# Learning Dynamic Multi-attribute Interest for Personalized Product Search

**Yutong Bai**
School of Information,
Renmin University of China
ytbai@ruc.edu.cn

**Zhicheng Dou**[*] **and Ji-Rong Wen**
Gaoling School of Artificial Intelligence,
Renmin University of China
{dou,jrwen}@ruc.edu.cn

## Abstract

Personalized product search aims to learn personalized preferences from search logs and adjust the ranking lists returned by engines. Previous studies have extensively explored excavating valuable features to build accurate interest profiles. However, they overlook that the user's attention varies on product attributes(e.g., brand, category). Users may especially prefer specific attributes or switch their preferences between attributes dynamically. Instead, existing approaches mix up all attribute features and let the model automatically extract useful ones from rather complex scenarios. To solve this problem, in this paper, we propose a dynamic multi-attribute interest learning model to tackle the influences from attributes to user interests. Specifically, we design two interest profiling modules: attribute-centered and attribute-aware profiling. The former focuses on capturing the user's preferences on a single attribute, while the latter focuses on addressing the interests correlated with multiple attributes within the search history. Besides, we devise a dynamic contribution weights strategy that sends explicit signals to the model to determine the impacts of different attributes. Experimental results on large-scale datasets illustrate that our model significantly improves the results of existing methods.

## 1 Introduction

With the rapid growth of e-commerce services, online shopping has become increasingly popular. In a common e-shopping scenario, the user formulates her demands into a query and selects the items she is interested in from the list retrieved by a product search engine. However, the query could be ambiguous and have multiple meanings, which makes it difficult to capture accurate user needs. For instance, the user could enter the query word "MAC" to purchase computers, but the search engine cannot distinguish the needs of computers from those of cosmetic brands. Furthermore, the query could also be broad (such as "laptop"), without specifying the brands and desired features of the products the user wants to buy. Personalized product search tasks address this challenge by learning interests from the user history. Researchers have tried excavating features from various views for accurate interest learning. A group of studies aim at improving the features capturing ability through practical algorithms, including simple embedding-based methods (Ai et al., 2017), attention-based methods (Ai et al., 2019), or transformer-based methods (Bi et al., 2020). Some studies pay attention to extracting features from multiple aspects. Modeling the impacts of short- and long-term history (Guo et al., 2019; Bennett et al., 2012; Shen et al., 2022) has been a popular search topic. Leveraging the product reviews (Bi et al., 2021) has also achieved satisfactory results. Some studies also attempt to use visual resources to model multi-modal preferences. Other works (Ai et al., 2020; Liu et al., 2020, 2022) explore relationships between user, items queries by constructing knowledge graphs with the help of product attributes (e.g., names, brands).

However, the studies mentioned above overlook that the users' interests in different product attributes (such as brands, categories, features, etc.) are sophisticated. Instead, they simply feed attribute features into the model and let it automatically learn interests from the mixed features. Such a paradigm neither distinguishes the characteristics of specific attributes nor explicitly models the influences among multiple attributes. Hence, we argue that previous methods do not sufficiently explore the potential within the attributes to reflect user interests. As the user behavior sequence shown in Table 1, the user reveals her special attention for the product brands while showing less consideration for the product names and categories. Intrinsically,

[*]Corresponding author.

Table 1: Example user histories. Attribute information, such as brand and category, of purchased products $p_1$, $p_2$, and $p_3$, and the candidate products $c_1$ and $c_2$ are listed.

| | Name | Brand | Category |
|---|---|---|---|
| ***Purchased Products*** | | | |
| $p_1$ | 2023 iMac | Apple | Computer |
| $p_2$ | iPhone 15 | Apple | Phone |
| $p_3$ | iPad 2018 | Apple | Tablets |
| ***Candidate Products*** | | | |
| $c_1$ | AirPods Max | Apple | Headphone |
| $c_2$ | WH-CH720N | Sony | Headphone |
| ***Current query: headphone*** | | | |

the interest in this case should depend more on the brand features. Whereas, existing studies send all types of features into the model where other features would bring more interference than contribution. The obtained interest representation would include much noise without properly enhancing important attributes, leading to an underestimation of the candidate product $c1$'s relevance. Aiming to resolve this problem, we propose explicitly enhancing important attribute features by learning multi-attribute interest for personalized product search.

To better build multi-attribute interest, we need to answer the following two questions: 1) how to represent the user's interests on specific attributes, and 2) how to effectively fuse the interests on multiple attributes. To resolve the first problem, we attempt to build item/query representations centered on each attribute. As for the second problem, we intend to address the attributes' contributions using two strategies. The first focuses on separately learning the user's attention for each attribute. To achieve this, we would observe the affinities of historical attribute-centered item representations. Higher affinities indicate that the user's tastes on that attribute are stable, so it is important to match her tastes again. The second strategy focuses on simultaneously learning the user interests that switch between attributes. We compress the attribute-centered representations from multiple attributes into attribute-aware representations. Then, we send a sequence of historical representations into one encoder and let the model draw the attribute correlations within the history.

