OPEN ACCESS Freely available online

PLOS ONE

Phrasal Paraphrase Based Question Reformulation for Archived Question Retrieval

Yu Zhang[1☯], Wei-Nan Zhang[1☯], Ke Lu[2], Rongrong Ji[3]*, Fanglin Wang[4]*, Ting Liu[1]

1 Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, Harbin City, Heilongjiang, China, **2** Graduate University of Chinese Academy of Sciences, Beijing City, China, **3** Department of Cognitive Science, Xiamen University, Xiamen City, Fujian, China, **4** School of Computing, National University of Singapore, Singapore, Singapore

Abstract

Lexical gap in cQA search, resulted by the variability of languages, has been recognized as an important and widespread phenomenon. To address the problem, this paper presents a question reformulation scheme to enhance the question retrieval model by fully exploring the intelligence of paraphrase in phrase-level. It compensates for the existing paraphrasing research in a suitable granularity, which either falls into fine-grained lexical-level or coarse-grained sentence-level. Given a question in natural language, our scheme first detects the involved key-phrases by jointly integrating the corpus-dependent knowledge and question-aware cues. Next, it automatically extracts the paraphrases for each identified key-phrase utilizing multiple online translation engines, and then selects the most relevant reformulations from a large group of question rewrites, which is formed by full permutation and combination of the generated paraphrases. Extensive evaluations on a real world data set demonstrate that our model is able to characterize the complex questions and achieves promising performance as compared to the state-of-the-art methods.

Citation: Zhang Y, Zhang W-N, Lu K, Ji R, Wang F, et al. (2013) Phrasal Paraphrase Based Question Reformulation for Archived Question Retrieval. PLoS ONE 8(6): e64601. doi:10.1371/journal.pone.0064601

Editor: Derek Abbott, University of Adelaide, Australia

Received February 14, 2013; **Accepted** April 15, 2013; **Published** June 21, 2013

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: rrji@ee.columbia.edu (RRJ); hardegg@gmail.com (FLW)

☯ These authors contributed equally to this work.

Introduction

As the blooming of Web 2.0, community question answering services (cQA) have emerged as popular means for knowledge dissemination and information seeking, such as Yahoo! Answers, WikiAnswer and Quora, etc. Over times, an overwhelming amount of QA pairs with high quality devoted by human intelligence has been accumulated as comprehensive knowledge base, which greatly facilitates users to seek precise information by querying in natural language, rather than issuing the key words and painstakingly browsing through large ranked lists of results in order to look for the correct answers.

However, question retrieval is nontrivial. One major reason is the lexical gap between the queried questions and the archived historical questions in repositories. This is due to the variability of languages, which directly leads to both of the content contributors and seekers conveying their intentions in different word forms, even describing the same meanings. As shown in Table 1, Q1 and Q2 are semantically similar but lexically different questions.

Paraphrasing techniques can gracefully bridge the lexical gap in question retrieval. As described in [1], paraphrases are alternative ways of conveying the same information. For example, the phrases ''catch a cold'' and ''get colds'' are paraphrases as well as the phrases ''expectant mother'' and ''pregnant''. The existing approaches generally fall into two categories according to different granularities. One is lexical-level [2–4], aiming to acquire word level paraphrases by extracting the synonyms from dictionaries or monolingual and bilingual corpora. The other is sentence-level, extracting semantically similar questions from query log or web search results [5,6]. However, the former fails to characterize the relations among terms. It means that the words are independent with their adjacent context. The latter still faces the challenges that are not easy to tackle, such as the deep understanding of the complex questions with sophisticated syntactic and semantic, and taking the context into consideration to generate the equivalent candidates. Therefore, new approaches towards paraphrasing questions in appropriate level of granularity are highly desired.

