Contrastive Fusion of Knowledge for Product Search Relevance

Anonymous EMNLP submission

Abstract

With the development of Knowledge Graph (KG) in e-commerce, more and more downstream tasks can be benefited. Product Search Relevance (PSR) is a fundamental and essential task in e-commerce. Due to the lack of information caused by short text in this task, external knowledge is usually necessary to help model understand query intent and product essence. Especially, in the life-service scenario, various products from thousands of categories with different synonyms make this task more challenging. Thus, how to effectively utilize KG to improve model performance on PSR is a problem worth exploring. However, conventional fusion approaches underutilize the internal relationship knowledge in KG, yielding sub-optimal performance. In this paper, we propose a novel framework to fuse knowledge in KG into PSR models based on contrastive learning called Knowledge Graph Contrastive Fusion (ConFu), which outperforms other baseline methods. Besides, in order to facilitate resolutions of this problem, we also release a Chinese PSR dataset along with specialized KG in our real life-service e-commerce scenario. [1]

1 Introduction

Knowledge Graph (KG) has been deployed in a large amount of companies nowadays. In e-commerce scenario, KGs can benefit a lot of downstream tasks, such as Product Search Relevance (PSR), recommendation, Question Answering (QA) etc. In this paper, we focus on utilizing KG to improve PSR task in e-commerce.

In life service scenario, huge variety of products makes PSR task more challenging. In such scenario, product category covers almost all products you can see in all markets and shops in one city, including all kinds of fruits, vegetables, snacks etc. One of the most challenging problem in such scenario is that you may not know what a product exactly is with only product name as they may have various synonyms, especially in Chinese. Apart from synonyms, typos also add difficulty to this task. All these challenges make external knowledge necessary for this task.

The core idea of fusing KG into PSR model is that with the help of KG, query intent and product essence can be comprehensively understood by the model, which means KG can help model understand objective things better. The overview of PSR task with KG in e-commerce is depicted in Figure **??**. Generally speaking, there is usually a hierarchical taxonomy to organize items and concepts in e-commerce KG. This taxonomy is usually manually defined and coarse-grained, for example, "Vegetable / Rhizomes / Potato" is a three-level path in taxonomy. To better understand user needs and describe essence of products, more fine-grained e-commerce related concept graph is usually constructed. For these concepts in KG, some necessary relations, such as *synonym* and *related_to*, are also defined to clarify the relationship between different concepts. Generally speaking, this KG framework is adopted in AliCoCo (**??**), Xiaomi product KG (**?**) and our scenario etc., although different companies have different design for their own scenario. Specifically, in our scenario, the product taxonomy is a manually constructed tree structure connected by isA relationships. Apart from the taxonomy, a fine-grained e-commerce concept graph is also constructed.

KG contains different kinds of knowledge to be mined. Lots of hidden knowledge, including hypernyms, hyponyms, synonyms, and KG structure, is waiting to be exploited. In downstream tasks, knowledge is usually utilized in a relatively straightforward and shallow way (**????**), such as string concatenation, and embedding fusion, etc.

[1] The code and part of the data is temporarily released at: https://anonymous.4open.science/r/confu_repo/, while the full version of the code and data will be released after the paper is accepted.

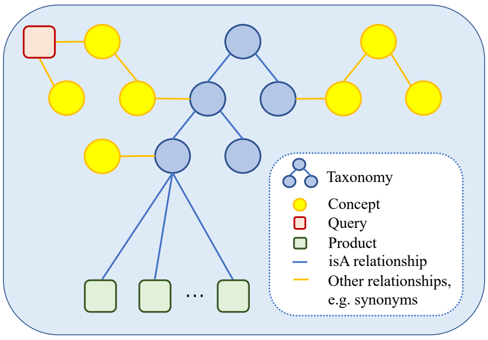

Figure 1: Overview of Product Search Relevance task with Knowledge Graph. In such task, given a pair of query and product, a PSR model needs to measure the relevance between the query and product with the help of KG.

