Pretraining Representations of Multi-modal Multi-query E-commerce Search

Xinyi Liu*
Xiamen University
Xiamen, China

Wanxian Guan
Alibaba Group
Hangzhou, China

Lianyun Li*
Xiamen University
Xiamen, China

Hui Li
Xiamen University
Xiamen, China

Chen Lin†
Xiamen University
Xiamen, China

Xubin Li†
Alibaba Group
Hangzhou, China

Si Chen
Alibaba Group
Hangzhou, China

Jian Xu
Alibaba Group
Hangzhou, China

Hongbo Deng
Alibaba Group
Hangzhou, China

Bo Zheng
Alibaba Group
Hangzhou, China

ABSTRACT

The importance of modeling contextual information within a search session has been widely acknowledged. However, learning representations of multi-query multi-modal (MM) search, in which Mobile Taobao users repeatedly submit textual and visual queries, remains unexplored in literature. Previous work which learns task-specific representations of textual query sessions fails to capture diverse query types and correlations in MM search sessions. This paper presents to represent MM search sessions by heterogeneous graph neural network (HGN). A multi-view contrastive learning framework is proposed to pretrain the HGN, with two views to model different intra-query, inter-query, and inter-modality information diffusion in MM search. Extensive experiments demonstrate that, the pretrained session representation can benefit state-of-the-art baselines on various downstream tasks, such as personalized click prediction, query suggestion, and intent classification.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

E-commerce search, query pretraining , graph contrastive learning

ACM Reference Format:
Xinyi Liu, Wanxian Guan, Lianyun Li, Hui Li, Chen Lin, Xubin Li, Si Chen, Jian Xu, Hongbo Deng, and Bo Zheng. 2022. Pretraining Representations of Multi-modal Multi-query E-commerce Search. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539200

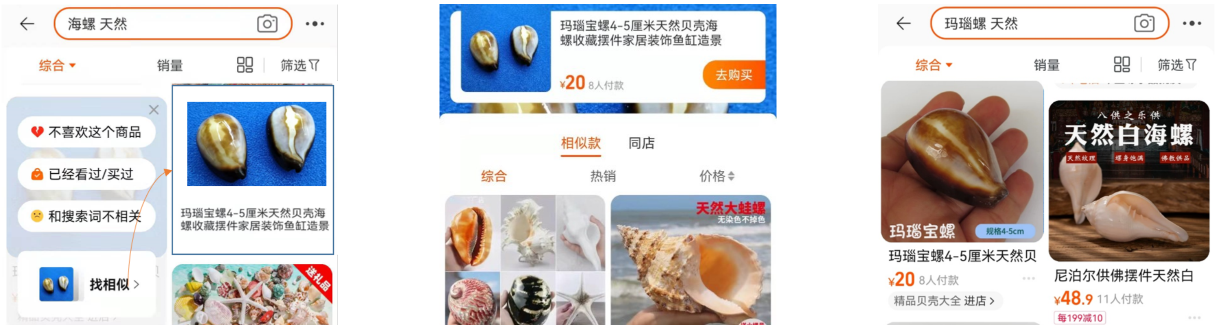
1 INTRODUCTION

In-site Product search [7, 21] has become an indispensable component in most E-commerce platforms. To provide better search experience, Mobile Taobao, one of China's most popular E-commerce apps, has deployed an important feature to allow consumers to search in different ways. Consumers can use the conventional way of typing textual *keyword queries*. When the results of keyword queries return, they can also click on one of the resulting product and "search similar products". A short piece of product descriptions extracted from the product title (i.e., *product query*) and an product image (i.e.,*image query*) will be input queries to the search system. The two types of queries can be repeated in a search session, generating an **M**ulti-modal **M**ulti-query search (**MM search**) session, as illustrated in Figure 1.

MM search plays a crucial role in Mobile Taobao. Currently, "search similar product" is a popular feature embraced by millions of daily active users. MM search is contributing an increasing part to the overall search traffic. Compared with conventional, purely textual search (i.e., a user only types in textual keyword queries and never uses "search similar product" in a search session), MM search provides an opportunity to encourage user engagements. As shown in Figure 2 (a), more than 55% conventional textual search sessions contain only one query, while MM search sessions tend to be longer, 9% sessions even contain more than ten queries. On the other hand, MM search reflect E-commerce users' complex search behaviors, whose information needs are more difficult to satisfy. As shown in Figure 2 (b), in all major product categories, number of clicks per query for MM search is significantly smaller than pure

[*] Xinyi Liu and Lianyun Li contribute equally.
[†] Chen Lin (chenlin@xmu.edu.cn) and Xubin Li (lxb204722@alibaba-inc.com) are the corresponding authors. This work is supported by the Natural Science Foundation of China (No. 61972328), and Alibaba Group through Alibaba Innovative Research program.

KDD '22, August 14–18, 2022, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539200



(a) Textual query “seashell natural"

(b) Image query and product query

(c) Refined query “cowrie shell natural"

Figure 1: A typical MM search session contains multiple occurrences of textual and visual queries. (a) The user first types in a keyword query containing two terms “seashell natural". He/she clicks on one of the preferred results and searches similar product. The product image and product descriptions extracted from product title, which are fed to the search system as the second query, are shown in the blue box. (b) Screenshot of the second query (i.e., the image query and product query) and search results. (c) Based on previous results, the user inputs a refine keyword query “cowrie shell natural".

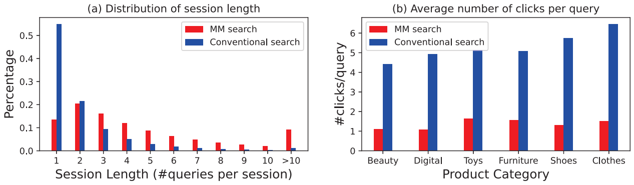


Figure 2: Characteristics of MM search v.s. conventional textual search. (a) Percentages of search sessions w.r.t. session length. (b) Average number of clicks per query

textual search, implying that the conversion rate of MM search remains a performance bottleneck in Mobile Taobao.

In the literature, learning representations of search sessions can be used for many downstream E-commerce tasks, including query performance prediction [19, 34], intent classification [11, 30, 36], query suggestion [5, 8, 24, 37], click prediction [3, 11, 12, 17, 35, 36], to name a few. Existing works are task-oriented, i.e., they encode information of the search sessions explicitly for the task. Furthermore, they focus on multi-query sessions of purely textual queries [3, 5, 8, 11, 12, 17, 24, 35–37], or multi-modality query in single-query sessions [21]. A question naturally arises, *can we enhance our understanding of MM search by pre-training to benefit various downstream tasks?*

Learning pre-trained representations for MM search is non-trivial, due to the following two intrinsic challenges.