Concretely, we propose a **M**ulti-**A**ttribute **I**nterest learning model (**MAI**) for personalized product search. It includes the following four parts: (1) Attribute-centered interest profiling. For each attribute, it obtains attribute-centered representations for queries and items by enhancing corresponding attribute features and feeding them to corresponding encoders to get the profiles. (2) Attribute-aware interest profiling. It attends attribute correlations within search history with combined attribute-centered item representations. (3) Multi-attribute interest fusion. We update attribute-centered contribution weights by observing the attention weights from the first part. According to these weights, we calculate the similarity score between the profiles and their corresponding candidate representations to obtain the final ranking score.

To summarize, the main contributions of this paper include:

(1) We propose a method of learning multi-attribute interest for personalized product search in a dynamic way.

(2) We design an attribute-centered interest profiling module that builds separate profiles by encoding item/query representations centered on corresponding attributes. We subsequently utilize an attribute-aware interest profiling module to learn user interests based upon multiple attributes.

(3) We propose a method for dynamic multi-attribute fusion that explicitly models the individual contributions of each attribute.

## 2 Related Work

### 2.1 Personalized Product Search

Personalized product search problems aim to improve the ranking quality retrieved from search engines by building accurate user interests from the purchase history. Many studies focus on exploiting interests in semantic latent space by leveraging deep learning technology. Guo et al. (2019) utilize attention networks to learn and integrate long- and short-term user preferences. Bi et al. (2019) study short-term clicks to represent users' hidden intents with a context-aware embedding model. Ai et al. (2019) devise a novel attention mechanism which enables the attention model to attend no input by introducing a zero vector. Such a zero attention model successfully allocates different attention to the users' search logs according to their current intent. Recently, since the transformer (Vaswani et al., 2017) architectures have succeeded in various fields, a group of studies attempt to employ it in personalized product search. For instance, Bi et al. (2021) design a review-level transformer-based model that matches the reviews from the user and item while allowing each review to have a dynamic impact based on the sequential context. Recently, Jagatap et al. (2024) explore query generation and interaction simulation to solve the cold start problem faced by new categories.

## 2.2 Aspect-based Interest Learning

There exist some personalized product search studies that try to learn user interests from various aspects (e.g., brands, categories, popularity, etc.). Early methods, as (Lim et al., 2010), require the aspects to be structurally organized so the algorithms can conduct accurate matching between the query and item. Recent methods obviate such requirements thanks to the deep learning technology's superiority in extracting semantic features from free-form text. Wu et al. (2017) blend multiple models into a stacking ensemble model where different sub-models are used for statistic features, query-item features and session features accordingly. Xiao et al. (2019) devise a Dynamic Bayesian Metric Learning model to represent semantic representations of different categories of users, products, and words and capture the affinities between them. Subsequent studies (Ai et al., 2020; Liu et al., 2022; Zhu et al., 2024) apply knowledge graphs to jointly model sophisticated relationships from structured and unstructured aspects of the user and item.

However, these works blindly send all attributes into a model and let it automatically extract useful features. This would inevitably bring noise to the learning process. In contrast, we efficiently enhance important attributes with explicit weighting for simultaneously and separately profiled attribute interests.

## 3 Methodology

To start with, the problem could be formulated as follows. Suppose that for each user, her search history $H$ includes $N$ purchased items , $H = \{h_1, \ldots, h_N\}$, where $h_i$ represents $i$-th historical purchased items. Given the current query $q$ and the candidate target item list $C = \{c_1, c_2, \ldots\}$ returned by the search engine, our objective is to model a ranking probability score for each candidate item $c$ in $C$ based on the current query $q$ and the purchased item sequence $H$.

The overview of our multi-attribute interest learning model is shown in Figure 1. Later, we will elaborate on the modeling details following the three stages: (1) attribute-centered interest profiling, (2) attributed-aware interest profiling and (3) multi-attribute interest fusion.

### 3.1 Attribute-centered Interest Profiling

As stated in Section 1, previous approaches could not efficiently detect interests centered on specific attributes, for they deteriorate the interesting learning procedure by feeding the models misleading signals from other attributes. This module focuses on solving this problem by preserving attribute-centered information through separate interest profiling.

#### 3.1.1 Query and Item Representation

***Base Query Representation.*** Following previous methods (Ai et al., 2019, 2017, 2020), we generate our base query representation $\mathbf{q}$ using a non-linear projection for the average word embeddings:

$$\mathbf{q} = \tanh(W_{\phi_q} \frac{\sum_{w_q \in q} \mathbf{w_q}}{|q|} + b_{\phi_q}), \qquad (1)$$

where $q$ is the current query, $d$ is the embedding size, $W_{\phi_q} \in \mathbb{R}^{d \times d}$ and $b_{\phi_q} \in \mathbb{R}^d$ are trainable parameters, $|q|$ is the length of $q$ and $\mathbf{w_q} \in \mathbb{R}^d$ is the embedding of word $w_q$ in $q$. These query representations would be used to extract features related to current intents in the attribute-aware interest profiling module.

