunable hyper-parameters to control the personalization impact and the embeddings dimension.

Dynamic Relation Embedding Model (DREM). DREM [10] is a state-of-the-art Neural Network-based model that, similarly to HEM, maps users, items, and queries in the same latent space. Unlike HEM, which makes use only of reviews, DREM takes advantage of many additional side information, such as item brands and item categories. DREM shares the same query representation model of HEM, and therefore it can be considered a purely semantic retrieval model. DREM delivers the best retrieval performance among the considered baselines at the state-of-the-art in Product Search. DREM relies on two tunable hyper-parameters to control the personalization impact and the embeddings dimension.

4.3. Model training and hyper-parameters tuning

To the aim of defining the user and item representations described in Section 3, both Node2Vec and PV-DBOW were trained using the hyper-parameters configurations proposed by their respective authors. Lemmatization and stop-words removal were performed on reviews before feeding them to PV-DBOW. The weights w_k in Eq. (7) and the root smoothing parameter for item popularity were tuned using the hyper-parameter optimization Python package Optuna [41]. The search

space for each parameter ranges from 0.01 to 1.0 — note that the n-root operation used for item popularity smoothing has been implemented as power-of-n. These values were sampled from a discrete uniform distribution, with a discretization step equal to 0.01. This means that our search space was composed of 10^{10} possible combinations. The number of trials was limited to 10. The decay constant λ in Eq. (4) was set to 0.1 for all datasets.

4.4. Evaluation metrics

To evaluate the proposed approach and compare it with the baselines, we employed three different evaluation metrics: (1) mean average precision (MAP), (2) mean reciprocal rank (MRR) and normalized discounted cumulative gain (NDCG). MAP is the arithmetic mean of the average precision values — the mean of the precision scores after each relevant item is retrieved. MRR of a query results list is computed as the inverse of the rank of the first relevant item. MRR gives information about the expected number of results a user needs to view before finding one she/he is interested in. NDCG gives insights into how good a ranked list is compared to the optimal one. MAP and MRR were computed on the top 100 items retrieved by each model, whereas NDCG was computed on the top 10: these cutoffs have been chosen accordingly to previous works [4,10] for evaluation consistency. Statistical significance testing was conducted with the Fisher randomization test [44] with $p \leq 0.01$.

5. Results and discussion

In this section, we present the results obtained on different Product Search benchmark datasets by our model as well as by the considered baselines. First, we discuss the retrieval performance of all the considered models and analyze the results in detail. Then, we conduct an ablation study of the side information employed by our re-ranking model and discuss the effect of personalization over the underlying retrieval function, BM25.

5.1. Retrieval performance

Table 2 shows the effectiveness of our model and those of the baselines on the datasets *Electronics*, *Kindle Store*, *CDs & Vinyl* and *Cell Phones & Accessories*. Our model's results refer to the application of the proposed re-ranking approach to the top-1000 results produced by BM25 in response to the user queries. HEM's and DREM's results refer to the best hyper-parameter configuration found by their respective authors on the test sets of the same benchmark datasets, while LSE's results refer to the best hyper-parameter configuration found by Ai at al. [4] for these same datasets. Note that the evaluation datasets used here as well as their training/test splits are the same of the previous works and so are the performance scores of previous models. BM25's hyper-parameters were tuned using Optuna [41], with a number of trials limited to 10.

BM25 performed consistently better than LSE, achieving a NDCG increase ranging from 54% on *Electronics* to 143% on *Kindle Store*. The employed lexical retrieval model outperforms LSE, although the latter was specifically designed for Product Search. We claim that the main reason for this is related to the fact that vocabulary mismatch — the main issue that drove the design of LSE — is not so severe in Product Search. Moreover, we think lexical matching is fundamental when searching for products as real-world objects usually have a well-defined related lexicon. Users and sellers usually know this lexicon, and they use it to describe their needs and the characteristics of the products they sell, respectively. Finally, as already pointed out by Guo et al. [45] in an ad-hoc retrieval scenario, "relevance matching requires proper handling of the exact matching signal" and, therefore, supporting vocabulary mismatch through semantic search while decreasing lexicon-based matching capabilities is not ideal.

