# Multi-modal Preference Modeling for Product Search

Yangyang Guo
Shandong University
guoyang.eric@gmail.com

Zhiyong Cheng†
National University of Singapore
jason.zy.cheng@gmail.com

Liqiang Nie†
Shandong University
nieliqiang@gmail.com

Xin-Shun Xu
Shandong University
xuxinshun@sdu.edu.cn

Mohan Kankanhalli
National University of Singapore
mohan@comp.nus.edu.sg

## ABSTRACT

The visual preference of users for products has been largely ignored by the existing product search methods. In this work, we propose a multi-modal personalized product search method, which aims to search products which not only are relevant to the submitted textual query, but also match the user preferences from both textual and visual modalities. To achieve the goal, we first leverage the *also_view* and *buy_after_viewing* products to construct the visual and textual latent spaces, which are expected to preserve the visual similarity and semantic similarity of products, respectively. We then propose a translation-based search model (*TranSearch*) to 1) learn a multi-modal latent space based on the pre-trained visual and textual latent spaces; and 2) map the users, queries and products into this space for direct matching. The *TranSearch* model is trained based on a comparative learning strategy, such that the multi-modal latent space is oriented to personalized ranking in the training stage. Experiments have been conducted on real-world datasets to validate the effectiveness of our method. The results demonstrate that our method outperforms the state-of-the-art method by a large margin.

## CCS CONCEPTS

• **Information systems** → **Personalization**; *Query representation*; *Learning to rank*;

## KEYWORDS

Product Search, Personalization, Multi-modal Fusion

## ACM Reference Format:

Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Xin-Shun Xu, and Mohan Kankanhalli. 2018. Multi-modal Preference Modeling for Product Search. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10. 1145/3240508.3240541

† Corresponding Author.


*MM '18, October 22–26, 2018, Seoul, Republic of Korea*

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240541

## 1 INTRODUCTION

The ease of e-commerce has greatly changed the way of how people purchase products. With the convenience of online shopping, e-commerce users could find almost any product that they are interested in by just a few clicks. Typically, current product search engines (provided by e-commerce websites such as Amazon[1] and Taobao[2]) require users to formulate their shopping needs as a textual query (i.e., a few keywords), and then return a list of products ranked according to their relevance to the query. The returned results provide both the meta information of each product (e.g., brand, price, descriptions) and the visual appearance, as well as comments from other users who have already purchased the product.

In order to purchase a desired product, a prudent user will screen each product based on all the provided information carefully in the ranking list, which is time-consuming. To enhance the user experience and increase the user loyalty, it is important to rank the products, which are not only relevant to the given query, but also match the user preferences on different aspects (e.g., visual preferences and quality requirements) at the top positions. However, the design of such search engines is non-trivial, because 1) the queries are often too short or even ambiguous, and thus they cannot express user's needs precisely; and 2) even for the same query, the preferred products of users could be very different due to the distinct user preferences on different aspects. In light of this, considering the user preferences in product search or the *personalized product search*, plays a pivotal role in boosting product search performance.

Traditional approaches on product search [8, 9, 34] often focus on the simple matching between queries and products without leveraging the user preference. However, ignoring the user preference will lead to sub-optimal performance owing to the diverse user expectations. As pointed out in [1, 7, 33], the purchase behaviors in online shopping could be highly personal. Based on this observation, Ai et al. [1] introduced personalization into product search algorithm and extended the model in [34] by adding the user preference to the same latent space with queries and products (represented by textual reviews). Indeed, textual reviews disclose product's characteristics on some aspects and thus could reflect user's preferences on those aspects. Taking a *Clothing* product for example, users may comment on aspects such as *material*, *size*, *whether it is comfortable to wear* or *worth to buy*. However, other aspects which could be directly observed from the product images are seldom mentioned in the reviews, such as *style* and *color*. As a result, relying merely

[1] https://www.amazon.com/.
[2] https://world.taobao.com/.

on textual reviews can only capture the user's partial preferences while ignoring their visual preferences. For many products, such as *Clothing* and *Shoes*, the visual appearance of products plays an important role on user's purchase behaviors [12, 13, 25]. For example, a user who likes square collar *T-shirt* will not purchase a round collar one, even if the latter one matches his/her requirement on other aspects (e.g., brand, price, quality). Therefore, incorporating user's visual preference into personalized modeling can capture the user preferences more comprehensively, and thus could further enhance the product search performance.

In this paper, we aim to develop a personalized product search method, which considers the user preferences from both textual and visual modalities when ranking products. To design such a method, it involves two research problems: 1) How to model the multi-modal user preferences; and 2) Given a textual query $q$ of a user $u$, how to measure the relevance of a product $p$ with respect to the query $q$ and the user $u$'s multi-modal preferences. Note that solving those problems step by step is not an optimal solution. For example, a typical sequential method is: 1) representing products multi-modal features (with multi-modal feature fusion methods); 2) extracting the multi-modal user preferences based on their purchased products; and 3) measuring "the relevance between the query and a product (<query, product>)" and "the relevance between the user preference and a product <user, product>" separately, and then using a combination or re-ranking method to obtain the final product list.[3] The problem of the above sequential method is two-folds: on the one hand, the constructed multi-modal feature space using above method is not ranking oriented, resulting in sub-optimal ranking results; on the other hand, measuring the relevance of <query, product> and <user, product> separately cannot fully capture the complex interactions among <user, query, product>, as measuring the relevance of one pair (e.g., <query, product>) totally ignores the other one (e.g., <user, product>). Therefore, how to design a unified solution for constructing a multi-modal and ranking-oriented feature space and performing product ranking in this space is a challenging problem.

