A Zero Attention Model for Personalized Product Search

Qingyao Ai*
School of Computing, UMass Amherst
Amherst, MA, USA
aiqy@cs.umass.edu

Daniel N. Hill
Amazon Search
Berkeley, CA, USA
daniehil@amazon.com

S. V. N. Vishwanathan
Amazon Search
Palo Alto, CA, USA
vishy@a9.com

W. Bruce Croft
CICS, UMass Amherst
Amherst, MA, USA
croft@cs.umass.edu

ABSTRACT

Product search is one of the most popular methods for people to discover and purchase products on e-commerce websites. Because personal preferences often have an important influence on the purchase decision of each customer, it is intuitive that personalization should be beneficial for product search engines. While synthetic experiments from previous studies show that purchase histories are useful for identifying the individual intent of each product search session, the effect of personalization on product search in practice, however, remains mostly unknown. In this paper, we formulate the problem of personalized product search and conduct large-scale experiments with search logs sampled from a commercial e-commerce search engine. Results from our preliminary analysis show that the potential of personalization depends on query characteristics, interactions between queries, and user purchase histories. Based on these observations, we propose a Zero Attention Model for product search that automatically determines when and how to personalize a user-query pair via a novel attention mechanism. Empirical results on commercial product search logs show that the proposed model not only significantly outperforms state-of-the-art personalized product retrieval models, but also provides important information on the potential of personalization in each product search session.

KEYWORDS

Product Search, Personalization, Attention Mechanism

ACM Reference Format:
Qingyao Ai, Daniel N. Hill, S. V. N. Vishwanathan, and W. Bruce Croft. 2019. A Zero Attention Model for Personalized Product Search. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, **??** pages. https://doi.org/10.1145/3357384.3357980

1 INTRODUCTION

Due to the increasing popularity of online shopping and a large number of products on e-commerce websites, product search has become one of the most popular methods for customers to discover products online. In a typical product search scenario, a user would first issue a query on the e-commerce website to get a list of relevant products, then browse the result page, and select one or more items to purchase. Therefore, the quality of product search results has a direct impact on customer satisfaction and the number of transactions on e-commerce websites.

Because purchasing is a personal behavior with a real financial cost, it is well-recognized that personal preferences could directly affect customer's purchase decisions [**?**]. Previous studies show that many purchase intents in product search can be influenced by a user's personal taste and experience [**? ?**]. Experiments on synthetic data also demonstrate that incorporating user information extracted from product reviews and purchase histories can significantly improve the performance of product retrieval models [**?**]. Thus, it is intuitive that personalization should have a significant potential in product search.

A major question is whether personalization always improves the quality of product search. Attractive as it seems to be, search personalization has been shown to potentially have negative effects by previous studies on Web search [**? ? ?**]. When a customer submits the query "toothpaste" to an e-commerce search engine, it is possible that they want personalized search results which suit their personal needs (e.g., sensitive teeth). It is also possible that they do not differentiate much between toothpastes, and simply purchase the best seller on the list. On the one hand, personalization with user information can help us better understand the user's search intent when there is limited information revealed in the query. On the other hand, incorporating unreliable personal information could exacerbate the problem of data sparsity and introduce unnecessary noise into a search model. When and how to conduct personalization is an important research question for product search in practice.

Despite its importance, the effect of personalization on real product search engines has not been extensively studied. To the best of our knowledge, existing work on personalized product search only uses user information as an additional feature in the retrieval model, and conducts undifferentiated personalization in all search sessions [**? ?**]. This research constructs user profiles with query-independent information and provides little insight on how personalization would benefit or damage the performance of product retrieval in different search scenarios.

In this paper, we explore the potential and risks of personalization for product search. Specifically, we focus on the question

*This work is done in Amazon Search during the first author's Ph.D. at UMass Amherst.

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11.
https://doi.org/10.1145/3357384.3357980

of when and how to do search personalization with a user's purchase history on e-commerce websites. We start from a theoretical analysis and identify two necessary conditions in order for personalization to benefit product search. Then, we verify the two conditions on large-scale search logs sampled from a commercial e-commerce search engine. We find that, while personalization appears to be useful for queries with medium or high frequency, it tends to be less beneficial on tail queries with low frequency. Also, we notice that the importance of personalization in product search often depends on the interactions between query context and the user's previous purchases. It is impossible to determine the usefulness of personalization without knowing the query and the user's purchase history simultaneously.

Based on our observations, we propose a Zero Attention Model for differentiated personalization in product search. Previous studies on personalized product search often model users and queries separately by assuming that they are independent [**?**]. This, however, makes it impossible to conduct query-specific personalization in each search session. To that end, we propose a Zero Attention Strategy that constructs user profiles as a weighted combination of their previously purchased items. The idea of the Zero Attention Strategy is to allocate different attention to the user's previous purchases according to their current search intent. More importantly, in contrast to a classic attention mechanism where the model must attend to at least one item in the input, the Zero Attention Strategy introduces a zero vector and allows the attention model to pay no attention to any input. As shown in Section **??**, this essentially creates a threshold function that enables our product retrieval model – the Zero Attention Model (ZAM) – to automatically determine when and how to personalize search results based on the current query and user information. Our experiments on real e-commerce search logs demonstrate that ZAM can significantly outperform state-of-the-art baselines for personalized product search. Also, the attention weight on the zero vector in ZAM is a good indicator for the potential of personalization in each search session.

In summary, the major contributions of this paper include:

- We present both a theoretical and empirical analysis of the potential of personalization in product search with large-scale search logs sampled from a commercial e-commerce search engine.
- We propose a novel Zero Attention Model for personalized product search by conducting differentiated personalization for different query-user pairs.
- We conduct comparisons over state-of-the-art personalization techniques for product search, and discuss their advantages and disadvantages in practice.

2 RELATED WORK

Our work in this paper is closely related to the research of product search, search personalization, and neural retrieval models.

Product Search. Product search is an important problem that has been widely studied in the research communities of Data Mining and Information Retrieval. Early studies mainly focus on how to construct effective retrieval systems that support search for structured product information [**? ? ?**]. For example, Lim et al. [**?**] propose to conduct product retrieval with a facet search engine built on structured representations of products in relational databases such as brands, prices, categories, etc. Later, product search studies move to more advanced retrieval models that support more complicated search data and optimization objectives. For instance, Van Gysel et al. [**?**] introduce a latent semantic entity model that matches products and queries in a latent semantic space; Guo et al. [**?**] propose a TranSearch model that can search product images with text queries. There are also a variety of studies on applying feature extraction and learning-to-rank techniques [**? ? ? ?**] to product search for the optimization of different product retrieval metrics [**?**]. In this paper, we focus on the problem of when and how to conduct search personalization in product retrieval.