Table 2
Effectiveness of our model and those of the baselines on four benchmark datasets. Best values are highlighted in boldface.

Model	*Electronics*			*Kindle Store*		
	MAP	MRR	NDCG	MAP	MRR	NDCG
BM25	0.320*	0.321*	0.368*†	0.015*	0.017*	0.017*
LSE	0.233	0.234	0.239	0.006	0.007	0.007
HEM	0.308*	0.309*	0.329*	0.029**	0.035**	0.033**
DREM	0.366**†	0.367**†	0.408**†	**0.057****†	**0.067****†	**0.067****†
Our	**0.405****†‡	**0.406****†‡	**0.451****†‡	0.046**†	0.054**†	0.055**†
Model	*CDs & Vinyl*			*Cell Phones & Accessories*		
	MAP	MRR	NDCG	MAP	MRR	NDCG
BM25	0.027*	0.031*	0.032*	0.205*†	0.205*†	0.212*†
LSE	0.018	0.022	0.020	0.098	0.098	0.084
HEM	0.034**	0.040**	0.040**	0.124	0.124	0.153*
DREM	0.074**†	0.084**†	0.086**†	0.249**†	0.249**†	0.282**†
Our	**0.077****†‡	**0.088****†‡	**0.092****†‡	**0.294****†	**0.294****†	**0.306****†

$*$, $\star$, $\dagger$, $\ddagger$ denote significant differences w.r.t. BM25, LSE, HEM and DREM respectively, in Fisher randomization test with $p \leq 0.01$.

Surprisingly, BM25 also outperformed HEM on two datasets out of four, achieving a NDCG increase of 12% on *Electronics* and 39% on *Cell Phones & Accessories*, respectively. We suspect HEM failed to deliver better performance than BM25 on two of the considered datasets, despite its personalization mechanism, due to its semantic-based retrieval function, similarly to LSE.

Our proposed model consistently improved BM25 performances by a considerable margin across all the benchmark datasets, demonstrating the effectiveness of the proposed re-ranking approach as well as of the employed user and item representations. Our approach also outperformed all the considered baselines across all datasets with the sole exception of DREM on the *Kindle Store* dataset. Our approach achieved a NDCG increase over LSE ranging from 89% on *Electronics* to 686% on *Kindle Store* and a NDCG increase over HEM ranging from 37% on *Electronics* to 130% on *CDs & Vinyl*. This again demonstrates the superiority of BM25 as a ranking function over the ranking function learned by LSE and HEM as well as the better performance of the proposed personalization approach over that of HEM.

Regarding the comparison with the state-of-the-art model DREM, our approach achieved strong improvements on the *Electronics* and the *CDs & Vinyl* datasets, +11% and +7% on the NDCG score, respectively, while performing similarly on the *Cell Phones & Accessories*, as no statistically significant difference was detected by the Fisher randomization test [44]. DREM achieves better results then our model on the *Kindle Store* dataset. By further analyzing the results, it is easy to spot that DREM outperforms our model on the dataset where BM25 achieved the worst performance. As a consequence of being a re-ranking approach, our model struggles when the recall of the underlying retrieval function is poor, despite being able to consistently improve the ranking of the initially retrieved items. This is because, by design, re-ranking approaches do not have full access to the product catalog but only to the initial result list. Therefore, if the underlying retrieval function fails to retrieve the relevant items, a re-ranking model cannot improve the result list. However, we suppose the poor performances of BM25 on *Kindle Store* and *CDs & Vinyl* are due to the synthetic queries contained in the benchmark datasets [4] used for conducting the comparative evaluation and the available information about the products in those datasets and not to the model itself. Following Gysel et al. [12], queries were generated from categorical information related to the items. There is no query related to authors or book titles for *Kindle Store* nor artist names or album titles for *CDs & Vinyl*. This seems to be largely unlikely as music can be listened to before buying on many web platforms and more. We guess that users often issue queries containing the title — or part of it — of a book or its author's name because they already know what they are looking for. Often people buy books because of word-of-mouth or because they watched their authors interviewed in

Table 3
Effectiveness of BM25 and our approach with the use of each kind of side information in isolation.

