A Data Augmentation Approach for Retrieving Synonymous Keywords in Sponsored Search

Yijiang Lian*
lianyijiang@baidu.com
Baidu

Zhenjun You
youzhenjun@baidu.com
Baidu

Fan Wu
wufan_aais@pku.edu.cn
Peking University

Wenqiang Liu
liuwenqiang01@baidu.com
Baidu

Jing Jia
jiajing01@baidu.com
Baidu

ABSTRACT

In sponsored search, retrieving synonymous *keywords* is of great importance for accurately targeted advertising. The extremely high precision requirements (>= 95%) and the semantic gap between queries and *keywords* are two major challenges to this task. Since the synonymous relationship between queries and *keywords* is quite scarce, conventional information retrieval methods in this scenario are quite inefficient. In this paper, we propose an idea of data augmentation to effectively retrieve synonymous *keywords*. Given a high-quality seed dataset, our approach includes two steps: translation-based retrieval and discriminant-based filtering. Firstly, we devise a Trie-based translation model to make a data increment. In this phase, a Bag-of-Core-Words trick is conducted, which increases the data increment's volume 4.2 times while keeping the original precision. Then we use a BERT-based discriminant model to filter out nonsynonymous pairs, which exceeds the traditional feature-driven GBDT model with 11% absolute AUC improvements. The approach has been successfully applied to Baidu's sponsored search system, which yields a significant improvement in revenue. In addition, a commercial Chinese dataset containing 500K synonymous pairs with a precision of 95% is released to the public for paraphrase study[1].

CCS CONCEPTS

• **Information systems → Computational advertising**.

KEYWORDS

Sponsored search, keyword matching, keyword retrieval, synonymous keywords retrieval, paraphrase generation, paraphrase identification

*Corresponding author
[1]http://ai.baidu.com/broad/subordinate?dataset=paraphrasing

WWW '21, The Web Conference , April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Yijiang Lian, Zhenjun You, Fan Wu, Wenqiang Liu, and Jing Jia. 2021. A Data Augmentation Approach for Retrieving Synonymous Keywords in Sponsored Search. In *WWW '21: The Web Conference , April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Sponsored search is one of the most major forms of online advertising and also the main source of revenue for most search engine companies. In sponsored search, there are three distinct roles involved: advertiser, user and search engine. Each advertiser submits ads (abbreviated for advertisements) and bids on a list of relevant *keywords* for each ad. To avoid ambiguity, *keywords* in this paper are particularly used to denote queries purchased by advertisers. When the search engine receives a query submitted by a user, it firstly retrieves a set of matched *keywords*. Then an auction is carried out to rank all corresponding ads, taking both the quality and bid price of each ad into account. Finally, the winning ads are presented on the search result page to the user.

Major search engine companies provide a structured bidding language, with which the advertisers can specify how would their purchased *keywords* be matched to the online queries. In general, three *match types* are supported: *exact, phrase, broad.* In the early days, *exact match* requires that query and *keyword* are exactly the same. Since queries are ever-changing, advertisers usually have to come up with a lot of synonymous forms of their *keywords* to capture more similar query flows. To ease their burden, modern search engines relax the *exact match*'s matching requirement to the synonymous level [2], which means under *exact match type*, the ad would be eligible to appear when a user searches for the specific *keyword* or its synonymous variants. For example, the *keyword how much is iPhone 11* would not only be matched to the identical query but also be matched to other queries like *the price of iPhone 11.* In *phrase match type*, the matched queries should include the *keyword* or the synonymous variants of the *keyword. Broad match type* further relaxes the matching requirements to the semantic relevance level.

The highly accurate matching makes *exact match type* greatly welcomed by most customers, and nowadays it still occupies a great portion of the *keywords* revenue for most search engine companies. In this paper, we focus on the synonymous *keywords* matching problem under the *exact match type.* To make it clear, for a

[2]https://support.google.com/google-ads/answer/2497825?hl=en

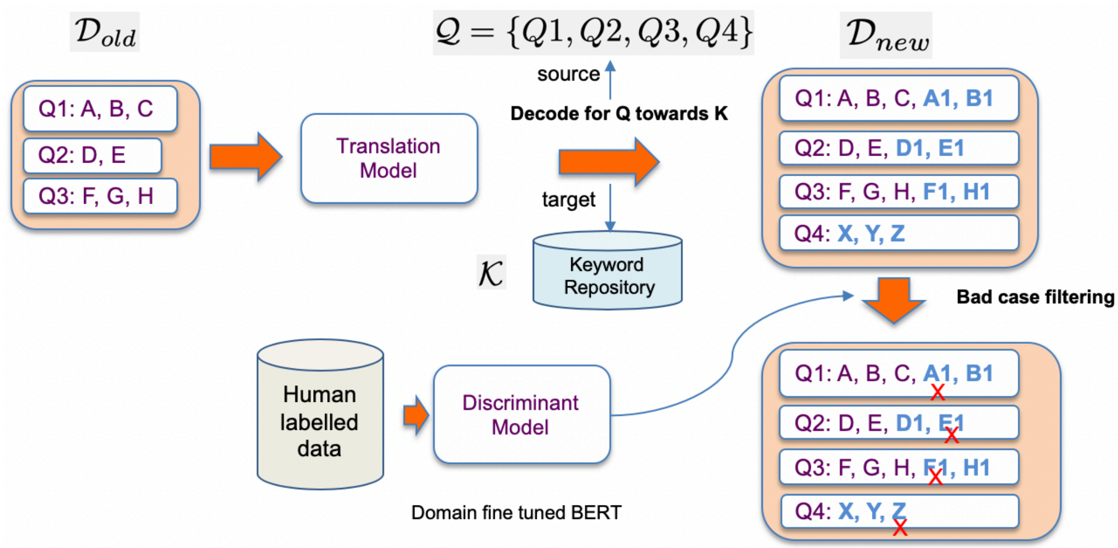

Figure 1: Our framework contains two steps. Firstly, a Bag-of-Core-Words translation model is trained on $\mathcal{D}_{old}$, and is used to make constrained decoding for frequent query set $\mathcal{Q}$ towards the *keyword* repository $\mathcal{K}$, which yields $\mathcal{D}_{new}$. Secondly, a BERT based synonym identification model is used to filter out nonsynonymous cases.

