Enhanced Representation with Contrastive Loss for Long-Tail Query Classification in e-commerce

Lvxing Zhu, Hao Chen, Chao Wei, Weiru Zhang
Alibaba Group, Hangzhou, China
{lvxing.zlx, ryan.ch, weichao.wc, weiru.zwr}@alibaba-inc.com

Abstract

Query classification is a fundamental task in an e-commerce search engine, which assigns one or multiple predefined product categories in response to each search query. Taking click-through logs as training data in deep learning methods is a common and effective approach for query classification. However, the frequency distribution of queries typically has long-tail property, which means that there are few logs for most of the queries. The lack of reliable user feedback information results in worse performance of long-tail queries compared with frequent queries. To solve the above problem, we propose a novel method that leverages an auxiliary module to enhance the representations of long-tail queries by taking advantage of reliable supervised information of variant frequent queries. The long-tail queries are guided by the contrastive loss to obtain category-aligned representations in the auxiliary module, where the variant frequent queries serve as anchors in the representation space. We train our model with real-world click data from AliExpress and conduct evaluation on both offline labeled data and online AB test. The results and further analysis demonstrate the effectiveness of our proposed method.

1 Introduction

In the e-commerce search engine, query classification is a task to assign one or multiple predefined product categories to each search query. It is a fundamental component that recognizes the intent of user query and retrieves relevant products. The task of query classification can be basically viewed as a multi-label short text classification problem.

Deep learning methods are the mainstream approaches for query classification tasks nowadays. Considering the massive amount of queries and categories, it's usually too expensive to collect train data by manually labeling. Therefore, utilizing the click-through data as implicit feedback signals to build a model is the most common approach that predicts the categories of query (Shen et al., 2009; Lin et al., 2018b) . Various deep models have achieved great success in query classification (Zhang et al., 2019; Yu and Litchfield, 2020; Zhang et al., 2021). To fully utilize the mutual information between the query and categories, some models convert the multi-label classification to a multiple binary classification task and obtain superior performance (Liu et al., 2017; Nam et al., 2014).

However, the long-tail distribution of queries in e-commerce websites brings challenges to deep models. Few high-frequency queries dominate in search input while low-frequency queries have a very low probability of occurrences. These low-frequency queries are what we call long-tail queries and others are frequent queries. The users' feedback logs of long-tail queries are usually difficult to obtain and insufficient training data also result in a serious data noise problem. Moreover, the product taxonomy in e-commerce websites usually consists of thousands of categories. The large amount of categories aggravates the sparsity of long-tail query-category feedback data. Therefore, the lack and noise of training data cause the lower performance for long-tail queries compared with frequent queries in the task of query classification.

Another problem is that queries with slight lexical differences may have totally different category intents (Zhang et al., 2021). Queries in e-commerce are typically short and ambiguous (Shen et al., 2009; Lin et al., 2018b). A modification of one word in the query could entirely change the corresponding category, such as "blouse collar" and "blouse with collar", or "pearl ring" and "pearl earring". This phenomenon hinders the deep models from classifying long-tail queries because there aren't enough examples to distinguish the intents of these lexically similar queries.

In summary, query classification in real-world e-commerce scenarios differs from common text classification tasks in at least two aspects. At First,

Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5), pages 141 - 150

the supervised information is not entirely reliable especially for long-tail queries, depending on the query frequency in search logs. Secondly, most queries are short and the textual information inside queries is very limited.

Inspired by the aforementioned observations, we propose a novel method to improve the performance of long-tail query classification. The basic idea of our method is to utilize the query frequency information and transfer knowledge from frequent queries to long-tail queries, which takes advantage of the fact that the click feedbacks of frequent queries are more reliable. For each query, we select several frequent queries as the variant queries, which are lexically similar to the original query. We use an auxiliary module coupled with a contrastive loss (Chopra et al., 2005) to enhance the representation of the original query by these variant queries. The variant queries serve as anchors in vector space while the auxiliary module aligns the representation vectors between original queries and variant queries in the view of category semantics. We conduct experiments on real-world click data from AliExpress [1] and also evaluate our method on the public dataset. The results suggest that significant improvement of long-tail query classification tasks on multiple metrics. We further conduct the comparison and visualization to verify the effect of our representation enhancement.

The major contributions of this article are summarized as follows:

- We propose a novel method for long-tail query classification by transferring knowledge from multiple variant frequent queries to long-tail queries.
- Our method enhances the representations of queries with the contrastive loss which bases on the category consistence among lexical similar queries.
- We validate the effectiveness of our method in a public dataset and real-world search scenarios.