Search Personalization. Studies on search personalization focus on providing different results to each individual to maximize their search experience [**?**]. In the context of Web search, this means re-ranking documents according to the personal needs of each user based on their locations, search histories, clicked documents, etc. [**? ? ? ?**]. While search users could have different behavior for the same query [**? ?**], it has been found that personalization is not always beneficial for the effectiveness of search engines. For example, in a large-scale log analysis of user behavior patterns on Live Search, Teevan et al. [**?**] identify multiple factors that affect the usefulness of personalization in Web search, such as result entropy, result quality, search tasks, etc.

In the scope of product search, however, the effect of search personalization has not been fully studied. While recent studies on e-commerce search logs [**? ?**] identify and analyze several types of product search intents qualitatively, there has been little work on quantifying the potentials and risks of personalization in product search. There are two notable exceptions. The first is Jannach and Ludewi's work on applying personalized recommendation techniques to product search [**?**]. The authors reduce the problem of product search to a recommendation task by ignoring the search query in the ranking. A second study by Ai et al. investigated personalized product search using Amazon Review datasets [**?**]. They conducted experiments with synthetic queries generated from product category information. To the best of our knowledge, our work is the first study that analyzes the impact of personalization on a real e-commerce search engine.

Neural Retrieval Models. The advance of deep learning techniques has attracted much attention in the IR community recently. Neural retrieval models are highly effective as they can automatically learn and incorporate embedding features in ranking optimizations [**? ? ? ? ?**]. For product search, Van Gysel et al. [**?**] propose a latent semantic entity model that jointly learns the embeddings of words, queries, and items so that products could be directly retrieved according to their similarities in the latent space. Later, Ai et al. [**?**] incorporate user information extracted from user purchases and product reviews into a neural generative framework and create a Hierarchical Embedding Model for product search.

Attention is an important deep learning technique that enables neural networks to allocate variable attention or weighting to different model components in the training or prediction process [**?**]. It has proved effective in many AI applications such as Computer Vision [**? ?**], Natural Language Processing [**? ?**], etc. Examples in e-commerce studies include the work of Chen et al. [**?**] which applies attention mechanisms to the construction of a personalized

recommender system. Their task, however, is fundamentally different from product search as users cannot specify their information needs explicitly in recommendation scenarios. In this paper, we focus on the problem of personalized product search and propose a new attention mechanism that allows a retrieval model to conduct differentiated personalization based on the previous purchases of the user and their relationship to the current query.

3 PRELIMINARY ANALYSIS

In this section, we present our preliminary analysis on the potential of personalization in product search. Previous work has demonstrated that incorporating user information is beneficial for the performance of product retrieval models on synthetic data generated from product category hierarchies [**?**]. However, to the best of our knowledge, there is no systematic study on why and how personalization could help product search in practice. To fill this gap, we conduct a theoretical analysis of the fundamental assumptions behind personalized product search, and validate them on the search logs sampled from a commercial e-commerce search engine. More details about the log data can be found in Section **??**.

The key difference between personalized and non-personalized product search is the modeling of the user's purchase behavior. Formally, let q be a query submitted by a user u. In a personalized product retrieval model, purchase behavior on an item i depends on both the current search context and the user's personal preferences. The probability of whether i would be purchased by u in the query q should be parameterized as $P(i|q,u)$. By contrast, non-personalized product retrieval models assume that user's purchase behavior only depends on the current search context (i.e., query q). It models the probability of item purchase as $P(i|q)$, which can be treated as the aggregated group preference over all possible users as

$$P(i|q) = \sum_{u \in \mathcal{U}} P(i|q,u) \cdot P(u)$$

where $\mathcal{U}$ is the universal user set, and $P(u)$ is the probability of observing a specific user u in the search logs. We assume that the distribution of u is independent of q for simplicity.

Given this formulation, we can identify the first necessary condition for personalization to benefit product search as:

Condition 1. *Personalization is beneficial only when the query carries diverse purchase intents.*

The proof is straightforward – $P(i|q,u)$ cannot be a better model than $P(i|q)$ when q only represents the intent to purchase a specific item. The more specific intent a query has, the less beneficial personalization could be in product search.

To analyze query specificity, we compute the purchase entropy of each query in the sampled e-commerce search logs as

$$Entropy(q) = -\sum_{i \in I_q} P(i|q) \log P(i|q) = -\sum_{i \in I_q} \frac{\#(i,q)}{|S_q|} \log_2 (\frac{\#(i,q)}{|S_q|}) \quad (1)$$

where I_q is the candidate item set for query q, S_q represents the search sessions for q, and $\#(i,q)$ refers to the number of sessions in S_q where item i has been purchased.

Figure **??** shows the purchase entropy of queries on *Beauty* products (e.g., facial cleanser) in our sampled search logs. Here, we rank queries according to their frequencies in logarithmic scale, and split them into three groups: the queries with low frequency (*LowFreq*), with medium frequency (*MedFreq*), and with high frequency (*HighFreq*). The group sizes are balanced so that the total number of sessions in each group is approximately the same. As depicted in the figure, purchase entropy shows a strong correlation with query frequency. When the number of sessions increases, we are likely to observe more purchases on different items, which usually indicates diverse purchase intents. Hence, it seems that queries with high frequencies have more potential for personalization.

This, however, may not be true. While a high purchase entropy indicates a high diversity of purchase intents, it doesn't necessarily mean that every user has unique preferences in search. Therefore, the second necessary condition for the effectiveness of personalization in product search is:

Condition 2. *Personalization is beneficial only when the personal preferences of individuals are significantly different from their aggregated group preference.*

In other words, personalized models can outperform non-personalized models only when the distribution of $P(i|q,u)$ is significantly different from $P(i|q)$.