given query and a *keyword* repository (which is a snapshot of all the purchased *keywords*), we want to retrieve as more synonymous *keywords* as possible, while keeping a high precision. (On the one hand, retrieving more qualified synonymous *keywords* can provide a more competitive advertisement queue for the downstream auction system; on the other hand, from the point of advertisers' fairness, each advertiser's synonymous *keywords* should be retrieved.)

There are several challenges to this problem. The first one is the extremely high precision (>=95%) required by *exact match type*. Under this precision, the recall rate of most traditional models is very low. The second one is the semantic gap between queries and keywords. Last but not least, since the volume of *keywords* and queries might reach billions, how to effectively detect the synonymous relationships between these huge numbers of queries and *keywords* is not an easy task.

As far as we know, almost all of the existing work about *keyword* matching focus on nonsynonymous matching scenarios happened *broad match* [1, 17, 20]. Few works have tried to tackle the synonymous matching problem in *exact match* scenario.

Here we want to emphasize that the framework for synonymous *keywords* retrieval should be carefully designed. As is common knowledge, *keywords* retrieval in *broad match* can easily fit into the standard information retrieval framework, where we can first analyze the query and find its most important terms, and then the intersection of the corresponding inverted doc(*keyword*) list of these tokens constitutes the candidates, finally, a query-keyword relevance model can be utilized to filter out irrelevant cases. Query rewriting techniques can be used to enlarge the retrieval performance.

However, since synonymous query-*keyword* pairs are quite scarce compared to relevant pairs, this framework is not suitable for *synonymous* situations. According to our statistics, about 70% of the inverted-list-retrieved candidates satisfy the *broad match* requirement. However, only about 1% of these candidates satisfy the synonymous requirement. Considering that the volumes of query and keywords are extremely large, and synonymous checking is quite time-consuming, we need to make the whole retrieval framework more efficient.

One way is to narrow down the candidates' scope, and find promising query-*keyword* pairs which are more likely to be synonymous. Following this idea, we introduce a data-augmentation approach to retrieving synonymous *keywords*, where the candidates' scope is controlled by a high-quality seed data and the augmentation model. As is illustrated in Figure 2, suppose we have already accumulated some high precision synonymous query-*keywords* pairs $\mathcal{D}_{old}$, we would utilize the generalization capability of machine learning models to expand $\mathcal{D}_{old}$ into $\mathcal{D}_{new}$, and $\Delta = \mathcal{D}_{new} - \mathcal{D}_{old}$ would be our newly retrieved results.

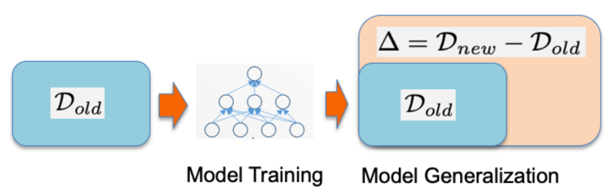

Figure 2: A schematic diagram for the data augmentation framework.

It is well known that search queries are highly skewed and exhibit a power-law distribution [27, 33]. Approximately 20% of frequent queries occupy 80% of the query volume. In an industrial environment, the retrieved synonymous *keywords* for frequent queries are stored in a Key-Value lookup table, which is computed and updated in an offline mode. When an ad hoc query arrives, the corresponding synonymous *keywords* can be retrieved immediately by looking it up on the table. In this paper, we consider the existing result stored on this lookup table as the seed dataset $\mathcal{D}_{old}$, which is a combined result of several data mining strategies.

Our approach contains two steps. The first one is translation-based retrieval, where a translation model is trained to fit $\mathcal{D}_{old}$. For brevity, let us denote the frequent query set as Q, and the *keyword* repository as $\mathcal{K}$. A constrained decoding [17] is conducted for Q towards the *keywords* repository $\mathcal{K}$ to get $\mathcal{D}_{new}$. To encourage the model to generate a larger Δ, a synonym keeping Bag-of-Core-Words transformation is applied to the source and target side of $\mathcal{D}_{old}$. The second step is bad case filtering. The translation model's generalization ability could increase recall, but might also introduce bad cases. To remove them, a strong synonym identification model based on BERT [7] is introduced to score the sentence pairs in $\mathcal{D}_{new}$. Pairs with scores lower than a given threshold are filtered out. As is shown in Figure 1, after translation-based retrieval, new *keywords* A1, B1 are retrieved and appended after Q1's retrieval list [A, B, C]. Then a discriminant-based filtering is conducted, and *keyword* A1 is finally removed from Q1's expanded retrieval list.