These knowledge fusion methods are good at fusing positive relationship knowledge to tell model what an unknown entity or concept is, but there are still problems that cannot be solved in our scenario. One limitation of these methods is that they have less ability to make use of negative relationship knowledge in KG to teach model to distinguish between two vague concepts or entities well when positive relationship knowledge is insufficient. This means relationships between concepts or entities in KG are not fused well into model. For example, for the query product pair "Green onion" and "Seaweed-flavored shallots green onion 65g" (a kind of snack), even though the positive relationship knowledge of the product in KG "Snacks / Fried food & puffed food / Others", is concatenated to the product name to tell the model they are not the same thing, the similarity between the query and product is still high, which means model ignores the fused positive relationship knowledge. This kind of knowledge ignorance problem is usually caused by lexical domination, which means the semantic of product is dominated by text that appears in both query and product. In this case, "green onion" dominates model's understanding of this product. This problem is more significant in our scenario as query and product name are usually short.

In order to solve the problem mentioned above, negative relationship knowledge is necessary to be further utilized to tell the model what the product is not. Thus, we design a novel KG fusion method with contrastive learning called Knowledge Graph **Con**trastive **Fu**sion (ConFu) to utilize different kinds of relationship knowledge in KG better. Our core idea is that model can learn relationships among query/product and other concepts or entities in KG through contrastive learning task to understand user intent and product essence. Specifically, with query-side contrastive learning, ConFu can learn knowledge like synonyms and hypernyms of query and learn to distinguish irrelevant queries. With product-side contrastive learning, ConFu can learn relationships among different products and have a stronger ability to distinguish ambiguous products. For the case we mentioned above, ConFu can solve the problem using contrastive learning by embedding "green onion" and the product into different regions in semantic space. Another advantage of ConFu is that it does not need external knowledge any more during inferece stage.

Generally speaking, our key contributions are summarized as follows:

- **A KG based PSR Dataset**. In this paper, we release a novel PSR dataset with KG in life-service scenario to call for better solutions. As far as we know, we are the first to release such a Chinese PSR data set with KG.

- **A Novel KG Fusion Method**. We propose a novel fusion method to inject knowledge in KG into PSR model. With query-side contrastive fusion and product-side contrastive fusion, knowledge in KG can be effectively fused into model to help model understand user need and product essence.

- **Effectiveness**. According to our experiments, ConFu can successfully fuse KG into PSR model and benefits PSR task. ConFu outperforms all other baseline methods on KGPSR task by more than 2 points on AUC score. Visualization and case study also demonstrate the effectiveness of ConFu.

2 Knowledge Graph based Product Search Relevance Task

The KGPSR task focuses on utilizing KG to improve PSR task. PSR task is clarified as follows. Given a query-product pair $< q, p >$, q is user query and p is the name of product. The goal of PSR task is to compute the relevance score of each query-product pair.

KGPSR task is a simple variant of PSR task. In KGPSR task, q and p are linked on one KG $G = \{V, E\}$, where V is a set of e-commerce concept nodes and E is a set of relationships. The goal of KGPSR task is to utilize KG as much as possible to benefit PSR task in e-commerce.

3 Knowledge Graph based MMM Product Search Dataset

In this section, we introduce the KG based MMM[2] Product Search dataset. We first introduce the background of releasing such a dataset and then introduce the construction of this dataset.

3.1 Dataset Background

Different from other e-commerce platforms like Amazon, Taobao, JD etc., MMM is a life-service e-commerce platform focusing on retailing of services and retailing of products. As a life-service e-commerce platform, it is closely related to people's daily life and is gradually becoming "the largest convenience store in China". In the "Top 100 List" of China's e-commerce in 2021, MMM ranked second with a total company value of 1,128.357 billion yuan, second only to Alibaba. Figure **??** is the user interface of MMM APP and its services include takeaway, vegetables, fruits, products in supermarkets and convenience stores, flowers, medicines and any other products that we can see in our daily life. Various kinds of products make PSR task in our secnario more challenging. Besides, as life-related products is closely related to people's daily life, one product can have a lot of synonyms, which adds more difficulty to PSR task. Thus, external knowledge is especially necessary for this task in our scenario. Against this background, we construct and release a PSR dataset with KG.

3.2 Dataset Construction

To evaluate the performance of PSR models with KG, especially in life-service scenario with various product expressions, we construct a challenging dataset called MMM Product Search (MMMPS) from our real business logs.