***Attribute-centered Query Representation.*** We intend to use the current query to enhance current intents during attribute-centered interest profiling. However, these base query representations are independent of attribute features, so the correlations between the query and attributes are complex to capture. As a result, these representations are inefficient in enhancing current intents and might even impede the interest learning from historical attribute features. As the example shown in Table 1, using the current query to emphasize the brand attributes will lead to a cluttered profile where the information of brands and the query are both contaminated.

Thus, we will reformulate the query according to the attributes. Formally, the attribute-centered query representations are obtained as follows. First, for attribute $a_k$, we obtain weighted word embeddings $\mathbf{w_q^{a_k}}$ of the query $q$ according to each word's

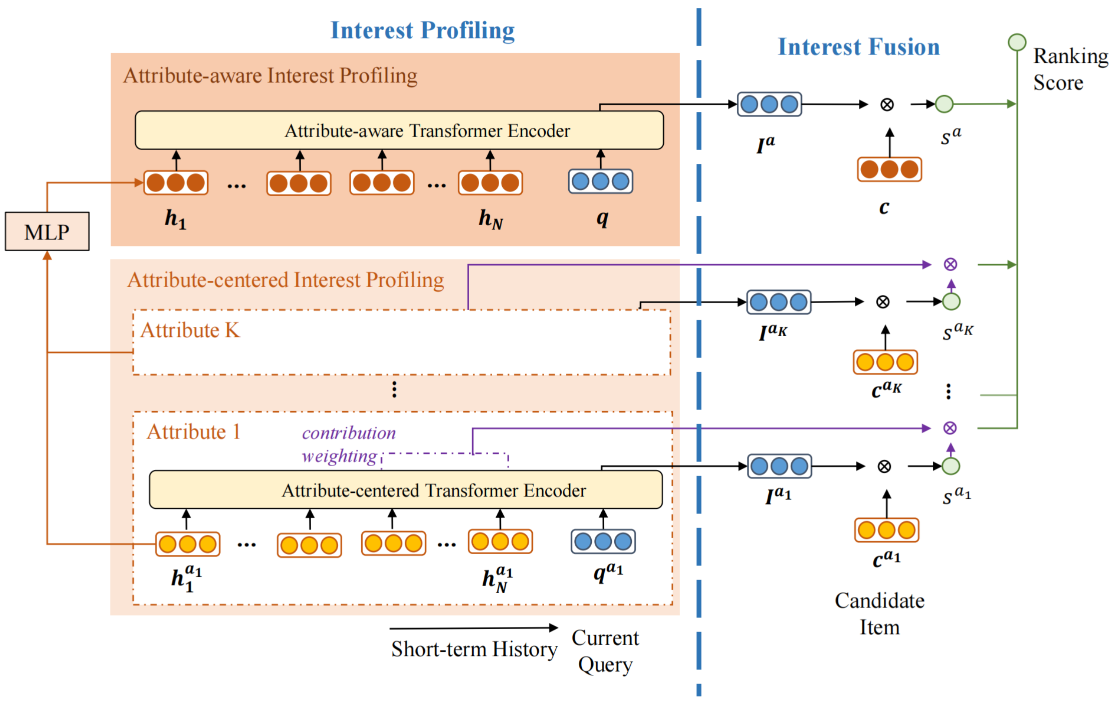

Figure 1: The overview of the proposed MAI. Given the user's purchased items and a current query, we first encode attribute information for each item and the current query into attributed-centered representations and then send them to corresponding attribute-centered interest profiling encoders. Besides, we also compress all the attributed-centered representations of each item to unified vectors and send them to an attribute-aware profiling encoder, where we model the interactions among attributes simultaneously. After obtaining the interest profiles from each profiling procedure, we match them with corresponding attribute representations of the candidate item. Finally, we weighted the matching scores according to contributing weights calculated during profiling.

relationships with recent historical information on attribute $a_k$:

$$\mathbf{w_q^{a_k}} = W_q^{a_k}\mathbf{w_q}, \tag{2}$$

where $W_q^{a_k} \in \mathbb{R}^{|q|}$ is calculated through multi-head attention. To measure the query-attribute relationship, we take the recent attribute representations as the query and the query word embeddings as the key and value. In this way, we enhance query features that are correlated with recent interests:

$$W_q^{a_k} = \text{MLP}([head_1^w, \dots, head_H^w]), \tag{3}$$

where MLP$(\cdot)$ refers to the multilayer perceptron (MLP) with softmax$(\cdot)$ function, $head_h^w$ is the attention weights of the $h$th head in total $H$ heads in the multi-head attention layer. The $head_h^w$ is obtained as follows:

$$head_h^w = \text{Attn}^w(h^{a_k,s}W_h^Q, q^wW_h^K, q^wW_h^V), \tag{4}$$
$$h^{a_k,s} = [\mathbf{h_M^{a_k}}, \dots, \mathbf{h_N^{a_k}}], \tag{5}$$

where $h^{a_k,s}$ is the sequence of short-term attribute representations centered on attribute $a_k$ from the $M$-th to the $N$-th purchased item. $\mathbf{h_i^{a_k}}$ refers to $i$-th item representation in the purchasing history centered on attribute $a_k$. The process of getting it will be explained later. $q^w$ is the word embedding sequence of the query. The projection matrices of each head $W_h^Q \in \mathbb{R}^{d\times d/H}$, $W_h^K \in \mathbb{R}^{d\times d/H}$ and $W_h^V \in \mathbb{R}^{d\times d/H}$ are learned during training. $Attn(\cdot)^w$ is the attention weights from each head:

$$\text{Attn}^w(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d/H}}). \tag{6}$$

At last, we send the weighted embeddings to the same representing procedure as the base query in Equation (1) to get the attribute-centered query representation $\mathbf{q^{a_k}}$. Such reformulated query representations emphasize the query words related to the corresponding attribute.

***Attribute-centered Item Representation.*** We average the term vectors of the item's corresponding attributes and apply the same non-linear function in query representing to get the attribute-centered representation $\mathbf{h_i^{a_k}}$ based on the $k$th attribute $a_k$. The process is the same as Equation (1). $W_{\phi_h}$ and $b_{\phi_h}$ are two different parameters used for item representing. For the candidate item $c$, we also encode its attribute-centered representation $c^{a_K}$ for the final comparison.

### 3.1.2 Attribute-centered Interest Learning

We join the attribute-centered item representations with the attribute-centered query representation to learn the attribute interests. Transformer encoders are used to capture the complex interactions within the history and query. This process can be formulated as follows:

$$\mathbf{I}^{\mathbf{a_k}} = \mathrm{Trm}^{\mathrm{last}}([\mathbf{h}_i^{a_k} + \mathbf{h}_i^{a_k,p}, \mathbf{q}^{a_k} + \mathbf{q}^{a_k,p}]), \quad (7)$$

where $\mathbf{I}^{\mathbf{a_k}}$ is the interest profile centered on attribute $a_k$, $\mathrm{Trm}^{last}$ notes the last outputs of the transformer encoder, which are the query outputs in this case. We leverage position information by adding the position embedding $\mathbf{h}_i^{a_k,p}$ and $\mathbf{q}^{a_k,p}$ for the items and query.

### 3.1.3 Attribute-centered Interest Weighting

In this part, we model the contributing level of each independent attribute-centered interest profile learned from previous parts. We inspect the attention weights of the short-term purchased items assigned by their previous items during the transformer encoding shown in Equation (7). The contribution weights $W^{a_k}$ of attribute $a_k$ in independent profiling are computed from the attention weights of short-term items:

$$W^{a_k} = \mathrm{MLP}([\mathrm{Trm}_M^w, \ldots, \mathrm{Trm}_N^w]), \quad (8)$$

where $\mathrm{Trm}_i^w$ denotes the last transformed encoder layer's attention weights assigned for the $i$th historical item from its previous $P$ items. As explained in Section 1 Higher attention weights suggest the interests on that attributes are more important. To make the model focus on recent interests, we only inspect weights from short-term items.

## 3.2 Attribute-aware Interest Profiling

As we discussed, user preferences may change between attributes within the history. Separately profiling attribute-centered interests would fail to capture such variations. To overcome this obstacle, in this module we model the information of all attributes simultaneously. As illustrated in Section 1, mixing up all attributes like existing methods will omit useful features. So, we concatenate and project the attribute-centered item representations to preserve multiple attribute features. The process of obtaining the $i$th attribute-aware item representation $\mathbf{h_i} \in \mathbb{R}^d$ from $K$ attributes is symbolized as follows:

$$\mathbf{h_i} = \mathrm{MLP}([\mathbf{h_i^{a_1}}, \ldots, \mathbf{h_i^{a_K}}]). \quad (9)$$

Then, we join the item representations with the base query representation $\mathbf{q}$ to build the attribute-aware interest profile $\mathbf{I}$:

$$\mathbf{I} = \mathrm{Trm}^{\mathrm{last}}([\mathbf{h_i} + \mathbf{h_i^P}, \mathbf{q} + \mathbf{q^P}]). \quad (10)$$

Similarly, $\mathbf{h_i^P}$ and $\mathbf{q^P}$ are the positional embeddings associated with the item and query based on their positions in the search sequence. Note that we directly use the base query representations because we want to protect all clues of current intents. Besides, using base query representations to influence the profiling for attribute-aware interests would not face the same problem stated in 3.1.1 since item information is completely preserved.