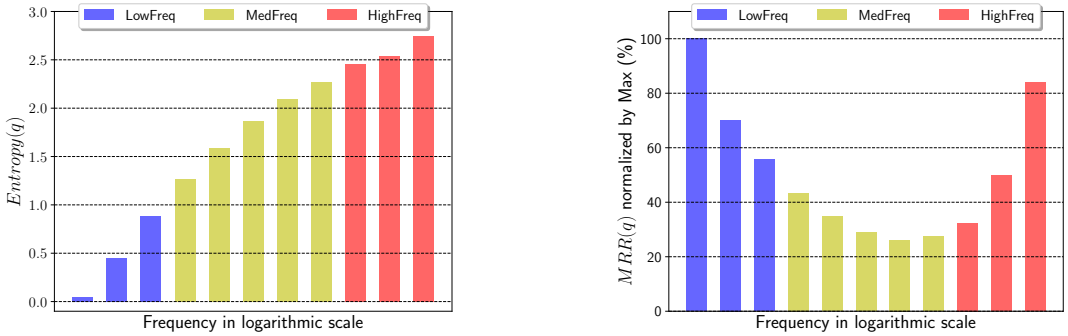
A simple method to evaluate the differences between $P(i|q,u)$ and $P(i|q)$ in search problems is to rank items with $P(i|q)$ and evaluate the performance with $P(i|q,u)$. Specifically, we use:

$$MRR(q) = \sum_{u \in \mathcal{U}} RR(P(i|q), P(i|q,u)) \cdot P(u) \quad (2)$$

where $RR(P(i|q), P(i|q,u))$ is the reciprocal rank of a ranked list produced by ranking with $P(i|q)$, using $P(i|q,u)$ as the ground truth. In other words, $MRR(q)$ can be computed as the mean reciprocal rank of a retrieval model that ranks products according to how many times they have been purchased given the query (i.e., *the Popularity Model*), using the actual user purchases in each session as the ground truth. The higher $MRR(q)$ is, the closer $P(i|q,u)$ is to $P(i|q)$ and the less beneficial search personalization can be.

Figure **??** shows the $MRR(q)$ of queries on *Beauty* products. We compute the Popularity Model based on one year of search logs and test it on the search data sampled from the following week. Numbers are normalized with respect to the maximum $MRR(q)$ of all query groups. As we can see in the figure, $MRR(q)$ shows a U shape with respect to query frequency. While tail queries and head queries tend to have high $MRR(q)$, queries with median frequencies usually have low $MRR(q)$ in our search logs. This indicates that, unlike purchase entropy, the similarity between $P(i|q,u)$ and $P(i|q)$ is not monotonically correlated with query frequency.

To further understand the results of Figure **??** and Figure **??** together, we extract some example queries from the search logs for each query group. In *LowFreq*, we observe many *spear-fishing* queries where the user directly specifies the name of the item in the query string, such as "stila stay all day waterproof liquid eyeliner", "mario badescu drying lotion", etc. When searching with these queries, users usually have exact items in mind. Thus, the query specificity is high and ranking by $P(i|q)$ achieves a good performance in general. In *MedFreq*, we observe queries on personal products like "body wash for women", "maybelline mascara", etc. Although the concept of relevance could be personal in these queries, users only provide vague descriptions of what items they



(a) Purchase Entropy **(b) Popularity Model Performance**

Figure 1: The purchase entropy $Entropy(q)$ and popularity model performance $MRR(q)$ for queries with different frequencies.

want in the query strings. As a result, the purchase entropy is high and the $MRR(q)$ of the Popularity Model is low. In *HighFreq*, while we observe many queries on popular products in daily activities, such as "sunscreen", "nail file", and "mouthwash", the user behavior patterns in these queries, however, are not consistent. For "sunscreen", we observe products with a variety of properties such as "oil-free", "water-resistant", etc. It seems easy to identify the need of each user using the products they purchased. Whereas, in "nail file" and "mouthwash", most candidate items are cheap and homogeneous. While the purchase entropy of these queries is high, we haven't identify any salient personal preferences over the candidate products among the customers we sampled. In many cases, the customers simply purchased one of the best sellers. Therefore, the potential of personalization varies significantly in different queries, and this motivates us to develop a more sophisticated model for personalization in product search.

4 ZERO ATTENTION MODEL

In this section, we propose a Zero Attention Model (ZAM) for personalized product search. ZAM is designed under an embedding-based generative framework. It conducts query-dependent personalization by constructing user profiles with a Zero Attention Strategy that enables it to automatically decide when and how to attend in different search scenarios. Theoretical analysis shows that our proposed attention strategy can create a dynamic threshold that controls the weights of personalization based on both the query and the user purchase history.

4.1 Embedding-based Generative Framework

Latent semantic models have proven effective for product search and recommendation [**? ?**]. Among different types of latent semantic models, neural embedding models have achieved the state-of-the-art performance on many benchmark product search datasets [**? ?**]. Specifically, Ai et al. [**?**] propose an embedding-based generative framework that can jointly learn the embedding representations of queries, users, and items by maximizing the likelihood of observed user purchases.

Let q be a query submitted by a user u, i be an item in the candidate set I_q for q, and α be the size of an embedding vector. Under an embedding-based generative framework [**? ?**], the probability of whether i would be purchased by u given q can be modeled as

$$P(i|u,q) = \frac{\exp(\boldsymbol{i} \cdot \boldsymbol{M_{uq}})}{\sum_{i' \in I_q} \exp(\boldsymbol{i'} \cdot \boldsymbol{M_{uq}})} \tag{3}$$

where $\boldsymbol{i} \in \mathbb{R}^{\alpha}$ is the embedding representation of i, and M_{uq} is a joint model of user-query pair (u,q). Products are ranked according to $P(i|u,q)$ so that the probability of user purchases in each search session can be maximized. Depending on the definition of M_{uq}, we could have multiple types of embedding-based retrieval models for product search. Here, we introduce two representative models from previous studies – the Query Embedding Model and the Hierarchical Embedding Model.

Query Embedding Model. The Query Embedding Model (QEM) is an embedding-based generative model for non-personalized product search [**?**]. It defines M_{uq} as

$$\boldsymbol{M_{uq}} = \boldsymbol{q} \tag{4}$$

where $\boldsymbol{q} \in \mathbb{R}^{\alpha}$ is the embedding representation of the query q.

Because queries usually are unknown beforehand, $\boldsymbol{q}$ must be computed in product search at request time. Previous studies [**? ?**] have explored several methods to construct query embeddings from query words directly. One of the state-of-the-art paradigms is to compute query embeddings by encoding query words with a non-linear projection function ϕ defined as

$$\boldsymbol{q} = \phi(\{w_q | w_q \in q\}) = \tanh(\boldsymbol{W_\phi} \cdot \frac{\sum_{w_q \in q} \boldsymbol{w_q}}{|q|} + \boldsymbol{b_\phi}) \tag{5}$$

where $\boldsymbol{w_q} \in \mathbb{R}^{\alpha}$ is the embedding of a word w_q in q, $|q|$ is the length of the query, and $\boldsymbol{W_\phi} \in \mathbb{R}^{\alpha \times \alpha}$ and $\boldsymbol{b_\phi} \in \mathbb{R}^{\alpha}$ are two parameters learned in the training process.

