Unsupervised Synonym Extraction for Document Enhancement in E-commerce Search

Hanqing Lu∗
luhanqin@amazon.com
Amazon.com Inc
Palo Alto, USA

Yunwen Xu∗
yunwenx@amazon.com
Amazon.com Inc
Palo Alto, USA

Qingyu Yin∗
qingyy@amazon.com
Amazon.com Inc
Beijing, China

Tianyu Cao
caoty@amazon.com
Amazon.com Inc
Palo Alto, USA

Boris Aleksandrovsky
skelab@amazon.com
Amazon.com Inc
Palo Alto, USA

Yiwei Song
ywsong@amazon.com
Amazon.com Inc
Palo Alto, USA

Xianlong Fan
fanxl@amazon.com
Amazon.com Inc
Beijing, China

Bing Yin
alexbyin@amazon.com
Amazon.com Inc
Palo Alto, USA

ABSTRACT
The vocabulary gap between search queries and product descriptions is an important problem in modern e-commerce search engines. Most of the existing methods deal with the vocabulary gap issues by rewriting user-input queries. In this work, we describe another way to address vocabulary gap issues in the e-commerce search systems. In particular, we propose an unsupervised synonym extraction framework for document enhancement. Comparing to existing methods, the purposed framework has two main differences: 1) instead of extending and rewriting user-input queries, we make the synonym tokens searchable by adding them into the text descriptions of products; 2) the whole process is unsupervised and doesn’t require any human labels. A two-phase unsupervised synonym discovery framework is proposed for extracting synonym rules from the search log data. We demonstrate the effectiveness of our approach by two experiments: 1) we do online A/B testing experiments in multiple countries, which show significant improvements in key business metrics; 2) we conduct a human audit to evaluate the quality of the extracted synonym rules, which indicates that 85.5% of the extracted rules are high quality pairs.

CCS CONCEPTS
• Information systems → Query intent; • Computing methodologies → Information extraction.

KEYWORDS
synonym, e-commerce, query understanding

∗Both authors contributed equally to this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WWW KMEcommerce’21, April, 2021, Virtual Event
© 2021 Association for Computing Machinery.

1 INTRODUCTION
E-commerce customers express their purchase intents in several ways, some of them may use a different vocabulary than that of the product description [15]. For instance, customers could use “cat rx food” to express their needs for “cat prescription food”. Though the word “rx” and “prescription” have the same meaning in the given context, the query “cat rx food” returns much less relevant products due to the vocabulary gap between query terms and product descriptions (as shown in Figure 1). Admittedly, the vocabulary gap issue heavily hurts the performance of keyword-based e-commerce search systems which perform in a “matching and ranking” manner. Typically, in the matching stage, a search engine relies on lexical similarity between product descriptions and customer keywords. In the above example, only “prescription” is indexed for most of the relevant products, but not “rx”. In other words, these products are not searchable with token “rx”. Therefore, the input query “cat rx food” can not find these products, which leads to matching failure and bad customer experience.

One straightforward way to alleviate this vocabulary gap issue is to rewrite query terms by using synonyms [16]. In this way, relevant products that match the customers’ search intent can be retrieved. In the previous example, if we replace the term “rx” in the query with its synonym “prescription”, more related products can be retrieved and shown to customers. In literature, query rewriting techniques have been widely explored [1, 11, 15, 25, 27]. Some of these methods utilize user behavior information in the form of click-through rate [5], query co-occurrence [12], and co-clicked query similarity [1, 12] to rewrite queries, and some others employ external knowledge base [17, 19, 22] to expand search keywords. Recently, [16] propose a novel query rewriting method by utilizing
We employ a BERT-based embedding model to filter out irrelevant n-gram pairs. This provides us with the synonym rules. To ensure semantic similarity of the n-gram terms in candidate pairs, we employ a BERT-based embedding model to filter out the irrelevant n-gram pairs. This provides us with the synonym rules.