C1: how to represent diverse query types and correlations in MM search sessions. The query keywords, product descriptive terms, and images in an MM search session show rich semantics and complex intra-query, inter-query, and inter-modality correlations. For example, as shown in Figure 1, a keyword query describes search intent in seashell species, while an image query, complementing to previous keywords, describes search intent in pattern and color. Most existing work, despite the end-to-end infrastructure, adopts sequential model at the query/session level. Thus, they are not able to represent diverse query types and complex correlations. To the best of our knowledge, representing a multi-modal and multi-query search session has never been explored in literature.

C2: how to design an informative pre-training task. Inspired by recent breakthrough in contrastive learning, we adopt contrastive loss to learn representations for MM search sessions. The effectiveness of contrastive loss relies on informative and feasible positive and negative samples. However, most existing contrastive learning methods are designed for language and visual domains [31, 38]. The positive and negative samples are sub-optimal for MM search. Furthermore, a search session can be interpreted from different views, while important semantics are shared across views. For example, the search session in Figure 1 can be interpreted from the view of how the user refines queries, or from the view of how different types of queries in the whole session complement each other. Considering only one interpretation can not fully encode the information in MM search sessions.

Our solution is a MUlti-View COntrastive Graph neural network framework (MUVCOG). To address challenge C1, we model an MM search session as a **H**eterogeneous **G**raph, and adopt Graph Neural Networks (GNNs) to encode diverse query types and interactions. Unlike other heterogeneous graph neural networks, we present two views to interpret MM search sessions. The first view aggregates globally correlated queries from different modalities. The second view interprets the session in a hierarchical manner and captures local correlations at query level. Thus, the graph level representation derived by these views capture complex local and global interactions among various query types of MM sessions. To address challenge C2, we present strategies to generate positive samples and hard negatives for MM sessions. Hard negatives enhance the difficulty of contrastive task. Then, the representations are contrasted across two views in pretraining, so that the learned representations are more effective and robust. We verify the effectiveness of the learned representations of MM sessions in various downstream tasks, including personalized click prediction, query suggestion, and intent classification.

In summary, our contributions are three-fold. (1) We study the novel problem of pre-training for Multi-modal Multi-query search in Mobile Taobao. We release the first public dataset of Multi-modal Multi-query search. (2) We present an effective contrastive pretraining framework MUVCOG to learn representations of MM search sessions based on multi-view heterogeneous graph networks. (3) We conduct extensive experiments to show that the pretrained representations of MM sessions benefit various downstream tasks, including personalized click prediction, query suggestion, and intent classification.

2 RELATED WORK

We identify two lines of related work: E-commerce search and contrastive pretraining.

2.1 E-commerce Search

Click prediction. Click prediction has received a lot of attention from both industry and academy. Nonetheless, multi-query search sessions are less extensively studied than single query search sessions. Previous work [3, 11, 12, 17, 35, 36] reveals that contextual information in a search session is helpful to predict clicked items. Most of these frameworks [11, 12, 36] adopt sequential models such as gated recurrent unit network (GRU) and attention mechanisms to encode a query based on preceding queries. Recurrent neural network (RNN) and attention are also adopted to represent inter query preference in cross-modal search, e.g., for image search based on multiple textual queries [32]. A few recent research employs Graph Neural Network (GNN). However, the graph is built beyond session-level and involves user feedback [4, 35]. Since click signals depend on items which are displayed, involving user feedback in the graph faces the risk of confusing GNN with noisy and incomplete user feedback. Previous work ignores contextual information for multi-modal queries. Single query which consists of an image of a product together with text to alter or add certain product attributes is studied in [21].

Query suggestion. When users are struggling to reform queries, search engines can provide assistance by predicting their next query, based on the contextual information or/and personal preference [37]. Again, sequential models and attention mechanism are heavily adopted in encoding session information. For example, hierarchical RNN in HRED [24], bidirectional RNN with attention in RIN [18], combined RNN in user-level and session-level in AH-NQS [5], GRU with feedback memories (i.e., search results and the clicked positions) in FMN [29], etc.. Pure attention models are presented, such as the multi-head self attention in HSCM [3], the flat-Transformer and hierarchical-Transformer [22] and the hierarchical attention mechanism and a copy network in ACG [8]. There is a trend of jointly learning for click prediction and query suggestion [1, 3, 6], where representation of the search session is shared to improve performance for multiple tasks.

Other tasks. Many other applications have been proposed. For example, session length prediction [14], user reformulation prediction [16] (i.e., whether users will reformulate their query before presenting the search results), user intent classification [25] (i.e., whether users explore in a search session or browses with strong purchase intent), query performance prediction [34], and multi-query evaluation [19]. In these studies, pre-retrieval features (i.e., features that can be obtained before the results and feedback are available) are preferred than post-retrieval features (e.g., user clicks), because pre-retrieval features can facilitate real-time prediction [16].

Summary. Representing contextual information in search sessions is important for a variety of tasks. However, representation learning for search sessions containing multiple textual and image queries has never been studied in these tasks. A recent large-scale query log analysis [7] is the only known study related to multi-query multi-modality search sessions. It points out that multi-modal search sessions are very different from textual search sessions and require special treatments.

2.2 Contrastive Pretraining

Self-supervised Pretraining. A major and ongoing thrust of research recently is on self-supervised pretraining, which automatically learns task-independent, useful features without expensive annotations. Pretraining-finetuning has become the predominant approach to achieve state-of-the-art results in NLP (natural language processing) and CV (computer vision) tasks. In NLP, popular pretraining tasks include masked token prediction (e.g., BERT [9]) and auto-regressive language modeling (e.g., GPT [2]), where the typical architecture is based on Transformer [26]. It is found that neither of the above objectives can produce better sentence embeddings than contrastive pretraining tasks [10]. In CV, both masked autoencoders [15] and contrastive learning [31], with different architectures, obtain promising results.

Contrastive learning. For any anchor data point, the contrastive loss pulls closer some positive points and push apart some negative points in the feature space. For graph data, positive data is obtained by a number of graph augmentations and negative data is usually randomly sampled [23, 33]. Cross-view contrastive learning which contrasts representation from one view to representation from another view is presented to represent heterogeneous graph [28].

Summary. Existing contrastive learning methods for graph data are sub-optimal for MM search pretraining. (1) They can not generate high quality positive and negative samples for MM search. (2) The contrastive task is not effective for MM search sessions and the discriminating power is weaken.

3 METHODOLOGY

3.1 Preliminaries

An MM (Multi-Modal Multi-query) search session, like a conventional search session, is a series of consecutive queries input by the same user, where the idle time between two adjacent queries is limited by a threshold. To represent an MM search session, it is important to capture different query types and complex interactions. Firstly, an MM search session contains three query types: keyword query, product query and image query. The former two query types are textual queries, i.e., keyword queries and product queries both consist of a set of words from the vocabulary. To distinguish different query types and capture complex correlations, we model each MM search session as a Heterogeneous Graph.