In QEM, item embedding is learned from their associated text data. Let T_i be a set of words associated with an item i (e.g., titles). Ai et al. [**?**] propose to learn $\boldsymbol{i}$ by optimizing the likelihood of observing T_i given i as

$$P(T_i|i) = \prod_{w \in T_i} \frac{\exp(\boldsymbol{w} \cdot \boldsymbol{i})}{\sum_{w' \in V} \exp(\boldsymbol{w'} \cdot \boldsymbol{i})} \tag{6}$$

where $\boldsymbol{w} \in \mathbb{R}^{\alpha}$ is the embedding of word w, and V is the vocabulary of all possible words. Note that it is important to learn $\boldsymbol{i}$ separately instead of representing it by averaging word embeddings because user purchases could be affected by information other than text [**?**].

Hierarchical Embedding Model. Similar to QEM, HEM [**?**] also computes query embeddings with the encoding function ϕ and item embeddings with their associated text T_i. However, in contrast to QEM, HEM defines M_{uq} in Eq. (**??**) as

$$\boldsymbol{M_{uq}} = \boldsymbol{q} + \boldsymbol{u} \tag{7}$$

where $\boldsymbol{u}$ is the embedding representation of the user u. In this way, HEM considers both query intents and user preferences in the ranking of items for product search.

In HEM, the embedding of user u is obtained by optimizing the likelihood of observed user text T_u given u as

$$P(T_u|u) = \prod_{w \in T_u} \frac{\exp(\boldsymbol{w} \cdot \boldsymbol{u})}{\sum_{w' \in V} \exp(\boldsymbol{w'} \cdot \boldsymbol{u})} \tag{8}$$

where T_u could be any text written or associated to u, such as product reviews or the descriptions of items that the user has purchased.

4.2 Attention-based Personalization

As discussed by Ai et al. [**?**], HEM is constructed based on the assumption that user preferences are independent of query intents in search. This, however, is not true in practice. For example, a customer who likes the baby products from *Johnson's* may not want to buy *Johnson's* baby shampoo when they search "shampoo for men". To address this problem, a better paradigm for personalization in product search is to consider the relationships between queries and a user's purchase history. Specifically, we apply an attention function over the user's purchase history to build user embeddings for product search. Let I_u be the set of items purchased by user u before query q, then we can compute the embedding of u as

$$\boldsymbol{u} = \sum_{i \in I_u} \frac{\exp(f(q, i))}{\sum_{i' \in I_u} \exp(f(q, i'))} \boldsymbol{i} \tag{9}$$

where $f(q, i)$ is an attention function that determines the attention weight of each item i with respect to the current query q. Similar to previous studies on attention models [**? ?**], we define $f(q, i)$ as

$$f(q, i) = \left(\boldsymbol{i} \cdot \tanh(\boldsymbol{W}_f \cdot \boldsymbol{q} + \boldsymbol{b}_f)\right) \cdot \boldsymbol{W}_h \tag{10}$$

where $\boldsymbol{W}_h \in \mathbb{R}^{\beta}$, $\boldsymbol{W}_f \in \mathbb{R}^{\alpha \times \beta \times \alpha}$, $\boldsymbol{b}_f \in \mathbb{R}^{\alpha \times \beta}$, and β is a hyper-parameter that controls the number of hidden units in the attention network. Given the attention-based user embedding $\boldsymbol{u}$, we can further conduct personalized product search using the same method described in Eq. (**??**) and Eq. (**??**). We refer to this model as the Attention-based Embedding Model (AEM).

In contrast to HEM, AEM conducts personalization with query-dependent user profiling. The embedding representations of each user are constructed according to their queries so that the model can better capture relevant user preferences in the current search context. This is beneficial especially when the user has purchased many products that are irrelevant to the current query. There is another attention-based product search model proposed by Guo et al. [**?**] that shares the similar idea of AEM. Unfortunately, for the computation of attention weights, their model assumes that each item in a user's purchase history should associate with at least one search query submitted by users, which is not true in our datasets. Thus, we ignore it in this paper.

There is, however, an important problem that limits the power of both HEM and AEM. As shown in Section **??**, different queries have different potential for personalization. Despite the efforts on query-dependent user profiling, the attention mechanism in Eq. (**??**) requires AEM to attend to at least one item in the user's purchase history, which means that it always personalizes. Ai et al. [**?**] have explored a naive solution that adds a hyper-parameter in Eq. (**??**) to control the weight of user embedding $\boldsymbol{u}$ in $\boldsymbol{M_{uq}}$, but this merely trades off the gains of personalization on some queries with its loss on other queries. To actually solve this problem, we need a model that can automatically determine when and how to do personalization in product search.

4.3 Zero Attention Strategy

The usefulness of personalization depends on both the query and the user's purchase history. For example, spear-fishing queries often result in the same item being purchased regardless of customer preferences. Similarly, a customer who is shopping in a certain category for the first time may have no relevant history on which to base personalization. To address this issue, we propose a Zero Attention Strategy that relaxes the constraints of existing attention mechanisms by introducing a Zero Vector in the attention process. Accordingly, we propose a Zero Attention Model (ZAM) that conducts differentiated personalization in product search based on search queries and user's purchase history.

Figure **??** shows the structure of ZAM. Similar to AEM, ZAM learns item embeddings based on their associated words, and conducts retrieval with both query embeddings and user embeddings. The main difference between ZAM and AEM is that, instead of attending to the user's previous purchases only, ZAM allows the attention network to attend to a Zero Vector, which we refer to as the Zero Attention Strategy. Formally, let $\boldsymbol{0} \in \mathbb{R}^{\alpha}$ be the Zero Vector where each element is 0. Then, in ZAM, the embedding representation of a user u is computed as

$$\boldsymbol{u} = \sum_{i \in I_u} \frac{\exp(f(q, i))}{\exp(f(q, \boldsymbol{0})) + \sum_{i' \in I_u} \exp(f(q, i'))} \boldsymbol{i} \tag{11}$$

where $f(q, \boldsymbol{0})$ is the attention score of $\boldsymbol{0}$ with respect to query q.

