Quotient Space-Based Keyword Retrieval in Sponsored Search

Yijiang Lian
Baidu Inc.
Beijing, China
lianyijiang@baidu.com

Shuang Li
Baidu Inc.
Beijing, China
lishuang15@baidu.com

Chaobing Feng*
Baidu Inc.
Beijing, China
fengchaobing@baidu.com

YanFeng Zhu
Baidu Inc.
Beijing, China
zhuyanfeng@baidu.com

ABSTRACT

Synonymous keyword retrieval has become an important problem for sponsored search ever since major search engines relax the *exact match* product's matching requirement to a synonymous level. Since the synonymous relations between queries and keywords are quite scarce, the traditional information retrieval framework is inefficient in this scenario. In this paper, we propose a novel quotient space-based retrieval framework to address this problem. Considering the synonymy among keywords as a mathematical *equivalence relation*, we can compress the synonymous keywords into one representative, and the corresponding quotient space would greatly reduce the size of the keyword repository. Then an embedding-based retrieval is directly conducted between queries and the keyword representatives. To mitigate the semantic gap of the quotient space-based retrieval, a single semantic siamese model is utilized to detect both the keyword–keyword and query-keyword synonymous relations. The experiments show that with our quotient space-based retrieval method, the synonymous keyword retrieving performance can be greatly improved in terms of memory cost and recall efficiency. This method has been successfully implemented in Baidu's online sponsored search system and has yielded a significant improvement in revenue.

KEYWORDS

Keyword Compressing, Quotient Space, Retrieval Model, Keyword Matching, Semantic Siamese Model

ACM Reference Format:
Yijiang Lian, Shuang Li, Chaobing Feng, and YanFeng Zhu. 2021. Quotient Space-Based Keyword Retrieval in Sponsored Search. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'21).* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Sponsored search advertising is one of the most popular advertising approaches because it can directly target the user's intentions. To match the user's queries with the advertiser's bidded keywords (Keyword in this paper is particularly used to denote queries purchased by the advertisers) plays a significant role in the sponsored search system. In general, three *match types* are supported: *exact match*, *phrase match*, and *broad match* [1]. Among them, *exact match* indicates that the ads are eligible to appear when a user searches for the specific keyword or its synonymous variants.

In this paper, we focus on the synonymous keyword matching problem under the *exact match*. For a given query and a keyword repository (a snapshot of all the keywords committed by the advertisers), the objective is to retrieve as many synonymous keywords as possible. This industrial problem has several unique challenges. The first challenge is the extremely high precision (>=95%) required by this commercial product. The second challenge is that compared with the large number of candidate keywords, the synonymous relations between queries and keywords are rare, which makes the traditional retrieval framework (e.g., the inverted term-doc link list) quite inefficient in this scenario, as most of the retrieved candidates do not satisfy the synonymous requirement. The third one is the stubborn semantic gap problem in natural language processing. And the last challenge is the extremely low latency required in an industrial sponsored search system.

Though the synonymous keywords for an ad-hoc query are sparse compared with the entire keyword repository, synonymous relations among keywords are quite common. The advertisers tend to enumerate the keyword variations to capture more similar flows. As a matter of fact, more than one thousand literally different but synonymous keywords of *the price of double eyelid surgery* can be found in Baidu's keyword repository, e.g., *how much is double eyelid surgery, how much does it cost to do a double eyelid surgery.* Our motivation is to use the synonymous equivalence relation to compress the huge keyword repository into a small quotient set, with which the query–keyword matching can be efficiently conducted between queries and the quotient space representatives.

The quotient retrieval puts forward a higher request for overcoming the *semantic gap* challenge. Suppose that there are two queries Q1, Q2, and two keywords K1, K2, and the existing synonym matching system can trigger K1 for Q1, and K2 for Q2 separately. If K2 and K1 are synonymous and compressed into one representative K1, then Q1 can be matched to {K1, K2} effortlessly. However, Q2 might retrieve nothing because of the semantic gap between K1 and Q2. An intuition to mitigate this problem is to keep

*Corresponding author.

SIGIR eCom'21, July 15, 2021, Virtual Event, Montreal, Canada

[1] https://support.google.com/google-ads/answer/7478529?hl=en

the synonym retrieval capability on the (*query*, *keyword*) pairs synchronized with that on the (*keyword*, *representative*) pairs.