For indexing, we apply the synonym rules to expand product title and the target word/phrase is from the related query. Therefore, by indexing additional informative tokens extracted from the related queries, the vocabulary mismatch between products and queries can be alleviated. At the first place, we generate highly engaged product title-query pairs from customer query logs. After that, we extract n-gram pairs (one from product title and the other from the associated query) as synonym candidates based on both the co-occurrence frequency and the transition probability. To ensure semantic similarity of the n-gram terms in candidate pairs, we employ a BERT-based embedding model to filter out the irrelevant n-gram pairs. This provides us with the synonym rules.

Figure 1 gives an overview of the proposed synonym discovery framework. In the high-level, our synonym rules are extracted from two different data sources: dictionaries and query logs. In this work, we employed two dictionaries to generate general synonym candidates: the WordNet synset [18] and Wikipedia redirects data. For query logging mining method, we generated synonym candidates by utilizing the related query-product pairs. In order to better utilize the customer feedback, we first characterized the relatedness of each query and product by using customer behavior information, i.e., click and purchase actions and then selected related query-product pairs. In this way, we can collect plenty number of related query and product pairs and then employ these parallel data to generate n-gram pairs, where the first n-gram term comes from the customer keywords and the other one comes from the correspondence product title. Among these pairs, we only selected ones with high co-occurrence frequency and transition probability as the candidate synonym pairs. Finally, the candidate synonym pairs
generated by using these two data sources were combined to be the ultimate candidate set.

In order to capture the similarity of n-gram terms in candidate synonym pairs at the semantic level, we employed BERT [6] to generate the representation vectors of n-gram tokens. Specially, we trained the BERT model by using the data collected from search queries and product titles. For each n-gram pair, we fed them into the BERT and apply the average-pooling over the output hidden states to generate correspondence representation vectors. We then calculated the cosine similarity between terms in each n-gram pair and kept the ones with scores bigger than the specific threshold. In this way, we can collect millions of synonym rules and apply these rules for document enhancement. In practice, we expanded the definitions of product by applying synonym rules on each word in the product title and made these additional words searchable. Consequently, these additional words can be utilized in the matching stage, so that the vocabulary gap between queries and products can be alleviated.

2.2 Synonym Candidate Generation

In this subsection, we describe how to generate the candidate synonym pairs from different data sources in details, including the dictionary method and query log mining.

2.2.1 Dictionary Method. We extracted general synonym candidates from the WordNet synset and Wikipedia redirects data. WordNet is one of the authoritative sources for English. It mostly provides synonyms for single words, such as [handbag, bag, wrinkle, purse, pocketbook] are all defined as synonyms. Many of them are good candidates for noun phrases. Wikipedia redirects data is based on Wikipedia title redirects links, and are also generated manually. Wikipedia covers a vast range of topics, some of them are helpful in search. An example of useful redirect pair is "cba" to 'Chinese basketball association", which links the acronym from âĂĲcbsâĂİ to its complete form. Then word alignment and phrase extraction are implemented to prepare substitute words (such as [cba, Chinese basketball association]).

2.2.2 Query Log Mining. Admittedly, query logs are from customers' historical searches and contain rich customer feedback information. In order to extract synonym pairs from query logs, we need to generate query and related product pairs in the first place. A straightforward way to accomplish this is to directly utilize query and purchased (or clicked) products. However, since the number of purchased products is limited, the coverage of synonym pairs generated by this approach will be relatively low. Therefore, we employ a query-product graph building method to generate plenty number of parallel query-product pairs.

Specifically, we build a query-product bipartite graph from query logs, where the nodes present queries and products. For a given query node and a product node, an edge exists only when at least one of click or purchase actions are made for the query and corresponding product. The value of each edge is the linear combination of the action attributes, including purchases and clicks. After that, queries and products nodes can be clustered by the label propagation algorithm [32]. In this way, we collected queries and products that are interconnected among them but not with other nodes in the graph.

---

1https://en.wikipedia.org
We employed the BERT model\cite{6} to filter candidate synonym pairs where Q is the query node and P presents the product node. A solid line indicates a direct connection and a dotted line denotes an indirect connection.