2 Related Work

2.1 Query Classification

There have been various works studying query classification in e-commerce recently. These works can be classified into three categories: statistical-based methods (Shen et al., 2009), traditional machine learning methods and deep learning methods. Lin et al. (2018b) introduce an unsupervised method to collect query classification data from click-through logs and apply several traditional methods such as SVM, XGBoost and fastText on this task. Zhang et al. (2019) design a progressively hierarchical classification framework to make use of the semantic information from a category tree and take TextCNN (Zhang and Wallace, 2015) as the base model. To incorporate information of category tree structure, Gao et al. (2020) proposes a deep hierarchical classification framework. The framework generates layer representation for each layer and shares the representation to lower layers. Yu and Litchfield (2020) propose a multi-objective method that optimizes hierarchical accuracy-depth trade-off across multi-level categories. Multi-objective optimization is adopted in post inference phase to select the deepest category whose prediction accuracy exceeds its corresponding threshold. Zhang et al. (2021) propose a framework that also focuses on long-tail query classification in e-commerce, which adds an auxiliary across-context attention module to extract external information by predicting the categories of variant queries.

2.2 Text Classification

Multi-label text classification can be viewed as the generalization of query classification, although most text classification tasks focus on long text such as web document, papers and news. CNN-based models, including traditional CNN (Liu et al., 2017) and graph-CNN (Peng et al., 2018), are the common approaches to classify text. Lin et al. (2018a) apply multi-level dilated convolution and attention-over-attention mechanism to generate higher-level semantic representations for text classification. Recurrent networks are also applied to this task, You et al. (2019) proposes a label BiLSTM-based deep learning model with multi-label attention named AttentionXML. AttentionXML uses a probabilistic label tree to handle extreme multi-label text classification (XML). X-Transformer (Chang et al., 2020) deals with XML by transformer models, which predicts labels in two steps: first, recalls the label clusters and then re-ranks the labels within the predicted clusters. LightXML (Jiang et al., 2021) and DECAF (Mittal et al., 2021) also follows the above two-stage

[1] https://www.aliexpress.com, a cross-border e-commerce platform of Alibaba

schema but make efforts in utilizing label metadata and dynamic negative sampling, respectively. Yang et al. (2018) and Lin et al. (2018a) apply a sequence-to-sequence model on multi-label classification to capture the correlations between label. Peng et al. (2018) use a regularized loss to model the dependency of hierarchical classes.

2.3 Contrastive Learning

The contrastive loss is first presented by Chopra et al. (2005) for the face verification task. By minimizing the contrastive loss with siamese networks, the similarity metric becomes small for pairs of faces from the same person and large for pairs from different persons. Oord et al. (2018) utilize a probabilistic contrastive loss in a universal unsupervised learning approach to extract useful representations from high-dimensional data, such as speech, images and text. Lian et al. (2018) introduce a deep model coupled with contrastive loss to learn discriminative audio representations. The principle that aligns the representation according to the categories is similar to contrast learning. Contrastive learning learns effective representations by pulling semantically close neighbors together and pushing apart non-neighbors so that "similar" points in input space are mapped to nearby points on the manifold (Hadsell et al., 2006). SimCLR(Chen et al., 2020) is a successful appliance of contrastive learning in computer vision fields and Gao et al. (2021) introduce SimCSE in NLP, which applies contrast learning to sentence embedding task of both unsupervised approach and supervised approach.

3 Methodology

We introduce our proposed method in this section. We first give an overview of our method in Section 3.1 and then demonstrate each module in detail from Section 3.2 to 3.5.

3.1 Overview

We first define the formal notation of our work. We cast query classification task as multi-label text classification in a binary manner. Given a query $q = [wq_1, wq_2, ..., wq_n]$, a sequence of words with length n, the task is to predict whether category $c \in C$ is a positive category for query q or not, where C is the set of predefined e-commerce categories. The category $c = [wc_1, wc_2, ..., wc_m]$ is also a sequence of words that is the description of the category. We denote training set as $T = \{\langle q_i, c_i, y_i\rangle | i = 1, 2, ..., N\}$, where $y_i \in \{0, 1\}$ is the label which indicates the pair $\langle q_i, c_i\rangle$ is a negative example or a positive example. We denote the set of all queries in training set as $Q_{ALL} = \{q | \langle q, *, *\rangle \in T\}$. We also define a frequent queries set Q_F, which is a subset of Q_{ALL} and consists of the queries with daily average page views more than $thres_q$. The other queries are regarded as long-tail queries.