## 3.3 Interest Fusion

So far, we have obtained $K$ attribute-centered interest profiles and one attribute-aware interest profile. With the guidance of contribution weights, we integrate the ranking scores as follows:

$$\begin{aligned} \mathrm{score}\,(q,H,c) = \mathrm{MLP}([&W^{a_1} s(\mathbf{I^{a_1}}, \mathbf{c^{a_1}}), \ldots, \\ &W^{a_k} s(\mathbf{I^{a_k}}, \mathbf{c^{a_k}}), s(\mathbf{I}, \mathbf{c})]). \end{aligned} \quad (11)$$

$s(\cdot)$ refers to the dot product similarity function. $\mathbf{c^{a_i}}$ is the attribute-centered candidate item representations. $\mathbf{c}$ the attribute-aware candidate item representations generated as Equation (9).

## 3.4 Model Optimization

Following previous methods (Ai et al., 2017, 2019; Bi et al., 2021), we optimize our model by maximizing the log-likelihood of the observed (candidate item, query, history) triples. The loss function can be formulated as:

$$\begin{aligned} \mathcal{L} &= \sum_{(q,H,c)} \mathcal{L}(q,H,c) \\ &= \sum_{(q,H,c)} (\log P\,(c|q,H) + \log P\,(q,\ H)) \\ &\approx \sum_{(q,H,c)} \log \frac{\exp(\mathrm{score}\,(q,H,c))}{\sum_{c' \in C} \exp(\mathrm{score}\,(q,H,c'))}, \end{aligned} \quad (12)$$

where $\log P\,(q,\ H)$ can be ignored for it is predefined as a uniform distribution. Similar to most methods (Ai et al., 2019, 2017, 2020), we adopt the negative sampling strategy (Le and Mikolov, 2014; Mikolov et al., 2013) to approximate the probability on large-scale data.

## 4 Experiment Setup

### 4.1 Datasets

We conduct extensive experiments on JDsearch dataset[1] (Liu et al., 2023)and Amazon dataset[2] to verify and analyze the functionalities of the proposed model. Three types of attributes, product name, product brand and product category, are chosen for experiments.

**JDsearch Dataset** The JDsearch dataset is a large-scale dataset collected for personalized product search from JD.com, a popular Chinese online shopping platform. Following Liu et al. (2023), we take the last behaviors issued on 2022-10-17 as the testing set and the ones before them as the training set.

**Amazon Dataset** We apply the Amazon dataset to the personalized product search task following existing works (Ai et al., 2019, 2020). We use the dense sub-datasets of the corpus where each user and each item has at least five associated reviews to collect sufficient information for user profiling. Since the Amazon datasets are categorized by the product's categories, we choose two sub-datasets that have multiple sub-categories, *CDs & Vinyl*, *Electronics*, to ensure our interest learning on the category attribute are fed with diverse preferences. We take the last search of the user history as the testing set, the former 20% as the validation set, and the rest as the training set. Queries are constructed from categories following existing works (Ai et al., 2017; Gysel et al., 2016). The top 100 items ranked by BM25 (Robertson and Zaragoza, 2009) according to all attribute text are taken as candidate items.

### 4.2 Model Settings and Evaluation metrics

The final parameters of the proposed model are set as follows: The embedding dimension is 128. The attribute-centered transformer encoder and the attribute-aware transformer encoder are 4 heads with 2 layers. The multi-head attention used for attribute-centered query representing is 2 heads. For the JDsearch dataset, the history length is 30, the short-term history length is 15, and the weighting window size noted as $P$ in Section 3.1.3 is 4. For the Amazon dataset, the history length is 10, the short-term history length is 5, and the weighting window size is 2. We compute MRR@200, Precision@1, and NDCG@10 for evaluation metrics to evaluate the models.

[1]JDsearch: `https://github.com/rucliujn/JDsearch`
[2]Amazon: `http://jmcauley.ucsd.edu/data/amazon/`

### 4.3 Baselines

We compare our model with ad-hoc models and personalized models listed as follows:

• **BM25** (Robertson and Zaragoza, 2009): It is a classical ad-hoc retrieval algorithm.

• **QEM** (Ai et al., 2019): It is an ad-hoc query embedding model, which gets ranking scores by matching items with the query.

• **HEM** (Ai et al., 2017): It learns the semantic representations for items and queries in latent space.

• **DREM** (Ai et al., 2020): It creates a dynamic knowledge graph based on search context and product metadata.

• **AEM, ZAM** (Ai et al., 2019): AEM is an attention-based embedding model representing users according to current queries. ZAM is an improvement of AEM, which introduces a zero vector in the attention process to conduct differentiated personalization.

• **TEM** (Bi et al., 2020): It dynamically controls the effects of personalization by encoding the user history and the query with transformers.

• **HGN** (Ai and Ramasamy, 2021): It builds knowledge graphs to explicitly construct user representations based on the user's purchase history.