Our main contributions are two folds: Firstly, a practical data-augmentation approach is proposed to address the synonymous *keywords* retrieval problem under *exact match type*, which includes translation-based retrieval and discrimi–nant-based filtering. The Bag-of-Core-Words transformation trick increases the Δ substantially while keeping the original precision. And the domain fine-tuned BERT's performance far exceeds the feature-driven GBDT model's. To the best of our knowledge, this is the first time to address this important commercial problem. Our approach has been successfully applied to Baidu's sponsored search, giving a significant improvement in revenue. Secondly, a high-quality Chinese commercial synonym data set containing 500K pairs has been published along with this paper, which might be used in paraphrasing or other similar tasks. As far as we know, this is the first published large scale high-quality Chinese paraphrase data set reaching a precision of 95%.

2 RELATED WORK

The semantic gap between users and advertisers is one of the most challenging problems in synonymous *keywords* retrieval. Some works introduced query-query transformation (query rewriting) [5, 10, 20, 43] and *keyword-keyword* transformation [1]. Direct query-*keyword* transformation has also been studied [16, 17]. [17] proposed a Trie-constrained translation method to make sure all generated sentences are valid commercial *keywords*.

Our framework consists of two main parts: a paraphrase generation model and a paraphrase identification model. Paraphrase generation (PG), a task of rephrasing a given sentence into another with the same semantic meaning, has been used in various Natural Language Processing applications, such as query rewriting [47], semantic parsing [2], and question answering [28]. Traditionally, it has been addressed using rule-based approaches [21, 45]. Statistical machine translation has been used in [40]. Recent advances in deep learning have led to more powerful data-driven approaches to this problem. [32] applied neural machine translation for paraphrase generation to improve Alexa's *ASK* user experience. The semantic augmented transformer seq2seq model has also been studied [38]. The Variational AutoEncoder based generation model is also a good option [11, 26].

Paraphrase identification (PI) aims to determine whether two natural language sentences have identical meanings. With the growing trend of PI, many English paraphrasing datasets have been made for this task, such as Quora Question pairs, and a lot of works have been developed based on them [4, 29]. Traditional methods mainly made use of handcrafted features [6, 19, 30, 37]. Recently, deep neural network (DNN) architectures have played a part in PI tasks. According to whether the inner interaction between a pair of sentences is modeled, there are mainly two types of methods. The first is encoder-based. RAE [31] is a pioneer that introduced recursive AutoEncoders to PI. Both ARC-I [12] and Hybrid Siamese CNN [23] adopted Siamese architecture introduced in [3] but used different loss functions. The other is interaction-based. ARC-II [12] and IIN [9] utilized the interaction space between two sentences while [25] viewed the similarity matrix between words in two sentences as an image and utilized a CNN to capture rich matching patterns. Bi-CNN-MI [41], ABCNN [42], BiMPM [39] and GSMNN [8] focused on the effect of introducing multi-granular and multi-direction matching. [44] showed that BERT [7] pre-trained on a large corpus and then fine-tuned with an additional layer worked quite well on PI tasks.

3 METHOD

Our method can be formalized as 2 main steps: translation-based retrieval and discriminant-based filtering. The seed paraphrasing dataset needed for the augmentation framework can be any dataset of short paraphrasing text pairs. In this paper, we won't go into much detail about paraphrase extraction techniques. Our augmentation framework aims to retrieve more *keywords* for each query while ensuring high precision.

3.1 Translation-Based Retrieval

Our translation model follows the common sequence to sequence learning encoder-decoder framework [35]. And we implemented it with the Transformer [36], considering its state-of-the-art performance in learning long-range dependencies and capturing the semantic structure of the sentences.

To increase the translation model's retrieval efficiency, a prefix tree for the *keyword* repository is built ahead of time, and all the beam search decoding is constrained on this prefix tree [17]. At each step of the beam search, the prefix tree will directly give the valid suffix tokens following the current hypothesis path, then a greedy top-N selection is performed within the valid tokens. This technique makes sure all the generated sequences are valid *keywords*.

Table 1: Some typical trivial synonymous translations for *How much does double eyelid surgery cost.*

How much does double eyelid surgery cost generally
How much does double eyelid surgery cost in general?
How much does double eyelid surgery cost probably

Paraphrases extracted from web data usually include some trivial patterns: reordering of words, insertion of function words and punctuation. A simple implementation of the translation model might generate too many trivial paraphrases, as is shown in Table 1. Common stop words removing method is too coarse to meet our need for synonym keeping, especially in Chinese. We carefully designed a synonym preserving data reduction method called Bag-of-Core-Words (abbreviated as *BCW*) transformation to reduce each sentence into a compact form without losing semantic information.

For each tokenized sentence, the *BCW* transformation consists of two sequential steps:

(1) **Core-Words Transformation**. Here we consider some part of speech (abbreviated as POS) tags (like interjections, modal particles, etc.) as redundant, which means removing it generally does not change the query's intention. Table 2 lists the typical redundant POS tags. Tokens with these tags would be removed. It is worth noting that some of the POS removing rules might not be universally applicable to other languages. For example, modal particles and interjections are quite common in Mandarin Chinese, however, these words are unusual in English. And the remaining tokens are considered as core words. For brevity, we refer to this step as *CW* in the following sections.

Table 2: Redundant part of speech tags in Chinese.