For each query and product pair, we can gain direct connections by using the edges. Nevertheless, we can also find indirect connections in this graph. To determine indirect matches, we vectorized the queries using a weight vector by using the connected products in the same graph and computed the cosine similarity between each pair of query nodes within a given cluster. Figure 3 shows an example of our approach where we show a cluster with various queries and products. In this example, Q1-P1, Q1-P2 and Q1-P3 are direct matches; the pair Q1-P4 is an indirect match. If the similarity is above a minimum threshold, we collected a set of products that are connected to at least one query in the given query pair, and then connect both queries to all the products in the set. Finally, we can gain plenty number of query-product pairs and the synonyms should appear within the user keywords and the correspondence product titles.

After generating query-product pairs, we extracted n-gram pairs directly from these parallel sequence of tokens: one from the user keywords and another one from product title. For a given pair of n-gram tokens, source A (from product title) and target B (from user keyword), we calculated both the co-occurrence frequency, \( \text{co}(A, B) \) and transition probability \( p(A, B) \). The transition probability can be obtained by using the following formulation:

\[
p(A, B) = \frac{\text{Count}(A, B)}{\sum_{B' \in S(A)} \text{Count}(A, B')}
\]

where \( S(A) \) presents all the n-grams associated with A. In particular, transition probability indicates the exclusive association of the given n-gram pairs, by which we can make sure the association is not very spread. For each n-gram pair, we calculated the multiplication of \( \text{co}(A, B) \) and \( p(A, B) \), and dropped out low-scoring ones.

\section{3\hspace{1em}DEPLOYMENT: DOCUMENT ENHANCEMENT WITH SYNONYMS}

For each product, we extracted its title and used the generated synonym rules to find the synonym tokens. We added all these synonym tokens to a new index field of a given product to make them searchable. We didn’t apply synonym expansion to all the product text fields because many text fields contain noisy/defect tokens, and applying synonym expansion to these tokens could introduce more defects.

\section{4\hspace{1em}EXPERIMENT}

\subsection{4.1\hspace{1em}End-to-end Online Experiments}

To evaluate the effectiveness of the document side synonym expansion method, we performed an online A/B test on a global e-commerce site. We first extracted over one million synonym pairs using the method described in Section 2. Then, we used these rules to expand the text in the product description. Specifically, in the treatment group, we indexed the products with additional synonym tokens, making them searchable even when different words were used in search queries. While the synonyms were not utilized in the control group. We considered the impact on both business metrics, such as sales, as well as recall improvement, characterized by the percent of searches resulting in less than one page of results (sparse result rate). As a guardrail, we also employed human evaluation for search defect on a few thousand randomly sampled top-ranked query, result pairs in each country\footnote{Each query, result pair is manually judged as relevant or irrelevant, the defect rate is defined by the percentage of irrelevant pairs.}.

In all countries, we found the proposed method statistically significantly improved key business metrics. These results indicated our method led to a better retrieval performance and can help customers fulfill their shopping missions when the vocabulary gap existed. Notably, the human evaluated defect rate was equal or better in all countries. This was a strong evidence that we injected reasonably relevant synonym tokens to the search index.

\subsection{4.2\hspace{1em}Synonym Quality Evaluation}

In addition to the overall recall improvement by applying the document enhancement in search, we also directly evaluated the quality of the synonyms generated by our approach detailed in Section 2.2 and 2.3. Since the process is unsupervised, we conducted human audits on a few hundred expanded texts focusing on the relatedness of synonyms. Specifically, for each applied synonym expansion example, we assigned one of the three labels to the rules and context: “irrelevant”, “related” (relevant in the context), or “fully interchangeable”. After audit, we found 14.5% expansions were irrelevant, 33.1% were relevant in the context. 52.4% were fully interchangeable. This means about 85.5% expansions are semantically related and are effective in addressing the vocabulary gap.