To solve the above problem, we proposed a Translation-based Product Search (dubbed as *TranSearch*) method. In our method, a user $u$'s preference, a textual query $q$, and a product $p$ are embedded into a latent multi-modal feature space and represented as vectors $\vec{\boldsymbol{u}}$, $\vec{\boldsymbol{q}}$, and $\vec{\boldsymbol{p}}$, respectively. In this space, our method translates the query $\vec{\boldsymbol{q}}$ with the user preference $\vec{\boldsymbol{u}}$ to be approximately equal to the final bought product $\vec{\boldsymbol{p}}$. The multi-modal feature space is initialized based on the textual and visual features of products, and then learned via the training process. To this end, our model can elegantly solve the personalized ranking problem that involves the complex interaction between <user, query, product>. Besides, the multi-modal feature space is also trained towards the personalized ranking. However, a practical problem is that in the training stage we need a large number of triplets <user, query, product>, which is usually very sparse. To tackle this problem, we adopt a pre-training strategy to first construct a visual and a textual feature spaces based on the rich data of *also_view* and *buy_after_viewing* [25]. The *also_view* and *buy_after_viewing* indicate the two products may be substitutable and closely related. Therefore, the constructed visual feature space is expected to preserve the visual similarity between products; and similarly, the textual feature space is expected to preserve the similarity between products from textual features. After the pre-training stage, in our *TranSearch* model , the two pre-trained feature spaces are fused and refined. At the same time, the users, queries, and products are mapped to this space for personalized product ranking.

To verify the effectiveness of our proposed model, we empirically evaluate *TranSearch* and compare its performance with state-of-the-art methods on the public Amazon product dataset[4]. Experimental results show that our approach can outperform the baselines significantly. Besides, we also quantitatively and qualitatively analyze the effectiveness of integrating textual and visual modality in preference modeling for product search.

In summary, the main contributions of this work are as follows:

- We propose a multi-modal translation-based method for personalized product search. To the best of our knowledge, this is the first work that models the user preferences from both textual and visual modalities for personalized product search.
- We propose a two-stage framework to solve the data sparsity problem in personalized product search. In the first stage, we leverage the rich data of *also_view* and *buy_after_viewing* to pre-train a product's textual and visual feature spaces; and then in the second state, we fuse the two feature spaces and refine the fused feature space based on a comparative learning method to achieve the better product ranking results.
- We have conducted extensive experiments on the real-world Amazon dataset to demonstrate the effectiveness of our proposed *TranSearch* model. Moreover, we released the codes and involved parameter settings to facilitate others to repeat this work[5].

## 2 RELATED WORK

### 2.1 Product Search

With the popularity of online shopping nowadays, the online product search attracts more and more attentions. Typically, the e-commerce product inventory information (e.g., product entity specifications) is structured and stored in relational databases. To fill the gap between the free-form keyword queries and the structured product entities, Duan et al. [8, 9] proposed a probabilistic retrieval model. The proposed method mines and analyzes the product search log data to solve the semantic mismatch between queries and structured product entities. In recent years, influenced by the development of online review platforms, researchers have attempted to extract information from the textual reviews to represent products by leveraging representation learning methods (e.g., word2vec [6, 26]). More specifically, Gysel et al. [34] introduced a latent semantic entity model to learn the distributed representations of words and entities (i.e., products and queries). A common problem of aforementioned methods is that they ignore the user's personal preference in product search. In fact, the user preference could be very different given the same query. Without personalization, the search engine will return the same list to distinct users,

[3]The combination method could be the combination of relevance score; the re-ranking method is to first rank the products based on one type of relevance and then re-ranking the top results based on the other type of relevance.

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

[5]https://github.com/guoyang9/TranSearch.

barely satisfying the diverse user needs. Some researchers have realized the importance of personalization in product search. For example, Duan et al. [7] and Su et al. [33] analyzed the user intent and satisfaction for product search. Recently, a personalized product search model has been proposed by Ai et al. [1]. They extended the model in [34] by mapping the user's personal preference into the same latent space with queries and products.

However, existing methods ignore the user's visual preference in product search. The importance of visual preference in product purchasing behaviors has been verified in previous works. For example, Macauley et al. [13, 25] considered the user's visual preference in recommender system to improve the product recommendation performance. Therefore, we believe that modeling the user's visual preference in product search could improve the search accuracy. Thus, in this paper, we propose a multi-modal preference modeling method for personalized product search, which has not been studied yet.

### 2.2 Multi-modal Deep Learning in IR

Deep learning techniques have been widely used in the multi-modal feature fusion [10, 16, 21–23, 27, 28, 37, 39, 40, 42] and recently been applied in the IR problems [5, 32, 43]. A comprehensive review of DL in multi-modal fusion and IR is out of the scope of this paper, we mainly focus on the deep multi-modal IR, which is most related to our work.

The key problem of deep multi-modal IR is to find an effective mapping mechanism to project data from different modalities into a common latent space and then match the query with the items in this latent space [36]. Previous methods (including deep cross-modal and intra-modal IRs) can be broadly categorized into hashing- and semantic-based ones. The former approaches [3, 4] map modalities in the original space to a Hamming space using hash functions, such that the distance between queries and items can be computed in this Hamming space. The latter [2, 18, 36] project the multi-modal data into a low-dimensional space by learning a mapping function, and then compute the semantic matching with the given query in this latent space. In [32] and [31], a fused multi-modal representation is generated from a learned probability density over the multi-modal space through Deep Boltzmann Machines and Deep Belief Networks, respectively. In [18], Laenen et al. introduced a multi-modal fashion search paradigm by using the e-commerce data. Nevertheless, in their problem setting, a query is comprised of keywords and images, which is inconsistent with current e-commerce product search scenarios that users typically provide only text queries. Besides, they have not considered the user preferences in their model.

## 3 PRELIMINARY

### 3.1 Research Problem

Given a textual query from a user, our target is to return a ranked list of products, which are ranked based on their relevance with respect to the query and the user preferences from both textual and visual modalities. In our problem setting, each product has its textual reviews and images. To achieve better search results, we embed users, their queries, and products (including visual and textual features) into the same latent space, so that the products could be directly matched with a given query and the corresponding user in this space. Therefore, we need to first preprocess the visual and textual product modalities, followed by the multi-modal feature fusion and user preference modeling, and finally a ranking mechanism. It is worth noting that the query and product textual modality are preprocessed in the same way in order to maintain their semantic relationship.

Before describing our proposed model, we first elaborate the textual and visual feature pre-processing.

### 3.2 Feature Preprocessing

**Textual Modality.** The PV-DM model [19] is adopted to extract textual vector representation for queries and products. PV-DM is an unsupervised method to learn continuous distributed vector representations for textual documents. It takes word sequence information into consideration and can preserve the semantic features of words. PV-DM takes text documents as inputs and outputs their vector representations in a latent semantic space. In our model, the textual modality of products is user reviews. And the queries only contain the textual modality. The textual modality of products and queries are mapped into the same latent space via PV-DM to learn their vector representations, which are then used as the inputs to our model.