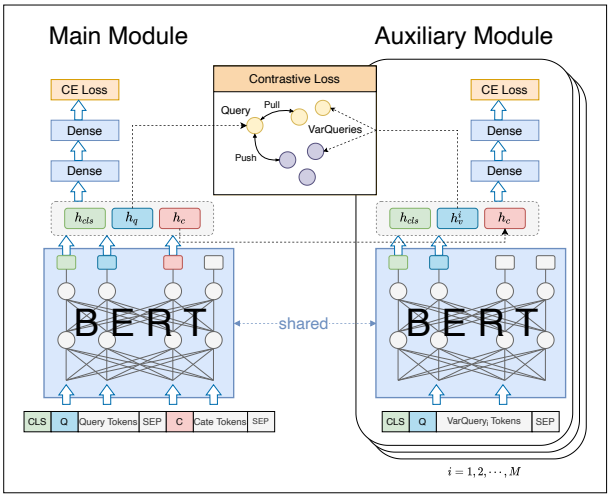

Figure 1: The overview of our proposed model.

Our proposed method consists of two components: the main module and the auxiliary module, as depicted in Figure 1. The main module is a standard text classification model which takes the tokenized sequence of query and category as input and predicts the relation between query and category in an interactive manner. The auxiliary module is to learn the representations of variant queries and transfer the representing ability to the main module with a contrastive loss function. The main and auxiliary modules are optimized simultaneously in training phase while only the main module is used in inference phase. Therefore, we do not add extra computation for prediction compared with base method. In this work, we adopt BERT (Devlin et al., 2018) as our base model because it achieves state-of-the-art performance among many NLP tasks and provides a high standard of baseline.

3.2 Main Module

The main module is basically a standard BERT to compute the relevance score between query and category via transformer architecture and we make a slight modification on input schema. Since each query and category share the same BERT model, the query input is typed by customers and the cate-

gory's textual description is usually more formally written, which differs in choice of words. According to ColBERT(Khattab and Zaharia, 2020), we distinguish the input sequences by adding a special token [Q] to queries and another token [C] to category descriptions. Given a training example $\langle q, c, y\rangle$, we get the concatenated input tokens $[CLS, [Q], wq_1, ..., SEP, [C], wc_1, ..., SEP]$. After feeding the input, we obtain the output vectors from BERT of all tokens. We denote the output vectors of $[Q]$-location, $[C]$-location and CLS-location as h_q, h_c and h_{cls}, which are expected to represent the query, the category and the interactive feature between query and category respectively.

We concatenate h_q, h_c and h_{cls} as a hidden vector and feed it to fully connected networks with binary cross-entropy loss to predict the target score $\hat{y} \in [0, 1]$. The target score indicates the relevance of the query and category, as follows:

$$\hat{y} = f(W_a \cdot (h_q \oplus h_c \oplus h_{cls}) + b_a), \quad (1)$$

$$L_M = -ylog(\hat{y}) - (1-y)log(1-\hat{y}), \quad (2)$$

where W_a and b_a is the weight and bias of the fully connected network and y denotes the label of pair $\langle q, c\rangle$.

3.3 Variant Query Selection

To transfer knowledge from frequent queries to long-tail queries, we introduce variant queries that are lexically similar to the original query with several different tokens. Despite the similarity in text, those slight tokens' differences between the original query and its variant queries can lead to totally different category intents. Since variant queries and original queries may be fused in semantic space because of the lexical similarity, we use the category information from variant queries to build a better latent representation for original queries with auxiliary tasks.

We select M variant queries for each query in the training set. All the variant queries are frequent queries that are selected from the candidate set Q_F. We propose a simple but effective method with a low computational cost to select variant queries. To measure the textual similarity between query q and candidate query q_c, let T_q and T_{qc} be the set of tokens of q and q_c respectively, a weighted token similarity is calculated as follows:

$$Sim(q, q_c) = \frac{\sum_{t_i \in T_q \cap T_{qc}} w_{i,qc}}{|T_{qc}|}, \quad (3)$$

where $w_{i,qc}$ is the weight score of token t_i in query q_c. We take TF-IDF (Salton and Buckley, 1988) as the weight score:

$$w_{i,qc} = TF_{i,qc} * IDF_i, \quad (4)$$

where IDF_i is the inverse document (i.e., query) frequency of token t_i. We order all the candidate queries by their similarity score $Sim(q, q_c)$ with q and select the top M of them as the set of variant queries, denoted as V^q.

The variant queries have similar text to the original query but they are not always have same categories. These queries with different category intents become hard negative examples which are then utilized by the contrastive loss. All the work in this subsection is done in data preparing phase.