Formally, a heterogeneous graph for an MM search session s is a directed, labeled graph $G^s = \{\mathcal{V}^s, \mathcal{E}^s, Q, \mathcal{A}, \phi^s, \psi^s\}$, where $\mathcal{V}^s = \{\mathbf{v}_1^s, \cdots, \mathbf{v}_{N^s}^s\}$ is a set of nodes, each node $\mathbf{v}_i^s \in \mathbb{R}^D$ is represented

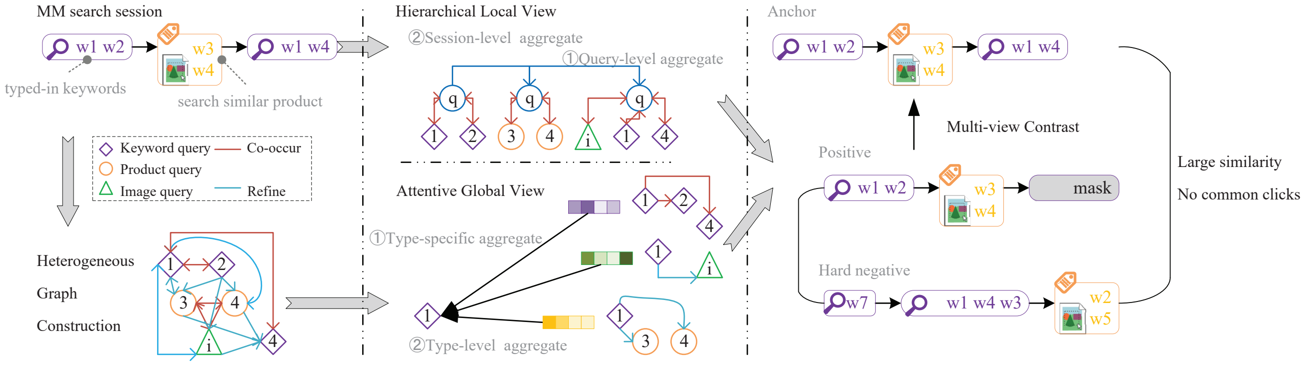


Figure 3: Overall framework of MUlti-View COntrastive Graph neural network (MUVCOG)

by a D−dimensional embedding vector. $\mathcal{E}^s = \{e_{i,j} | i \in \mathcal{V}^s, j \in \mathcal{V}^s\}$ is a set of edges. $\mathcal{Q}$ is the set of distinct node types and $\phi^s : \mathcal{V}^s \rightarrow \mathcal{Q}$ is the node mapping function. $\mathcal{A}$ is the set of distinct edge types and $\psi^s : \mathcal{E}^s \rightarrow \mathcal{A}$ is the edge mapping function.

Next, we design two views to construct heterogeneous graph for MM search sessions and learn node embeddings by heterogeneous graph neural network.

3.2 AGV: Attentive Global View

Under attentive global view, we construct G^s with $\mathcal{Q} = \{KQ, PQ, IQ\}$, which corresponds to keyword query, product query and image query, respectively. For keyword queries and product queries, we add a node for each distinct word in the graph, since words are the basic units of a query. Note that we treat words in keyword queries and product queries differently. For example, word ‘w4" in Figure 3 appears in both a keyword query and a product query, we add two nodes, one is assigned with node type KQ and the other is with PQ. The motivation is that, keyword queries are manually input by users, while product queries are extracted from product descriptions, thus they are distinguished to reflect different user intents. For each distinct image query, we add a node and assigne it with type IQ. We consider two edge types $\mathcal{A} = \{CO, RF\}$, corresponding to co-occur and refine relations, respectively. If two words i, j co-occur in the same keyword query or product query, we add two edges $e_{i,j}, e_{j,i}$ and assign the edge type $\psi(e_{i,j}) = \psi(e_{j,i}) = CO$. For any pair of nodes i, j, if node i appears in a query and node j appears in an immediate subsequent query, we add a refine edge $e_{i,j}, \psi(e_{i,j}) = RF$.

We initialize the node embeddings by pretrained word embeddings and image embeddings. Note that since visual queries and textual queries lie in different feature spaces, we apply a projection function in the initialization to transform visual nodes into the textual space, i.e., $\mathbf{v}_i = \sigma(\mathbf{W}^e\tilde{\mathbf{v}}_i + \mathbf{b})$, where $\tilde{\mathbf{v}}_i$ is the initialized image embedding for image i.

In updating the node embeddings, intuitively, contributions of neighboring nodes depend on the query types and the semantics of nodes. Inspired by [28], we employ attention mechanism to first aggregate information from different nodes with the same type, and then aggregate information from different query types. Specifically, we update the hidden state vector of node i w.r.t to type p by:

$$\mathbf{v}_i^p = \sigma(\sum_{j\in\mathcal{N}_i,\phi(j)=p} \alpha_{i,j}^p \mathbf{v}_j), \tag{1}$$

where $p \in \{KQ, PQ, IQ\}$ is the query type, $\mathcal{N}_i$ is the neighbor set, i.e., $\forall j \in \mathcal{N}_i$, there exists an edge $e_{j,i}$. σ is the nonlinear activation function, $\alpha_{i,j}^p$ is the type-specific attention, computed by:

$$\alpha_{i,j}^p = \frac{exp\Big(\sigma(\mathbf{W}^a[\mathbf{v}_i, \mathbf{v}_j])\Big)}{\sum_{l\in\mathcal{N}_i,\phi(l)=p} exp\Big(\sigma(\mathbf{W}^a[\mathbf{v}_i, \mathbf{v}_l])\Big)}, \tag{2}$$

where [] is the concatenation operation.

Then, we utilize attention to aggregate different types, to obtain the node embedding under AGV.

$$\mathbf{z}_i^{AGV} = \sum_p \beta^p \mathbf{v}_i^p \tag{3}$$

where $\mathbf{z}_i^{AGV}$ is the node embedding for node i under the attentive global view. The computation of type-specific attention involves *globally* information diffusion from nodes across queries. β^p is the weight of each query type p.

$$\beta^p = \frac{exp \sum_{\mathbf{v}_i\in\mathcal{V}} \sigma(\mathbf{W}^g\mathbf{v}_i^p + \mathbf{b}^g)}{\sum_p exp \sum_{\mathbf{v}_i\in\mathcal{V}} \sigma(\mathbf{W}^g\mathbf{v}_i^p + \mathbf{b}^g)} \tag{4}$$

Finally, the session representation under the attentive global view is computed by :

$$\mathbf{g}_s^{AGV} = Avg(\mathbf{z}_i^{AGV} | i \in \mathcal{V}^s), \tag{5}$$

where Avg is the mean pooling operation, $i \in \mathcal{V}^s$ means the session representation is averaged over all nodes (including words and images) in the session.