Firstly, we collect <query, product> pairs with their exposure frequency from search logs. To alleviate noisy logs, we filter the queries using a query corpus that auto-collected from logs and product titles with NER. Queries appear in query corpus will be kept and queries do not appear in the corpus will be dropped. Note that, the size of the query corpus is much larger than the size of the concept graph. Then, we randomly sample a subset

[2]MMM (anonymized for blind review) is a major e-commerce platform in China.

Figure 2: User interface of MMM.

of pairs by their exposure frequency to ensure pairs with both head and tail frequencies are maintained. To further focus on the hard scenarios, we use a group of models to validate these sampled pairs and sample a hard dataset from different buckets with different consistency scores or average validation scores to ensure both easy and hard pairs are sampled. Finally, we recruit experienced annotators to manually label the dataset. Data groups with over 95% accuracy in quality checking are used in our final dataset.

Apart from the label of <query, product> pair, we also provide a subset of the concept graph associated with queries and a subset of the hierarchical category taxonomy associated with products. The concept graph is a fine-grained knowledge graph that contains hypernym and synonym relations among concepts, e.g. "murphy" is a synonym of "potato". The hierarchical category graph is a coarse-grained KG that contains three level categories, e.g. "Vegetable / Rhizomes / Potato". Figure **??** shows a fragment of our three-level tree-structured taxonomy. Note that although both KGs have been reviewed by annotators to ensure the quality, noisy data cannot be completely avoided, and that is the real scenario we are facing.

4 Knowledge Graph Contrastive Fusion

In this section, we will introduce our proposed ConFu method for PSR task with KG. Figure **??** is the overview of ConFu, which adopts a two-tower architecture for faster retrieval and ranking in real-time search. Different from previous knowledge fusion methods, ConFu uses contrastive learning to

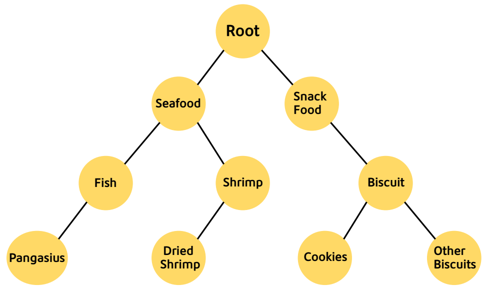

Figure 3: A Fragment of Taxonomy.

inject knowledge from KG into the model during fine-tuning stage. ConFu is a multi-task learning framework and there are three tasks: one classification task, one query contrastive learning task and one product contrastive learning task. ConFu is based on Sentence-BERT (SBERT) (?), which uses a classification objective function during fine-tuning and computes similarity score in the inference stage. Two contrastive learning tasks are designed to inject query-side knowledge and product-side knowledge in KG into the model respectively to help model understand user intent and product essence.

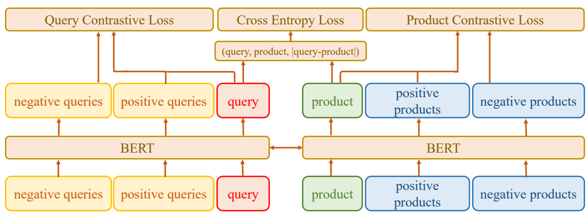

Figure 4: Overview of *ConFu*.

4.1 Query Contrastive Knowledge Fusion

In the query side, we utilize KG to explore positive and negative samples as knowledge for Query Contrastive Knowledge Fusion (QCKF). As a query is usually short and most queries can perform as concepts in KG, the concept graph in KG can be the main source of knowledge for query. The basic idea is that synonyms, hypernyms, hyponyms and other relevant concepts of query can be selected as its positive samples. Irrelevant concepts or other concepts that are far from the concept can be selected as negative samples for the query. With these positive and negative samples, PSR model will be able to learn knowledge behind KG to understand user query better.

Specifically, for each query q, one positive sample q_{pos} and one negative sample q_{neg} are selected from KG. Other queries in the same batch will be treated as random negative samples. The query contrastive loss is displayed as Equation **??**.

$$\begin{aligned}\mathcal{L}_{QCKF} &= -\log \frac{\exp(\mathbf{q}\cdot\mathbf{q}_{pos}/\tau)}{Q_{Pos}+Q_{Neg}}.\\ Q_{Pos} &= \exp(\mathbf{q}\cdot\mathbf{q}_{pos}/\tau),\\ Q_{Neg} &= \exp(\mathbf{q}\cdot\mathbf{q}_{neg}/\tau) + \sum_{i=1}^{B-1}\exp(\mathbf{q}\cdot\mathbf{q}_{neg,i}/\tau),\end{aligned} \tag{1}$$

where B is batch size.