3.4 Auxiliary Module

The auxiliary module is also a BERT-based model that predicts the relevance between the given category and the variant queries. The auxiliary module shares parameters of BERT with the main module. However, to obtain the pure representation of queries, the auxiliary module only receives the tokens of variant queries as input. For each variant query $q_i^v \in V^q$, we add the special token $[Q]$ ahead to indicates a query sequence, i.e., the input sequence is $[CLS, [Q], wq_1^v, wq_2^v, ..., wq_n^v, SEP]$. We take the output of $[CLS]$-location and $[Q]$-location from BERT as the representation of variant query q_i^v , which is denoted as h_{cls}^i and h_v^i. we obtain representation of category c from the main module and concatenate h_{cls}^i and h_v^i with h_c for downstream fully connected networks. Finally, the relevance score is predicted as follows:

$$\hat{y}_i = f(W_b \cdot (h_{cls}^i \oplus h_v^i \oplus h_c) + b_b), \quad (5)$$

$$L_A = -\sum_M y_i log(\hat{y}_i) + (1-y_i)log(1-\hat{y}_i), \quad (6)$$

where $\hat{y}_i$ is the prediction value and y_i is the label that indicates whether category c is related to the variant query q_i^v. The value of y_i is from the training set where $y_i = 1$ if $\langle q_i^v, c, 1\rangle \in T$ otherwise $y_i = 0$. Through the training of the auxiliary module, we build the representations of variant queries which play an important role in the calculations of contrastive loss.

3.5 Contrastive Loss

Since we have obtained h_q and $h_i^v, i = 1, .., M$, which are the representations of original query q and its variant queries q_i^v, we apply a contrastive loss function to align their representations according to their relevance with the category c. We follow the definition of contrastive loss in (Lian et al., 2018). If the variant query q_i^v and original query q belong to the same category c, the representations of the two queries should be pulled together. Otherwise, the representations should be pushed apart. The above process is adopted by adding the contrastive loss as follows:

$$L_C = \sum_{i \in 1,...,M} y \cdot L_C^i, \tag{7}$$

$$L_C^i = \begin{cases} \|h_q - h_i^v\|_2 & y \bigwedge y_i = 1 \\ max(0, m - \|h_q - h_i^v\|_2) & y \bigwedge y_i = 0 \end{cases}, \tag{8}$$

where $\|\cdot\|$ is the L2 norm and m is the margin.

Finally, the total loss is calculated as follows, re-weighted by parameters λ_A and λ_C:

$$L = L_M + \lambda_A L_A + \lambda_C L_C. \tag{9}$$

The auxiliary module aims at transferring knowledge from frequent queries to long-tail queries. Since all the variant queries are frequent queries that have a large number of click feedback, the supervised signal derived from those feedback is more reliable and their representations are more reasonable in feature space. With the constraint of contrastive loss between the original query and lexically similar variant queries (shown in Figure 1), queries obtain better representations to recognize different category intents.

It's noteworthy that our proposed method enhances the query representations in long-tail query classification task without bringing in external information. we neither augment the training examples nor use other data except the training set.

4 Experiments

We introduce the training and evaluation data set and the setting of our experiments in section 4.1. We discuss the performance and effectiveness of our proposed method with other methods in section 4.2 and 4.3.

4.1 Data and Setting

To collect training data, we sample search queries and their clicked products' categories in recent 3 months logs from AliExpress, a cross-border e-commerce platform of Alibaba. We collect 5,000,000 $\langle query, category \rangle$ pairs and the numbers of distinct queries and categories are 3,620,000 and 6,300 respectively. All of the queries and categories are in English. Queries with daily average page view less than $thres_q = 100$ are defined as long-tail queries, which include almost 97% of queries and occupy only 56% of the whole clicked pairs. The rest 3% of queries are defined as frequent queries that contributes 44% of clicks. For each query, we choose its corresponding categories with high click-through rates as our positive training examples and replace the query or category randomly from a different pair to generate negative examples. Finally, we obtain a training data set which consists of about 33,400,000 pairs. We list several examples from the training set in Table 1 including the variant queries and their labels. The columns named "Query", "Category" and "Label" are the inputs of the main module, where the "Label" column denotes whether the pair of query and category is relevant or not. The columns named "Variant Q1/Q2/Q3" and corresponding "Label 1/2/3" are the inputs of the auxiliary module.

For evaluation, We randomly sample another 2,000 long-tail queries and collect a total of 78,226 correspondings clicked $\langle query, category \rangle$ pairs from search engine. Each pair in the evaluation set are labeled as relevant or irrelevant by human annotators.