3.3 HLV: Hierarchical Local View

Different from the attentive global view, the assumption of hierarchical local view is to first capture intra-query dependencies and then capture inter-query dependencies in a hierarchical manner. Thus, as shown in Figure 3, we construct G^s with four node types, $\mathcal{Q} = \{KQ, PQ, IQ, VQ\}$, where VQ is a “virtual" word which corresponds to the aggregated query. Thus, the HLV contains more nodes than the AGV graph. HLV also has two edge types $\mathcal{A} = \{CO, RF\}$. We add CO edges between a VQ node and all of its query words/images. We add a RF edge from a VQ node to its subsequent query's VQ node.

In initialization, we first initialize KQ, PQ, IQ nodes with the corresponding pretrained word embeddings and image embeddings. We use a linear transformation to word and image embeddings $\mathbf{v}_i, \phi(i) \in \{KQ, IQ, PQ\}$ and obtain a more expressive hidden state $\mathbf{h}_i = \mathbf{W}^b \mathbf{v}_i$, where $\mathbf{W}^b$ is a learnable weight matrix. We initialize VQ nodes as zero vectors.

Then, we apply GCN [20] hierarchically. First we obtain the node embedding of a VQ node $\mathbf{h}_q, \phi(q) = VQ$ by query-level aggregation:

$$\mathbf{h}_q = \sum_{\psi(e_{j,q})=CO} \sigma\big(\frac{1}{\sqrt{d_j d_q}} \mathbf{W}^c \mathbf{h}_j\big), \tag{6}$$

where $\sum_{\psi(e_{j,q})=CO}$ means the information is aggregated over query words and images that belong to the current query q, d_j is the degree of node j, $\mathbf{W}^c$ is the learnable weight matrix.

Then we update the node embedding by inter-query aggregation:

$$\mathbf{z}_q^{HLV} = \sum_{\psi(e_{q',q})=RF} \sigma\big(\frac{1}{\sqrt{d_{q'} d_q}} \mathbf{W}^d \mathbf{h}_{q'}\big), \tag{7}$$

where $\sum_{\psi(e_{q',q})=RF}$ means the information is aggregated over a receptive area of the query sequence, $\mathbf{W}^d$ is the learnable weight matrix.

Finally we obtain the session representation under hierarchical local view by averaging all VQ node embeddings:

$$\mathbf{g}_s^{HLV} = Avg\big(\mathbf{z}_q^{HLV} | q \in \mathcal{V}^s, \phi(q) = VQ\big) \tag{8}$$

3.4 Multi-view Contrast

With the two view-specific session representation, we employ a contrastive learning across these two views. The motivation is, for each anchor session s, we want its similar session (i.e., positive sample) to be close in the representation space, and the dissimilar session (i.e., negative sample) to be far-apart, even under a different view. Thus, we employ a Multi-Layer Perceptron (MLP) to predict the label (i.e., positive or negative) of a pair of sessions (s, s'). The MLP takes the concatenation of $[\mathbf{g}_s, \mathbf{g}_{s'}]$ as input, and adopts l layers of transformation, $MLP(\mathbf{g}_s, \mathbf{g}_{s'}) = \sigma\Big(f^l\big(\cdots f^1([\mathbf{g}_s, \mathbf{g}_{s'}])\cdots\big)\Big)$, where in each layer $f^l(\mathbf{x}) = ReLU\big(\mathbf{W}_{MLP}^l \mathbf{x} + b_{MLP}^l\big)$. Then we optimize the binary cross entropy loss:

$$L = \sum_{<s,p,n>\in\mathcal{B}} BCE(\mathbf{g}_s^{AGV}, \mathbf{g}_p^{HLV}, \mathbf{g}_n^{HLV}) + BCE(\mathbf{g}_s^{HLV}, \mathbf{g}_p^{AGV}, \mathbf{g}_n^{AGV}) \tag{9}$$

where $BCE(i, j, k) = \big[log(MLP(i, j)) + log\big(1 - MLP(i, k)\big)\big]$ is the binary cross entropy loss, $\mathcal{B}$ is a mini-batch of sampled sessions, s is every anchor session in the batch, p is the positive session for s. The positive sample is obtained by randomly masking a query (including all query words and images in this query) in the session. n is a negative session for s.

Hard negative sample. To generate the negative sample n, we first count the number of common clicks between each pair of sessions. Within the batch $\mathcal{B}$, for each session s' which shares zero common clicked items with the anchor session s, we compute their similarity by the aforementioned $MLP(s, s')$, and select the session with the largest similarity, i.e., $n = arg\max_{s'\in\mathcal{B}} MLP(s, s')$. Thus, the negative sample is the most difficult sample to distinguish for the contrastive learner.

4 EXPERIMENT

In this section, we conduct three important tasks in E-commerce, namely personalized click prediction (Section 4.2), query suggestion (Section 4.5), and intent classification (Section 4.6). We evaluate the effectiveness of the pretrained MUVCOG framework in extracting useful session representations on these tasks.

4.1 Experimental Setup

Dataset. We gather search logs of Mobile Taobao in a period of seven days in November 2021 in three product categories: clothes, beauty and digital. Clothes is the largest product category in Mobile Taobao with the most items and search traffic. Beauty and digital are two popular product categories which target different market demographics. For each category, we divide search sessions. As most of previous research [7, 14, 16, 25], we use the 30 minute threshold for idle time. Then, we delete purely textual sessions which contain no image queries, as we focus on multi-modal multi-query sessions in this paper. This step also removes all single query sessions, because a product query and image query can only be issued after a keyword query. Finally, for the convenience of click prediction task, we collect click events, and the corresponding user IDs and item IDs, for each resulted search session.

We report the key statistics of the three datasets in Table 1. Beauty and Digital differ with Clothes in scale (i.e., number of sessions/users/items/clicks). In all datasets, keyword queries are slightly more than product queries. Note that a product query always accompanies an image query, i.e., #PQ equals #IQ. The three categories show different characteristics of search behavior. We compute the time span of each search session, i.e., $T(s)$ is the difference of timestamps between the first query and the last query in search session s. As shown in Table 1, MM search sessions in clothes (averagely 42 minutes) are longer than digital (36 minutes) and beauty (27 minutes). We compute the number of queries in each session (i.e., $L(s)$). Averagely, each clothes session (7.23 queries) is longer than beauty session (6.13 queries) and digital session (6.47 queries). Finally we compute the number of words in each product query ($L(PQ)$) and keyword query ($L(KQ)$). We can see that averagely, digital products (20.17 words) tend to use more words in their titles than beauty products and clothes (approximately 14 words). Users averagely use more keywords to search for clothes (2.84 words) than beauty products (2.42 words) and digital products (2.77 words).