**Visual Modality.** In general, each product has its own visual appearance and users have their unique tastes or preferences on the appearances of products, especially for the ones whose visual appearances are important features to attract customers, e.g., clothing. In this work, the dataset we adopted contains one image for each product and the visual features [25] of the image. To be specific, the visual features are extracted via the Caffe reference model [15]. The architecture has 5 convolutional layers followed by 3 fully connected layers, and has been pre-trained on 1.2 million ImageNet (ILSVRC2010) images. In this work, we take the output of the second fully connected layer, resulting in a 4096-D feature vector as the visual features for each product.

## 4 OUR PROPOSED METHOD

In order to return a personalized ranking list for a query from a user, we map the users, queries, and products into the same latent space, where the products could be directly matched with the queries and user preferences. However, directly training a model (as the one shown in Figure 2) may suffer from the data sparsity problem of <user, query, product> triplets. Insufficient data may hinder the learning of a desired feature space, resulting in inferior performance (as demonstrated in Section 6.2). To deal with the problem, we adopt a pre-training stage (shown in Figure 1), which leverages the abundant *also_viewed* and *buy_after_viewing* product data to first learn good feature spaces for visual and textual modalities. Then the learned feature space is used to extracted products' visual and textual features, which are used in our personalized product search model, called Translation-based product search (*TranSearch*) model. In our *TranSearch* model, the feature space will be refined towards personalized product ranking.

### 4.1 Feature Space Construction

To construct desired feature space to preserve the visual or semantic similarity between products, we take advantage of the rich data

of *also_viewed* and *buy_after_viewing* product sets[6], in which a product is paired with another product. In both sets, the paired products are similar either visually or semantically based on user's *also_viewed* or *buy_after_viewing* behaviors. We adopt the advanced deep autoencoder networks to learn the feature space, which has been proven to be effective in many latent space learning tasks [35].

There are two components in the autoencoder networks: the encoder and decoder [30]. The former can learn a new representation where an input can be reproduced by the latter. As shown in Figure 1, in our network, the inputs are $\boldsymbol{p}_a$, $\boldsymbol{p}_{a+}$ and $\boldsymbol{p}_{a-}$, among them, $\boldsymbol{p}_a$ and $\boldsymbol{p}_{a+}$ are the feature vectors of a paired products in *also_viewed* or *buy_after_viewing* product sets, $\boldsymbol{p}_{a-}$ is the feature vector of an irrelevant product with respect to $\boldsymbol{p}_a$. The autoencoder network is to construct a latent feature space, in which the relationship of $f_a$, $f_{a+}$, and $f_{a-}$ is preserved, namely:

$$d(f_a, f_{a+}) < d(f_a, f_{a-}), \tag{1}$$

where $d$ is a pair-wise distance function which can be cosine similarity, dot product, Euclidean and Manhattan distances. In our experiments, we observed that Euclidean distance yields relatively better performance. Accordingly, this autoencoder network is trained by minimizing the reconstruction errors with constraints of observed product relationships. The objective function is:

$$\begin{aligned}\mathcal{L}_{pre} = \sum [&max(0, \gamma_{pre} + d(f_a, f_{a+}) - d(f_a, f_{a-})) \\ &+\beta(\mathcal{L}(\boldsymbol{p}_a, \hat{\boldsymbol{p}}_a) + \mathcal{L}(\boldsymbol{p}_{a+}, \hat{\boldsymbol{p}}_{a+}) + \mathcal{L}(\boldsymbol{p}_{a-}, \hat{\boldsymbol{p}}_{a-}))] \\ &+\lambda_{pre}\|\boldsymbol{\Theta}_{pre}\|,\end{aligned} \tag{2}$$

where $\gamma_{pre}$ is the margin parameter that regularizes the gap between the squared Euclidean distance among the relevant products, irrelevant products and the anchor products, $\beta$ is the trade-off between the reconstruction error and relationship constraint error, $\lambda_{pre}$ is the $\ell_2$ regularization hyperparameter, and $\boldsymbol{\Theta}$ denotes all the parameters in our pre-training model. In this way, the decoder recovers the raw input $\boldsymbol{p}_a$ to $\hat{\boldsymbol{p}}_a$ to minimize the reconstruction error as follows,

$$\mathcal{L}_{rec}(\boldsymbol{p}_a, \hat{\boldsymbol{p}}_a) = \frac{1}{2}\|\boldsymbol{p}_a - \hat{\boldsymbol{p}}_a\|. \tag{3}$$

Without loss of generality, both the visual and textual latent spaces are constructed through this autoencoder framework and constrained to the same dimension $k$. After the feature space construction, we can obtain a semantic space learned from the textual reviews and a visual space learned from the product visual appearance. The two spaces could preserve the relevance of products from different perspectives. For example, the textual space may preserve the semantic similarity of two products based on their reviews, such as their materials and brands; and the visual space could preserve the visual similarity among products, such as style and color.

## 4.2 Translation-based Product Search

Personalized search needs to match the product's features and user preferences from both textual and visual modalities. Notice that matching them on each modality separately and then combing them ignores the interactions between the visual and textual features, which may lead to sub-optimal results. Besides, the query is only in the text modality, which cannot be directly matched with the products of two modalities. Therefore, we need to design a multi-modal fusion method and map both the user preferences and query into the same space. To achieve this goal, we propose a *TranSearch* model, comprised of three components: *Multi-modal Feature Fusion*, *Embedding*, and *Comparative Learning*. In the next, we will introduce each component in sequence.

[6]More details can be found on Section 5.1.