We use the BERT-Tiny pre-trained model released by google research as our base model. We process the queries and categories by WordPiece tokenization (Wu et al., 2016). The number of variant queries M is 3. The number of frequent queries N is 100,000. Margin m in contrastive loss is 32.0 . The loss reweight parameter λ_A and λ_C are 0.1 and 0.02 respectively. We use Adam optimization method (Kingma and Ba, 2014) and set the learning rate to 1e-6 while the batchsize is 1024. All hyper-parameters are adjusted according to the performance on the held-out validation set. We train the model for a fixed number of global steps (640,000) and save models at regular intervals. Then we choose the best performance achieved by these models as the result.

Table 1: Examples of training set.

Query	Category	Label	Variant Q1	Label1	Variant Q2	Label2	Variant Q3	Label3
video game consoles 16bit	Handheld Game Players	1	game consoles	1	video game consoles	1	video	0
lure glass rattles	Fishing Lures	1	lure	1	glass	0	fishing lure	1
fine point heels	Women's Pumps	1	point	0	heels	1	fine jewelry	0
huawei p 40 lite 5g cover	Mobile Phone Cases & Covers	1	huawei	0	cover	1	huawei phone	0
612 bundles with frontal	Hair Bundles with Closures	1	bundles	1	bundles with frontal	1	613 bundles	1
0.25 eyelashes	Body Foundation	0	eyelashes	0	rover 25	0	false eyelashes	0
1 birthday boy clothes	Audio Intercom	0	birthday	0	birthday boy	0	boy birthday	0
2000s aesthetic sunglasses	Wax Fabrics	0	2000s	0	2000s aesthetic	0	sunglasses	0
pink porcelain plate	Men's Socks	0	porcelain	0	porcelain plate	0	plate porcelain	0

4.2 Performance

We use AUC (Area Under Curve), AP (Average Precision), Prec (Precision), Recall and F1 score as our evaluation metrics. The Average precision (Turpin and Scholer, 2006) is the area under the precision-recall curve and it is independent of threshold as well as AUC.

We conduct experiments on the aforementioned data set and compare our method with the baseline and existing approaches in Table 2:

- "Base" is the standard BERT model which takes the same setting as our proposed method (only preserve the main module).
- "AC(LSTM)" (Zhang et al., 2021) is the latest related work for long-tail query classification which couples with an auxiliary task to provide across-attention information. The original paper uses LSTM (Hochreiter and Schmidhuber, 1997) as the encoder and the base queries and variant queries differ in encoders.
- "AC(BERT)" is the modified version we implemented, which shares the same BERT encoder for original queries and variant queries following our proposed method setting for a fair comparison.
- "Proposed" is our proposed method.

As shown in Table 2, our proposed method outperforms the baselines by a statistically significant margin on all of the metrics. Compared with method named "Base", our method achieves +1.92% improvement on AUC, +2.48% improvement on AP and +2.27% improvement on F1 score. Our method also outperforms "AC(LSTM)" and "AC(BERT)" in all of the metrics with steady margins (range from +1.19% to +1.68%) which reflect the effectiveness. The AC methods only use one variant query to adjust the original query representation implicitly. In contrast, our proposed method clearly establishes the pulling or pushing relations between the original query and variant queries by contrastive constraints.

Table 2: Performance comparison between our method and baselines on evaluation set.

Method	AUC	AP	Prec	Recall	F1 Score
Base	0.7610	0.3110	0.3367	0.3463	0.3414
AC(LSTM)	0.7622	0.3132	0.3214	0.3727	0.3452
AC(BERT)	0.7681	0.3190	0.3396	0.3632	0.3510
Proposed	**0.7802**	**0.3358**	**0.3522**	**0.3767**	**0.3641**

To better understand the contribution of each key component of our method, we conduct several ablation tests and each method is described as follows:

- "Proposed-Without LC" removes the contrastive loss from the proposed method.
- "Proposed-Only CLS" only uses h_{cls} as feature instead of concatenating h_q, h_c and h_{cls} in main module.
- "Proposed-Sym" means the auxiliary module uses the same input schema as the main module, which takes both variant query tokens and category tokens as input.
- "Proposed-Entire Query" takes the whole queries in the training set as variant query candidates rather than only the frequent queries.