In this study, we propose a question reformulation scheme that makes an intelligent use of paraphrases in phrase-level of the queried questions. It is able to increase the likelihood of finding the relevant or similar questions which are well answered or voted by other users. As shown in Figure 1, the scheme consists of three components. Given a question, the first component detects the involved key-phrases by jointly exploring the corpus-dependent knowledge and question-aware cues. Second, it automatically paraphrases each identified key-phrase utilizing multiple online translation engines. The third component selects the most relevant reformulations from a large group of question rewrites, which is formed by full permutation and combination of the generated paraphrases. By conducting extensive experiments on a large data set, we demonstrate that our proposed scheme yields significant gains in question retrieval performance and remarkably outperforms other state-of-the-art technologies.

The remainder of this paper is organized as follows. The Related Work section briefly reviews the related work. In Sections of the Key-Phrase Detection and Proposed Phrasal Paraphrase

Table 1. Representative questions for lexical gap illustration.

Query:
Q1: Can you catch a cold from cold temperature?
Expected:
Q2: Does cold weather affect actually catching a cold?
Not Expected:
Q3: How can you catch a cold?
Q4: Can you catch a cold from getting your head wet?

doi:10.1371/journal.pone.0064601.t001

Extraction, we introduce the key-phrase identification and paraphrase-candidates generation, respectively. We then describe the paraphrase selection and question reformulation approach in Section Question Reformulation Generation. Experimental results and analysis are presented in Section Experimental Results, followed by the conclusion and future work in the last Section.

Related Work

Key Phrase Detection

Question sentences in cQA are usually surrounded by various description sentences, and expressed by informal languages such as question mark etc. Key phrase detection is important for not only QA but also other tasks, such as tag-based image retrieval [7,8], tweet summarization [9], and social media analysis [10–12]. In 2009, Wang et al. [13] proposed a syntactic tree matching model to find similar questions, which was demonstrated to be robust against grammatical errors. One year later, they [14] exploited salient patterns to improve question retrieval results. Bendersky et al. [15] captured the lexical, statistical and n-grams features to identify key concepts (phrases) from verbose queries. Later, [16] proposed a learning method to estimate concept weights in verbose queries using Markov random field model. Then they parameterized concept weights and integrated the weights into query expansion model [17]. Recently, they modeled concepts dependencies in queries using hypergraphs [18]. However, these work only focuses on distinguishing key concepts from other non-key concepts and the differences between key concepts are not considered. In this paper, we propose a ranking based method to tackle the problem and improve the performance of key concept detection approach.

Question Term Weighting

The queries are usually depicted in form of natural languages including various sophisticated syntactic and semantic features, rather than the simple key words supported by current dominant web search engines. Therefore, one of the major challenges is how to capture the syntactic and semantic relations among query terms. Song et al. [19] and Srikanth et al. [20] shifted from the unigram to bigram and bi-term in language model to capture the term dependence in queries. An advanced dependency language model was proposed in [21], which exploited term relations using dependency parsing and integrated the dependency relations into the traditional language model. Further, Cui et al. [22] tried to capture the similarity between different dependency relation paths of the two same terms using translation model.