We follow (Liu et al., 2023) to implement all models. For a fair comparison, we fed aspect-based baselines (DREM, HGN) with the same attribute information used in our model. For other baselines, we feed them with concatenated attribute words.

## 5 Results and Analysis

### 5.1 Overall Performance

The overall results on the three datasets are reported in Table 2. we find that:

(1) **Our model significantly outperforms all baseline models with paired t-test at p<0.05 level on every dataset.** Specifically, compared to the state-of-the-art model TEM, our model MAI improves the ranking results on the JDseach dataset by 4.31% in terms of MRR and 1.84% in terms of NDCG. These results verify that building multi-attribute interests is more effective in achieving personalized product search.

(2) Compared to other aspect-based models (i.e., DREM, HGN), our model achieves apparent improvements. Generally, personalized models show superiority over ad-hoc models, proving the necessity of learning interests from history. The poor results from some KG-based models on JDsearch

Table 2: Overall performance. The best results are shown in bold. '†' indicates the model significantly outperforms all baseline models with paired t-tests at p<0.05 level.

| Dataset | | JDsearch | | | *CDs & Vinyl* | | | *Electronics* | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | | MRR | Prec | NDCG | MRR | Prec | NDCG | MRR | Prec | NDCG |
| Ad-hoc | BM25 | 0.1114 | 0.0402 | 0.0940 | 0.01 | 0.0001 | 0.0001 | 0.0194 | 0.0096 | 0.0096 |
| | QEM | 0.1774 | 0.0728 | 0.1705 | 0.1953 | 0.1327 | 0.211 | 0.2409 | 0.1421 | 0.2659 |
| Person-alized | HEM | 0.1955 | 0.0847 | 0.1905 | 0.2896 | 0.2236 | 0.41 | 0.2000 | 0.1248 | 0.3448 |
| | DREM | 0.1647 | 0.0632 | 0.1578 | 0.2482 | 0.1549 | 0.3823 | 0.1807 | 0.0916 | 0.3316 |
| | HGN | 0.1662 | 0.0634 | 0.1591 | 0.2583 | 0.1734 | 0.3873 | 0.2096 | 0.1152 | 0.3373 |
| | AEM | 0.1971 | 0.0851 | 0.1920 | 0.2977 | 0.2227 | 0.3207 | 0.2571 | 0.1635 | 0.2890 |
| | ZAM | 0.1969 | 0.0849 | 0.1920 | 0.2828 | 0.2056 | 0.3022 | 0.2600 | 0.1716 | 0.2838 |
| | TEM | 0.2229 | 0.1049 | 0.2192 | 0.3558 | 0.2853 | 0.3734 | 0.2234 | 0.1302 | 0.2487 |
| Ours | MAI | **0.2672**$^{\dagger}$ | **0.1233**$^{\dagger}$ | **0.2778**$^{\dagger}$ | **0.3845**$^{\dagger}$ | **0.2864**$^{\dagger}$ | **0.4207**$^{\dagger}$ | **0.2871**$^{\dagger}$ | **0.1497**$^{\dagger}$ | **0.3483**$^{\dagger}$ |

Table 3: Ablation experiments on the JDsearch dataset.

| Model | MRR | Prec | NDCG |
|---|---|---|---|
| MAI | **0.2672** | **0.1233** | **0.2778** |
| w/o. AC | 0.1737 | 0.0680 | 0.1694 |
| w/o. AA | 0.2110 | 0.0969 | 0.2026 |
| w/o. QR | 0.2602 | 0.1177 | 0.2026 |
| w/o. CW | 0.2560 | 0.1135 | 0.2654 |

might be due to the dataset's characteristics, where the relationships between entities are too sparse to extract useful features. Thanks to parallel interest profiling and explicit weighting, our MAI overcomes this obstacle by successfully preserving and modeling attribute features.

(3) Compared to other transformer-based and attention-based models (i.e., TEM, AEM, ZAM), our model boosts the ranking results on each dataset. It is illustrated that these models yield the best results owing to these structures' excellent ability to capture latent features. Instead of simply applying the structures to represent items or encoding history sequences, we leverage the attention weights to reflect relevance.

## 5.2 Ablation Analysis

We test the functionalities of the four major components with several ablations models:

**MAI w/o. AC**. We abandon the attribute-centered profiling (AC) described in Section 3.1.

**MAI w/o. AA**. We delete the attribute-aware profiling (AA) part described in Section 3.2.

**MAI w/o. QR**. We substitute the attribute-centered query representations (QR) in Section 3.1.1 with base query representations.

**MAI w/o. CW**. We strip off the contribution weighting (CW) in Section 3.1.3.

As the results reported in Table 3, all the ablation models are inferior to the MAI model. Particularly, we can find that:

(1) The most significant performance drop is observed when removing the AC module. Without AC, the model simply aggregates all attribute information, similar to other aspect-based models. Similarly, it faces performance drops due to the same reason: the lack of relationships among products and attributes. With AC, the attribute features are clearly distinguished through separate profiling and explicit weighting.