4.2 Product Contrastive Knowledge Fusion

In KG, products are usually organized by a hierarchical taxonomy. Thus, we mainly utilize this taxonomy to generate positive and negative samples for products, which means homonyms are selected as positive samples and products in other leaf nodes will be treated as negative samples. With both positive and negative samples sampled from KG, different relationship knowledge and KG structure knowledge can be injected to our model.

Referring to Self-Tuning(?), We adopt Group Contrastive Learning to perform Product Contrastive Knowledge Fusion (PCKF). For each product, we selected a group of positive samples from KG and use products from other groups in the same batch as negative samples. These contrastive groups are arranged in each batch. One advantage of such kind of arrangement is that there will be no false negative samples given labels (KGPs) are correct.

$$\begin{aligned}\mathcal{L}_{PCKF} &= -\frac{1}{D}\sum_{d=1}^{D}\log \frac{\exp(\mathbf{p}\cdot\mathbf{p}_{pos,d}/\tau)}{P_{Pos}+P_{Neg}}.\\ P_{Pos} &= \sum_{d=1}^{D}\exp(\mathbf{p}\cdot\mathbf{p}_{pos,d}/\tau),\\ P_{Neg} &= \sum_{i=1}^{B-D}\exp(\mathbf{p}\cdot\mathbf{p}_{neg,i}/\tau),\end{aligned} \tag{2}$$

where D is the number of positive samples in a contrastive group, B is batch size.

4.3 Partial Multi-task Training Strategy

For those samples hard to find knowledge to fuse due to long tail effect, instead of throwing away these data, we choose to use a training strategy called Partial Multi-task Training Strategy to keep all data samples.

In detail, we arrange samples with contrastive knowledge into contrastive batches and samples without contrastive knowledge into non-contrastive batches. For contrastive batches, the overall loss of ConFu is:

$$L = L_{CE} + \lambda_q L_{QCKF} + \lambda_p L_{PCKF}, \quad (3)$$

For non-contrastive batches, λ_q and λ_p will be set to zero, which means the loss will be the simplified as traditional Cross Entropy Loss.

$$L = L_{CE} \quad (4)$$

5 Experiments

In this section, we discuss the experiments settings of ConFu. We report the result of ConFu and other baseline methods on two datasets. To further demonstrate the effectiveness of ConFu, we use TSNE (?) to visualize query embedding and product embedding by models. Case study is also performed to show how ConFu solves problems in our scenario.

5.1 Datasets

In order to demonstrate the effectiveness of ConFu, we perform experiments on two PSR datasets.

5.1.1 MMMPS

MMM Product Search (MMMPS or MMM) is the dataset from our own life-service scenario and will be released. MMMPS dataset is a binary dataset and for each query-product pair, the corresponding label is relevant (1) or irrelevant (0). For this data set, we use Area Under the receiver operating characteristic Curve (AUC) to evaluate the performance of different models.

Table 1: Dataset statistics of MMMPS dataset

# training	# dev	# test	# 1st cate.	# 2nd cate.	# 3rd cate.[1]
100k	10k	19,959	27	199	1,577

[1] The hierarchical taxonomy of MMMPS dataset is a three-level tree-structured taxonomy.

Statistics of MMMPS dataset is listed in Table **??**. In MMMPS dataset, there are 119,845 different kinds products from 1,577 different categories, which means most products appear few times in the dataset and the ratio of unique products in the dataset can reach around 92%. MMMPS covers most product categories we can see in all kinds of shops in our daily life, which makes it a challenging dataset. For most queries, their connected concepts like synonyms and hypernyms in KG are also provided.

5.1.2 WANDS

WANDS dataset (?) is also a product search relevance dataset released by wayfair. In this data set, hierarchical category and product class are provided for products. Query class is provided for query. The hierarchical categories and product/query classes can be taken as KG of this dataset. There are three labels in this dataset: Exact, Partial and Irrelevant. In order to make it consistent with MMMPS dataset, we drop the Partial labels and make it a binary class dataset, denoted as WANDS-binary. We also drop duplicate data and split data into training set, dev set and test set. The evaluation metric of WANDS-binary dataset is AUC. Note that although problems we claimed are not significant in this dataset, it is still a good dataset to evaluate and analyze different methods.