POS tag	Typical terms
Interjection	哼 (humph), 嗯 (em), 嘿 (hey), 嘘 (shh)
Auxiliary word	等等 (and so on), 一般 (generally)
Punctuation	comma, colon, question mark
Modal particle	啊 (ah), 哇 (wow), 呦 (yo), 耶 (yeah)

(2) **Bag-of-Words Transformation**. In most cases word order does not affect the meaning of a sentence. In fact, we sample 600 commercial queries from the ad weblog and find that in 94% of cases, the original query and its Bag-of-Words form have the same meaning. Based on this consideration, we sort the remaining core tokens in the sentence literally to remove the order's effect in the model training process. To make it more accurate, additional rules have been made to exempt the special cases. For example, *Flights from New York to Beijing* and *Flights from Beijing to New York* have the same Bag-of-words form, but their meanings are different. The same is true for *Does hypertension cause hyperlipidemia* and *Does hyperlipidemia cause hypertension*. So when sentences have two location tokens or two disease entity tokens or other token pairs with causality, temporal relation, etc., the order of the paired tokens remains unchanged.

The *BCW* transformation is simple, fast and very effective. Applying it to our Chinese training data effectively reduces the dataset size by nearly 20% with a synonym precision of 98%. Based on the *BCW* transformation, we devised a data-augmentation method to retrieve synonymous *keywords*, as is illustrated in Algorithm 1.

Algorithm 1: Retrieve synonymous *keywords* in a data augmentation way.

Input: Synonymous query-*keyword* pair dataset $\mathcal{D}_{old}$, *keyword* repository $\mathcal{K}$, frequent queries set Q, beam size B

Output: Expanded query-*keyword* pair dataset $\mathcal{D}_{new}$

Apply the *BCW* transformation to $\mathcal{D}_{old}$ to get $\widetilde{\mathcal{D}}_{old}$

Train a neural machine translation model M on $\widetilde{\mathcal{D}}_{old}$

Apply the *BCW* transformation to $\mathcal{K}$ to get $\widetilde{\mathcal{K}}$ and the corresponding relationship for elements in $\widetilde{\mathcal{K}}$ and $\mathcal{K}$ is stored in a lookup table T

Set the expanded dataset $\mathcal{D}_{new}$ to be a copy of $\mathcal{D}_{old}$

for *each* q *in* Q **do**

  Apply the *BCW* transformation to q to get $\widetilde{q}$

  Using M to translate $\widetilde{q}$ towards $\widetilde{\mathcal{K}}$ with a beam size of B, which results retrieved *keywords* set $\mathcal{R}_{\widetilde{q}}$

  Using T to make inverse *BCW* transformation on $\mathcal{R}_{\widetilde{q}}$, which results $\mathcal{R}_q$

  for *each* k *in* $\mathcal{R}_q$ **do**

    Merge query-*keyword* pair $< q, k >$ into $\mathcal{D}_{new}$

  end

end

3.2 Discriminant-Based Filtering

In business applications like sponsored search's matching product, high precision is essential. The BERT-based classifier we use to further filter out bad cases has a similar structure with the classifier used in the sentence pair classification task illustrated in [7]. So we skip the exhaustive background description of the architecture and the training progress of the underlying model.

4 EXPERIMENTS

4.1 Dataset

Dataset for translation model. The seed data $\mathcal{D}_{old}$ is extracted by calculating query-query similarity based on same URL click-through information from the search engine's weblog [46], *keyword-keyword* similarity based on same advertiser purchase information from the ad database and synonyms replacement. This data is splitted into 3 parts: $\mathcal{D}^{PG}_{train}$ for training, $\mathcal{D}^{PG}_{dev}$ for developing, and $\mathcal{D}^{PG}_{test}$ for testing. The detailed statistics are shown in Table 3. The *keyword* repository $\mathcal{K}$ contains 102,025,475 *keywords*.

Table 3: Statistics of datasets for the translation model.

Dataset	Query Number	Total Pairs	Average Pairs (/query)
$\mathcal{D}^{PG}_{train}$	16,578,545	110,687,827	6.67
$\mathcal{D}^{PG}_{dev}$	93,467	613,143	6.56
$\mathcal{D}^{PG}_{test}$	99,588	656,602	6.59

Dataset for discriminant model. To make a balanced domain dataset for human evaluation, three kinds of query-*keyword* matching weblogs $\mathcal{D}_{exact}$, $\mathcal{D}_{phrase}$, $\mathcal{D}_{broad}$ are used, which correspond

Table 4: Results of different strategies.

Strategy	Beam Size	Diff ratio	BLEU-2	Dist-1/2	Precision	Decoding Time (ms/query)
BASE-M	30	17.089%	0.446	0.0014/0.047	83.5%	262.5
BASE-M	120	69.753%	0.356	0.0007/0.032	62.0%	2988.9
CW-M	30	40.510%	0.404	0.0015/0.058	83.0%	238.8
BCW-M	30	72.522%	0.387	0.0018/0.061	82.5%	254.1

to the *exact match*, *phrase match*, and *broad match* respectively. Among them, $\mathcal{D}_{exact}$ is probably synonymous and provides potential positive examples, while $\mathcal{D}_{phrase}$ and $\mathcal{D}_{broad}$ mainly contribute negative examples. These three data sources are merged with the proportion of 2:1:1. Then 170,000 data denoted as $\mathcal{D}_h^{PI}$ is sampled from it and sent to professionals for synonymous binary human evaluation. According to the human labels, 42.8% of $\mathcal{D}_h^{PI}$ are positive samples and 57.2% are negative. $\mathcal{D}_h^{PI}$ is further split into three parts: 90% of it is used as the domain specific data for BERT fine-tuning, which is denoted as $\mathcal{D}_{train}^{PI}$; and 5% of it is used for development, denoted as $\mathcal{D}_{dev}^{PI}$ and the remaining 5% denoted as $\mathcal{D}_{test}^{PI}$ is used for testing.