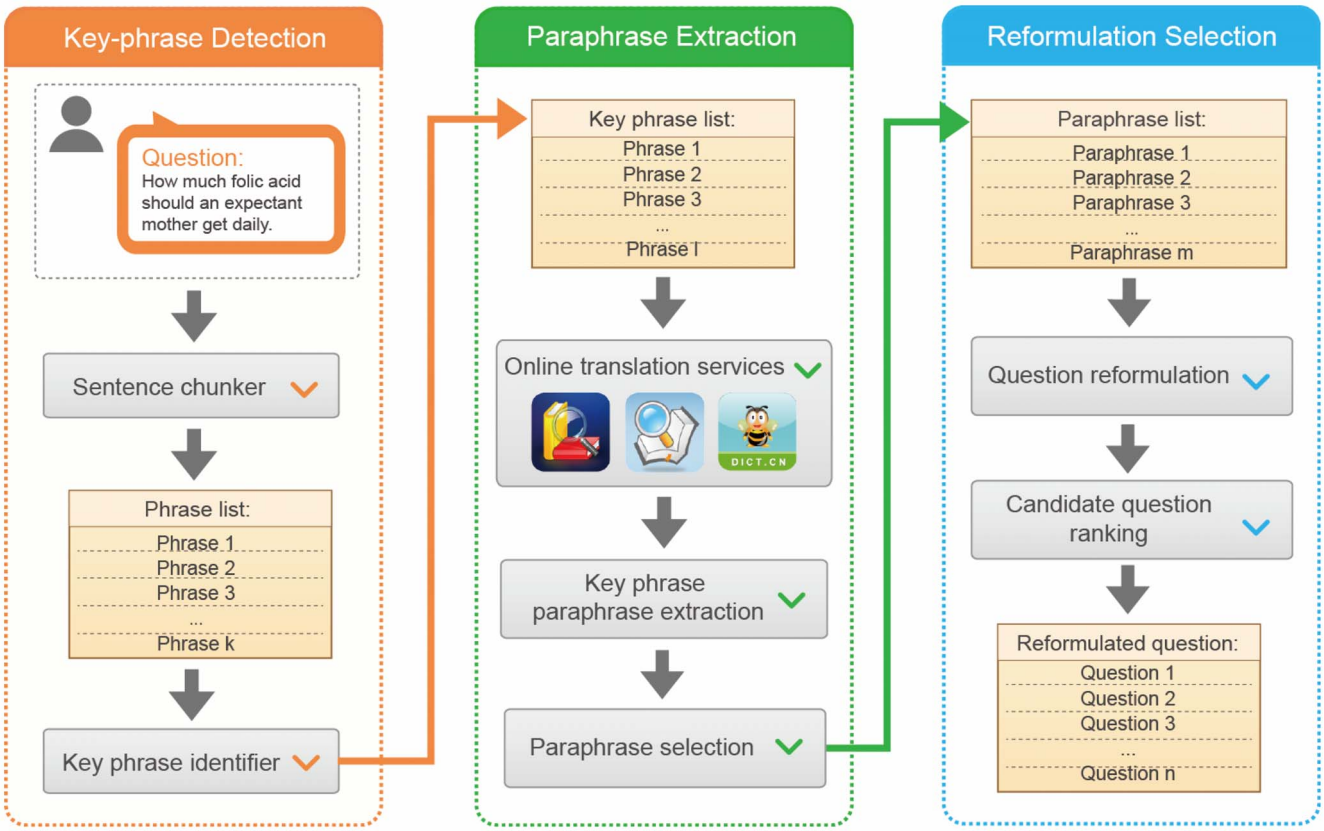

Figure 1. The schematic illustration of the proposed question reformulation scheme.
doi:10.1371/journal.pone.0064601.g001

To compare with other indexed documents, the archived questions in current cQA forums are usually very short, which are hard to be matched by conventional lexicon and statistics based approaches. Hence, the research community in cQA retrieval faced the challenge of question expansion. Some researchers utilized the lexical dictionary, such as WordNet, to fuzzily match the synonyms between the queried questions and the reservoirs [2,3,23]. Some others [1,24,25] employed the term co-occurrence to expand queries and the translation model to estimate the semantic similarities among words.

Pivot Paraphrasing Approach

The diversity of expressions in natural language for the similar questions directly results in mismatch in the question retrieval task. This is the so-called lexical gap. Word-level [2–4] and sentence-level [5,6] question paraphrasing technologies have been proposed to relieve this problem, but obtained not very satisfying results.

Key-Phrase Detection

A key-phrase in a given question is generally one or more words constituting an indispensable part of the question meaning. Beyond key-phrases, questions tend to contain several redundant chunks, which have grammatical meanings for communication among humans to help understanding users' intents. However, they are almost useless in describing the key concepts. For instance, for the question "How much folic acid should an expectant mother get daily?", the two phrases "folic acid" and "expectant mother" essentially express the meaning of the original question and "daily" is temporal expression. They play the important roles in question match. The use of other chunks may bring unpredictable noise to question retrieval. Therefore, to distinct the key-phases from others, we propose a heuristic framework as illustrated in Figure 2.