Based on this consideration, we propose the following quotient space-based retrieval approach: Firstly, a semantic siamese model (SSM) [3] is trained to calculate the synonymy matching distance between queries and keywords, as well as between keywords and keywords. Secondly, approximate nearest neighbor (ANN) search based on the SSM is conducted offline to discover the synonymous relations between keywords. Then, a quotient space, namely a partition of the original keywords, is built based on the synonymous relations, and each partition is denoted by a representative keyword. Finally, when an ad-hoc query arrives, an ANN search is performed online between the query and the keyword representatives, and all the synonymous keywords in the same quotient space are supplied.

We have successfully applied the proposed method in Baidu's sponsored search system. Experimental results show that using the proposed method, the compression ratio of the keyword space can reach a point of 5:1, the retrieval efficiency is improved, and significant revenue has yielded.

2 METHOD

2.1 Retrieval Framework

As is shown in Figure 1, our quotient space-based keyword retrieval framework comprises four steps: The first step is *keyword compressing*. Since the synonymous relationship between keywords are reflexive, symmetric and transitive, it conforms to a mathematical equivalence relationship. Based on this fact, synonymous keyword compressing is conducted for the keyword repository $\mathcal{K}$. And for each partition, one single keyword is selected as its representative. The set of all the representatives is denoted as $\widetilde{\mathcal{K}}$. The second step is an embedding-based ANN *retrieval*, which is directly performed between a query q and the keyword representatives $\widetilde{\mathcal{K}}$ to obtain synonymous representative candidates. The third step is *discrimination*, where a single-tower-like discriminant model is utilized to score the query–representative pairs, only synonymous representatives verified by the discriminant model remain. And the final step is keyword *mapping*. For the obtained synonymous representatives, all keywords in the synonym partition indexed by the representatives are fetched out.

2.2 Semantic Siamese Model

The SSM plays a core role in our framework, which undertakes two missions: keywords compression and matching between queries and the keyword representatives. It is double-tower-like as in [4]. And each tower's function is to encode the input sentence into a semantic embedding vector, and then these output vectors go through a distance function to measure their synonymous matching degree. Since there are no interactions at the encoding stage, this structural restriction makes the SSM slightly weaker than a single-tower-like model with full interactions. Based on this consideration, a single-tower-like model is initially trained as our discriminant model. And it plays as a teacher model to guide the training of the SSM. In this paper, all of the towers are implemented under the transformer [10] network structure.

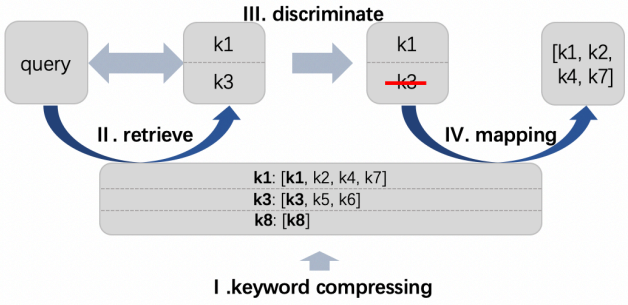

Figure 1: Quotient space-based keyword retrieval framework. (I) The original keyword repository $\mathcal{K}$ is compressed into quotient space representatives $\widetilde{\mathcal{K}}$. (II) ANN-based retrieval is conducted between the query and $\widetilde{\mathcal{K}}$, yielding candidate representatives k1 and k3. (III) The discriminant model screens out bad representative k3. (IV) All of the keywords represented by k1 are returned.

Discriminant Model. A two-stage fine-tuning is conducted to match our synonymy discriminating task. The first stage is performed on the training data extracted from the weblog, where the data volume is large (more than 200 million) and the label accuracy is approximately 70%. The second stage training data is human-annotated, which is 95% accurate and scarce (less than a million). The AUC(area under the curve) of the trained discriminant model is around 97%, and the recall approximates 75% under the precision of 95%.

Semantic Siamese Model. To optimize the SSM's retrieval performance, we adopt the triplet loss [15]:

$$L = \sum_{i=1}^{N} max(0, D(q^i, k_+^i) - D(q^i, k_-^i) + m) \tag{1}$$

where (q, k_+) and (q, k_-) represent positive pairs and negative pairs, respectively, $D(q, k)$ denotes the synonymy distance between *query*, *keyword* embedding vector and m is the margin used to separate the positive and negative pairs.

The SSM's training is guided by the discriminant model, which comprises three main steps:

1 Large-scale pretraining on confident instances. In this step, the trained discriminant model is used to score the query–keyword pairs shown in the sponsored weblog $\mathcal{D}$, and the label-confident instances are denoted as $\mathcal{D}_{sure}$, thereby only instances whose scores are higher than the prescribed upper bound or less than the lower bound are retained. Then the SSM is trained on $\mathcal{D}_{sure}$.
2 Self-training. Given a discriminant model, a query set, and a keyword set, this step forms an iteration loop:
   – Based on the current SSM, ANN search is conducted between the query set and the keyword set, and top K approximate nearest keywords are obtained.
   – The discriminant model is used to score these retrieved query–keyword pairs, and the non-synonymous pairs are selected as hard negative examples.