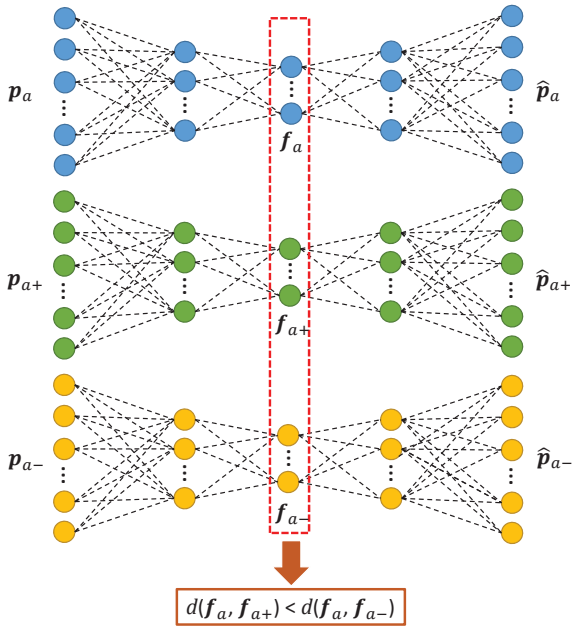


**Figure 1: The architecture of our pre-trained model.**

*4.2.1 Multi-modal Feature Fusion.* This component first concatenates the visual and textual features together, and then leverages a deep neural network (DNN) to fuse them in a complex non-linear way. The concatenation operation is:

$$\boldsymbol{c}_0 = [f^v; f^t], \tag{4}$$

where $f^v$ and $f^t$ are the visual and textual features obtained by our autoencoder model in Figure 1, respectively.

To model the interactions between the visual and textual features and obtain better fusion features, we refer to DNNs, which introduces multi-layer of non-linear interactions and has been proven to be very effective in the feature fusion tasks [41]. Specifically, the fully connected layers are:

$$\begin{aligned}\boldsymbol{c}_1 &= \phi(\boldsymbol{W}_1\boldsymbol{c}_0 + \boldsymbol{b}_1), \\ \boldsymbol{c}_2 &= \phi(\boldsymbol{W}_2\boldsymbol{c}_1 + \boldsymbol{b}_2), \\ &\ldots\ldots, \\ \boldsymbol{c}_L &= \phi(\boldsymbol{W}_L\boldsymbol{c}_{L-1} + \boldsymbol{b}_L),\end{aligned} \tag{5}$$

where $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$ denote the weight matrix and bias vector for the $l$-th fully connected layer, respectively. $\phi(\cdot)$ is the activation function, which could be sigmoid, hyperbolic tangent (tanh), rectified linear unit (ReLU) , leaky ReLU or exponential linear unit (ELU). In our experiment, we find that the ELU function can achieve better performance. To be more specific, our network structure follows a tower pattern, where the bottom layer is the widest and each

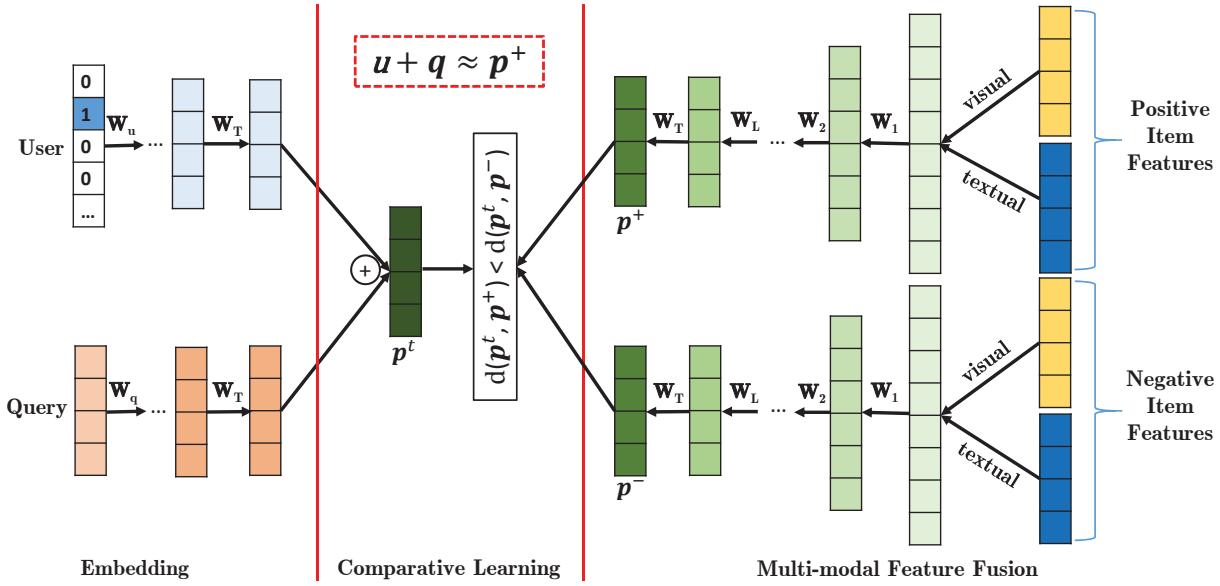


**Figure 2: The architecture of our proposed *TranSearch* model.**

successive layer has a smaller number of neurons [14]. Ultimately, the output from the last layer $\boldsymbol{c}_L$ has the dimension of $k$, equal to the visual and textual inputs.

After fusing the two features together, we then leverage another embedding matrix $\boldsymbol{W}_T \in \mathbb{R}^{k \times k}$ to map products into a latent space,

$$\boldsymbol{p} = \phi(\boldsymbol{W}_T \boldsymbol{c}_L + \boldsymbol{b}_T). \tag{6}$$

In this way, we can obtain the product representation $\boldsymbol{e}$.

*4.2.2 Embedding.* **User Embedding.** Each user is represented as a one-hot vector and then converted into a dense representation $\bar{u}$ via the embedding method. We use the same embedding matrix $\boldsymbol{W}_T$ with the products to translate the user into the same latent space with products,

$$\boldsymbol{u} = \phi(\boldsymbol{W}_T \bar{\boldsymbol{u}} + \boldsymbol{b}_T), \tag{7}$$

where $\bar{\boldsymbol{u}}$ is the dense embedding representation, $\phi$ is the activation function of ELU, and $\boldsymbol{u}$ is the final user preference representation.