The experimental results are listed in Table 3. The results show that our proposed method outperforms the others significantly. As contrast, the performance of "Proposed-Without LC" decreases significantly, which shows that the contrastive loss plays a key role in enhancing representations. The performance of "Proposed-Only CLS" decreases remarkably compared with our proposed method, which shows that the representations h_q and h_c can provide extra useful information for query classification. The performance of "Proposed-Sym" is dramatically lower than the results of "Proposed" method and has only slight improvement compared to the "Base" method. The result indicates that the

Table 3: Ablation test of our proposed method on evaluation set.

Method	AUC	AP	Prec	Recall	F1
Base	0.7610	0.3110	0.3367	0.3463	0.3414
Proposed-Sym	0.7650	0.3148	0.3124	**0.3791**	0.3425
Proposed-Without LC	0.7584	0.3147	0.3240	0.3780	0.3489
Proposed-Only CLS	0.7708	0.3245	0.3339	0.3699	0.3510
Proposed-Entire Query	0.7645	0.3209	0.3304	0.3589	0.3441
Proposed	**0.7802**	**0.3358**	**0.3522**	0.3767	**0.3641**

representation alignment effect of contrastive loss weakens using query tokens and category tokens simultaneously in the auxiliary module. With inputs of extra category tokens, the representation of the variant query h_i^v loses its independence and becomes sensitive to disturbance of category texts, which makes h_i^v an unstable anchor for the original query. The decreased performance of "Proposed-Entire Query" shows choosing frequent queries as variant queries can lead to better representations for long-tail queries, which implies frequent queries serve as better anchors in hidden spaces because of sufficient training data.

To investigate the effect of our method on long-tail queries, we split the evaluation set of long-tail queries into 3 sets according to their frequency levels in search logs. The set of relatively high-frequency queries is named "Long-tail Head", the set of queries with middle-frequency level is named "Long-tail Mid" and the set of rest queries is "Long-tail Tail". To compare with performance on those long-tail query sets, we also randomly sample a set of queries named "Top Freq" from frequent queries as defined in Section 4.1, which includes 28,389 pairs of 512 queries. The performance of base method and our method on the above evaluation sets are listed in Table 4. Our method outperforms the baseline by a significant margin in all the groups of long-tail queries, where the AUC improvements are +1.99%, +1.84% and +2.22% respectively. The improvement on "Long-tail Tail" set is greater than other sets, which means the greatest improvement is achieved on the queries at the far end of the tail. We notice that the improvement on "Top Freq" queries is much smaller with +0.28% on AUC, +0.37% on AP and +0.67% on F1 Score, compared with the remarkable improvement on long-tail queries. The results indicate that our method improves the effectiveness on long-tail queries more than frequent queries.

Furthermore, We evaluate our method on the public dataset released by the personalized e-commerce search challenge of the CIKM Cup 2016 [2]. This dataset contains query searching and browsing logs and product metadata including the product categories information (Wu et al., 2017). We process the data files and collect a total of 500,000 $\langle query, category \rangle$ pairs as training data for the query classification task, which have 26,137 distinct queries and 1,213 distinct categories. To collect test data, we sample 3000 long-tail queries and remove the corresponding pairs from the training set. The detailed data process is described in Appendix A. As shown in Table 5, our proposed method substantially improves multiple metrics including AUC, AP and F1 Score on this dataset, with +4.3% and +2.6% absolute improvement on AP compared with the base method and AC(LSTM) method respectively. Considering that the dataset is more sparse and its tokenization is different from the common wordpiece model (Wu et al., 2016), these performances demonstrate the effectiveness and generalizability of our method.

4.3 Discussion

To verify whether our proposed method is able to recognize the category intents of lexically similar queries, we calculate two distances: 1) the average Euclidean distance between representation vectors of original queries and their category-consistent variant queries and 2) the same metric between original queries and their category-inconsistent variant queries. We visualize the distances between the proposed method and baseline with increasing training steps in Figure 2. The curves named "PD" and "PS" are derived from our proposed method and the curves named "BD" and "BS" are from the base method. In Figure 2, the gap between the curve "BD" and "BS" are very close all the time while the curve "PD" and "PS" gradually separate from each other with a certain margin. As we mentioned ahead, the phenomenon indicates that our proposed model distinguishes different category well by pulling original query and category-consistent variant queries together and pushing them of different category further apart.

To verify whether our proposed method generates better representations for long-tail queries, we visualize the representation vectors (i.e. vector [Q] from the model) of the baseline method and our method in Figure 3. We randomly sample 100,000 long-tail queries and project their representations

[2] https://competitions.codalab.org/competitions/11161

Table 4: Performance improvement of queries grouped by frequency in search logs.