Model	*Electronics*			*Kindle Store*		
	MAP	MRR	NDCG	MAP	MRR	NDCG
BM25	0.320	0.321	0.368	0.015	0.017	0.017
Review	0.320	0.321	0.372	0.023	0.028	0.027
Interaction	0.340	0.340	0.387	**0.057**	**0.068**	**0.068**
Category	0.371	0.371	0.422	0.016	0.018	0.018
Popularity	0.346	0.347	0.396	0.016	0.018	0.018
Model	*CDs & Vinyl*			*Cell Phones & Accessories*		
	MAP	MRR	NDCG	MAP	MRR	NDCG
BM25	0.027	0.031	0.032	0.205	0.205	0.212
Review	0.047	0.054	0.056	0.230	0.230	0.231
Interaction	0.066	0.077	0.079	0.200	0.200	0.216
Category	0.034	0.039	0.040	0.240	0.240	0.264
Popularity	0.029	0.034	0.035	0.214	0.214	0.218

television programs. A similar discussion can be done for *Electronics* and *Cell Phones & Accessories* where, for example, brands have a high impact on sales. However, the categorical information used to generate the queries for *Electronics* and *Cell Phones & Accessories* always contains item types, such as "screen protector" or "bluetooth speaker", that are highly discriminative and resemble real user queries, despite the lack of more specific information, such as the storage capacity for an external Hard Drive. On the other hand, *Kindle Store* and *CDs & Vinyl* contains very few item types and the categorical information is mostly based on literature and music genres, respectively. In addition, *Kindle Store* and *CDs & Vinyl* have a very high number of missing item titles (100% and 23%, respectively) and descriptions (75% and 27%, respectively), probably due to how the item-related data were obtained by McAuley et al. [42]. Finally, it is worth mentioning that DREM makes use of item categories as a source of item-related information when computing their latent representations. Despite the author of DREM *anonymized* item categories when computing item representations, their model can still learn to draw strong relationships between the information about item categories and the terms of the queries, that following [12] were constructed from the item categories, as extensively described in Section 4.1. BM25 would suffer from a very similar problem if we index the item categories as text along with the other item-related information (titles, descriptions, and reviews). Therefore, to not trivialize the retrieval task, we did not index the item categories. However, it would probably be a good practice in a real-world scenario as users often search for a product using "terms that describe the category of the product" [15]. As our re-ranking approach does not have direct access to the query terms and a fixed formula is used to compute user and item category-based representations, as discussed in Section 3, it was not affected by the aforementioned issue

5.2. Ablation study

In this section we analyze the contribution of each of the employed compatibility scores and item popularity on our proposed personalized re-ranking process, by leveraging them in isolation. In Table 3 the results of the ablation study are reported. *Review*, *Interaction*, *Category* and *Popularity* refer to our personalized re-ranking approach with the sole use of *Review-based representations*, *Interaction-based representations*, *Category-based representations* and *Item popularity*, respectively.

As shown in the table, the chosen side information used for re-ranking purpose always increases BM25's effectiveness also when used in isolation. There is no clear insight on which of those information is the strongest/weakest for re-ranking products, as their effect vary on a dataset bases. Interestingly, the sole use of the *Interaction-based representations* in the *Kindle Store* dataset allows our approach to achieve state-of-the-art performance even on the toughest of the benchmark dataset. This mainly indicates the necessity of a smarter score fusion approach, able to dynamically select and weight the contribution of the side information when computing the re-ranking scores. We leave this for future studies.

Table 4
Effectiveness of our model and those of its variants using the alternative categories weighing schemes. Best values are highlighted in boldface.