We now show how this simple modification achieves differentiated personalization in product search. Let $\boldsymbol{x} \in \mathbb{R}^{|I_u|}$ be a vector formed from $\{f(q, i) | i \in I_u\}$. Then, Eq. (**??**) can be reformulated as

$$\boldsymbol{u} = \frac{\exp(\boldsymbol{x})}{\exp(f(q, \boldsymbol{0})) + \exp^{+}(\boldsymbol{x})} \cdot \boldsymbol{I_u} \tag{12}$$

where $\boldsymbol{I_u}$ is a matrix consisting of all item embeddings in the user's purchase history, and $\exp^{+}(\boldsymbol{x})$ is the element-wise sum of $\exp(\boldsymbol{x})$. In Eq. (**??**), $\exp(f(q, \boldsymbol{0})) = 1$, so the factor of $\boldsymbol{I_u}$ in Eq. (**??**) is actually a sigmoid function of $\boldsymbol{x}$. In other words, the introduction of the Zero Attention Strategy creates an activation function that controls the influence of user purchase history in the current search context. The value of $\exp^{+}(\boldsymbol{x})$ is the cumulative attention received by the user's previous purchased items given the query, and $\exp(f(q, \boldsymbol{0}))$ is essentially a threshold for personalization. Although our formulation is constant, this threshold can be query-dependent by defining f with a more complicated function. In any case, the user embedding $\boldsymbol{u}$ should only have strong influence in ZAM if the user shows consistent and salient interests on products related to the current query. This enables ZAM to conduct differentiated personalization in different search scenarios.

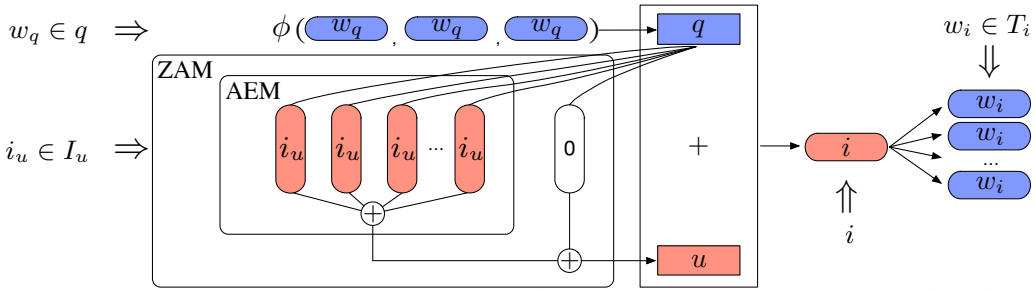


Figure 2: The structure of the Attention-based Embedding Model (AEM) and the Zero Attention Model (ZAM). I_u is the purchase history of user u, i is a candidate item for query q, and w_q and w_i are words in q and the text associated with i (T_i), respectively.

4.4 Model Optimization

Similar to previous studies [? ? ?], we optimize both AEM and ZAM by maximizing the log likelihood of observed user purchases and item information. Specifically, we represent each item with their titles, each user with their previously purchased items, and each query with their query words. Let T_i be a list of words in the title of item i, then the log likelihood of an observed user-query-item triple can be computed as

$$\begin{aligned}\mathcal{L}(T_i, u, i, q) &= \log P(T_i|i) + \log P(i|u, q) + \log P(u, q) \\ &\approx \sum_{w_i \in T_i} \log \frac{\exp(\boldsymbol{w}_i \cdot \boldsymbol{i})}{\sum_{w' \in V} \exp(\boldsymbol{w}' \cdot \boldsymbol{i})} \\ &\quad + \log \frac{\exp(\boldsymbol{i} \cdot (\boldsymbol{q} + \boldsymbol{u}))}{\sum_{i' \in I_q} \exp(\boldsymbol{i}' \cdot (\boldsymbol{q} + \boldsymbol{u}))}\end{aligned} \tag{13}$$

where $\boldsymbol{w}$ and $\boldsymbol{i}$ are learned parameters, $\boldsymbol{q}$ is computed with Eq. (??), $\boldsymbol{u}$ is computed with Eq. (??) (for AEM) or Eq. (??) (for ZAM), and $\log P(u, q)$ is ignored because training examples are sampled uniformly from the logs.

Computing $\mathcal{L}(T_i, u, i, q)$, however, is often infeasible because of the large number of words in V and items in I_q. For efficient training, we adopt a negative sampling strategy [? ? ?] to estimate Eq. (??). Specifically, for each softmax function, we sample k negative samples to approximate its denominator. The final optimization function of AEM and ZAM can be formulated as

$$\begin{aligned}\mathcal{L}' &= \sum_{(u,q,i)} \mathcal{L}(T_i, u, i, q) \\ &\approx \sum_{(u,q,i)} \sum_{w_i \in T_i} \left(\log \sigma(\boldsymbol{w_i} \cdot \boldsymbol{i}) + k \cdot \mathbb{E}_{w' \sim P_w} [\log \sigma(-\boldsymbol{w}' \cdot \boldsymbol{i})] \right) \\ &\quad + \log \sigma\big(\boldsymbol{i} \cdot (\boldsymbol{q} + \boldsymbol{u})\big) \\ &\quad + k \cdot \mathbb{E}_{i' \sim P_{I_q}} \big[\log \sigma\big(- \boldsymbol{i}' \cdot (\boldsymbol{q} + \boldsymbol{u})\big)\big]\end{aligned} \tag{14}$$

where $\sigma(x)$ is a sigmoid function, and P_w and P_{I_q} are the noise distribution of word w and item i, respectively. In our experiments, we define P_w as the unigram distribution raised to the 3/4 power [?], and P_{I_q} as a uniform distribution. Also, as shown in Figure ??, words in queries and item titles and items in I_u and I_q share a common embedding space. We experimented with different types of regularization techniques such as L2 regularization, but none made a significant difference to the performance of the models. This indicates that, given the large-scale training data we have, overfitting is not an issue for model optimization in our experiments. Thus, we ignore the regularization terms in Eq. ?? for simplicity.

Table 1: Some basic statistics of our experimental datasets.

	Beauty	Grocery	Hpc
Unique Words	168,831	132,032	191,640
Unique Users	999,130	998,655	997,989
Unique Queries	1,369,893	918,453	1,171,631
Unique Products	3,940,175	2,928,612	3,868,178
Unique Sessions	3,985,150	4,381,715	3,809,994

5 EXPERIMENTAL SETUP

In this section, we describe our experimental settings for the log analysis and product search experiments. We first introduce how we collect search logs from a commercial product search engine, and then the details of our baselines and the parameter settings for different product retrieval models.