Group	base AUC	base AP	base Prec	base Recall	base F1 Score	proposed AUC	proposed AP	proposed Prec	proposed Recall	proposed F1 Score
Top Freq	0.7408	0.4615	0.4149	0.4852	0.4473	0.7436(+0.28%)	0.4652(+0.37%)	0.4177(+0.28%)	0.4973(+1.21%)	0.4540(+0.67%)
Long-tail Head	0.7558	0.3469	0.3359	0.3823	0.3576	0.7757(+1.99%)	0.3514(+0.45%)	0.3564(+2.05%)	0.3887(+0.64%)	0.3719(+1.43%)
Long-tail Mid	0.7461	0.3102	0.3214	0.3862	0.3508	0.7645(+1.84%)	0.3344(+2.42%)	0.3287(+0.73%)	0.4005(+2.43%)	0.3611(+1.03%)
Long-tail Tail	0.7742	0.2926	0.3103	0.3539	0.3307	0.7964(+2.22%)	0.3211(+2.85%)	0.3484(+3.81%)	0.3736(+1.97%)	0.3606(+2.99%)

Table 5: Performance comparison between our method and baselines on the public CIKM Cup 2016 dataset.

Method	AUC	AP	Prec	Recall	F1 Score
Base	0.9096	0.6465	0.6571	0.5716	0.6061
AC(LSTM)	0.9122	0.6637	0.6736	0.5732	0.6193
AC(BERT)	0.9129	0.6668	**0.6845**	0.5636	0.6181
Proposed	**0.9160**	**0.6897**	0.6738	**0.5969**	**0.6330**

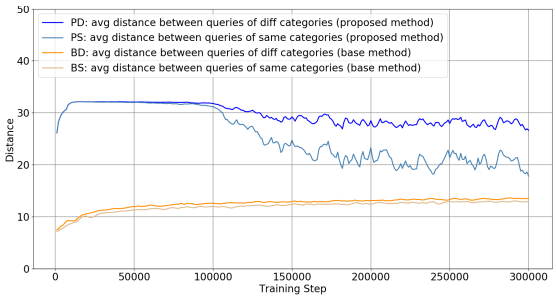

Figure 2: The Euclidean distance of category-consistent queries and category-inconsistent queries representation vector.

to 2-dimensional space by UMAP(McInnes et al., 2018) (neighbors=40, epochs=400). Each point in Figure 3 is a long-tail query colored according to its category. If a query is relevant to multiple categories, the category which has the most click logs is chosen. To simplify the figure, we only reserve queries that are relevant to the top-10 hot categories. As shown in the left box of Figure 3, a lot of the query representations are scattered around the space, and categories groups are overlapped with each other, which means the baseline method fails to preserve the category consistency of long-tail queries. In contrast, the query representations in the right figure form clusters according to their categories spontaneously. Most of these clusters are cohesive and keep away from other clusters. There are still some overlaps between query clusters, probably due to that some queries are naturally interested with multiple categories, such as T-shirt (Men) and T-shirt (Women). The visualization results indicate that our method is able to obtain reasonable representations corresponding to the category semantics of queries.

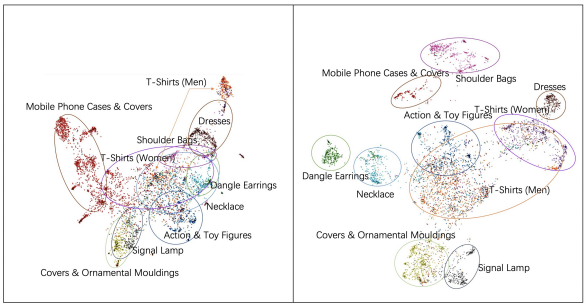

Figure 3: The visualized representation vectors of long-tail queries generated by Base model (left) and our proposed model (right).

Table 6: Online performance evaluation.

Method	CTR	RPM
Base	-	-
Proposed	+0.63%	+1.06%

4.4 Online Evaluation

We deploy online evaluation in search advertising system of AliExpress. Instead of comparing each query with the whole 6,300 categories online, we predicted categories offline beforehand and generated a query-category cached table that covers over 80% of page views. The predicted categories of queries serve as the filters of vector-based product retrieval and influence the relevance score between the queries and products in our e-commerce sponsored search system. We conducted standard A/B testing for 5 days and selected 5% of the search traffic as the test group to evaluate our proposed method. Two common metrics are calculated for evaluation: CTR (click-through rate) and RPM (revenues per mille). As shown in Table 6, the results suggest that our proposed method improves the online performance on tens of millions of user visits. The gains of CTR and RPM reflect that our method increases valid exposures of the advertisement with better quality, which finally results in the growth of users' clicks and platform revenue.