Model	*Electronics*			*Kindle Store*		
	MAP	MRR	NDCG	MAP	MRR	NDCG
Our	**0.405**	**0.406**	0.451	**0.046**	**0.054**	**0.055**
Reversed	0.401	0.402	**0.455**	0.042	0.050	0.051
Flat	0.402	0.403	0.446	0.045	0.053	0.054
Model	*CDs & Vinyl*			*Cell Phones & Accessories*		
	MAP	MRR	NDCG	MAP	MRR	NDCG
Our	**0.077**	**0.088**	**0.092**	**0.294**	**0.294**	**0.306**
Reversed	0.071	0.082	0.085	0.286	0.286	0.295
Flat	0.073	0.084	0.087	0.282	0.282	0.291

We also performed an ablation study of the procedure we employed to build the category-based representations described in Section 3.4. In particular, we changed the category weighing scheme, trying two different alternatives. Firstly, we reversed the category importance, giving more weight to more specific categories instead of the broader ones. Secondly, we applied a flat weighing scheme so that each category has the same importance. Table 4 shows the results of this analysis. *Reversed* refers to using the compatibility scores obtained by reversing the weighing scheme, while *Flat* refers to using a flat weighing scheme instead. The results of this analysis confirm that the formulation we proposed is more effective.

5.3. Analysis of the efficiency of the proposed approach

In this section, we analyze the overhead deriving from the employed re-ranking model on top of BM25 and the time needed to train the review-based and the interaction-based representation models. To conduct this evaluation, we selected 1000 user–query pairs from the *Electronics* dataset. We assumed the various user and item representations to be previously computed and stored in the system's RAM. Note that the size of the representations of all the *Electronics*' users (192k) and items (63k) is less than 40MB in total. The average time required by BM25 to retrieve a set of documents from the *Electronics* dataset in response to a user query amount to 152 ms in our test system. Preparing all the compatibility scores required by our re-ranking function, computing the new scores, and re-order the retrieved list of documents only takes 17 ms. Therefore, the overhead deriving from our personalization approach at run-time is negligible and should not negatively impact the user experience while increasing the overall retrieval performances. Regarding the training time of the neural models employed by the proposed approach (PV-DBOW and Node2Vec), they are very computationally efficient, and only one hour is required to train each model on a single CPU core (Intel® Core i7-4790k) on the largest of the evaluation datasets (*Electronics*). For comparison, training HEM [4], the smaller model among the personalized baselines, on an NVidia® Titan X GPU usually requires 7-8 h on the same dataset.

6. Conclusion and future work

In this paper, we addressed the problem of personalized results re-ranking in the context of Product Search. In particular, we investigated the use of four different user/item representations to enhance BM25 performances on the top 1000 results. We employed representations derived from user-generated content, user purchasing behavior, categorical information, and item popularity. Our empirical evaluations show that the proposed approach consistently enhances BM25 and outperforms recently proposed Neural Network-based models specifically designed for Product Search on multiple benchmark datasets. As

future work, we intend to further improve the proposed approach by accommodating new representations, by refining the existing ones and by improving our re-ranking function to dynamically weight users and items contextual information on a per-query basis.

CRediT authorship contribution statement

Elias Bassani: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Gabriella Pasi:** Conceptualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Statista, Number of digital buyers worldwide from 2014 to 2021, 2020, URL https://www.statista.com/statistics/251666/number-of-digital-buyers-worldwide.

[2] Statista, Retail e-commerce sales worldwide from 2014 to 2024, 2021, URL https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/.

[3] Statista, E-commerce share of total global retail sales from 2015 to 2024, 2021, URL https://www.statista.com/statistics/534123/e-commerce-share-of-retail-sales-worldwide/.

[4] Q. Ai, Y. Zhang, K. Bi, X. Chen, W.B. Croft, Learning a hierarchical embedding model for personalized product search, in: N. Kando, T. Sakai, H. Joho, H. Li, A.P. de Vries, R.W. White (Eds.), Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017, ACM, 2017, pp. 645–654, http://dx.doi.org/10.1145/3077136.3080813.

[5] Y. Guo, Z. Cheng, L. Nie, X. Xu, M.S. Kankanhalli, Multi-modal preference modeling for product search, in: S. Boll, K.M. Lee, J. Luo, W. Zhu, H. Byun, C.W. Chen, R. Lienhart, T. Mei (Eds.), 2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018, ACM, 2018, pp. 1865–1873, http://dx.doi.org/10.1145/3240508.3240541.