**Query Embedding.** Similar to the user embedding, we first convert the query into $k$-dimensional, and then project it into the same latent space with the users and products,

$$\begin{aligned} \tilde{\boldsymbol{q}} &= \phi(\boldsymbol{W}_q \bar{\boldsymbol{q}} + \boldsymbol{b}_q), \\ &\cdots\cdots, \\ \boldsymbol{q} &= \phi(\boldsymbol{W}_T \tilde{\boldsymbol{q}} + \boldsymbol{b}_T), \end{aligned} \tag{8}$$

where $\bar{\boldsymbol{q}}$ is the query representation learned via PV-DM framework. It is transformed into $k$ dimensional $\tilde{\boldsymbol{q}}$ via $\boldsymbol{W}_q$. Finally, we can obtain the query representation $\boldsymbol{q}$ by $\boldsymbol{W}_T$. Notice that $\boldsymbol{u}$, $\boldsymbol{q}$, and $\boldsymbol{p}$ are embedded into the latent space using the same embedding matrix $\boldsymbol{W}_T$.

*4.2.3 Comparative Learning.* After embedding the users, queries and products into the same latent space, we then rank the candidate products according to an effective ranking mechanism. Let $\boldsymbol{p}^+$ and $\boldsymbol{p}^-$ be the feature vectors of a positive product and a negative product, respectively. The corresponding purchased product with respect to the query is regarded as the positive product. A negative product is randomly sampled from other non-purchased products. And then we argue that the positive product representation $\boldsymbol{p}^+$ should be closer than the negative one $\boldsymbol{p}^-$ to the desired product representation $\boldsymbol{p}^t$, where $\boldsymbol{p}^t$ is inferred by:

$$\boldsymbol{p}^t = \boldsymbol{u} + \boldsymbol{q}. \tag{9}$$

Our objective is to minimize a margin-based ranking criterion:

$$\mathcal{L} = \sum [max(0, \gamma + d(\boldsymbol{p}^t, \boldsymbol{p}^+) - d(\boldsymbol{p}^t, \boldsymbol{p}^-)] + \lambda \|\Theta\|, \tag{10}$$

where $\gamma$ is the margin parameter that regularizes the gap between the $d(\boldsymbol{p}^t, \boldsymbol{p}^+)$ and $d(\boldsymbol{p}^t, \boldsymbol{p}^-)$, $\lambda$ is the $\ell_2$ regularization hyperparameter, and $\Theta$ denotes all the parameters in our model.

To this end, our model learns user preference representation, query representation, and product representation based on a learning-to-rank framework, leading to better personalized product ranking. Besides, the fusion of multi-modal feature is refined to the ranking orientation simultaneously. In implementation, we train the model via the Adam [17] optimization method.

## 5 EXPERIMENTAL SETUP

### 5.1 Datasets

We experimented on the public Amazon product dataset. The dataset contains product reviews and metadata (e.g., product images) from Amazon. In our experiments, we adopted the 5-core version provided by McAuley et al. [24], whereby the remaining users and products both have at least 5 reviews. Besides, we selected four categories: *Office Products*, *Women's Clothing*, *Men's Clothing*, and *Toys & Games*. Following the strategy in [25], we extracted the user's product purchasing behaviors based on their reviews, i.e., the products they reviewed are the ones they purchased. The dataset provides four categories of relations: 1) *also_viewed* (e.g., users who viewed X also viewed Y); 2) *buy_after_viewing* (e.g., users who

**Table 1: Statistics over the four 5-core evaluation datasets. A_V represents also_viewed product pairs and B_A_V represents buy_after_viewing product pairs.**

| Datasets | #Users | #Products | #Queries | #Feedback | #<U, Q, P> Triplets | Av. Query Length | Av. Review Length | Av. (#A_V + #B_A_V) |
|---|---|---|---|---|---|---|---|---|
| Men's Clothing | 21,530 | 5,254 | 681 | 65,402 | 171,619 | 6.67 | 37.78 | 4.2 |
| Women's Clothing | 35,245 | 14,659 | 1,245 | 179,193 | 441,171 | 6.77 | 32.04 | 6.6 |
| Toys & Games | 19,412 | 11,838 | 396 | 166,600 | 168,298 | 6.93 | 89.63 | 19.5 |
| Office Products | 4,905 | 2,406 | 292 | 53,081 | 53,421 | 8.41 | 136.39 | 8.1 |

**Table 2: Performance comparisons between *TranSearch* and baselines over four Amazon datasets. Symbols † denotes the statistical significance with two-sided t-test of $p < 0.01$, compared to the best baseline. The best performance is highlighted in boldface.**

| | Men's Clothing | | | Women's Clothing | | | Toys & Games | | | Office Products | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Model | HR | MRR | NDCG | HR | MRR | NDCG | HR | MRR | NDCG | HR | MRR | NDCG |
| QL | 0.157 | 0.050 | 0.073 | 0.126 | 0.038 | 0.057 | 0.111 | 0.031 | 0.048 | 0.352 | 0.092 | 0.149 |
| UQL | 0.167 | 0.051 | 0.076 | 0.126 | 0.038 | 0.057 | 0.121 | 0.032 | 0.051 | 0.352 | 0.092 | 0.149 |
| LSE | 0.047 | 0.008 | 0.017 | 0.031 | 0.007 | 0.012 | 0.055 | 0.015 | 0.024 | 0.362 | 0.099 | 0.159 |
| HEM | 0.182 | 0.050 | 0.079 | 0.109 | 0.029 | 0.048 | 0.277 | 0.070 | 0.117 | 0.747 | 0.261 | 0.375 |
| TranSearch | **0.332**$^\dagger$ | **0.093**$^\dagger$ | **0.145**$^\dagger$ | **0.197**$^\dagger$ | **0.051**$^\dagger$ | **0.083**$^\dagger$ | **0.282**$^\dagger$ | **0.075** | **0.120**$^\dagger$ | **0.813**$^\dagger$ | **0.284**$^\dagger$ | **0.401**$^\dagger$ |

viewed X eventually bought Y); 3) *also_bought* (e.g., users who bought X also bought Y); and 4) *bought_together* (e.g., users bought X and Y simultaneously) [25]. Critically, categories 1 and 2 indicate that two products may be substitutable, namely, they are similar in some way (visually or semantically). According to Amazon's own technical report [20], the aforementioned relationships are collected by ranking products according to the cosine similarity of the sets of users who purchased/viewed them. Based on that, we extracted categories 1 and 2 to form the related product set for our pre-training model (in Figure 1). Besides, we removed the words with low frequency. The basic statistics of our used datasets are shown in Table 1.