The core part in this framework is an automatically constructed term-weight pair vocabulary. Given a question, we extract its constituent key-phrases as follows:

1. We segment the given question into chunks using the openNLP (http://incubator.apache.org/opennlp/) toolkit.
2. We construct a vocabulary containing term-weight pairs by leveraging the corpus-dependent knowledge. Specifically, we archive all the terms selected from our data set and their estimated weights (to be introduced in Section Question Terms Weighting).
3. For each chunk, we fetch each term weight from our constructed vocabulary. If one of its term weights satisfies the question-dependent constraints (to be introduced in Section Question-aware Constraints), the chunk is classified as a key-phrase. The detected key-phrase is used as the basic unit for paraphrasing.

Question Terms Weighting

In this paper, we quantify question terms using the following equation inspired by BM25 [26] and LM [27],

$$w(t_i) = \log(tf(t_i) + \lambda) \times \log \frac{N}{df(t_i) + \lambda} \quad (1)$$

where t_i represents the i-th term in question, $tf(t_i)$ and $df(t_i)$ represent term frequency and document frequency, respectively. In question retrieval task, a document usually indicates a candidate question for retrieval. N is the total number of questions in our corpus. λ is a smoothing parameter. It is observed that term weight is independent of a specific question, but aware of corpus knowledge. Later, we will present the term weighting based key-phrase detection approach.

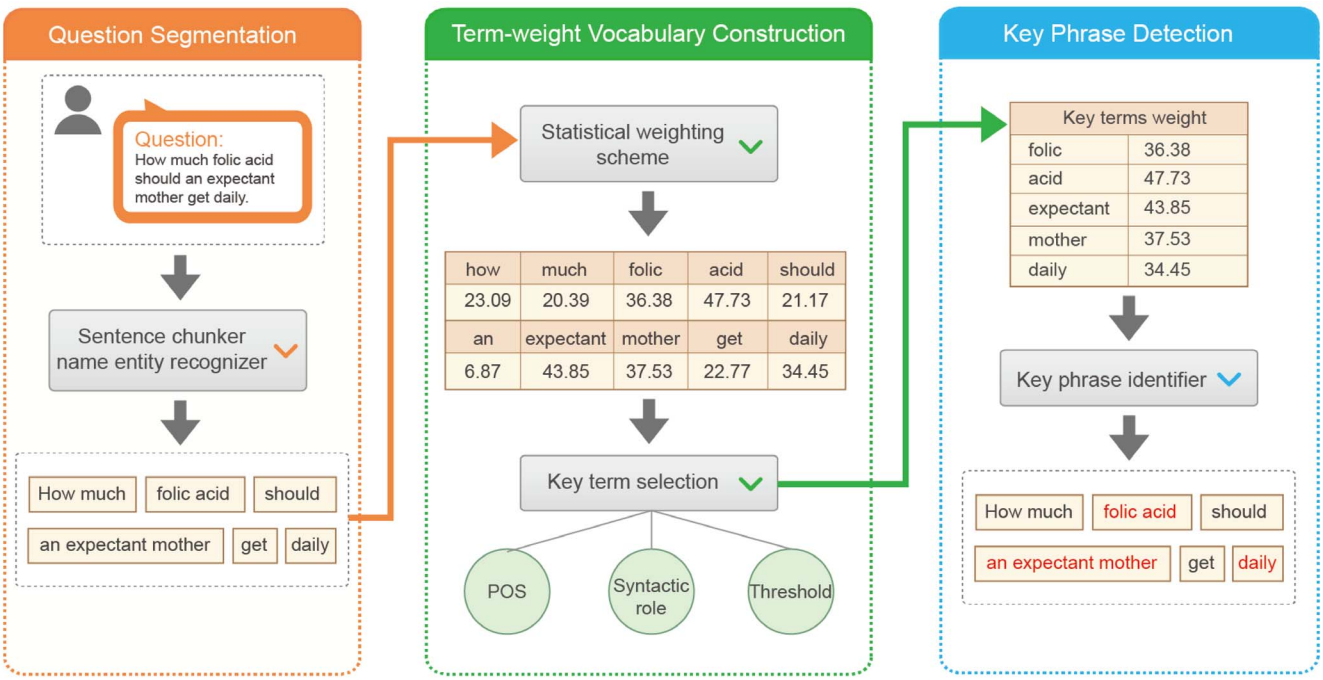

Figure 2. An illustration of key-phrases detection from a given question in natural languages.
doi:10.1371/journal.pone.0064601.g002