[6] Y. Zhang, X. Chen, Q. Ai, L. Yang, W.B. Croft, Towards conversational search and recommendation: System ask, user respond, in: A. Cuzzocrea, J. Allan, N.W. Paton, D. Srivastava, R. Agrawal, A.Z. Broder, M.J. Zaki, K.S. Candan, A. Labrinidis, A. Schuster, H. Wang (Eds.), Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018, ACM, 2018, pp. 177–186, http://dx.doi.org/10.1145/3269206.3271776.

[7] Q. Ai, D.N. Hill, S.V.N. Vishwanathan, W.B. Croft, A zero attention model for personalized product search, in: W. Zhu, D. Tao, X. Cheng, P. Cui, E.A. Rundensteiner, D. Carmel, Q. He, J.X. Yu (Eds.), Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019, ACM, 2019, pp. 379–388, http://dx.doi.org/10.1145/3357384.3357980.

[8] K. Bi, Q. Ai, Y. Zhang, W.B. Croft, Conversational product search based on negative feedback, in: W. Zhu, D. Tao, X. Cheng, P. Cui, E.A. Rundensteiner, D. Carmel, Q. He, J.X. Yu (Eds.), Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019, ACM, 2019, pp. 359–368, http://dx.doi.org/10.1145/3357384.3357939.

[9] H. Zamani, W.B. Croft, Joint modeling and optimization of search and recommendation, in: O. Alonso, G. Silvello (Eds.), Proceedings of the First Biennial Conference on Design of Experimental Search & Information Retrieval Systems, Bertinoro, Italy, August 28-31, 2018, in: CEUR Workshop Proceedings, 2167, CEUR-WS.org, 2018, pp. 36–41, URL http://ceur-ws.org/Vol-2167/paper2.pdf.

[10] Q. Ai, Y. Zhang, K. Bi, W.B. Croft, Explainable product search with a dynamic relation embedding model, ACM Trans. Inf. Syst. 38 (1) (2020) 4:1–4:29, http://dx.doi.org/10.1145/3361738.

[11] Y. Guo, Z. Cheng, L. Nie, Y. Wang, J. Ma, M.S. Kankanhalli, Attentive long short-term preference modeling for personalized product search, ACM Trans. Inf. Syst. 37 (2) (2019) 19:1–19:27, http://dx.doi.org/10.1145/3295822.

[12] C.V. Gysel, M. de Rijke, E. Kanoulas, Learning latent vector spaces for product search, in: S. Mukhopadhyay, C. Zhai, E. Bertino, F. Crestani, J. Mostafa, J. Tang, L. Si, X. Zhou, Y. Chang, Y. Li, P. Sondhi (Eds.), Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, in, USA, October 24-28, 2016, ACM, 2016, pp. 165–174, http://dx.doi.org/10.1145/2983323.2983702.

[13] G.W. Furnas, T.K. Landauer, L.M. Gomez, S.T. Dumais, The vocabulary problem in human-system communication, Commun. ACM 30 (11) (1987) 964–971, http://dx.doi.org/10.1145/32206.32212, URL http://doi.acm.org/10.1145/32206.32212.

[14] S.E. Robertson, S. Walker, Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval, in: W.B. Croft, C.J. van Rijsbergen (Eds.), Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum), ACM/Springer, 1994, pp. 232–241, http://dx.doi.org/10.1007/978-1-4471-2099-5_24.

[15] J. Rowley, Product search in e-shopping: A review and research propositions, J. Consum. Mark. 17 (2000).

[16] O. Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E.J. Shekita, B. Sznajder, S. Yogev, Beyond basic faceted search, in: M. Najork, A.Z. Broder, S. Chakrabarti (Eds.), Proceedings of the International Conference on Web Search and Web Data Mining, WSDM 2008, Palo Alto, California, USA, February 11-12, 2008, ACM, 2008, pp. 33–44, http://dx.doi.org/10.1145/1341531.1341539.