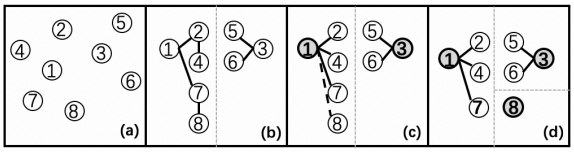

Figure 2: Keyword compressing Procedures: (a) The original keyword repository. (b) *Synonymous relation detection* and *connected subgraph generation.* (c) *Representative selection* and *non-synonymous relation elimination.* (d) Representative cluster generation. (The nodes represent keywords. The grey nodes represent keyword representatives. The solid lines represent synonymous relations, while the dotted lines represent non-synonymous relations.)

- The hard negative examples are fused into the new train set with the previous data $\mathcal{D}_{sure}$, and the SSM is retrained on this newly obtained dataset.

3 Fine-tuning on human-annotated training data.

2.3 Keyword Compressing

Keywords compressing might be simply implemented by inferencing all keyword pairs' synonymous relations based on the discriminant model. However, it is unrealistic as the keyword's volume is extremely large and the discriminant model's inference is quite time-consuming. The embedding-based SSM (i.e., Sec. 2.2) can be used here to accelerate the process. In the meanwhile, the consistency between the retrieval model and the keyword compression model can also mitigate the semantic gap between queries and keyword representatives.

Algorithm 1 Compressing the keywords into representatives

Input: Keyword repository $\mathcal{K}$, SSM M, discrimination model D, and the number of elements retrieved by ANN K
Output: Synonym clusters C of $\mathcal{K}$

```
Obtain embeddings of the keywords in K by encoder of M;
Build ANN index for embedding representation of K;
for each keyword k in K do
    Obtain top K relations of k by ANN;
    Filter out non-synonymous relations of k through D;
end for
Construct connected subgraphs by synonymy, then each subgraph corresponds to a synonym cluster c in C;
for each c in C do
    Select a representative for c by the relation degree of k;
    Filter out non-synonymous relations between representative and each k in c through D;
end for
```

As is illustrated in Figure 2, the keyword compressing consists of three steps. In the first step (*synonymous relations detection*), ANN is performed between each keyword and the repository to obtain the synonymous keyword candidates. The second step (*connected subgraph generation*) is to generate the connected subgraphs based on the detected synonymous relations. The last step (*representative selection* and *non-synonymous relation elimination*) is to select a representative for each cluster. In our experiments, the element having the largest relation degree (the number of synonymous relations) is selected to represent the whole cluster. Since the errors might propagate during the subgraph generation, further discrimination is conducted between the representative and the other keywords in the same cluster based on the discriminant model. The whole process is elaborated in Algorithm 1.

3 EXPERIMENTS

In this section, we focus on the performance of the SSM and the overall performance of the quotient space-based retrieval.

3.1 Offline Experiments

3.1.1 Experiments for the Semantic Siamese Model.

Dataset. A total of 400 thousand query–keyword pairs are randomly sampled from the sponsored search engine's one week's weblog and professionally annotated for their synonymity. This dataset $\mathcal{D}^h$ is further split into three parts with a ratio of 8:1:1, which are used as training ($\mathcal{D}^h_{train}$), developing ($\mathcal{D}^h_{dev}$), and testing data ($\mathcal{D}^h_{test}$) for the SSM respectively. We focus on two indicators: the Area Under the Curve (AUC) and the recall rate under the precision of 95% (R@P95) on $\mathcal{D}_{test}$.

Implementation Details. The SSM and discriminant model are built based on the state-of-the-art bert-like ERNIE [14], which has been well pretrained on large-scale corpora with multitask learning. Each of the two towers of the SSM is implemented with 4 transformer layers, 4 self-attention heads and the hidden dimension size is 128. These two towers shared their parameters. The discriminant model is implemented with 24 transformer layers, 16 self-attention heads and the hidden dimension size is 1024. During fine-tuning stages, both models are optimized by Adam [5] with an initial learning rate of 1×10^{-4} and a batch size of 128.