5.2 Baselines

5.2.1 Sentence-BERT

Sentence-BERT (SBERT) (?) is a two-tower or bi-encoder structure model, which is widely adopted in industry scenarios including ours. The bi-encoder structure is very efficient due to its ability to embed query and product separately.

5.2.2 SBERT+KGP

One of the most direct and common approaches to utilize KG is to concatenate the knowledge in text form from KG on the original text (**??**). In PSR task, we implement such a method by concatenating Knowledge Graph Path (KGP) of query and product. In SBERT+KGP, knowledge in KG of specific query and product is concatenated to query and product respectively.

5.2.3 SBERT+KGE

Another commonly used method to utilize KG is fusing knowledge in KG through Knowledge Graph Embedding (KGE), which shows competitive performance in (**???**). The advantage of KGE is that it concerns the overall structure of KG. Thus, we adopt this method as one of our baseline methods. In SBERT+KGE, KGE of nodes in KG trained by PairRE (?) is concatenated to the embedding of query and product.

5.3 Implementation Details

BERT-related models are initialized from the pre-trained Google BERT-base (Chinese) and tuned with 2e-5 learning rate and 128 batch size. The optimizer is AdamW. In model selection, for each model, we trained enough epochs for each model and select the best performing checkpoint on the dev set to be the final trained model. Each experiment is repeated three times with different random seeds and corresponding scores are averaged.

5.4 Overall Results

The overall experiment results on two datasets are displayed in Table **??**. Result on both dev set and test set is reported.

Table 2: The AUC score of our *ConFu* framework and other baseline methods on MMMPS and WANDS-binary. KGP and KGE stands for concatenating Knowledge Graph Path and Knowledge Graph Embedding respectively. Query/product only means only Query/Product Contrastive Knowledge Fusion is adopted in ConFu.

Methods	MMM_{dev}	MMM_{test}	$WANDS_{dev}$	$WANDS_{test}$
SBERT	0.764	0.765	0.955	0.960
SBERT+KGP	0.768	0.771	0.956	0.962
SBERT+KGE	0.756	0.756	0.958	0.962
ConFu	0.800	0.800	0.980	0.981
ConFu (query only)	0.771	0.769	0.982	0.984
ConFu (product only)	0.779	0.779	0.927	0.927
ConFu+KGP	0.797	0.793	0.944	0.947
ConFu+KGE	0.795	0.795	0.981	0.982

5.4.1 Results on MMMPS

On MMMPS dataset, our experiment results illustrate the effectiveness of ConFu compared with vanilla SBERT, SBERT+KGP and SBERT+KGE. According to the experiment result of SBERT+KGP and SBERT+KGE, we can see that simply concatenating KGP or KGE can bring limited improvement on model performance. With contrastive learning, ConFu can effectively inject knowledge in KG into PSR model. Specifically, QCKF and PCKF in ConFu can complement each other in such case where both query and product are ambiguous. Note that though ConFu can also be applied on Model+KGP or Model+KGE, our experiment result shows that ConFu only is sufficient for effective knowledge fusion.

Besides, we also perform T-test on our results with a critical value 0.05 and the following conclusions can be made:

1. ConFu and its variants (ConFu+KGP and ConFu+KGE) are significantly better than baseline methods with p-value 0.004, 0.016 and 0.025 on test set respectively, which demonstrates the effectiveness of ConFu.

2. Though ConFu (query only) and ConFu (product only) cannot significantly outperform baseline methods, they can complement each other well and so that ConFu can achieve significant improvement.

3. Confu, ConFu+KGP and ConFu+KGE have similar performance, which means ConFu is sufficient for effective knowledge fusion.