## 5.2 Experimental Settings

**Query Extraction.** As Rowley described in [29], a typical scenario of the product searching is to use *a producer's name, a brand or a set of terms which describe the category of the product* as the query in retrieval. Based on this observation and following the strategy of [1, 34], for each product a user purchased, we extracted the corresponding search query from the categories to which the product belongs. The query extraction procedure is detailed as follows. We first obtained the category information for each product from its metadata. And then we concatenated the terms from a single hierarchy of categories to form a string. Finally, we removed the punctuation, stop words and duplicate words from this string. While eliminating the duplicate words, we maintained the terms from the sub-categories, since these terms carry more important information compared to their parent-categories. For example, in the dataset of *Women's Clothing*, the extracted query for *Women → Accessories → Sunglasses → Eyewear Accessories → Sunglasses* would be "*women eyewear accessories sunglasses*".

**Data Split.** We partitioned each of the four datasets into two sets: the training and testing sets. Specifically, we first constructed the user-product pairs from the user reviews, and then extracted the queries for these products. We finally obtained the valid user-query-product triplets. For each dataset, we randomly selected 80% user-query-product triplets of each user for training, and the remaining 20% for testing. For our *TranSearch* model, we trained it on the training set and reported the final results on the testing set. Note that in the testing stage, we have not used the reviews, because in real scenarios the reviews is unavailable before the purchase behaviors.

**Parameter Settings.** At the pre-training autoencoder phase, for each anchor product, a positive product is paired with 20 randomly sampled negative ones to learn a better feature space. The parameters are initialized by the *xavier* method [11] and the optimization method is Adam [17]. The number of layers for both encoder and decoder is 2.

We carefully tuned the dimension of the final latent vector from 16 to 512. In the training stage of *TranSearch*, the parameters are also initialized with the *xavier* and then optimized with the Adam. The learning rate is tuned in the range of [0.00001, 0.0001, 0.001, 0.01], regularizer is [0.000001, 0.00001, 0.0001, 0.001], and the margin parameter is [0.001, 0.01, 0.1, 1]. The number of negative samples for each positive training data is set to 5 and the batch size is 512.

**Evaluation Metrics.** We applied three standard metrics in evaluation: Hit Ratio (HR), Mean Reciprocal Rank (MRR), and Normalized Discounted Cumulative Gain (NDCG), where the first one indicates the percentage of queries which are hit correctly, while

the other two consider the position of positive items in the ranking list. Without special specification, we truncated the ranking list at 20 for all the three metrics.

### 5.3 Compared Baselines

We compared the proposed *TranSearch* model with different retrieval approaches from two categories: 1) traditional methods based on bag-of-words representations, such as *Query-likelihood Model* [38] and *Extended Query Likelihood with User Models* [1]; and 2) representation learning approaches based on the latent space modeling, such as *Latent Semantic Entity* [34] and *Hierarchical Embedding Model (HEM)* [1].

**Query-likelihood Model (QL).** It first estimates a language model for each document, and then ranks the documents by the likelihood of generating the query according to the estimated model [38].

**Extended Query Likelihood with User Models (UQL).** It was first introduced to personalized product search by Ai et al. [1]. Specifically, it extends QL by choosing the most frequent words[7] of reviews submitted by a user, then leverages a coefficient parameter to control the weights between the user and query.

**Latent Semantic Entity (LSE).** This method is specially designed for product search [34]. It projects words and products into the same latent space and learns a mapping function between them. Then the objective is to directly maximize the similarity between the products vector representation and $n$-gram's latent vector representation of its corresponding reviews.

**Hierarchical Embedding Model (HEM).** This model (HEM) proposed by [1] is the state-of-the-art approach for the personalized product search. It extends LSE [34] by adding the element of user preference to the product search. Similar to UQL, HEM also uses a coefficient to control the weight between the query model and the user model. HEM learns the distributed representations of queries, users and products by maximizing the likelihood of observed user-query-product triplets.

## 6 EXPERIMENTAL RESULTS

In this section, we report and analyze the experimental results. In particular, we focused on the following research questions:

- **RQ1:** Can our model outperform the state-of-the-art product search baselines?
- **RQ2:** Are the integration of the two modalities and the pre-training stage helpful to the final results?
- **RQ3:** How does the embedding size of users, queries and products affect the model performance?

### 6.1 Performance Comparison (RQ1)

Table 2 shows the performance of our *TranSearch* model and the baselines on four Amazon datasets. We also conducted pairwise significance test between our model and the baseline with the best performance. The key observations are as follows:

- Overall, our proposed method outperforms all the baselines across the four datasets consistently and significantly. The performance of *Office Products* is much better than the other three ones, which is mainly because the number of products and users in *Office Products* is smaller; The better performance of *Men's Clothing* over *Women's Clothing* maybe because the taste of women for clothing is more dynamic or complicated.
- For both traditional bag-of-words (i.e., QL and UQL) and state-of-the-art representation learning methods (i.e., LSE and HEM), personalized product search consistently surpasses the non-personalized ones. This demonstrates the importance of the personalization in the product search.
- On the strongly visual related datasets *Men's Clothing* and *Women's Clothing*, our method exceeds the state-of-the-art baseline HEM by a large margin. For example, the improvement of NDCG over *Men's Clothing* is 0.066 (84%), *Women's Clothing* is 0.035 (73%), while Toys & Games is 0.003 (3%) and Office Products is 0.026 (7%). This indicates that for those datasets where the user's visual preference is more important, incorporating the visual modality of products can boost the product search performance more greatly.