Question-aware Constraints

Actually, we regard the key-phrase detection task as a key term identification problem. To be specific, if a chunk contains a key term, it will be classified as a key-phrase. Meanwhile, we predict whether a term is a key term by exploring multi-faceted cues, such as the part-of-speech (POS) of the term, as well as the term syntactic role in the question dependency parsing tree. Here, we use the Stranford CoreNLP [28] toolkit to get the POS and the syntactic roles of terms.

Given a question q, we employ three question-dependent thresholds to capture the average weights inside of q as follows.

The first one is the quadratic mean value based weight. Quadratic mean value of a set of values is the square root of the arithmetic mean (average) of the squares of the original values. Here, we represent the quadratic mean value as w_q which can be calculated as follows:

$$w_q(q) = \sqrt{\frac{1}{N}\sum_{t_i \in q} w(t_i)^2} \tag{2}$$

where N is the question length on words. t_i represents the i-th term in q, and $w(t_i)$ represents the weight of t_i, which is computed by Equation 1. The second threshold is the arithmetic mean weight which is represented as w_a and computed as follows:

$$w_a(q) = \frac{1}{N}\sum_{t_i \in q} w(t_i) \tag{3}$$

The arithmetic mean is the central tendency of a collection of numbers taken as the sum of the numbers divided by the size of the collection.

The third threshold is geometric mean weight which is represented as w_g and computed as follows:

$$w_g(q) = \sqrt[N]{\prod_{t_i \in q} w(t_i)} \tag{4}$$

The geometric mean, in mathematics, is a type of mean or average, which indicates the central tendency or typical value of a set of numbers.

As the weight of each term is positive and when question q is given, there exists a relationship among the above three thresholds in mathematics as follow, the relationship may be special when question length equals to 2. Actually, the average question length in our data set equals to 10.8 and even the shortest question still contains 5 words.

$$w_q(q) \geq w_a(q) \geq w_g(q) \tag{5}$$

Then, we use a heuristic method to identify the key terms.

In Table 2, t_i represents the i-th term in question q. $w(t_i)$ indicates the weight of term t_i; $POS(t_i)$ indicates the POS of term t_i; I_{SR} indicates the set of important syntactic role. Here, we empirically define the "nsubj" and "dobj" as the important syntactic roles, where in Stanford CoreNLP toolkit, dependency parsing labels "nsubj" and "dobj" represent the noun subject and object of predicate respectively.

In Table 2, the three thresholds are used as the question-aware constraints for key term identification. Given a query question and its original term weights, these thresholds can indicate the average term importance inside of the question in three measurements. By using the average term importance, we actually want to capture the different distributions of the term weights and select the salient terms as the key terms. As the three measurements have different scales, we add the external information, which is the POS and syntactic role, to distinguish the key and non-key terms.

Table 2. The heuristic method for key-phrase detection.

Key-term Decision Rules	Result
$w(t_i) \geq w_q(q)$	True
$POS(t_i) == \text{VB}$ and $I_{SR}(t_i) == \text{True}$ and $w(t_i) \geq w_a(q)$	True
$POS(t_i) == \text{NN}$ and $I_{SR}(t_i) == \text{True}$ and $w(t_i) \geq w_g(q)$	True
Otherwise	False

doi:10.1371/journal.pone.0064601.t002

Here, VB and NN represent two POS sets respectively. For instance, VB contains the POS of VBP, VBZ, VBG etc. NN has similar conditions with VB. As the relationship of $w_a(q) \geq w_g(q)$ exists and NN has been proven to be more reliable [15,29,30] in information retrieval, we empirically consider that NN is more important than VB as the decision rules representing. As there exists the relation in 5 and the different importance between NN and VB, we set the priority of the decision rules as shown in Table 2.