Implementation. We made our code and data publicly available[1]. MUVCOG parameters are initialized with Glorot initialization and trained using Adam SGD optimizer. For all models, including all baselines and competitors, the input word embeddings are 50−dimensional vectors extracted from a word2vec model trained on in-house large-scale E-commerce corpus, the input image embeddings are 512−dimensional vectors extracted from a metric learning

[1] https://github.com/XMUDM/MMsession

Table 1: Dataset statistics: total number of sessions, number of keyword queries (#KQ), number of product queries (#PQ), number of users, number of items, number of clicks, average time-span of each session, average number of queries in each session, average number of words in each product query, average number of words in each keyword query

Dataset	#Sessions	#KQ	#IQ(#PQ)	#Users	#Items	#Clicks	Avg.$T(s)$	Avg.$L(s)$	Avg.$L(PQ)$	Avg.$L(KQ)$
Clothes	8,729,238	34,622,269	28,453,958	6,379,040	11,965,004	245,587,692	42 min	7.23	13.95	2.84
Beauty	851,250	2,865,293	2,349,832	778,451	1,704,675	13,967,180	27 min	6.13	13.84	2.42
Digital	713,445	2,597,696	2,017,968	655,022	1,253,623	13,886,197	36 min	6.47	20.17	2.77

model trained on Taobao visual search dataset. The session embedding size for the MM session is 32 for a single view. The MLP in MUVCOG has three layers with $[64, 32, 1]$ neurons.

4.2 Personalized Click Prediction

Experimental protocol. Personalized click prediction is treated as a supervised learning problem where each training sample is a five-tuple $< \mathbf{q}_i, \mathbf{u}_i, \mathbf{p}_i, \mathbf{s}_i, c_i >$, where $\mathbf{q}_i$ is a query, $\mathbf{u}_i$ is a user, $\mathbf{p}_i$ is an item, $\mathbf{s}_i$ is a search session (optional), and c_i is a corresponding relevance label (i.e., $c_i = 1$: click or $c_i = 0$: non-click). Most existing work [3, 11, 12, 17, 35, 36] follows a two-stage framework, where the first stage is to *extract* feature representations for user/item/query/session, and the second stage is to compute a *matching* score. Our goal is to evaluate whether an MM search session can provide more contextual information and boost click prediction accuracy. Towards this goal, we extract a representation for the MM search session and integrate it to into different baseline click prediction models. For a fair comparison, we do not devise a specialized integration for each baseline. To be specific, the session representation $\mathbf{s}_i$ is concatenated with the other representations learned (e.g., for user/item/query, etc.) by each model's feature extractor, immediately before the matching module.

Baselines. We adopt the following recent deep learning based click prediction models with publicly available source codes and their default hyper-parameter settings. These baselines use different network architectures to extract features and model user-query-item correlations. (1) TranSearch [13][2] uses feed forward network to convert user preference, query, and item into a latent multi-modal feature space. It translates the user preference with the query to the item based on a comparative learning strategy. The vanilla TranSearch does not consider session information. (2) ALSTP [12][3] uses GRU and attention mechanism to extract long-term user preference and short-term purchase inclination from sessions and their corresponding purchases. Prediction is based on cosine similarity between item and query representation upgraded by user intentions. ALSTP considers not only immediately preceding queries but also historical queries of the user. (3) HSCM [3][4] is originally presented for document search. It uses self-attention to extract session representation. It aggregates representations of the current query, search session, feedback within the session, and query-item click bipartite graph in multi-task learning, i.e., document ranking and query suggestion. In implementation, we use the product titles as the documents. HSCM captures not only query interactions within a session, but also cross-session dependencies.

Feature representation of MM sessions. We use different methods to extract an embedding vector of the MM session and concatenate the embedding with other features in the baselines to feed the matching stage. (1) Vanilla: the vanilla baselines without any special treatments for MM search sessions. Note that the vanilla ALSTP and HSCM take into account preceding queries in the session, however, they do not distinguish query types, i.e., they ignore the image queries in the MM sessions. (2) LSTM: we use a Long Short Term Memory network on the MM search session (truncated at the current query). Note that the LSTM encoder is trained end-to-end with the baseline. (3) MUVCOG-T: we remove all image queries in each MM search session and pretrain MUVCOG on purely textual queries. Then we use the pretrained model to extract embedding vector for each session (truncated at the current query). (4) MUVCOG-M: the session representation extractor is pretrained on MM sessions. Note that for MUVCOG-T and MUVCOG-M, we keep the extracted session embeddings fixed and do not update them when training the baselines.

Evaluation Metrics. We apply a random $80 - 20$ train-test split to each dataset. For each session in the testing set, we predict item labels (click or not clicked) for each query. We adopt commonly used evaluation metrics [3, 11, 12, 17, 35, 36] for click prediction, i.e., Normalized Discounted Cummulative Gain (NDCG)@10, Hit Ratio (HR)@10 (HitRatio), and Mean Reciprocal Rank (MRR)@10. Higher metric values indicate more accurate click predictions.

Results and Analysis. We report the results of all baselines with different MM session representations. We compute performance improvements of MUVCOG-M over vanilla baselines in Table 2. Note that we are not comparing the performances among baselines. Instead, we are interested in comparing the effect of different representations of MM sessions. We have the following observations. (1) MUVCOG-M significantly improves various vanilla baselines in terms of NDCG@10, HR@10, and MRR@10, on all datasets. Given the wide coverage of feature extraction and matching mechanisms adopted in various baselines, it verifies our assumption that *incorporating contextual information in MM search session can robustly and effectively enhance click prediction performance.* (2) Even baselines which already encode contextual information in previous queries, i.e., ALSTP and HSCM, can benefit from the proposed MM session representations. The underlying reason is that existing methods only consider conventional purely textual sessions and ignore image queries. Image queries are an important supplement to textual queries, because some information need aspects are difficult to be verbally expressed. (3) Pretrained MUVCOG (without task-specific finetuning) outperforms LSTM which encodes MM sessions in an end-to-end fashion. It suggests that the complex correlations between different types of queries can not be captured by

[2] https://github.com/guoyang9/TranSearch
[3] https://github.com/guoyang9/ALSTP
[4] https://github.com/xuanyuan14/HSCM-master

a sequential model such as LSTM, while the heterogeneous graph neural network is superior. (4) In most cases, MUVCOG-M outperforms MUVCOG-T, which verifies the importance of encoding visual contextual information in MM sessions.

4.3 Ablation Study

Next, we study the contributions of different components in MUVCOG. We choose ALSTP as the baseline, because it is single task learning with contextual information. Thus it can show how MUVCOG can provide additional information of MM session on click prediction.