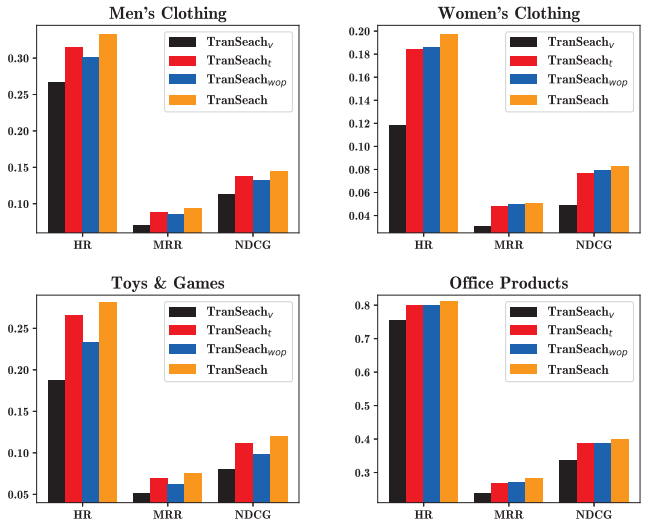


**Figure 3: Comparison of variants of our model.**

### 6.2 Model Ablation (RQ2)

To verify the effectiveness of the fusion of modalities and the pre-training, we compared our *TranSearch* model with three variants:

- **TranSearch$_t$:** It ignores the visual modality and only takes the textual modality into consideration.
- **TranSearch$_v$:** In contrast to **TranSearch$_t$**, it removes the textual modality in **TranSearch** and only considers the visual modality.
- **TranSearch$_{wop}$:** Instead of pre-training the visual and textual modalities, it takes the raw visual and textual features as inputs in *TranSearch* and trains on the user-query-product triplets data in an end-to-end way.

**Modality Comparison.** As we can see in Figure 3, the integration of visual and textual modalities outperforms the single modality ones over all the four datasets. In this experiment, *TranSearch$_t$* achieves better results than *TranSearch$_v$*. This indicates that even for strongly visual related datasets *Men's Clothing* and *Women's*

[7]Here, we define words appearing more than 50 times as the most frequent words.

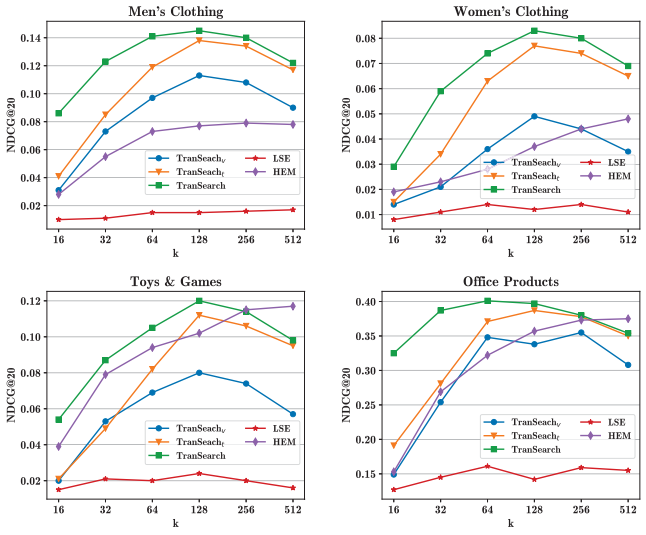


**Figure 4: The influence of latent embedding size.**

*Clothing*, the textual modality is still very important, since it provides users with crucial information such as price or material.

**Utility of Pre-training.** To demonstrate the utility of the pre-training for our *TranSearch* framework, we compared the performance of *TranSearch*$_{wop}$ and *TranSearch*. As shown in Figure 3, on all the four datasets, the *TranSearch* achieves superior performance, demonstrating the usefulness of the pre-training stage. An interesting observation is that on the datasets of *Men's Clothing* and *Toys & Games*, the *TranSearch*$_{wop}$ even performs worse than *TranSearch*$_t$. One possible reason is that the training samples are too sparse to train the model well, which further validates the importance of pre-training.

## 6.3 Influence of Embedding Size (RQ3)

To analyze the effect of the embedding size on the baselines of LSE and HEM, our model and its two variants, we show the results of these models with different embedding sizes over all datasets. Figure 4 shows the performance variation with the increase of the embedding sizes. It can be observed for all the methods, with the increase of the embedding size, the performance improves firstly, and then starts to deteriorate. Generally, the larger embedding size will lead to better representation capability, while it will result in over-fitting when the embedding size is too large. From the Figure 4, we can see that 128 is a good embedding size for our model.

## 6.4 Case Analysis (RQ4)

In this subsection, we will give some examples to illustrate the advantage of our method considering visual preference than the best baseline method HEM, which only models user preference from textual modality.

From Figure 5, we can observe that for each query submitted by a specific user (e.g., user1: *Women clothing active pants*), our method could return the desired product at a higher rank in the return list. Besides, all the top results returned by our method is relatively more relevant than HEM. For example, for the first query, *Women clothing active pants*, the HEM returns three top clothes


**Figure 5: Search results of four query examples on four distinct Amazon datasets. The first row represents different users with their example queries; the second row represents our model and the state-of-the-art baseline, followed by their top 5 results for the given query, where the products with the red box are the purchased ones.**

for the user who wants pants. The return of less relevant results at the top positions may result in the decline of trust towards the system. Besides, because of the consideration of the user's visual preference, the returned results of our method looks more visually similar. This indicates that our method could return the products with visual appearances preferred by the users, which may increase the user satisfaction for the product search engine.

## 7 CONCLUSION

In this paper, we proposed a translation-based personalized product search model *TranSearch*, which models the user preferences from both visual and textual modalities. Different from the previous works that mainly focus on the textual modality in product search, we argue that the visual preference of users also plays an important role in product selection. In particular, we adopted a pre-training stage to construct the visual and textual feature spaces for products firstly. In the next step, we proposed a translation-based comparative learning framework to refine the feature space and map the users, queries, and products into this space for the personalized product ranking. To the best of our knowledge, this is the first attempt of considering the visual preference in personalized product search. Comprehensive experiments have been conducted to verify the effectiveness of the proposed method.

## ACKNOWLEDGMENTS

This work is supported by the National Basic Research Program of China (973 Program), No.: 2015CB352502; National Natural Science Foundation of China, No.: 61772310, No.:61702300, and No.:61702302; the Project of Thousand Youth Talents 2016; the Tencent AI Lab Rhino-Bird Joint Research Program, No.:JR201805; and the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centre in Singapore Funding Initiative.