5 Conclusion

In this paper, we propose a novel method for query classification which focuses on the long-tail queries in e-commerce. Our method consists of a main

module and an auxiliary module that aims at utilizing reliable information from frequent queries to help the classification of long-tail queries. The results of extensive experiments show that our proposed method outperforms the baselines by a substantial margin. Further analysis demonstrates our method can obtain better representations for long-tail queries and discriminate different category intents from lexically similar queries. In the future, We will generalize our idea to more situations in e-commerce, such as multi-language query classification and other tasks such as product retrieval or relevance classification.

References

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3163–3171.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dehong Gao, Wenjing Yang, Huiling Zhou, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep hierarchical classification for category prediction in e-commerce system. *ECNLP 3*, page 64.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7987–7994.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Zheng Lian, Ya Li, Jianhua Tao, and Jian Huang. 2018. Speech emotion recognition via contrastive loss under siamese networks. In *Proceedings of the Joint Workshop of the 4th Workshop on Affective Social Multimedia Computing and First Multi-Modal Affective Computing of Large-Scale Multimedia Data*, pages 21–26.

Junyang Lin, Qi Su, Pengcheng Yang, Shuming Ma, and Xu Sun. 2018a. Semantic-unit-based dilated convolution for multi-label text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4554–4564.

Yiu-Chang Lin, Ankur Datta, and Giuseppe Di Fabbrizio. 2018b. E-commerce product query classification using implicit user's feedback from clicks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1955–1959. IEEE.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.

Anshul Mittal, Kunal Dahiya, Sheshansh Agrawal, Deepak Saini, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. Decaf: Deep extreme classification with label features. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 49–57.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale multi-label text classification—revisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 world wide web conference*, pages 1063–1072.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Dou Shen, Ying Li, Xiao Li, and Dengyong Zhou. 2009. Product query classification. In *Proceedings of the 18th ACM conference on information and knowledge management*, pages 741–750.

Andrew Turpin and Falk Scholer. 2006. User performance versus precision measures for simple search tasks. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18.

Chen Wu, Ming Yan, and Luo Si. 2017. Ensemble methods for personalized e-commerce search challenge at cikm cup 2016. *arXiv preprint arXiv:1708.04479*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. Sgm: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in Neural Information Processing Systems*, 32.

Hang Yu and Lester Litchfield. 2020. Query classification with multi-objective backoff optimization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1925–1928.

Hongchun Zhang, Tianyi Wang, Xiaonan Meng, Yi Hu, and Hao Wang. 2019. Improving semantic matching via multi-task learning in e-commerce. In *eCOM@ SIGIR*.

Junhao Zhang, Weidi Xu, Jianhui Ji, Xi Chen, Hongbo Deng, and Keping Yang. 2021. Modeling across-context attention for long-tail query classification in e-commerce. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 58–66.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

A Data Process for CIKM Cup 2016 Dataset

We extract pairs of $\langle query, productID \rangle$ from the file named `train-queries.csv`, which includes user sessions from e-commerce search engine logs. Only the query-full cases are selected and each query is represented as a list of hashed tokens. The queries which appear in more than $thres_q$ sessions are regarded as frequent queries while others are long-tail queries. We then map the pairs of $\langle query, productID \rangle$ to $\langle query, categoryID \rangle$ according to the content of file `product-categories.csv` and denote the set of pairs as S. We denote the number of occurrences of query q in S as f_q and the number of $\langle q, c \rangle$ pairs in S as $f_{q,c}$ where c is the category. The pair $\langle q, c \rangle$ is regarded as a positive example when it meets the requirements of both absolute number and relative ratio, which are $f_{q,c} > thres_N$ and $f_{q,c} > \frac{f_q}{M}$. We collect all positive examples as set S^P and then generate negative examples from S^P. For each $\langle q, c \rangle$ in S^P, we replace the q and c by a random query q' and category c' respectively and repeat R times. Finally, we randomly select L long-tail queries and then take all the corresponding pairs in positive set and negative set as the test set, while the rest of pairs are training set.

Considering there is no category description in the original dataset, we group the products based on their categories and take the top-K most frequent tokens in product names as the description of the category (the hashed product name tokens are obtained from `products.csv`).

The values of above parameters are listed in Table 7. The hyper-parameters of the model are the same as Section 4.1.

Table 7: Parameters in process of CIKM Cup 2016 dataset.

Name	value	Name	value
$thres_q$	5	R	4
$thres_N$	2	L	3000
M	16	K	10