Impact of multi-view contrastive learning. We first study the impact of multi-view heterogeneous graph neural network. The competiters include: (1) vanilla ALSTP without MM session encoding. (2) GAT: the MM search sessions are represented as homogeneous graphs and apply graph attention network [27] which employes Graph Attention Network (GAT) as the pretraining backbone. (3) AGV: the MM search sessions are represented as heterogeneous graphs and only the attentive global view is applied. (5) HLV: the hierarchical local view is applied on the heterogeneous network. (6) MUVCOG: the proposed multi-view pretraining framework.

From Figure 4, we have the following observations. (1) Incorporating MM session encoding can significantly improve ALSTP's prediction performance, despite the pretrain settings. (2) MUVCOG consistently outperforms single-view contrastive learning (i.e., AGV and HLV), suggesting that single-view is incomplete for MM sessions. (3) HLV produces higher accuracy than AGV and GAT, suggesting HLV is more powerful to capture the intent revealed by query refining and reforming. However, the difference between AGV and GAT is less significant.

Impact of hard negatives. Then, based on multi-view contrastive learning MUVCOG, we compare the results obtained by different negative sampling strategies. (1) random: the negative samples are randomly drawn to form a batch. (2) hard: the proposed hard negative sampling strategy.

From Figure 5, we observe that hard negative sampling strategy is better than random negative sampling strategy on all datasets. It verifies the necessity of using hard negatives to increase the difficulty of contrastive learning and improve the quality of the learned session representations.

4.4 Case Study

To demonstrate that MUVCOG learns good representations for MM sessions, we show a real search session in Figure 6 (a), where the user first types in a query and then searches for similar product. We can see that the image query provides visual characteristics (e.g., diamonds, layered) that are not articulated by the keyword query. Figure 6 (b) plots the embedding vectors of the session learned by different models. (1) LSTM trained jointly with ALSTP, (2) MUVCOG-T, and (3) MUVCOG-M, both are pretrained and extract session embeddings off-the-shelf. The user clicks one item in this session. We also plot the embedding vectors of the clicked item and a randomly sampled non-clicked item. For better illustration, we exhibit the product images of the clicked item and the non-clicked item. We can see that the clicked item is indeed a layered diamond necklace, with properties that are only revealed in the image query and product query. We observe that MUVCOG-M and MUVCOG-T are able to learn a representation of the session which is closer to the clicked item and apart from the the non-clicked item. MUVCOG-M is better than MUVCOG-T as it minimizes the discrepancy between the session and the click. On the contrary, LSTM performs poorly and the session embedding is closer to the non-clicked item.

4.5 Query Suggestion

Given an MM search session $s_i =< q_1, \cdots, q_i >$ of i queries, the goal of query suggestion is to predict the next (i.e., q_{i+1}) query the session will contain. Note that we do not consider user interactions (e.g., click history) in this task. To deliver predictions, we encode the session s_i and feeds the encoding to a feed-forward network with one hidden layer and sigmoid activation function.

Baselines. To isolate the impact of image queries in MM sessions, the comparative study is carried on two settings, with image queries and without image queries. "Without image queries" (corresponding to the top three rows in Table 3) means we remove the image queries in all MM sessions, and compare MUVCOG-T, i.e., pretrained on textual sessions, with (1) LSTM which is adopted in [37], and (2) Transformer which is adopted in [22]. We use the flat-transformer in [22], since it is reported that flat-transformer works better for longer sessions. LSTM and Transformer are trained end-to-end with the prediction layer, while MUVCOG-T is fixed during the training for query suggestion. We also experiment with the image queries in the MM sessions (corresponds to the bottom three rows in Table 3). We apply mean pooling to all image queries, and the LSTM and Transformer encodings, respectively. They are compared against MUVCOG-M, i.e., pretrained on full MM sessions.

Evaluation Metrics. Ranking metrics are commonly adopted to evaluate query suggestion performance. We adopt NDCG@10, HR@10, and MRR@10, to measure the ranking accuracy of top 10 predicted query words. An additional metric is the average cosine similarity between the embedding vectors of predicted query words and actual query words (Avg.Sim). We find that similar search intents may be represented by a diverse vocabulary, which makes predicting the exact query word difficult. For example, the word "shells" and "seashells" can replace each other. Thus we adopt Avg.Sim to eliminate the effect of synonyms.

Results and Analysis. As shown in Table 3, the variants of MUVCOG outperform LSTM and Transformer in terms of all evaluation metrics, on all datasets, with and without image queries. The proposed pretraining framework can extract more useful contextual information than existing sequential encoders. We also observe that all encoding methods perform better with image queries, which again indicates the importance of encoding image queries in MM search sessions.

4.6 Intent Classification

Intent classification task aims to classify a completed MM search session based on the category of products clicked in this session.

Baselines. Since our goal is to investigate whether MUVCOG can automatically extract useful session features, we keep the classification framework as simple as possible. The classifier is XGBoost, which is a gradient boosting algorithm that has been widely used in various classification tasks. We use three petrained models to

Table 2: Personalized click prediction performance of different baselines with various MM session representations. Best results for each baseline and dataset are shown in bold fonts. Improvements of MUVCOG-M over vanilla baselines are presented.

Baseline	DataSet	Beauty			Clothes			Digital		
	Feature	NDCG@10	HR@10	MRR@10	NDCG@10	HR@10	MRR@10	NDCG@10	HR@10	MRR@10
TranSearch	vanilla	0.4991	0.7365	0.4208	0.5074	0.7721	0.4201	0.4709	0.7147	0.3915
	LSTM	0.5009	0.7458	0.4206	0.5370	0.8112	0.4468	0.4787	0.7199	0.4011
	MUVCOG-T	0.5162	0.7756	0.4306	0.5417	0.8127	0.4527	0.4756	0.7249	0.3947
	MUVCOG-M	**0.5362**	**0.8024**	**0.4485**	**0.5632**	**0.8314**	**0.4753**	**0.4959**	**0.7463**	**0.4150**
	Improv.	↑ 7.4%	↑ 8.9%	↑ 6.6%	↑ 11.0%	↑ 7.7%	↑ 13.1%	↑ 5.3%	↑ 4.4%	↑ 6.0%
ALSTP	vanilla	0.4410	0.7140	0.3566	0.4063	0.7239	0.3090	0.2267	0.4152	0.1697
	LSTM	0.4500	0.7468	0.3589	0.4295	0.7353	0.3357	0.2461	0.4418	0.1869
	MUVCOG-T	0.5864	0.8932	0.4905	0.4876	0.7795	0.3973	0.4245	0.6671	0.3500
	MUVCOG-M	**0.6049**	**0.9002**	**0.5123**	**0.5116**	**0.8014**	**0.4216**	**0.4674**	**0.7468**	**0.3819**
	Improv.	↑ 37.2%	↑ 26.1%	↑ 43.7%	↑ 25.9%	↑ 10.7%	↑ 36.4%	↑ 106.2%	↑ 79.9%	↑ 125.1%
HSCM	vanilla	0.2902	0.5796	0.2010	0.3474	0.7224	0.2314	0.3057	0.6073	0.2124
	LSTM	0.3208	0.6704	0.2161	0.3484	0.7297	0.2304	0.3084	0.6136	0.2143
	MUVCOG-T	0.3230	0.6603	**0.2213**	0.3571	**0.7328**	0.2407	0.3105	0.6206	0.2151
	MUVCOG-M	**0.3251**	**0.6766**	0.2200	**0.3590**	0.7317	**0.2435**	**0.3278**	**0.6767**	**0.2232**
	Improv.	↑ 12.0%	↑ 16.7%	↑ 9.4%	↑ 3.3%	↑ 1.3%	↑ 5.2%	↑ 7.2%	↑ 11.4%	↑ 5.1%