5.1 Datasets and Evaluation

Our log analysis and retrieval experiments are carried on products from three categories – *Beauty*, *Grocery*, and *Health&Personal Care*. *Beauty* consists of products for beauty purposes, such as facial cleanser and hair care; *Grocery* consists of food purchases like chips and vegetables; and *Health&Personal Care* contains more items on non-beauty personal care, such as vitamins, dental cleaning products, etc. Any search sessions result in a purchase on products that belong to a certain category would be classified as a search in that category.

To construct our experimental testbed and preserve the privacy of customers at the same time, we anonymize all customer information beforehand and randomly sampled one million customers for each product category from the search logs of a commercial e-commerce search engine in 2018. We collected the search sessions of each user in the corresponding category from a one year period. We also collected the top 100 items retrieved by an initial product retrieval system for each query as our candidate item sets. Each item is represented with its title and a unique product ID. In this paper, we focus on predicting user purchases based on the user's current query and their purchase histories. Only the purchased items are considered relevant in each session, and we manually exclude the sessions with no purchase. Search sessions are divided with 30-minutes inactivity or query changing. Our training and test sets are formed by splitting search sessions chronologically on a specific date to achieve a train/test ratio of 7 to 1. The history of a user is formed by their purchases and queries in the corresponding category before the current search session. Test users that have never appeared in the training set were removed. We also remove words that have appeared less than 5 times in the whole dataset

and simply ignore OOV words in the training and testing process. The basic statistics of our datasets are shown in Table **??**.

For ranking evaluation, we compute Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain at 10 (NDCG@10), and Hit ratio at 10 (Hit@10) in session level. MRR focuses on the position of the first retrieved item that the user has purchased; NDCG evaluates the overall ranking performance over all purchased items; and Hit@10 is a recall-focused metric that measures the percentage of sessions where users have purchased at least one item in the top 10 results. Note that Significant differences are computed based on student t-test with $p \leq 0.01$, and model performances are reported as improvement percentages over the standard language modeling approach for IR (i.e. the Query Likelihood model).

5.2 Baselines

In our experiments, we include six types of product search baselines:

- **Query Likelihood Model** (QL): The standard language modeling approach for IR [**?**], which ranks products based on the log likelihood of queries in the item's unigram language model constructed with title words.
- **Query-dependent Popularity** (Pop_q): The popularity model that ranks items according to how many times they have been purchased given the query q in the training data.
- **Query-independent Popularity** (Pop_i): The popularity model that ranks items according to how many times they have been purchased in all sessions in the training data.
- **Query Embedding Model** (QEM): The product retrieval model based on the embedding-based generative framework described in Section **??**, which can be treated as a non-personalized version of the Hierarchical Embedding Model.
- **Hierarchical Embedding Model** (HEM): The personalized product search model proposed by Ai et al. [**?**] described in Section **??**. In this paper, we use the purchased item's titles as the text associated with each user.
- **Attention-based Embedding Model** (AEM): A personalized product search model proposed in Section **??**, which constructs query-dependent user embeddings with a classic attention mechanism over the user's purchase history.

Among the six baseline methods above, QL is a classic term-based retrieval model, Pop_q/Pop_i are statistical models, and QEM, HEM and AEM are embedding-based neural retrieval models. The first four are non-personalized product search models, while the last two are personalized product search models.

5.3 Parameter Settings

Both the neural baselines and our Zero Attention Model (ZAM) are trained on an NVIDIA Tesla K40 GPU with 12 GB memory. We initialized all embedding vectors randomly, and trained each neural model for 20 epochs in total (most models converged after 10 epochs). We used the Adagrad optimizer with an initial learning rate of 0.5. We set the batch size as 256 and the number of negative samples k as 5. For QL, we tuned the Dirichlet smoothing factor μ from 10 to 100 (the average length of item titles is shorter than 100). For AEM and ZAM, we tuned the number of hidden units β (in Section **??**) from 1 to 5, and set it as 3 in the final results. And for all embedding models, we tuned the embedding size α from 100 to 300. Large embedding size did not introduce any significant performance improvements in our experiments, so we only report the results with $\alpha = 100$ for all embedding models. Note that all model hyper-parameters are tuned on the training data. We will release our code after the paper is published.

6 RESULTS AND DISCUSSION

We now report our experimental results on personalized product search. We first describe the overall performance of different non-personalized baselines and personalized product retrieval models, and then discuss the effect of personalization on different query groups. After that, we conduct a case study to shed some light on the advantages and disadvantages of each personalization method.

6.1 Overall Performance

Table **??** shows the overall performance of each product search model in our experiments. As we can see, QL is significantly worse than other baselines in all categories, which is consistent with previous studies [**? ?**]. There are significant language gaps between how users formulate queries and how sellers write product descriptions [**?**]. In addition, simple queries will often map to a large number of items (e.g, "phone case"), which makes it difficult for a term-based retrieval model to differentiate the relevance of these products. In contrast, QEM achieves the best performance among the four non-personalized product retrieval models and is nearly two times better than QL. This demonstrates the effectiveness of the embedding-based generative models in product retrieval.

If we compare the performance of personalized product search models with QEM, we can see that HEM is better than QEM on *Grocery*, but significantly worse on *Beauty* and *Health & Personal Care*. As discussed in Section **??**, different queries have different potential for personalization. HEM conducts personalization by adding query-independent user embeddings to every query equally, which can hurt the system performance when a user's global purchase preferences are not related to their current search. After applying the attention mechanism, AEM conducts query-dependent personalization and achieves better ranking performance than HEM. Nonetheless, it still personalizes all queries with equal weights and has a limited improvement over QEM in *Beauty*.

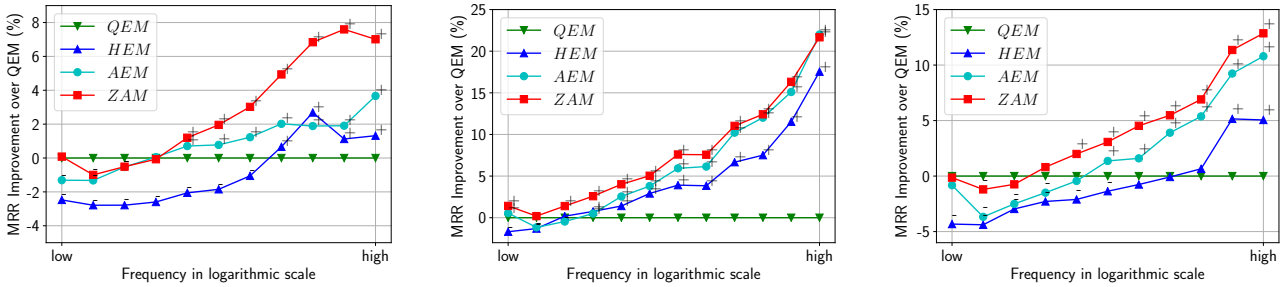
In our experiments, ZAM performs the best among all product search models. On all categories, ZAM achieves more than 2% improvement over QEM on both MRR and NDCG@10. It also significantly outperforms HEM and AEM. This demonstrates that the introduction of the Zero Attention Strategy indeed improves the overall effectiveness of personalized product search models.