4.2 Implementation Details

For the translation model, the word embeddings are randomly initialized. The vocabulary contains 100,000 most frequent tokens in the training data $\mathcal{D}_{train}^{PG}$. The word embedding dimension and the number of hidden units are both set to 512. For multi-layer and multi-head architecture of the Transformer, 4 encoder and decoder layers and 8 multi-attention heads are used. And during the training, all layers are regularized with a dropout rate of 0.2. And the model's cross-entropy loss is minimized with an initial learning rate of 5×10^{-5} by Adam [15] with a batch size of 128.

The paraphrase discriminant model is implemented with BERT [7], which takes a query-*keyword* pair separated by a special token as input and predicts a synonymy label. The model contains 12 layers, 12 self-attention heads, and the hidden dimension size is 768. We initialize it with ERNIE [34], which learns Chinese lexical, syntactic and semantic information from a number of pretrained tasks, and fine-tuned it on $\mathcal{D}_{train}^{PI}$. The fine-tuned loss is minimized with an initial learning rate of 1×10^{-6} by Adam [15] with a batch size of 64. We evaluate our model after each epoch and stop training when the validation loss on $\mathcal{D}_{dev}^{PI}$ does not decrease after 3 epochs.

All of the experiments are run on a machine equipped with a 12-core Intel(R) Xeon(R) E5-2620 v3 clocked at 2.40GHz, a RAM of 256G and 8 Tesla K40m GPUs.

4.3 Results of Translation Model

For the translation model, we compared the retrieval performances of 3 different strategies. For all of the strategies, decoding is conducted in a Trie-constrained mode. For the sake of convenience, these strategies are abbreviated as follows:

- *BASE-M*: This is our base strategy, where the translation model is trained on the original training data $\mathcal{D}_{train}^{PG}$. And the prefix tree is built on $\mathcal{K}$.
- *CW-M*: In this strategy, the model is trained on the *CW*-transformed data. And the prefix tree is built on *CW*-trans–formed $\mathcal{K}$. The final result is made by joining the generated hypotheses with $\mathcal{K}$ based on the *CW* transformation.
- *BCW-M*: In this strategy, the model is trained on the *BCW*-transformed data. And the prefix tree is built on *BCW*-trans–formed $\mathcal{K}$. The final result is made by joining the generated hypotheses with $\mathcal{K}$ based on the *BCW* transformation.

Each of the trained models is utilized to decode towards those queries in $\mathcal{D}_{test}^{PG}$ to get a result data set $\mathcal{D}_1^{PG}$. Under the data augmentation framework, we expect the translation model could make a large data increment Δ while keeping a high precision. There are three major concerns: the size of Δ, the precision of Δ, and the decoding time for generating Δ. The following indicators are considered for evaluation:

- **Diff ratio** is defined as $\frac{|\mathcal{D}_1^{PG} - \mathcal{D}_{test}^{PG}|}{|\mathcal{D}_{test}^{PG}|}$, which is an indicator of the generalization ability.
- **BLEU-n** is an indirect indicator of the generation quality. Since most sentences in our scenario are short texts, we only consider BLEU-2.
- **Dist-n** measures the number of distinct N-grams within the set of generated data, which indicates the diversity among the generated paraphrases. Following previous studies [26], we use Dist-1,2.
- **Precision** indicates the proportion of synonymous pairs in generated data. Concretely, for each strategy, 400 query-*keyword* pairs are sampled from the generated results $\mathcal{D}_1^{PG}$ for binary human evaluation. We denote this dataset as $\mathcal{D}_{1h}^{PG}$ for later reference.

Rows 1, 3 and 4 in Table 4 show the retrieval performances for these three different strategies with a beam size of 30. We can see that: *BASE-M* could already make a certain amount of Diff ratio, which proves the feasibility of the data-augmentation-like framework. *CW-M* and *BCW-M* further enlarge the Diff ratio to 2.4 times and 4.2 times, compared with the base method. Meanwhile, the precision of *CW-M* and *BCW-M* are almost the same as that of *BASE-M*. The Dist indicator shows that the results' diversity has been improved by *CW-M* and *BCW-M*.

For further analysis, we evaluate the decoding results of *BASE-M* with a beam size of 120. As is shown in Table 4, although the diff ratio increases up to 4 times, which is nearly equal to *BCW-M* with a beam size of 30, the precision and diversity drop significantly. What's more, *BASE-M* has to spend more than 10 times of time to make it.

To conclude, *BCW-M* greatly enlarges the Diff ratio while maintaining a high level precision. It is fast and almost little extra time is consumed.

4.4 Results of Discriminant Model

GBDT (Gradient Boosting Decision Tree) [14] is commonly used in industry because of its fast online prediction and strong interpretability. Our baseline model is a GBDT model trained on a collection of human designed text similarity features. The features are listed as follows:

(1) Token level matching degree: max matching length, the proportion of matching and missing tokens, BM25, BLEU1 and BLEU2.
(2) Named entity similarity: whether the named entities in query and *keyword* are matched.
(3) Simple Approximate Bigram Kernel [24] based on dependency parsing tree.
(4) Document class similarity: whether query and *keyword* belong to the same document class.
(5) Semantic similarity: DSSM [13] trained with query-*keyword* click-noclick pairs shown in the weblog, and Word2vec based cosine similarity [22].
(6) Translation likelihood: the *BASE-M* translation score which is calculated by $P(keyword \mid \text{query})$.