5.4.2 Results on WANDS-binary

On WANDS-binary dataset, ConFu can also outperforms baseline methods. However, one difference is that PCKF fails on WANDS-binary dataset and the performance gain is basically contributed by QCKF. One possible reason why PCKF of ConFu does not work on WANDS-binary is that product name in WANDS-binary has already contained information in taxonomy and products in each categories share similar names, and external knowledge from Knowledge Graph is not that informative and necessary enough. For instance, the KGP of products “totham rocking chair”, “ossu outdoor rocking chair with cushions”, “sonnenberg rocking chair” and “fordyce rocking chair” is “Outdoor / Outdoor & Patio Furniture / Outdoor Seating & Patio Chairs / Patio Seating / Patio Rocking Chairs & Gliders” and “rocking chair” appears in all four products. This point can also be demonstrated by product embedding visualization in the following section. Additionally, WANDS-binary dataset is simpler than MMMPS dataset as MMMPS dataset contains more kinds of products and more fine-grained Chinese introduces more ambiguity while WANDS-binary dataset contains only Furniture & Home products. Thus, additional PCKF can easily cause overfitting on product embedding learning with coarse-grained taxonomy on WANDS-binary with well self-separated products. Different from product, query in WANDS-binary is usually short and needs more knowledge, so QCKF can effectively improve model’s understanding of user query.

Similarly, the low AUC score of ConFu+KGP on WANDS-binary is also caused by such kind of overfitting. Besides, because the taxonomy path of product in WANDS-binary is relatively longer, it will also causes nodes in upper layers appears in most products. Directly concatenating the whole taxonomy path after product name will introduce irrelevant knowledge or noise and interfere with contrastive learning. Instead, if we only concatenate the leave node (the last category on path), the problem can be alleviated.

Generally speaking, combined with our T-test results, the following conclusions can be made:

1. ConFu, ConFu (query only) and ConFu+KGE are significantly better than baseline methods with p-value 0.012, 0.010 and 0.015 on test set respectively, which demonstrates the effectiveness of ConFu.

2. ConFu+KGE is not significantly better than ConFu, which means ConFu only can satisfy our needs.

3. ConFu+KGP performs poorly due to the long KGP problem and overfitting problem caused by information overlapping between product name and KGP. The overfitting problem is also why ConFu (product only) performs poorly. This means QCKF and PCKF in ConFu can handle datasets with ambiguous and difficult cases better, QCKF or PCKF can be overfitted by datasets with easy queries or products. Note that though PCKF fails on WANDS-binary, ConFu can still achieve good performance, which means ConFu has fault-tolerance ability to a certain extent.

5.5 Embedding Visualization in Knowledge Graph Fusion

In order to further demonstrate the effectiveness of ConFu, we display the visualization of embedding learned by different models on two datasets. Product embedding visualization is performed on two datasets. Because query class for each query is provided in WANDS-binary dataset, query embedding visualization is provided based on this dataset.

5.5.1 Product Embedding Visualization

Figure ?? (a) is the product embedding on MMMPS dataset. Comparing product embedding learned by SBERT and ConFu, we can see that products in MMMPS dataset are very ambiguous and ConFu can effectively inject knowledge in KG into model by optimizing product embedding based on taxonomy.

Figure ?? (b) is the product embedding on WANDS-binary dataset. Different from MMMPS

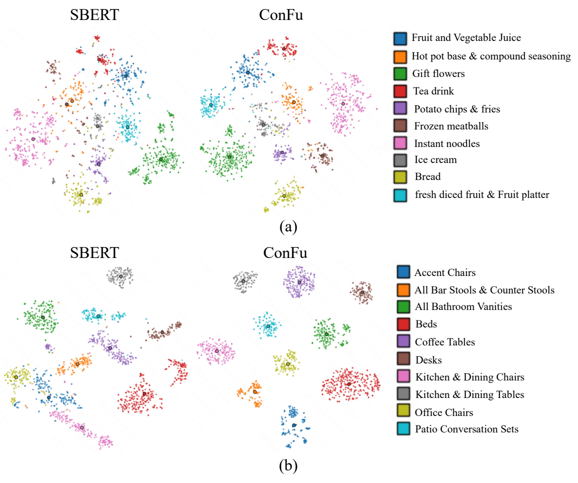

Figure 5: Product embedding visualization. (a): Product embedding visualization of MMMPS dataset. (b): Product embedding visualization of WANDS-binary dataset.

dataset, it is easy to see that product embedding in WANDS-binary is less ambiguous than MMMPS's and that's because most product names already contain information in taxonomy as mentioned earlier. Thus, with ConFu, products from different categories can be easily separated. This also demonstrates that PCKF is not necessary for WANDS-binary dataset.