6.2 Effect of Personalization

In this section, we describe a quantitative analysis of how personalization changes the performance of product retrieval models on different query groups. Figure **??** depicts the MRR improvements of HEM, AEM, and ZAM with respect to QEM on queries with different frequencies. In our experiments, QEM performs well and achieves the best performance on queries with low frequencies in most cases. HEM and AEM, on the other hand, performs worse than QEM on these queries after the incorporation of user information. As discussed in Section **??**, low-frequency queries usually

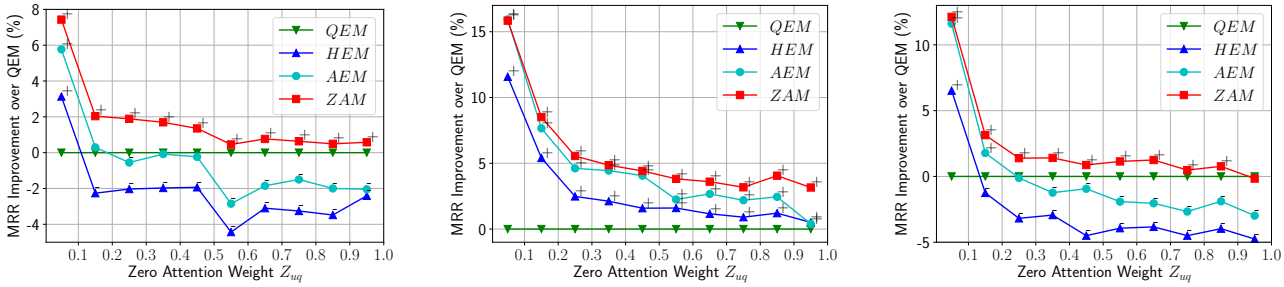
Table 2: The ranking performance of different product search models in each category. Metrics are reported as the improvement percentages over the best non-personalized baseline (i.e., QEM). + and − denote significant improvements or degradation over QEM, and ‡ denotes significant improvements over all the baselines. The best performance is highlighted in boldface.

Model		*Beauty*			*Grocery*			*Health & Personal Care*		
		MRR	NDCG@10	Hit@10	MRR	NDCG@10	Hit@10	MRR	NDCG@10	Hit@10
Non-personalized	QL	-62.59%$^-$	-63.19%$^-$	-56.17%$^-$	-70.86%$^-$	-70.76%$^-$	-60.92%$^-$	-67.66%$^-$	-68.06%$^-$	-60.18%$^-$
	Pop_q	-49.34%$^-$	-47.32%$^-$	-35.81%$^-$	-50.61%$^-$	-46.84%$^-$	-32.10%$^-$	-55.16%$^-$	-52.36%$^-$	-38.68%$^-$
	Pop_i	-27.08%$^-$	-24.70%$^-$	-16.05%$^-$	-31.78%$^-$	-27.36%$^-$	-14.66%$^-$	-34.13%$^-$	-29.28%$^-$	-15.45%$^-$
	QEM	0%	0%	0%	0%	0%	0%	0%	0%	0%
Personalized	HEM	-0.94%$^-$	-1.38%$^-$	-1.79%$^-$	6.05%$^+$	4.70%$^+$	1.48%$^+$	-0.43%	-1.18%$^-$	-2.23%$^-$
	AEM	0.96%$^+$	0.51%	-0.29%	8.50%$^+$	6.80%$^+$	2.54%$^+$	2.71%$^+$	1.85%$^+$	0.07%
	ZAM	**2.77%**$^{+‡}$	**2.10%**$^{+‡}$	**0.59%**	**9.46%**$^{+‡}$	**7.60%**$^{+‡}$	**2.91%**$^+$	**4.57%**$^{+‡}$	**3.36%**$^{+‡}$	**0.71%**$^{+‡}$



(a) *Beauty* **(b) *Grocery*** **(c) *Health&Personal Care***

Figure 3: The performance of different embedding-based product retrieval models with respect to query frequencies. + and − denote significant improvements or degradation with respect to QEM.



(a) *Beauty* **(b) *Grocery*** **(c) *Health&Personal Care***

Figure 4: The performance of different embedding-based product retrieval models with respect to the Zero Attention Weight Z_{uq} in Eq. (??). + and − denote significant improvements or degradation with respect to QEM.

have a spear-fishing intent. People who submit these queries are likely to have exact items in mind. Thus, in these cases, the risk of personalization is larger than its potential. On queries with higher frequencies, we observe more sessions where personalized product search models outperform the non-personalized model (i.e., QEM), especially in *Grocery*. One explanation is that people often have strong personal tastes on products retrieved for the head queries of *Grocery*. For example, while different customers may buy different flavors of snack chips when they searched for "chips", we observed that many of them would purchase the same chips that they have purchased previously. Among different personalization models, ZAM always produces the best performance and, in most cases, is significantly better than HEM and AEM.

In ZAM, we naturally create an activation function that controls the effect of personalization by allowing the model to attend to the Zero Vector **0**. The attention weights of **0** in Eq. (??), namely the Zero Attention Weight, could be a valuable indicator for the potentials of personalization in each product search query. We define the Zero Attention Weight of a user-query pair (u, q) in ZAM as

$$Z_{uq} = 1 - \sum_{i \in I_u} \frac{\exp(f(q,i))}{\exp(f(q,\mathbf{0})) + \sum_{i' \in I_u} \exp(f(q,i'))} \tag{15}$$

When Z_{uq} is 0, ZAM conducts maximum personalization on the query, which means that the query has great potentials for personalization; when Z_{uq} is 1, ZAM conducts no personalization. To verify this hypothesis, we plot the performance of both non-personalized and personalized embedding models with respect to Z_{uq} in Figure ??.