GBDT is also trained on $\mathcal{D}^{PI}_{train}$, and validated on $\mathcal{D}^{PI}_{dev}$. And the model hyperparameters are optimized by grid search. For evaluation, we use two metrics: the area under an ROC curve (AUC) and recall under 95% precision. Table 5 shows the models' performances on $\mathcal{D}^{PI}_{test}$. To our surprise, BERT greatly exceeds GBDT's performance. For the AUC indicator, BERT outperforms GBDT by 11.4 percentage points. Under the precision of 95%, BERT's recall exceeds GBDT by 45 percentage points.

Table 5: Model performances of *BERT* VS *GBDT* on $\mathcal{D}^{PI}_{test}$. Recall indicates the recall ratio under the precision of 95%.

Strategy	AUC	Recall
GBDT	84.4%	21.8%
BERT	95.8%	66.8%

Table 6: Model performances on $\mathcal{D}^{PG}_{1h}$ generated by *BCW-M*. Recall indicates the recall ratio under the precision of 95%.

Strategy	AUC	Recall
GBDT	75.0%	25.2%
BERT	94.7%	91.3%

We also considered the models' performances on $\mathcal{D}^{PG}_{1}$ generated by *BCW-M*, since our motivation is to remove bad cases in the translations. Here we use the previously human evaluated dataset $\mathcal{D}^{PG}_{1h}$ for testing. As is shown in Table 6, BERT achieves a recall of 91.3% under the precision of 95%, which also far exceeds the GBDT's performance. (The difference of recall ratios in Table 6 and Table 5 comes from the difference of data distribution between $\mathcal{D}^{PG}_{1h}$ and $\mathcal{D}^{PI}_{test}$. In $\mathcal{D}^{PG}_{1h}$, positive examples account for 82.5% , while in $\mathcal{D}^{PI}_{test}$, positive examples account for 42.8%.)

BERT's dramatic improvement might come from the following points: a) BERT has a super large parameter space in contrast to the simple semantic similarity features like Word2vec and DSSM. The huge capacity and abundant pretraining make BERT being able to learn a lot of external semantic similarity knowledge, which is especially useful to alleviate the vocabulary mismatch in paraphrase identification. b) The paraphrase relationship between two sentences is usually judged by the alignment of their tokens. To some extent, the multi-head attention mechanism in the Transformer might make soft alignments in multiple semantic spaces.

4.5 Case Study and Discussion

4.5.1 Bag-of-Core-Words Transformation. Table 7 shows some typical cases for the *BCW-M* strategy. Due to space constraints, we only present the top 5 decoding results. The second column in Table 7 shows the final results of *BASE-M*, and the last column shows the raw generated results of *BCW-M* without joining the *keyword* repository, which is presented as Bag-of-Words form and the symbol '|' is used as the token separator. We can see that with *BCW-M*, the number of redundant translations decreases a lot. For example, before *BCW-M* is applied, most of the generated hypotheses for query (金的市场价格, market price of gold) are quite trivial: 金市场价格 just removes the auxiliary token 的, and 市场金价格 simply further reorders these tokens. When *BCW-M* is applied, nontrivial bag-of-words paraphrases like (多少 | 黄金 | 钱, Gold| how much) and (查询 | 价格 | 黄金, Gold |price| query) emerged into the top 5 hypotheses list. Similarly, for query (化妆的培训学校, Training school of makeup), four nearly identical translations (化妆培训学校, 化妆的培训学校，化妆培训的学校，培训化妆的学校) are generated under the base strategy. When *BCW-M* is applied, the results are not dull anymore.

4.5.2 Good Cases and Bad Cases. Table 8 shows some sampled results generated by our translation model. We can see that in most cases our model greatly captures the synonym relationship between query and *keyword*. The vocabulary mismatch problem has been alleviated to some extent. For example, the synonym relationships between (减肥, weight loss) and (瘦身, slimming), (油污, oil) and (油渍, greasy dirt), etc. have been captured. Some complex paraphrases like (什么导致宝宝长胎记, 婴儿身上胎记是什么原因) (What causes a birthmark on a baby, what is the reason of a birthmark) have been generated, which could not be simply accomplished by synonymous phrase replacement.

In the meanwhile, bad cases might also be generated. Sometimes, important intentions in the query might be discarded. For example, in the case of (怎么在跑步机上跑步, 怎么跑步减肥) (How to run on a treadmill, How to lose weight through running), the intention of 'treadmill' is lost.

4.5.3 Online Key Transformation. As mentioned in the introduction, the expanded data $\mathcal{D}_{new}$ is saved in the Key-Value lookup table for online usage. When an ad hoc query comes, we would first seek the raw query on the table, if the table is hit, its synonymous results would be retrieved immediately. Since *BCW* transformation is fast and synonym-keeping, it can be implemented online to make the lookup table hit by more queries. To make it clear, all the keys in the lookup table can be transformed based on *BCW* ahead of time. When an ad hoc query q comes, we can change it into Bag-of-Core-Words form $\widetilde{q}$, and use $\widetilde{q}$ to look up the table.

Table 7: Top-5 beam search results of *BASE-M* VS *BCW-M*.