Results. We conduct an ablation study on the SSM training strategies mentioned in Sec.2.2. Four models are compared. All of them go through a final fine-tuning step on $\mathcal{D}^h_{train}$, and their differences are as follows:

- M_0 (i.e., the baseline of SSM) is pretrained on the raw weblog data $\mathcal{D}$, where *exact-matched* query–keyword pairs are used as positive examples, and the others are considered as negatives;
- M_1 is pretrained on $\mathcal{D}_{sure}$, which is the confident sample subset of $\mathcal{D}$ obtained by the discriminant model;
- M_2 is pretrained on $\mathcal{D}$ and further self-trained based on the discriminant model.
- M_3 is pretrained on $\mathcal{D}_{sure}$ and further self-trained as M_2, i.e., M_3 implements the full training strategies.

The results are listed in Table 1. By comparing M_1 with M_0, and M_2 with M_0, we can observe that both the discriminant-model's guidance and the self-training bring positive effects on AUC and R@P95. Besides, M_3 gains a further improvement over M_1 as well as M_2, resulting in 95.14% on AUC and 55.36% on R@P95 (+1.68% on AUC, +11.44% on R@P95 over M_0), which demonstrates the best performance of our full training strategies.

3.1.2 Experiments for the Quotient Framework.

Table 1: Evaluation of the SSM on $\mathcal{D}^h_{test}$. M_0 indicates the baseline model pretrained on $\mathcal{D}$. M_1 is pretrained on $\mathcal{D}_{sure}$. M_2 is pretrained on $\mathcal{D}$ and adopts self-training. M_3 applies our full training strategy.

	AUC	R@P95
M_0	93.46%	43.92%
M_1	94.76%	49.60%
M_2	94.16%	50.52%
M_3	**95.14%**	**55.36%**

Table 2: The recall evaluation of the quotient-based method on $\mathcal{D}^h_{quo}$.

	R@10	R@100
Baseline	31.34%	80.27%
Quotient-Based	78.75%	96.50%

With a well-trained SSM M_3, we examine the retrieval efficiency of the quotient-based method in this subsection. Our baseline method is the traditional ANN search conducted between queries and the original keyword repository $\mathcal{K}$. The quotient-based method implements the quotient space-based keyword compressing, and conducts ANN search between queries and the keyword representatives $\widetilde{\mathcal{K}}$.

Dataset. Our experimental keyword repository $\mathcal{K}$ is a 12 million keyword set sampled from one day's whole keyword repository, and 10 thousand queries are sampled from one day's weblog. For each query q, up to 10 synonymous keywords are obtained by coarse-selection with the discrimination model and refined-selection with human evaluation. This dataset is denoted as D^h_{quo}. Metric R@K is leveraged to measure the recall rate for D^h_{quo} when top K ANN search is performed, while lantency@K measures the average CPU time cost for conducting the top K ANN search.

Implementation Details. We adopt HNSW [9] both in our offline simulations and online settings for its high efficiency in large-scale index processing. The *number of links for an element* is set to 16, the *size of the dynamic list for the nearest neighbors* is set to 200, and the thread number is set to 10. The test is conducted on a machine equipped with 28-core Intel(R) Xeon(R) Gold 5117 CPU clocked at @ 2.00GHz with a RAM of 250G.

Results. Table 2 shows that our quotient-based method can greatly boost recall of the synonymous keywords, where R@10 increases from 31.34% to 78.75%, approximately 1.5 times improvement. When top 100 retrieval is conducted, recall rate achieves an increase by 16.23 percentage points.

Table 3 evaluates the latency and memory cost. The results show that our quotient-based method can greatly reduce the retrieval latency as well as the memory consumption. Compared with the baseline method, 1/3 latency time for the top 100 retrieval can be saved, and only 23.8% of the indexing space is consumed.

3.2 Online Experiments

This section examines the performance of the overall quotient space-based retrieval framework. The whole process is implemented as the pipeline in Figure 1 and deployed on Baidu's sponsored search system. We finally select 460 million active keywords to conduct the keywords compressing, and the ANN index is built upon 80 million keyword representatives correspondingly. Mappers from quotient representatives to keywords are implemented with a lookup table service. The online A/B experiments show that the quotient space-based retrieval framework leads to a 1.3% increase in CPM (average revenue per thousand searches), which is compelling for a large sponsored search system currently. Human evaluations on the shown ads show that the user's experience is not degraded than before.

Table 3: Latency and memory cost test for the quotient-based method. Latency is measured in milliseconds, and memory is measured in gigabytes.

	Latency@10	Latency@100	Memory
Baseline	0.26	0.90	8.4
Quotient-Based	0.16	0.60	2.0

4 RELATED WORK

Paraphrase. Paraphrases are widely used for various NLP applications, such as document retrieval [17], question answering [13]. Recent works [6, 12] proposed generative query–keyword matching methods. [7, 8, 16] contributed paraphrase datasets for model improvement. Our work is driven by the high precision and efficiency demand in practical industrial application, and dedicates to embedding-based query–keyword synonym retrieval in commercial search scenarios.