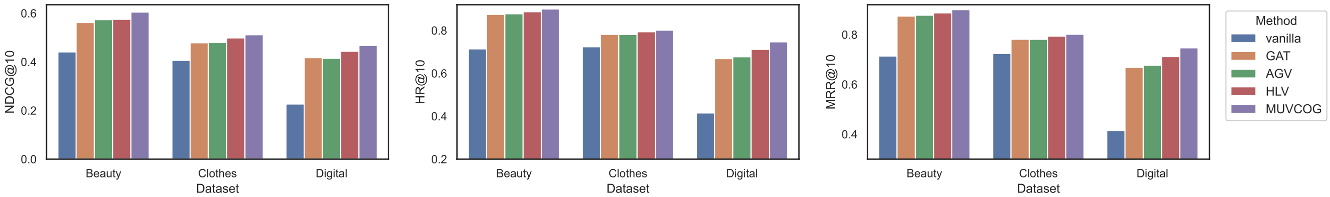


Figure 4: Click prediction performance of ALSTP with MM session encoding by different pretrain settings

Table 3: Query suggestion performance by different session encoding methods with and without image queries (IQ)

IQ	Method	Beauty				Clothes				Digital			
		NDCG@10	HR@10	MRR@10	Avg.Sim	NDCG@10	HR@10	MRR@10	Avg.Sim	NDCG@10	HR@10	MRR@10	Avg.Sim
No	LSTM	0.0707	0.1435	0.0489	0.8208	0.0463	0.0936	0.0322	0.7979	0.073	0.1441	0.0519	0.8322
	Transformer	0.1333	0.2488	0.0993	0.8463	0.0845	0.1536	0.0635	0.8248	0.1259	0.2322	0.0937	0.8504
	MUVCOG-T	0.1698	0.2879	0.1229	0.8792	0.1344	0.2259	0.0948	0.8589	0.1843	0.3159	0.1397	0.8891
Yes	LSTM	0.1284	0.2278	0.0982	0.8211	0.0689	0.1288	0.0507	0.8147	0.1217	0.2267	0.0904	0.8291
	Transformer	0.1442	0.2647	0.1045	0.8591	0.1028	0.1827	0.0722	0.8360	0.1439	0.2697	0.1147	0.8714
	MUVCOG-M	**0.1721**	**0.3019**	**0.1337**	**0.8823**	**0.1527**	**0.2551**	**0.1082**	**0.8663**	**0.2017**	**0.3413**	**0.1444**	**0.895**

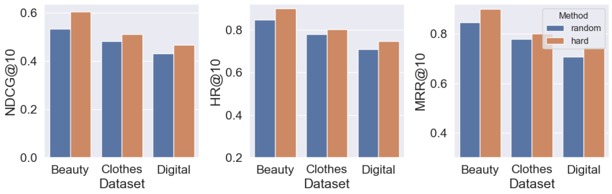
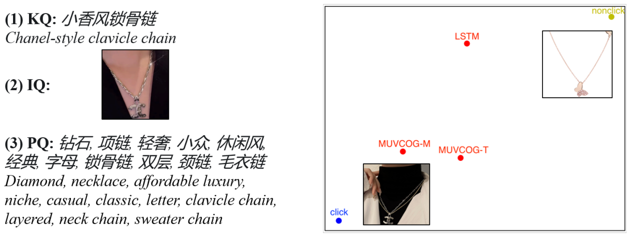


Figure 5: Click prediction performance by different negative sample strategy of MUVCOG


(1) KQ: 小香风锁骨链
Chanel-style clavicle chain
(2) IQ:
(3) PQ: 钻石, 项链, 轻奢, 小众, 休闲风, 经典, 字母, 锁骨链, 双层, 颈链, 毛衣链
Diamond, necklace, affordable luxury, niche, casual, classic, letter, clavicle chain, layered, neck chain, sweater chain
nonclick
LSTM
MUVCOG-M
MUVCOG-T
click

(a) An MM search session **(b) T-SNE visualization of embedding vectors**

Figure 6: A real MM search session and its session embedding vectors learned by different models.

extract session features for XGBoost. (1) word2vec+IQ: we first obtain the word embeddings extracted by a word2vec pretraining model (i.e., the input word embeddings), and apply mean pooling over all query words in the session to obtain the textual features. We then apply mean pooling to image embeddings for all image queries in the session (i.e., the input image embeddings) and obtain visual features. Finally, the textual features are concatenated with the visual features. (2) BERT+IQ: we obtain the word embeddings extracted by a BERT pretraining model on all sessions. The textual features are also obtained by averaging all query words in the session, and the textual features are concatenated with visual features in the same manner as word2vec+IQ. (3) MUVCOG-M: the session

Table 4: Intent classification results for XGBoost with features extracted by different pretrain models

Features	AUC	Acc	P	R	F1
word2vec+IQ	0.9701	0.9523	0.9476	0.9537	0.9506
BERT+IQ	0.9821	0.9742	0.9617	0.9738	0.9677
MUVCOG-M	**0.9977**	**0.9963**	**0.9948**	**0.9969**	**0.9958**

features are extracted by the graph neural network pretrained using multi-view contrastive loss on all MM sessions. Note that for all methods, once the session features are extracted, they are fixed and only parameters of XGBoost will be updated.

Evaluation Metrics. We use Area Under the ROC Curve (AUC), average Accuracy per class (ACC), Macro-Precision (P), Macro-Recall (R) and Macro-F1 as the evaluation metrics.

Results and Analysis. As shown Table 4, MUVCOG-M achieves optimal results, in terms of all evaluation metrics. It shows that the proposed pretrain model MUVCOG can produce universally useful session features to represent user intents.