Generally speaking, Figure ?? demonstrates the ability of product embedding learning of ConFu with PCKF. In other words, taxonomy structure knowledge or relationship knowledge can be effectively fused into ConFu to some extent with PCKF.

5.5.2 Query Embedding Visualization of WANDS-binary

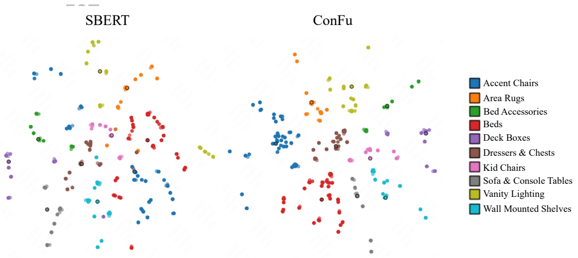

Figure 6: Query embedding visualization of WANDS-binary data set.

Figure ?? is the query embedding on WANDS-binary dataset. According to the query embedding learned by SBERT, it is easy to see that the query embedding of WANDS-binary is chaotic. For instance, "Accent Chair" queries appears in both upper left corner and lower right corner in the figure. ConFu can optimize query embedding by clustering queries from the same query class, making queries with the same query class appear in the same region. Thus, for ConFu, "Accent Chair" queries appear in the same region in Figure **??**.

5.6 Case Study

Table **??** shows some bad cases that cannot be solved by baseline methods solved by ConFu on our MMMPS dataset. Note that baseline methods in this section refers to SBERT and SBERT+KGP since they have better performance. Case 1, 2 and 5 are typical examples of knowledge ignorance problem caused by lexical domination. Specifically, query exactly appears in product name, making it difficult for model to distinguish the difference between query and product. Even though taxonomy path is provided, baseline methods sometimes ignore concatenated knowledge and still cannot perfectly solve these cases, meaning the query-product case is dominated by co-occuring words. With contrastive learning, ConFu can utilize both positive and negative relationship knowledge in KG by sampling positive and negative samples from KG and learn the essence of different products, so that the problem can be alleviated. The other three cases illustrate ConFu's ability to learn synonym knowledge in KG. With QCKF and PCKF, ConFu can inject such kind of knowledge fundamentally through embedding optimization. For instance, ConFu can learn that "Sesame cai" and "Stinky cai" are two synonyms of arugula so that Case 4 can be solved. Case 3 and 6 are similar to Case 4.

6 Related Work

Our work focuses on solving problems in PSR task with KG using contrastive learning, so it is relevant to PSR, Knowledge Fusion, Term-to-Term Similarity with KG and contrastive learning.

6.1 Product Search Relevance

PSR is a very important task in e-commerce platforms. For this task, a two-tower architecture is widely adopted to perform candidate retrieval and ranking etc. With a two-tower architecture, query and product can be encoded separately. Thus, products can be encoded offline and search can be accelerated in real-time services(**?**).

In order to improve PSR task, many researchers try to optimize this task in different ways. Priyanka Nigam et al. (**?**) study the problem of semantic matching in product search and train a deep learning model for semantic matching using customer behavior data to overcome problems of inverted index. Zheng Liu et al. (**?**) research on the generalization ability of product search models by performing text similarity pre-training on search click log. Han Zhang et al. (**?**) designs a query tower with multi-heads and utilizes user information to achieve better personalized and semantic retrieval. Multi-Grained Deep Semantic Product Retrieval (MGDSPR) model (**?**) is proposed to dynamically capture the relationship between user query semantics and personalized behaviors. They train a deep learning model for semantic matching using customer behavior data. Nurendra Choudhary et al. (**?**) formulates PSR as a multi-class classification problem and proposes a graph-based solution to classify a given query-item pair as exact, substitute, complement, or irrelevant (ESCI). Han Zhang et al. (**?**) propose Poeem to unify embedding learning and index building this two separate steps to reduce additional indexing time and improve retrieval accuracy. Shaowei Yao et al. (**?**) focus on how to train a product relevance model from the weak supervision of click-through data. These works solved different problems in PSR task, but work focusing on injecting KG knowledge into PSR task is still relatively less.