<table>
<tr><th>Query</th><th>BASE-M</th><th>BCW-M</th><th>Query</th><th>BASE-M</th><th>BCW-M</th></tr>
<tr><td rowspan="5">金的市场价格
Market price of gold</td><td>金市场价格
Gold price in the market</td><td>价格 | 金 | 市场
gold| market|price</td><td rowspan="5">化妆的培训班
Training school of makeup</td><td>化妆培训学校哪家好
Which makeup training school is good?</td><td>化妆 | 培训 | 学校
makeup | school| training</td></tr>
<tr><td>价格 | 金 | 市场
gold| market|price</td><td>多少 | 黄金 | 钱
how much|gold</td><td>化妆培训学校
Makeup training school</td><td>班 | 化妆 | 学习
course |learning| makeup</td></tr>
<tr><td>金价格
Gold price</td><td>价格 | 金
price|gold</td><td>化妆的培训学校
Training school of makeup</td><td>化妆 | 机构 | 培训
institution | makeup| training</td></tr>
<tr><td>金的价格
Price of gold</td><td>查询 | 价格 | 金
gold |price| query</td><td>化妆培训的学校
School of makeup training</td><td>彩妆 | 培训 | 学校
cosmetic| school| training</td></tr>
<tr><td>市场金价格
Market gold price</td><td>黄金 | 价格 | 最新
current |gold | price</td><td>培训化妆的学校
School of training makeup</td><td>化妆 | 学 | 学校
learning| makeup| school</td></tr>
</table>

Table 8: Some typical synonymous *keywords* generated by the translation model.

Query	Generated *Keywords*	Label
男士减肥机构	男性瘦身机构	Good
Men's Weight Loss Agency	Agency for men slimming	Good
厨房油污怎样清除	厨房油渍如何去除	Good
How to clean off oil in the kitchen	How to remove kitchen greasy dirt	Good
什么导致宝宝长胎记	婴儿身上胎记是什么原因	Good
What causes a birthmark on a baby	What is the reason of a birthmark	Good
怎么在跑步机上跑步	怎么跑步减肥	Bad
How to run on treadmill	How to lose weight through running	Bad

4.6 The published Dataset

Our work has a close relationship with paraphrase generation and paraphrase identification. Most of the existing large scale paraphrasing datasets (MSCOCO, SNLI, etc.) are in English and high-quality Chinese dataset is extremely scarce in this domain. The most related dataset is a 24K sized dataset LCQMC [18]. However, it focuses on general intent matching rather than paraphrasing. To promote the Chinese paraphrasing research, we decide to publish a large scale high-quality dataset containing 500K commercial synonymous short text pairs along with this paper. This dataset is produced in the following steps: A translation model trained with the *BASE-M* strategy is used to make unconstrained decoding for real-world frequent queries, where the prefix tree constraint is discarded. Then our discriminant model is used to filter out bad cases. Finally, some heuristic rules are devised to filter out trivial translations. Manual sampling evaluation shows that this dataset has a precision of 95%.

4.7 Online Experiments

A real online A/B test experiment is deployed on Baidu's commercial advertising system, where two fractions of search flow are sent to the experimental group and the control group independently. $\mathcal{D}_{old}$ is used as the Key-Value table in the control group and $\mathcal{D}_{old} + \Delta$ in the experimental group. We use 10,000,000 frequent queries, and the translation model is trained with *BCW-M* strategy.

For each group, #{searches} is used to denote the total number of queries it received, #{clicks} to denote the corresponding number of clicks, and revenue to denote the search company revenue. The following metrics are calculated independently for each group to evaluate the performance of our method.

- SHOW denotes the total number of ads shown to users.
- CTR $= \frac{\#\{\text{clicks}\}}{\#\{\text{searches}\}}$, which denotes the average clicks received by the search engine.
- ACP $= \frac{\sum \text{price}}{\#\{\text{clicks}\}} = \frac{\text{revenue}}{\#\{\text{clicks}\}}$, which denotes the average click price paid by the advertisers.
- CPM $= \frac{\text{revenue}}{\#\{\text{searches}\}} \times 1000 = \text{CTR} \times \text{ACP} \times 1000$, which denotes the average revenue received by the search engine for 1000 searches.
- Quality refers to the query-*keyword* synonym relationship. For each side of this A/B experiment, 600 query-*keyword* cases under the *exact match type* (excluding literally identical ones) are sampled from the system's ad weblog and are sent for binary human evaluation. And the quality score is calculated as the proportion of the synonymous pairs.

Table 9: Online A/B Test performance of our method.

SHOW	CTR	ACP	CPM	Quality
+0.8%	+1.02%	+0.62%	+1.64%	+1.2%

Table 9 shows the relative improvements in these indicators. We can see that the incremental dataset Δ increases the CPM by 1.64%, which is a significant improvement for our revenue. We give our intuitive explanation from the demand-supply perspective. The sponsored search system aims to match query flows to the advertisers' demands. Detecting more synonym relationships between queries and *keywords* helps to get more ads into the downstream auction phase. The ACP's 0.62% growth shows the competition in the ad queue has been intensified. Finally, the number of shown ads increased by 0.8%, and the clicks increased by 1.02%. The growth in clicks and prices combined to bring about the final CPM growth. In the meanwhile, the human evaluation demonstrates that the ads' relevance quality has not been deteriorated.

5 CONCLUSIONS

In this paper we have developed a simple but effective data-augmentation approach for addressing the synonymous *keyword* retrieval problem for frequent queries in sponsored search. Based on a seed paraphrasing dataset, a sequence to sequence translation model is trained and used to decode out more synonymous pairs to *expand* the data. To ensure high precision for commercial usage, a domain fine-tuned BERT is used to filter out bad cases. During the translation phase, a novel scheme is introduced to make the decoding more effective: Bag-of-Core-words transformation, which enlarges the diff 4.2 times while almost keeping the original precision. During the discrimination phase, BERT outperforms the traditional feature-driven GBDT model by 11 percentage points. Our approach has been successfully applied in Baidu's sponsored search. To the best of our knowledge, this is the first work to address the synonymous *keywords* retrieval problem. Moreover, as a byproduct, 500K high quality commercial Chinese synonymous pairs have been published along with this paper.