Embedding Representation and Retrieval. Embedding representation has been studied and applied in different fields and tasks in the age of deep learning [1]. Bi et al. [2] adopted a transformer-based embedding retrieval for personalized product search. Huang et al. [4] introduced their context-aware embedding representation and retrieval in Facebook vertical searches. As for retrieval efficiency, various ANN algorithms have been proposed. Many of the related works focus on the distribution characteristics of the representation vectors and attempt to reduce the scale of the index, such as principal component analysis (PCA), production quantization [11]. Our work differs from the ANN algorithms in that we compress the origin candidates into task-oriented partitions, that is, candidates in a partition are synonymous. Under our retrieval framework, the ANN accelerating algorithms can further work on the retrieval operation smoothly.

5 CONCLUSIONS

This paper presents a novel quotient space-based retrieval framework to address the synonymous keyword retrieval problem in sponsored search. The basic idea is to compress the large keyword space into a small quotient space based on the synonymous equivalence relation, and the keywords matching is directly conducted between the queries and the keyword representatives. A siamese semantic model is trained for embedding-based query–keyword retrieval. Keyword compressing implemented by the same SSM can avoid exacerbating the semantic gap in quotient-based retrieval. The offline evaluations and real experiments on Baidu's online sponsored search have proved the effectiveness of this work.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[2] Keping Bi, Qingyao Ai, and W. Bruce Croft. 2020. *A Transformer-Based Embedding Model for Personalized Product Search.* Association for Computing Machinery, New York, NY, USA, 1521–1524. https://doi.org/10.1145/3397271.3401192

[3] Davide Chicco. 2021. Siamese neural networks: An overview. *Artificial Neural Networks* (2021), 73–94.

[4] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-Based Retrieval in Facebook Search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2553–2561. https://doi.org/10.1145/3394486.3403305

[5] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)* (2015).

[6] Mu-Chu Lee, Bin Gao, and Ruofei Zhang. 2018. Rare Query Expansion Through Generative Adversarial Networks in Search Advertising. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 500–508. https://doi.org/10.1145/3219819.3219850

[7] Yijiang Lian, Zhenjun You, Fan Wu, Wenqiang Liu, and Jing Jia. 2020. Retrieve Synonymous keywords for Frequent Queries in Sponsored Search in a Data Augmentation Way. arXiv:2008.01969 [cs.IR]

[8] Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1952–1962.

[9] Y. A. Malkov and D. A. Yashunin. 2020. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (2020), 824–836. https://doi.org/10.1109/TPAMI.2018.2889473

[10] Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. 2020. Reranking for efficient transformer-based answer selection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1577–1580.

[11] Yusuke Matsui, Yusuke Uchida, Hervé Jégou, and Shin'ichi Satoh. 2018. A survey of product quantization. *ITE Transactions on Media Technology and Applications* 6, 1 (2018), 2–10.

[12] Weizhen Qi, Yeyun Gong, Yu Yan, Jian Jiao, Bo Shao, Ruofei Zhang, Houqiang Li, Nan Duan, and Ming Zhou. 2020. ProphetNet-Ads: A Looking Ahead Strategy for Generative Retrieval Models in Sponsored Search Engine. arXiv:2010.10789 [cs.IR]

[13] Fabio Rinaldi, James Dowdall, Kaarel Kaljurand, Michael Hess, and Diego Mollá. 2003. Exploiting Paraphrases in a Question Answering System. In *Proceedings of the Second International Workshop on Paraphrasing - Volume 16* (Sapporo, Japan) *(PARAPHRASE '03)*. Association for Computational Linguistics, USA, 25 – 32. https://doi.org/10.3115/1118984.1118988

[14] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 05 (Apr. 2020), 8968–8975. https://doi.org/10.1609/aaai.v34i05.6428

[15] Furong Xu, Wei Zhang, Yuan Cheng, and Wei Chu. 2020. Metric Learning with Equidistant and Equidistributed Triplet-based Loss for Product Image Search. In *Proceedings of The Web Conference 2020*. 57–65.

[16] Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 1298–1308.

[17] Ingrid Zukerman and Bhavani Raskutti. 2002. Lexical Query Paraphrasing for Document Retrieval. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1* (Taipei, Taiwan) *(COLING '02)*. Association for Computational Linguistics, USA, 1–7. https://doi.org/10.3115/1072228.1072389