\section{5\hspace{1em}RELATED WORK}

In this section, we will give a brief introduction about the recent work on 1) query rewrite and 2) synonym discovery.
5.1 Query Rewrite

Query Rewriting has been shown to be effective in improving search results on queries with the vocabulary gap. A variety of customer feedback based techniques have been proposed for commercial search engines [31]. Basically, these method utilize implicit user feedback in the form of click-through rate [5], query co-occurrence [12], and co-clicked query similarity [1, 12] to rewrite queries. However, such a method may fail for queries with no or few results, which is known as pseudo-relevance feedback. In recent years, deep learning techniques have been extensively studied for query rewriting [9, 11, 15, 23]. [23] first regard the query rewriting task as machine translation, where a monolingual translation system is for generating expansion terms. Together with query-snippet pairs from user query logs, these extracted terms used to train a statistical machine translation model, which is employed to generate the final rewritten queries. However, most of the existing query rewriter models are sub-optimal for search engines. To better deal with this issue, [11] propose a learning-based query rewrite method that consists of a candidate generating phase and a candidate ranking phase. Experimental results on a Yahoo search engine demonstrate the effectiveness of their method. [15] propose a novel query rewriting method, aiming at rewriting vocabulary gap queries to well-performing ones. Specially, a Co-Attentional MalSTM is proposed to better measure the similarity between query pairs at the semantic level. Though these methods gain great performance in solving vocabulary gap issues in the search engine, we would like to try to deal with the vocabulary gap problem by directly adding the searchable tokens into product description information, which we think, is a more straightforward way of bridging the vocabulary gap between queries and products.

5.2 Synonym Discovery

Synonym discovery has been a popular research topic in natural language processing for years. A straightforward way of dealing with synonym discovery problem is to utilize the dictionaries such as WordNet [18] and ConceptNet [14] that contain many common and general sense synonyms constructed manually based on domain knowledge. After that, these extracted synonyms can be used to generate alternative keywords candidates for query expansion [29], or index expansion [4]. Meanwhile, some methods learn to generate synonyms in historical search logs [29, 30] and expand search queries by synonym patterns. With the recent success in deep learning technology [2, 6], deep neural network models for synonym discovery have been widely explored [10, 24]. [10] introduce a novel word-embedding-based approach to address the synonym extraction of multi-word terms. [16] propose an end-to-end synonym generation system that learns to generate synonym candidates by using a set of filtering components. A machine learning classifier is then employed to select correct synonym rules from the candidates. Finally, the generated synonym dictionary is incorporated for improving retrieval.

5.3 Document Expansion

Other than directly addressing vocabulary gap issues existing in search engine with enhancing query representations, there are also many studies focus on enriching document representations [3, 8, 20, 21, 26, 28]. [26] first illustrated the effectiveness of document expansion method for speech retrieval, where a text collection related to the speech is utilized for enhancing the contents of the speech. [3] showed the advantage of using document expansion by comparing with query expansion in the same framework. [28] focused on the task of language model information retrieval and proposed a document expansion technique to deal with the insufficient sampling issues, where the documents are expanded with their neighborhood information. [8] further showed the effectiveness of aggressive document expansion for short documents. In recent study, [20] proposed a method of using neural network models for document expansion. In particular, a sequence-to-sequence model is employed to predict the possible queries for a given documents and these queries are then used as the external information to expand the contents of the documents. Compared with existing work, in our study, we focus on document expansion for e-commerce search, where the documents (products) are always short and the vocabularies used in the queries could be noisy. In addition, we explore to expand the contents of products by using only synonyms. As far as we know, this is the first time that synonyms are employed for document expansion.

6 CONCLUSION AND FUTURE WORK

In this paper, we presented an unsupervised synonym extraction framework for enhancing the product document. Specifically, we proposed a two-phase unsupervised synonym discovery framework that extracts synonym rules from user behavior log data. The extracted synonyms were used to expand the product titles. We did multiple experiments to demonstrate the effectiveness of our approach. In the online A/B testing experiment, it showed significant increased purchase and conversion in all six countries. In the offline human audit, we identified that 85.5% of extracted rules are high quality pairs. For the next step, we will focus on incorporating knowledge graphs to further improve the quality of the extracted synonym rules.

REFERENCES