[17] S.C.J. Lim, Y. Liu, W.B. Lee, Multi-facet product information search and retrieval using semantically annotated product family ontology, Inf. Process. Manag. 46 (4) (2010) 479–493, http://dx.doi.org/10.1016/j.ipm.2009.09.001.

[18] D. Vandic, J. van Dam, F. Frasincar, Faceted product search powered by the semantic web, Decis. Support Syst. 53 (3) (2012) 425–437, http://dx.doi.org/10.1016/j.dss.2012.02.010.

[19] D. Vandic, F. Frasincar, U. Kaymak, Facet selection algorithms for web product search, in: Q. He, A. Iyengar, W. Nejdl, J. Pei, R. Rastogi (Eds.), 22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013, ACM, 2013, pp. 2327–2332, http://dx.doi.org/10.1145/2505515.2505664.

[20] H. Duan, C. Zhai, J. Cheng, A. Gattani, Supporting keyword search in product database: A probabilistic approach, Proc. VLDB Endow. 6 (14) (2013) 1786–1797, http://dx.doi.org/10.14778/2556549.2556562, URL http://www.vldb.org/pvldb/vol6/p1786-duan.pdf.

[21] H. Duan, C. Zhai, J. Cheng, A. Gattani, A probabilistic mixture model for mining and analyzing product search log, in: Q. He, A. Iyengar, W. Nejdl, J. Pei, R. Rastogi (Eds.), 22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013, ACM, 2013, pp. 2179–2188, http://dx.doi.org/10.1145/2505515.2505578.

[22] H. Duan, C. Zhai, Mining coordinated intent representation for entity search and recommendation, in: J. Bailey, A. Moffat, C.C. Aggarwal, M. de Rijke, R. Kumar, V. Murdock, T.K. Sellis, J.X. Yu (Eds.), Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015, ACM, 2015, pp. 333–342, http://dx.doi.org/10.1145/2806416.2806557.

[23] J.M. Ponte, W.B. Croft, A language modeling approach to information retrieval, SIGIR Forum 51 (2) (2017) 202–208, http://dx.doi.org/10.1145/3130348.3130368.

[24] N. Parikh, N. Sundaresan, Beyond relevance in marketplace search, in: C. Macdonald, I. Ounis, I. Ruthven (Eds.), Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011, ACM, 2011, pp. 2109–2112, http://dx.doi.org/10.1145/2063576.2063902.

[25] J. Yu, S. Mohan, D. Putthividhya, W. Wong, Latent dirichlet allocation based diversified retrieval for e-commerce search, in: B. Carterette, F. Diaz, C. Castillo, D. Metzler (Eds.), Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014, ACM, 2014, pp. 463–472, http://dx.doi.org/10.1145/2556195.2556215.

[26] K. Aryafar, D. Guillory, L. Hong, An ensemble-based approach to click-through rate prediction for promoted listings at etsy, in: Proceedings of the ADKDD'17, Halifax, NS, Canada, August 13 - 17, 2017, ACM, 2017, pp. 10:1–10:6, http://dx.doi.org/10.1145/3124749.3124758.

[27] S.K.K. Santu, P. Sondhi, C. Zhai, On application of learning to rank for E-commerce search, in: N. Kando, T. Sakai, H. Joho, H. Li, A.P. de Vries, R.W. White (Eds.), Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017, ACM, 2017, pp. 475–484, http://dx.doi.org/10.1145/3077136.3080838.

[28] Y. Hu, Q. Da, A. Zeng, Y. Yu, Y. Xu, Reinforcement learning to rank in E-commerce search engine: Formalization, analysis, and application, in: Y. Guo, F. Farooq (Eds.), Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018, ACM, 2018, pp. 368–377, http://dx.doi.org/10.1145/3219819.3219846.

[29] Y. Lin, A. Datta, G.D. Fabbrizio, E-commerce product query classification using implicit user's feedback from clicks, in: N. Abe, H. Liu, C. Pu, X. Hu, N.K. Ahmed, M. Qiao, Y. Song, D. Kossmann, B. Liu, K. Lee, J. Tang, J. He, J.S. Saltz (Eds.), IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018, IEEE, 2018, pp. 1955–1959, http://dx.doi.org/10.1109/BigData.2018.8622008.