The term weight threshold design is inspired by [31,32], and this approach is widely used in [33,34]. In addition, there are many research efforts focused on key term selection, ranking and weighting in document and other scenes, such as [25,35,36]. Although the three thresholds are designed heuristically, they are based on the intra-question features which are depended on the average term weights inside of a question. Hence, they are robust to the key term detection task.

Figure 3 shows an example of key term detection result of question "How much folic acid should an expectant mother get daily?". The words in red color are viewed as key terms, followed by the POS tags, terms' weights and the values of three thresholds.

Proposed Phrasal Paraphrase Extraction

In this section, we introduce our proposed online translation engine-based approach to paraphrasing each identified key-phrase from a given question. The original idea of paraphrase extraction is machine translation process within monolingual [37,38] and bilingual [1,39–41] parallel corpora. We extend the state-of-the-art technology by using online translation engines as pivot language generators. Specifically, we first translate the detected key-phrase e_j into pivot language phrase f in the form of intermediate language through multiple online translation engines, such as iciba (http://www.iciba.com/), youdao (http://dict.youdao.com/) and dict (http://dict.cn/). Here, we only use Chinese as pivot language, multi-pivot languages can be used in our future work for paraphrase extraction. We then translate f back to English and obtain the paraphrases of the original detected key-phrase. Formally, we estimate the paraphrase probability between the key-phrase and its corresponding paraphrase as:

$$p(e_i|e_j) = \sum_f p(f|e_j)p(e_i|f) \tag{6}$$

where $p(f|e_j)$ represents the probability of the original language phrase e_j translated into the pivot language phrase f. While $p(e_i|f)$ stands for the translation probability of paraphrase candidate e_i given f. These two conditional probabilities can be estimated in a

How	much	folic	acid	should	an	expectant	mother	get	daily
WRB	JJ	JJ	NN	MD	DT	JJ	NN	VB	RB
23.09	20.39	36.38	27.03	21.17	16.87	43.85	26.57	27.77	24.45
w_q=27.81			w_a=26.76				w_g=25.80		

Figure 3. An example for illustrating term weighting scheme and key term selection.
doi:10.1371/journal.pone.0064601.g003

unified form as:

$$p(f|e)=\frac{\text{count}(e,f)}{\sum_f \text{count}(e,f)} \tag{7}$$

To make the paraphrase model more robust, we integrate it with the language model and restate it as:

$$p(e_i|e_j)=p(e_i)\sum_f p(f)p(f|e_j)p(e_i|f) \tag{8}$$

$$p(e_i)=\frac{\text{count}(e_i)}{\sum_{e_i}\text{count}(e_i)} \tag{9}$$

It can be intuitively interpreted as follows: the probability of the phrase e_i is the ratio of its frequency to the sum of all the phrases frequency. Where, $p(f)$ can be estimated in the same way as $p(e_i)$.

Here, we present an instance to show how our paraphrases generation component works. For the detected key-phrase "expectant m other", we first send it to the three dictionary systems as a query, and then we collect its translations, dictionary explanations and the network interpretations. According to our statistics, "孕妇" appeared 3 times and "待产孕妇" appeared 1 time. Following that, we translated these Chinese phrases back to English. Taking "孕妇" as an example, we obtained "expectant mother", "pregnant mother" and "gravida" 3 times, respectively. Therefore, count("孕妇", "expectant mother") = 6, count("孕妇", "gravida") = 3.

Question Reformulation Generation

As shown in Figure 4, for the question "How much folic acid should an expectant mother get daily?", 179 question reformulating candidates are generated with shuffling of the paraphrases in phrase level. However, some rewrites may drift away from the original meaning or not satisfy the common ways of language expression, such as this candidate "How much folate should a mother get daily".