(2) The "MAI w/o. AA" model also damages the results by 5.62% on MRR. This verifies the necessity of attending the correlations of attribute-aware features simultaneously. Without AA, the model blocks the information flow among different attributes within history, which happens for most users, leading to poor results of the "MAI w/o. AA".

(3) The "MAI w/o. QR" model causes the performance decline by 0.70% on MRR. This reveals that reformulating queries according to attributes helps enhance current intents in attribute-centered profiles.

(4) The apparent drops caused by "MAI w/o. CW" model proves our contribution weighting module helps the model determine the importance of attributes.

## 5.3 Case Study

In this section, we illustrate the functionalities of contribution weights with an example. As the JDsearch dataset only has anonymized term IDs, we use the category *Electronics* from the Amazon dataset. The attribute "category" weights are

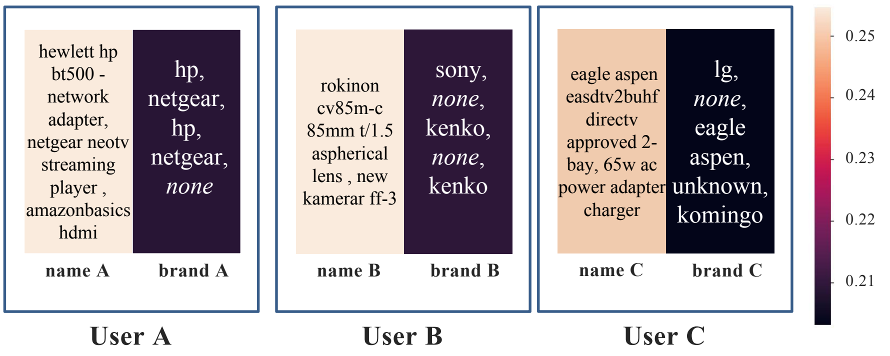

Figure 2: The contribution weights on attribute "name" and "brand" of users A, B, and C. A lighter area indicates a larger weight. Corresponding text of short-term items is shown in the frames. For the attribute "name", we list some terms from the long text. For "brand", we present all content. "," separates the text of different items, while "*none*" means the item does not have the corresponding attribute.

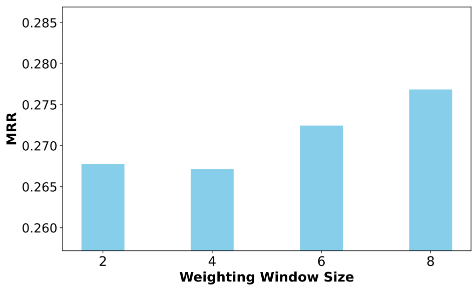

Figure 3: Results of different weighting window sizes.

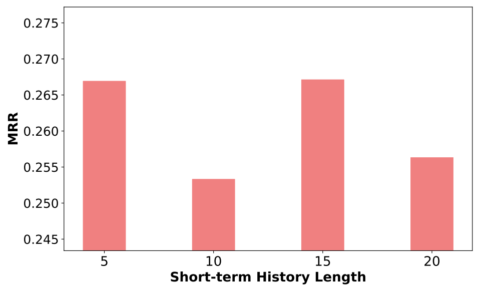

Figure 4: Results of different short-term history lengths.

much higher than other attributes since the user's purchased items are initially highly-related on category. So, we only present the weights of "name" and "brand" to show apparent changes.

As illustrated in Figure 2, we present the contribution weights from three users. It is observed that the weights of each attribute are limited in a particular range, which is caused by the MLP from Equation (8) that is learned from the whole training set. Despite the ranges based on a global view, the contribution weights could still adjust the attributes' importance from a personalized view. Take the "brand" attribute for example, its weights of user A and B are obviously higher than user C. This is because in user A and B, the short-term item shares more similarities with their neighboring items than in user C. This verifies that these weights successfully help the model determine the attributes' contributions according to the user's personal tastes.

### 5.4 Effects of Attribute-centered Interest Profiling

Now, we will explore the attribute-centered interest profiling by studying the impacts of weighting window size and short-term history lengths. The former is used for reflecting the attribute correlations, while the latter is used for extracting recent interests.

As shown in Figure 3, the results generally grow as the window size increases. It indicates our model's ability to leverage the attention weights from more items. From Figure 4, we can see that higher results are obtained at small and medium lengths. Perhaps it is because when increasing the length, the extracting becomes more challenging because the recent interests correlate with long-term interests. At a certain length, 15 in this case, the extracting may be efficient because of the successful distinguishing of the recent, long-term interests. But at a larger length, the extracting soon fails again with too much noise.