6.2 Knowledge Graph Fusion

For some tasks, especially domain-specific task, external knowledge is necessary. KG contains a wealth of knowledge and is beneficial to those downstream tasks. How to effectively fuse knowledge into model becomes a popular research direction in recent years. KBERT (**?**) injects triples into the sentences as domain knowledge. Ning Bian et al. (**?**) enhance Commonsense QA model via concatenation with knowledge extracted from KG. Following these works, we use KGP as one of our baseline models. KI-BERT (**?**) infuse knowledge context from multiple knowledge graphs for conceptual and ambiguous entities into transformer-based language models during fine-tuning. In e-commerce field, researchers from Alibaba (**?**) propose a multi-task encoder-decoder Knowledge Graph Embedding (KGE) framework to provide representations for nodes and edges from AliCoCo2 and so that downstream tasks can be improved. Inspired by their solution, we adopt KGE-based method as one of our baseline methods.

Table 3: Case study.

Query	Product	Taxonomy Path	Label
Cake (蛋糕)	Oreo Lemon Cheesecake (芝士蛋糕) Flavor 95g/box	Snack Foods / Biscuit / Crispy Biscuits & Sandwich Biscuits	0
Black-bone chicken	4 farm Black-bone chicken eggs about 250g/serving	Raw Meat & Raw Poultry & Raw Eggs / Raw Eggs / Chicken Eggs	0
Sea rice (海米)	About 250g dried shrimp	Seafood / Shrimp / Dried Shrimp	1
Sesame cai (芝麻菜)	Stinky cai (臭菜) 200g	Vegetables & Soy Products / Leafy Vegetables / Other Leafy Vegetables	1
Roast Duck	Vegetarian Beijing Roast Duck 100g	Snack Foods / Braised and Spicy Food / Soy Products	0
Niuzhan (牛展)	[2 catties of cooked beef tendon meat] Halal Gumei five-spice sauce marinated yellow beef leg meat ready to eat for fitness	Cooked food & Fresh Food / Delicatessen / Braised	1

[1] Sea rice is another name of dry shrimp in Chinese.
[2] Sesame cai and Stinky cai are synonyms of arugula in Chinese. Cai means "a kind of vegetable".
[3] Niuzhan is another name of beef tendon in Chinese.

6.3 Term-to-Term Similarity with Knowledge Graph

Term-to-Term Similarity task is similar with PSR task and some researchers also try to utilize KG to assist this task. Roy Rada et al. (**?**) measure the similarity between two terms in KG with the length of the shortest path connecting the two terms. Philip Resnik (**?**) uses information content of the least common ancestor node of the two terms in the tree-structured taxonomy to evaluate their semantic relatedness. Marco A. Alvarez (**?**) measuring the semantic similarity between pairs of words by constructing a rooted weighted graph. Peipei Li et al. (**?**) research on how to compute term similarity by large probabilistic isA knowledge. They map two terms into the concept space, and compare their similarity in the space. Unfortunately, these methods mainly rely on traditional algorithms and are relatively far away from current PSR technologies. Besides, query and product are naturaly different from simple terms and these methods rarely concerns the application in e-commerece scenario, so they can hardly be smoothly applied on current PSR task.

6.4 Contrastive Learning

As an effective way to optimize embedding, Contrastive Learning has been successfully utilized in different fields. In SimCSE (**?**), authors designs a simple yet effective contrastive learning method using dropout. Similarly, ConSERT (**?**) explores various effective text augmentation strategies to generate views for contrastive learning. Self-Tuning (**?**) designs pseudo group contrastive learning to help simultaneously exploring both labeled and unlabeled data. Following their work, the group contrastive learning is also adopted in ConFu. Zhiping Luo et al. (**?**) use contrastive learning to help learn KGE. Yuhao Yang et al. (**?**) utilize contrastive learning with KG to alleviate the information noise for KG-enhanced recommendation systems.

7 Conclusion

In this paper, we explore an effective method to inject KG knowledge into PSR models called ConFu. With Confu, relationship knowledge in KG can be better fused into PSR model. Our proposed ConFu method can perform well on PSR task and we are looking forward to see that ConFu can be successfully transferred to other domains or tasks in the future, which is also what we plan to do next. Besides, in order to collect better solutions to this task, we release a PSR dataset called MMMPS in life-service scenario with specialized knowledge from KG.

A Example Appendix

This is a section in the appendix.