In Figure ??, the y axis is the MRR improvement of each model with respect to QEM, and the x axis is the Z_{uq} in ZAM ranged from 0 to 1. Note that Z_{uq} is not a hyper-parameter but a variable automatically developed by ZAM during training. As shown in Figure ?? and ??, QEM significantly outperforms HEM and AEM on

Table 3: Example sessions on the query of "vitamin c serum" and the recent purchases of each user (sorted by time) in *Beauty*.

Session 1: "vitamin c serum"			Session 2: "vitamin c serum"			Session 3: "vitamin c serum"		
	QEM	HEM		HEM	AEM		AEM	ZAM
Purchased Item	Rank in list		Purchased Item	Rank in list		Purchased Item	Rank in list	
claire vitamin c serum ... eye treatment 1 fl oz	7	1	**tolb vitamin c serum for face ... skin 1 fl oz**	19	2	**tolb vitamin c serum for face ... skin 1 fl oz**	7	2
Previous Purchases of the User	Attention		Previous Purchases of the User	Attention		Previous Purchases of the User	Attention	
6 plastic afro pik ... lift hair comb detangle	–	–	perfume bottle travel refillable ... perfume atomizer spray	–	<0.001	**baebodi eye gel for dark circl ... 1.7 fl oz**	**0.744**	0.0325
essie nail polish, cuticle car... primers and finishers	–	–	**super vitamin c serum for face ... huge 2 fl oz**	–	**0.964**	makeup brush set ... 32 pieces professional makeup	<0.001	<0.001
kiss products ... piece nail kit, just for you	–	–	labs under eye serum,drk circl, 1 FZ	–	<0.001	roller for face and body ... microneedle kit	0.198	<0.001
claire vitamin c serum ... eye treatment 1 fl oz	–	–	premium retinol cream face moisturizer with hyaluronic	–	0.034	**Zero Vector 0**	–	**0.960**

queries where Z_{uq} is larger than 0.5, while HEM and AEM achieve large improvements over QEM on queries with Z_{uq} is smaller than 0.1, even when the overall performance of HEM is significantly worse than QEM. Also, in Figure **??**, we observe that the relative improvements of HEM and AEM with respect to QEM show a strong negative correlation with Z_{uq}. This demonstrates that the weight on the Zero Vector **0** in ZAM is indeed a good indicator of whether a product search query is suitable for personalization.

6.3 Case Study

To understand how different personalization methods benefit or damage the performance of product retrieval, we conduct case studies on *Beauty* and extract three representative sessions for "vitamin c serum" in Table **??**. For each session, we show the final purchased item, its position in the ranked list of each model, and the most recent search purchases of the user. In Session 2 and 3, we also show the attention weights on each previously purchased item computed by AEM and ZAM. For privacy concerns, we randomly replace the item titles in Table **??** with titles from other products with similar properties so that no user information would be revealed.

Session 1 in Table **??** is a typical example where personalized models (i.e., HEM) outperform non-personalized models (i.e., QEM). As shown in the table, before the start of Session 1, the user had already purchased "claire vitamin c serum ... eye treatment 1 fl oz" in a former search session. Without personalization, QEM cannot use this information, and it ranks the item at position 7 for this user. HEM, on the other hand, identifies that "claire vitamin c serum ... eye treatment 1 fl oz" is relevant to the user and it is likely that the user would purchase it again. Thus, HEM ranks it at position 1 in Session 1. This example indicates that personalization is particularly beneficial for re-finding or repeated purchase intents.

Session 2 in Table **??** shows an example where the introduction of attention mechanisms improves the performance of personalization. In HEM, all purchased items in a user's purchase history have equal importance for the construction of the user profile. In Session 2, however, most of the user's previous purchases are not directly related to the current query. As a result, HEM fails to promote the rank of the final purchased item "tolb vitamin c serum for face ... skin 1 fl oz" and places it at position 19. In contrast, AEM builds user embeddings based on the relevance of each item with respect to the current query. It identifies that "super vitamin c serum for face ... huge 2 fl oz" is the only item in the user's purchase history that is relevant to "vitamin c serum", and thus allocates 96.4% of the attention weights to the item. Since "super vitamin c serum for face ... huge 2 fl oz" was no longer available on our experimental e-commerce website and "tolb vitamin c serum for face ... skin 1 fl oz" is a common alternative to it, AEM puts "tolb vitamin c serum for face ... skin 1 fl oz" at the second position in the final rank list.

The last example in Table **??**, Session 3, depicts how the Zero Attention Strategy benefits personalized product search. In this session, none of the user's previous purchases are directly related to vitamin c serum, which means that personalization is not likely to be helpful here. Nonetheless, because the attention mechanism in AEM requires it to construct the user embedding with at least one item in the purchase history, AEM allocates considerable attention on the most "relevant" item – "baebodi eye gel for dark circl ... 1.7 fl oz". We expect a user's tastes in eye gel provide little information on their preferences for "vitamin c serum", so AEM fails to achieve a good performance in Session 3. With the Zero Attention Strategy, ZAM successfully recognizes that Session 3 is not suitable for personalization and places 96.0% of the attention weights on the Zero Vector **0**. This limits the effect of personalization to only 4.0% and results in the promotion of "tolb vitamin c serum for face ... skin 1 fl oz" from position 7 in AEM to position 2 in ZAM.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose a Zero Attention Model (ZAM) for personalized product search. The proposed model constructs user embeddings with a Zero Attention Strategy that enables it to conduct differentiated personalization for product search based on both the query characteristics and user purchase histories. As shown in the experiments, the introduction of the Zero Attention Strategy not only improves the performance of personalized product search models, but also provides important information on the potential of personalization in each product search query.

As a next step, we are interested in studying the potential of ZAM for explainable information retrieval. Although the threshold $f(q, \mathbf{0})$ in the activation function of ZAM (i.e., Eq. (**??**)) could be query-dependent, the attention function used in this paper (i.e., Eq. (**??**)) actually creates the same values for $f(q, \mathbf{0})$ in all queries. On the one hand, this design naturally makes the attention scores of each item comparable among different queries and sessions, which

could be useful for the generation of search explanations and the development of an explainable product retrieval model. On the other hand, learning query-dependent personalization threshold could be benefitial for the retrieval performance of the system. Besides, in our preliminary log analysis, we only analyze the potential of personalization with respect to query frequency. More analysis on other query characteristics such as query length and average candidate price would be interesting. For example, we observed that many customers have consistent price preferences when purchasing products in the *Clothing* category, which could be useful for personalizing their search experience.

8 ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by a gift from Amazon Search. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.