## 6 Conclusion

This work proposes a product search model that dynamically captures multi-attribute interests. In this model, we explore the potential of attribute features by modeling the user's preference on parallel profiling parts, where attribute interests are modeled independently and simultaneously. For each profiling part, we feed it with item/query representations that are enhanced by specific attributes accordingly. At the interest fusion stage, we use contribution weights obtained from profiling parts to help the model determine the importance of each attribute. Experiments demonstrate our model significantly outperforms existing models.

## Limitations

This work has several limitations. First, the performance drops obviously with a single attribute-centered profiling or attribute-aware profiling part. Although it is comprehensible as we explained in 5.2, it still indicates that both parts could be further improved. Take the attribute-aware profiling for example, mindlessly compressing all attribute information for all items neglects the fact that the user interests do not keep switching on all attributes at any time. A more efficient strategy could be designed to enhance or eliminate certain attribute features dynamically. We will leave this to our future work. Second, the contribution weights could not directly reflect the importance of corresponding attributes in the user's completed interests. It overlooks that the matching quality should influence the user's purchasing choices. For instance, if the user shows stable preferences on "brand", while in the current search, the product "name" perfectly matches her interests, using the contributing weights to reduce the impacts of "name" is problematic. Our work uses an MLP layer at the interest fusion stage to alleviate this problem. More efforts could be made to address this, and a more interpretable contribution weighting strategy could be designed.

## Acknowledgments

This work was supported by National Natural Science Foundation of China No. 62272467, and Public Computing Cloud, Renmin University of China. The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

## References

Qingyao Ai, Daniel N. Hill, S. V. N. Vishwanathan, and W. Bruce Croft. 2019. A zero attention model for personalized product search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 379–388. ACM.

Qingyao Ai and Lakshmi Narayanan Ramasamy. 2021. Model-agnostic vs. model-intrinsic interpretability for explainable product search. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 5–15. ACM.

Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W. Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 645–654. ACM.

Qingyao Ai, Yongfeng Zhang, Keping Bi, and W. Bruce Croft. 2020. Explainable product search with a dynamic relation embedding model. *ACM Trans. Inf. Syst.*, 38(1):4:1–4:29.

Paul N. Bennett, Ryen W. White, Wei Chu, Susan T. Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the impact of short- and long-term behavior on search personalization. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 185–194. ACM.

Keping Bi, Qingyao Ai, and W. Bruce Croft. 2020. A transformer-based embedding model for personalized product search. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 1521–1524. ACM.

Keping Bi, Qingyao Ai, and W. Bruce Croft. 2021. Learning a fine-grained review-based transformer model for personalized product search. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 123–132. ACM.

Keping Bi, Choon Hui Teo, Yesh Dattatreya, Vijai Mohan, and W. Bruce Croft. 2019. Leverage implicit feedback for context-aware product search. In *Proceedings of the SIGIR 2019 Workshop on eCommerce, co-located with the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, eCom@SIGIR 2019, Paris, France, July 25, 2019*, volume 2410 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan S. Kankanhalli. 2019. Attentive long short-term preference modeling for personalized product search. *ACM Trans. Inf. Syst.*, 37(2):19:1–19:27.

Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 165–174. ACM.

Akshay Jagatap, Srujana Merugu, and Prakash Mandayam Comar. 2024. Improving search for new product categories via synthetic query generation strategies. In *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 29–37. ACM.

Quoc V. Le and Tomás Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1188–1196. JMLR.org.

Soon Chong Johnson Lim, Ying Liu, and Wing Bun Lee. 2010. Multi-facet product information search and retrieval using semantically annotated product family ontology. *Inf. Process. Manag.*, 46(4):479–493.

Jiongnan Liu, Zhicheng Dou, Guoyu Tang, and Sulong Xu. 2023. Jdsearch: A personalized product search dataset with real queries and full interactions. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 2945–2952. ACM.

Jiongnan Liu, Zhicheng Dou, Qiannan Zhu, and Ji-Rong Wen. 2022. A category-aware multi-interest model for personalized product search. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 360–368. ACM.

Shang Liu, Wanli Gu, Gao Cong, and Fuzheng Zhang. 2020. Structural relationship representation learning with graph embedding for personalized product search. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 915–924. ACM.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *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*, pages 3111–3119.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Qijie Shen, Hong Wen, Jing Zhang, and Qi Rao. 2022. Hierarchically fusing long and short-term user interests for click-through rate prediction in product search. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, pages 1767–1776. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Chen Wu, Ming Yan, and Luo Si. 2017. Ensemble methods for personalized e-commerce search challenge at CIKM cup 2016. *CoRR*, abs/1708.04479.

Teng Xiao, Jiaxin Ren, Zaiqiao Meng, Huan Sun, and Shangsong Liang. 2019. Dynamic bayesian metric learning for personalized product search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pages 1693–1702. ACM.

Qiannan Zhu, Haobo Zhang, Qing He, and Zhicheng Dou. 2024. Query-aware explainable product search with reinforcement knowledge graph reasoning. *IEEE Trans. Knowl. Data Eng.*, 36(3):1260–1273.