[30] P. Sondhi, M. Sharma, P. Kolari, C. Zhai, A taxonomy of queries for E-commerce search, in: K. Collins-Thompson, Q. Mei, B.D. Davison, Y. Liu, E. Yilmaz (Eds.), The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, ACM, 2018, pp. 1245–1248, http://dx.doi.org/10.1145/3209978.3210152.

[31] S. Manchanda, M. Sharma, G. Karypis, Intent term selection and refinement in e-commerce queries, 2019, CoRR abs/1908.08564, URL http://arxiv.org/abs/1908.08564.

[32] S. Manchanda, M. Sharma, G. Karypis, Intent term weighting in E-commerce queries, in: W. Zhu, D. Tao, X. Cheng, P. Cui, E.A. Rundensteiner, D. Carmel, Q. He, J.X. Yu (Eds.), Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019, ACM, 2019, pp. 2345–2348, http://dx.doi.org/10.1145/3357384.3358151.

[33] N. Su, J. He, Y. Liu, M. Zhang, S. Ma, User intent, behaviour, and perceived satisfaction in product search, in: Y. Chang, C. Zhai, Y. Liu, Y. Maarek (Eds.), Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018, ACM, 2018, pp. 547–555, http://dx.doi.org/10.1145/3159652.3159714.

[34] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, in: JMLR Workshop and Conference Proceedings, vol. 32, JMLR.org, 2014, pp. 1188–1196, URL http://proceedings.mlr.press/v32/le14.html.

[35] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: B. Krishnapuram, M. Shah, A.J. Smola, C.C. Aggarwal, D. Shen, R. Rastogi (Eds.), Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, ACM, 2016, pp. 855–864, http://dx.doi.org/10.1145/2939672.2939754.

[36] S.E. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Found. Trends Inf. Retr. 3 (4) (2009) 333–389, http://dx.doi.org/10.1561/1500000019.

[37] Z.S. Harris, Distributional structure, Word 10 (2–3) (1954) 146–162.

[38] J.R. Firth, A synopsis of linguistic theory, 1930–1955, Stud. Linguist. Anal. (1957).

[39] M. Sahlgren, The distributional hypothesis, Italian J. Disabil. Stud. 20 (2008) 33–53.

[40] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C.J.C. Burges, L. Bottou, Z. Ghahramani, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States, 2013, pp. 3111–3119, URL https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

[41] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, G. Karypis (Eds.), Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, ACM, 2019, pp. 2623–2631, http://dx.doi.org/10.1145/3292500.3330701.

[42] J.J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in: R. Baeza-Yates, M. Lalmas, A. Moffat, B.A. Ribeiro-Neto (Eds.), Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015, ACM, 2015, pp. 43–52, http://dx.doi.org/10.1145/2766462.2767755.

[43] R. Krovetz, Viewing morphology as an inference process, in: R. Korfhage, E.M. Rasmussen, P. Willett (Eds.), Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Pittsburgh, PA, USA, June 27 - July 1, 1993, ACM, 1993, pp. 191–202, http://dx.doi.org/10.1145/160688.160718.

[44] M.D. Smucker, J. Allan, B. Carterette, A comparison of statistical significance tests for information retrieval evaluation, in: M.J. Silva, A.H.F. Laender, R.A. Baeza-Yates, D.L. McGuinness, B. rn Olstad, Ø.H. Olsen, A.O.F. ao (Eds.), Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007, ACM, 2007, pp. 623–632, http://dx.doi.org/10.1145/1321440.1321528.

[45] J. Guo, Y. Fan, Q. Ai, W.B. Croft, A deep relevance matching model for ad-hoc retrieval, in: S. Mukhopadhyay, C. Zhai, E. Bertino, F. Crestani, J. Mostafa, J. Tang, L. Si, X. Zhou, Y. Chang, Y. Li, P. Sondhi (Eds.), Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, in, USA, October 24-28, 2016, ACM, 2016, pp. 55–64, http://dx.doi.org/10.1145/2983323.2983769.