5 CONCLUSION

This paper studies the problem of modeling multi-query multi-modal (MM) search sessions in Mobile Taobao. We present a multi-view contrastive learning framework to pretrain heterogeneous graph neural network and learn feature representations of the MM sessions. We show that the pretrained session representations, without task-specific fine-tuning, are effective for a number of important Ecommerce downstream tasks such as personalized click prediction, query suggestion, and intent classification.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context Attentive Document Ranking and Query Suggestion. In *SIGIR*. ACM, 385–394.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *NIPS*, Vol. 33. Curran Associates, Inc., 1877–1901.

[3] Jia Chen, Jiaxin Mao, Yiqun Liu, Ziyi Ye, Weizhi Ma, Chao Wang, Min Zhang, and Shaoping Ma. 2021. A Hybrid Framework for Session Context Modeling. *ACM Trans. Inf. Syst.* 39, 3, Article 30 (may 2021), 35 pages.

[4] Jia Chen, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. *A Context-Aware Click Model for Web Search.* ACM, 88–96.

[5] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2018. Attention-Based Hierarchical Neural Query Suggestion. In *SIGIR*. ACM, 1093–1096.

[6] Qiannan Cheng, Zhaochun Ren, Yujie Lin, Pengjie Ren, Zhumin Chen, Xiangyuan Liu, and Maarten de de Rijke. 2021. *Long Short-Term Session Search: Joint Personalized Reranking and Next Query Prediction.* ACM, 239–248.

[7] Arnon Dagan, Ido Guy, and Slava Novgorodov. 2021. An Image is Worth a Thousand Terms? Analysis of Visual E-Commerce Search. In *SIGIR*. ACM, 102–112.

[8] Mostafa Dehghani, Sascha Rothe, Enrique Alfonseca, and Pascal Fleury. 2017. Learning to Attend, Copy, and Generate for Session-Based Query Suggestion. In *CIKM*. ACM, 1747–1756.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186.

[10] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *NACCL*. Association for Computational Linguistics, 879–895.

[11] Long Guo, Lifeng Hua, Rongfei Jia, Binqiang Zhao, Xiaobo Wang, and Bin Cui. 2019. Buying or Browsing?: Predicting Real-Time Purchasing Intent Using Attention-Based Deep Network with Multiple Behavior. In *SIGKDD*. ACM, 1984–1992.

[12] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Yinglong Wang, Jun Ma, and Mohan Kankanhalli. 2019. Attentive Long Short-Term Preference Modeling for Personalized Product Search. *ACM Trans. Inf. Syst.* 37, 2, Article 19 (jan 2019), 27 pages.

[13] Yangyang Guo, Zhiyong Cheng, Liqiang Nie, Xin-Shun Xu, and Mohan Kankanhalli. 2018. Multi-Modal Preference Modeling for Product Search. In *MM*. ACM, 1865–1873.

[14] Shashank Gupta and Subhadeep Maji. 2020. Predicting Session Length for Product Search on E-Commerce Platform. In *SIGIR*. ACM, 1713–1716.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. 2021. Masked Autoencoders Are Scalable Vision Learners. *CoRR* abs/2111.06377 (2021).

[16] Sharon Hirsch, Ido Guy, Alexander Nus, Arnon Dagan, and Oren Kurland. 2020. Query Reformulation in E-Commerce Search. In *SIGIR*. ACM, 1319–1328.

[17] Yujing Hu, Qing Da, Anxiang Zeng, Yang Yu, and Yinghui Xu. 2018. Reinforcement Learning to Rank in E-Commerce Search Engine: Formalization, Analysis, and Application. In *SIGKDD*. ACM, 368–377.

[18] Jyun-Yu Jiang and Wei Wang. 2018. RIN: Reformulation Inference Network for Context-Aware Query Suggestion. In *CIKM*. ACM, 197–206.

[19] Evangelos Kanoulas, Ben Carterette, Paul D. Clough, and Mark Sanderson. 2011. Evaluating Multi-Query Sessions. In *SIGIR*. ACM, 1053–1062.

[20] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR (Poster)*. OpenReview.net.

[21] Katrien Laenen, Susana Zoghbi, and Marie-Francine Moens. 2018. Web Search of Fashion Items with Multimodal Querying. In *WSDM*. ACM, 342–350.

[22] Agnès Mustar, Sylvain Lamprier, and Benjamin Piwowarski. 2021. On the Study of Transformers for Query Suggestion. *ACM Trans. Inf. Syst.* 40, 1, Article 18 (oct 2021), 27 pages.

[23] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. *GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training.* ACM, 1150–1160.

[24] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A Hierarchical Recurrent Encoder-Decoder for Generative Context-Aware Query Suggestion. In *CIKM*. ACM, 553–562.

[25] 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.

[26] 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 *NIPS*. 5998–6008.

[27] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR (Poster)*. OpenReview.net.

[28] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. *Self-Supervised Heterogeneous Graph Neural Network with Co-Contrastive Learning.* ACM, 1726–1736.

[29] Bin Wu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Query Suggestion with Feedback Memory Network. In *WWW*. International World Wide Web Conferences Steering Committee, 1563–1571.

[30] Liang Wu, Diane Hu, Liangjie Hong, and Huan Liu. 2018. Turning Clicks into Purchases: Revenue Optimization for Product Search in E-Commerce. In *SIGIR*. ACM, 365–374.

[31] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-parametric Instance Discrimination. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 3733–3742.

[32] Xiaohui Xie, Jiaxin Mao, Yiqun Liu, Maarten de Rijke, Qingyao Ai, Yufei Huang, Min Zhang, and Shaoping Ma. 2019. Improving Web Image Search with Contextual Information. In *CIKM*. ACM, 1683–1692.

[33] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *NIPS*, Vol. 33. Curran Associates, Inc., 5812–5823.

[34] Oleg Zendel, J. Shane Culpepper, and Falk Scholer. 2021. Is Query Performance Prediction With Multiple Query Variations Harder Than Topic Performance Prediction?. In *SIGIR*. ACM, 1713–1717.

[35] Yuan Zhang, Dong Wang, and Yan Zhang. 2019. Neural IR Meets Graph Embedding: A Ranking Model for Product Search. In *WWW*. ACM, 2390–2400.

[36] Jiashu Zhao, Hongshen Chen, and Dawei Yin. 2019. A Dynamic Product-Aware Learning Model for E-Commerce Query Intent Understanding. In *CIKM*. ACM, 1843–1852.

[37] Jianling Zhong, Weiwei Guo, Huiji Gao, and Bo Long. 2020. *Personalized Query Suggestions.* ACM, 1645–1648.

[38] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. *Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking.* ACM, New York, NY, USA, 2780–2791.