## REFERENCES


[1] Qingyao Ai, Yongfeng Zhang, Keping Bi, Xu Chen, and W Bruce Croft. 2017. Learning a hierarchical embedding model for personalized product search. In *SIGIR*. ACM, 645–654.

[2] Saeid Balaneshin-kordan and Alexander Kotov. 2018. Deep neural architecture for multi-Modal retrieval based on joint embedding space for text and images. In *WSDM*. ACM.

[3] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S Yu. 2016. Deep visual-semantic hashing for cross-modal retrieval. In *SIGKDD*. ACM, 1445–1454.

[4] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Qiang Yang. 2017. Transitive hashing network for heterogeneous multimedia retrieval. In *AAAI*. AAAI, 81–87.

[5] Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S Kankanhalli. 2018. Aˆ 3NCF: An adaptive aspect attention model for rating prediction. In *IJCAI*. Morgan Kaufmann, 3748–3754.

[6] Zhiyong Cheng, Jialie Shen, Lei Zhu, Mohan S Kankanhalli, and Liqiang Nie. 2017. Exploiting music play sequence for music recommendation. In *IJCAI*. Morgan Kaufmann, 3654–3660.

[7] Huizhong Duan and ChengXiang Zhai. 2015. Mining coordinated intent representation for entity search and recommendation. In *CIKM*. ACM, 333–342.

[8] Huizhong Duan, ChengXiang Zhai, Jinxing Cheng, and Abhishek Gattani. 2013. A probabilistic mixture model for mining and analyzing product search log. In *CIKM*. ACM, 2179–2188.

[9] Huizhong Duan, ChengXiang Zhai, Jinxing Cheng, and Abhishek Gattani. 2013. Supporting keyword search in product database: A probabilistic approach. *VLDB* 6, 14 (2013), 1786–1797.

[10] Golnoosh Farnadi, Jie Tang, Martine De Cock, and Marie-Francine Moens. 2018. User Profiling through Deep Multimodal Fusion. In *WSDM*. ACM.

[11] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.

[12] Ruining He, Chen Fang, Zhaowen Wang, and Julian McAuley. 2016. Vista: a visually, socially, and temporally-aware model for artistic recommendation. In *RecSys*. ACM, 309–316.

[13] Ruining He and Julian McAuley. 2016. VBPR: Visual bayesian personalized ranking from implicit feedback. In *AAAI*. AAAI, 144–150.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. ACM, 173–182.

[15] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *MM*. ACM, 675–678.

[16] Peiguang Jing, Yuting Su, Liqiang Nie, Xu Bai, Jing Liu, and Meng Wang. 2017. Low-rank multi-view embedding learning for micro-video popularity prediction. *TKDE* (2017).

[17] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[18] Katrien Laenen, Susana Zoghbi, and Marie-Francine Moens. 2018. Web search of fashion items with multimodal querying. In *WSDM*. ACM.

[19] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. ACM, 1188–1196.

[20] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[21] Meng Liu, Liqiang Nie, Meng Wang, and Baoquan Chen. 2017. Towards micro-video understanding by joint sequential-sparse modeling. In *MM*. ACM, 970–978.

[22] Meng Liu, Xiang Wang, Liqiang Nie, Xiangnan He, Baoquan Chen, and Tat-Seng Chua. 2018. Attentive moment retrieval in videos. In *SIGIR*. ACM, 15–24.

[23] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2014. Deep captioning with multimodal recurrent neural networks (m-rnn). In *arXiv preprint arXiv:1412.6632*.

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

[25] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*. ACM, 43–52.

[26] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781*.

[27] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *ICML*. ACM, 689–696.

[28] Liqiang Nie, Xiang Wang, Jianglong Zhang, Xiangnan He, Hanwang Zhang, Richang Hong, and Qi Tian. 2017. Enhancing Micro-video Understanding by Harnessing External Sounds. In *MM*. ACM, 1192–1200.

[29] Jennifer Rowley. 2000. Product search in e-shopping: a review and research propositions. *Journal of consumer marketing* 17, 1 (2000), 20–35.

[30] Xuemeng Song, Fuli Feng, Jinhuan Liu, Zekun Li, Liqiang Nie, and Jun Ma. 2017. NeuroStylist: Neural compatibility modeling for clothing matching. In *MM*. ACM, 753–761.

[31] Nitish Srivastava and Ruslan Salakhutdinov. 2012. Learning representations for multimodal data with deep belief nets. In *ICML workshop*, Vol. 79. ACM.

[32] Nitish Srivastava and Ruslan R Salakhutdinov. 2012. Multimodal learning with deep boltzmann machines. In *NIPS*. MIT Press, 2222–2230.

[33] Ning Su, Jiyin He, Yiqun Liu, Min Zhang, and Shaoping Ma. 2018. User intent, behaviour, and perceived satisfaction in product search. In *WSDM*. ACM, 547–555.

[34] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *CIKM*. ACM, 165–174.

[35] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *SIGKDD*. ACM, 1225–1234.

[36] Wei Wang, Xiaoyan Yang, Beng Chin Ooi, Dongxiang Zhang, and Yueting Zhuang. 2016. Effective deep learning-based multi-modal retrieval. *The VLDB Journal* 25, 1 (2016), 79–101.

[37] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. 2015. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *MM*. ACM, 461–470.

[38] Chengxiang Zhai and John Lafferty. 2004. A study of smoothing methods for language models applied to information retrieval. *TOIS* 22, 2 (2004), 179–214.

[39] Hanwang Zhang, Zawlin Kyaw, Shih-Fu Chang, and Tat-Seng Chua. 2017. Visual translation embedding network for visual relation detection. In *CVPR*. IEEE, 3107–3115.

[40] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. 2018. Grounding referring expressions in images by variational context. In *CVPR*. IEEE, 4158–4166.

[41] Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, and Tat-Seng Chua. 2014. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In *MM*. ACM, 187–196.

[42] Hanwang Zhang, Zheng-Jun Zha, Yang Yang, Shuicheng Yan, Yue Gao, and Tat-Seng Chua. 2013. Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval. In *MM*. ACM, 33–42.

[43] Yongfeng Zhang, Qingyao Ai, Xu Chen, and W Bruce Croft. 2017. Joint representation learning for top-n recommendation with heterogeneous information sources. In *CIKM*. ACM, 1449–1458.