REFERENCES

[1] Javad Azimi, Adnan Alam, and Ruofei Zhang. 2015. Ads keyword rewriting using search engine results. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 3–4.

[2] Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1415–1425.

[3] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. 2014. A semantic matching energy function for learning with multi-relational data. *Machine Learning* 94, 2 (2014), 233–259.

[4] Andreas Chandra and Ruben Stefanus. 2020. Experiments on Paraphrase Identification Using Quora Question Pairs Dataset. *arXiv preprint arXiv:2006.02648* (2020).

[5] Xiuying Chen, Daorui Xiao, Shen Gao, Guojun Liu, Wei Lin, Bo Zheng, Dongyan Zhao, and Rui Yan. 2019. RPM-Oriented Query Rewriting Framework for E-commerce Keyword-Based Sponsored Search. *arXiv preprint arXiv:1910.12527* (2019).

[6] Dipanjan Das and Noah A Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 468–476.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018).

[8] Miao Fan, Wutao Lin, Yue Feng, Mingming Sun, and Ping Li. 2018. A globalization-semantic matching neural network for paraphrase idnetification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2067–2075.

[9] Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348* (2017).

[10] Mihajlo Grbovic, Nemanja Djuric, Vladan Radosavljevic, Fabrizio Silvestri, and Narayan Bhamidipati. 2015. Context-and content-aware embeddings for query rewriting in sponsored search. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 383–392.

[11] Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[12] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*. 2042–2050.

[13] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2333–2338.

[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*. 3146–3154.

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

[16] 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)*. ACM, New York, NY, USA, 500–508.

[17] Yijiang Lian, Zhijie Chen, Jinlong Hu, Kefeng Zhang, Chunwei Yan, Muchenxuan Tong, Wenying Han, Hanju Guan, Ying Li, Ying Cao, et al. 2019. An end-to-end Generative Retrieval Method for Sponsored Search Engine–Decoding Efficiently into a Closed Target Domain. *arXiv preprint arXiv:1902.00592* (2019).

[18] 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.

[19] Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 182–190.

[20] Azarakhsh Malekian, Chi-Chao Chang, Ravi Kumar, and Grant Wang. 2008. Optimizing query rewrites for keyword-based advertising. In *Proceedings of the 9th ACM conference on Electronic commerce*. ACM, 10–19.

[21] Kathleen R McKeown. 1983. Paraphrasing questions using given and new information. *Computational Linguistics* 9, 1 (1983), 1–10.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[23] Massimo Nicosia and Alessandro Moschitti. 2017. Accurate sentence matching with hybrid siamese networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2235–2238.

[24] Şaziye Betül Özateş, Arzucan Özgür, and Dragomir Radev. 2016. Sentence similarity based on dependency tree kernels for multi-document summarization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2833–2838.

[25] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[26] Sunghyun Park, Seung-won Hwang, Fuxiang Chen, Jaegul Choo, Jung-Woo Ha, Sunghun Kim, and Jinyeong Yim. 2019. Paraphrase Diversification Using Counterfactual Debiasing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 6883–6891.

[27] Casper Petersen, Jakob Grue Simonsen, and Christina Lioma. 2016. Power Law Distributions in Information Retrieval. *ACM Trans. Inf. Syst.* 34, 2, Article 8 (Feb. 2016), 37 pages.

[28] 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*. 25–32.

[29] Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. Natural Language Understanding with the Quora Question Pairs Dataset. *arXiv preprint arXiv:1907.01041* (2019).

[30] Yatian Shen, Jifan Chen, and Xuanjing Huang. 2016. Bidirectional long short-term memory with gated relevance network for paraphrase identification. In *Natural Language Understanding and Intelligent Applications*. Springer, 39–50.

[31] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in neural information processing systems*. 801–809.

[32] Alex Sokolov and Denis Filimonov. 2019. Neural Machine Translation For Paraphrase Generation. (2019).

[33] Amanda Spink, Dietmar Wolfram, Major B. J. Jansen, and Tefko Saracevic. 2001. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology* 52, 3 (2001), 226–234.

[34] Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced Representation through Knowledge Integration. *CoRR* abs/1904.09223 (2019). arXiv:1904.09223 http://arxiv.org/abs/1904.09223

[35] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3104–3112.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you

need. In *Advances in neural information processing systems*. 5998–6008.

[37] Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take the'para-farce'out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop 2006*. 131–138.

[38] Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2019. A task in a suit and a tie: paraphrase generation with semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 7176–7183.

[39] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814* (2017).

[40] Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2010. Paraphrase Generation as Monolingual Translation: Data and Evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*. https://www.aclweb.org/anthology/W10-4223

[41] Wenpeng Yin and Hinrich Schütze. 2015. Convolutional neural network for paraphrase identification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 901–911.

[42] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics* 4 (2016), 259–272.

[43] Wei Vivian Zhang, Xiaofei He, Benjamin Rey, and Rosie Jones. 2007. Query rewriting using active learning for sponsored search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 853–854.

[44] Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130* (2019).

[45] Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - ACL-IJCNLP '09* (2009). https://doi.org/10.3115/1690219.1690263

[46] Shiqi Zhao, Haifeng Wang, and Ting Liu. 2010. Paraphrasing with search engine query logs. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 1317–1325.

[47] Ingrid Zukerman and Bhavani Raskutti. 2002. Lexical query paraphrasing for document retrieval. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1–7.