In this paper, we seamlessly integrate the Viterbi algorithm [42] with our proposed language model to filter out the "incorrect" reformulations. The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states, which is called Viterbi path, that results in a sequence of observed events. Before introducing our approach, we first define some notations. For a given question q, t denotes the detected key-phrase position starting from 0. And Syns(t) is its corresponding paraphrase set. $\delta_t(\text{syn}_{t,i})$ is the Viterbi variables meaning the maximum sentence probability from the beginning of the question to the current t-th key-phrase, which is replaced with $syn_{t,i}(\text{syn}_{t,i}\in\text{Syns}(t))$. The Viterbi value is formally defined as:

$$\delta_t(\text{syn}_{t,i})=\max_{\substack{\text{syn}_{t,i}\in\text{Syns}(t)\\ \text{syn}_{t-1,j}\in\text{Syns}(t-1)}}\delta_{t-1}\left(\text{syn}_{t-1,j}\right)\times p\left(\text{syn}_{t,i}|\text{syn}_{t-1,j}\right)\times p\left(\text{syn}_{t,i}\right) \tag{10}$$

where $p(\text{syn}_{t,i})=p(\text{syn}_{t,i}|e_t)$ which can be estimated by Equation 8, e_t is the t-th key-phrase in question q.

To estimate the conditional probability $p\left(\text{syn}_{t,i}|\text{syn}_{t-1,j}\right)$, a novel bi-gram language model in phrase level is proposed by exploiting the external knowledge, Google N-gram, which was extracted from 1T web corpus (http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13). To be specific, supposing $\text{syn}_{t,i}$ and $\text{syn}_{t-1,j}$ are segmented into the word sequences $w_{i1},w_{i2},...,w_{in}$ and $w_{j1},w_{j2},...,w_{jm}$ respectively. The conditional probability $p\left(\text{syn}_{t,i}|\text{syn}_{t-1,j}\right)$ can be formulated as:

$$\begin{aligned}p\left(\text{syn}_{t,i}|\text{syn}_{t-1,j}\right)&=\frac{p\left(\text{syn}_{t,i},\text{syn}_{t-1,j}\right)}{p\left(\text{syn}_{t-1,j}\right)}=\frac{p\left(\text{syn}_{t-1,j}|\text{syn}_{t,i}\right)p(\text{syn}_{t,i})}{p\left(\text{syn}_{t-1,j}\right)}\\ &=\frac{p(\text{syn}_{t,i})\,\Pi_{w_{jk}\in\text{syn}_{t-1,j}}p\left(w_{jk}|\text{syn}_{t,i}\right)}{\Pi_{w_{jk}\in\text{syn}_{t-1,j}}p\left(w_{jk}\right)}=\frac{\Pi_{w_{jk}\in\text{syn}_{t-1,j}}p\left(\text{syn}_{t,i}|w_{jk}\right)}{p(\text{syn}_{t,i})^{m-1}}\\ &=\frac{\Pi_{w_{il}\in\text{syn}_{t,i}}\Pi_{w_{jk}\in\text{syn}_{t-1,j}}p\left(w_{il}|w_{jk}\right)}{p\left(\text{syn}_{t,i}\right)^{m-1}}\\ &=\frac{\Pi_{w_{il}\in\text{syn}_{t,i}}\Pi_{w_{jk}\in\text{syn}_{t-1,j}}p\left(w_{il}|w_{jk}\right)}{\left(\Pi_{w_{il}\in\text{syn}_{t,i}}p(w_{il})\right)^{m-1}}\end{aligned} \tag{11}$$

where $l=1,2,..n$ and $k=1,2,...m$. Meanwhile, $p\left(w_il|w_jk\right)$ is estimated by Equation 12 with add-delta(Lidstone's law) smoothing, where δ is the smoothing parameter and $|V|$ is the size of vocabulary.

$$p\left(w_{il}|w_{jk}\right)=\frac{\text{count}\left(w_{il},w_{jk}\right)+\delta}{\sum_i\text{count}\left(w_{il},w_{jk}\right)+\delta\times|V|} \tag{12}$$

Now, we have a ranked list of question reformulations for each queried question.

At last, we get the k best reformulated questions as the final results. We will present how to fix k in Section Experimental Results.

For instance, "How much folic acid should an expectant mother get daily?", the top five reformulated questions of the original question are shown in the following Figure 5.

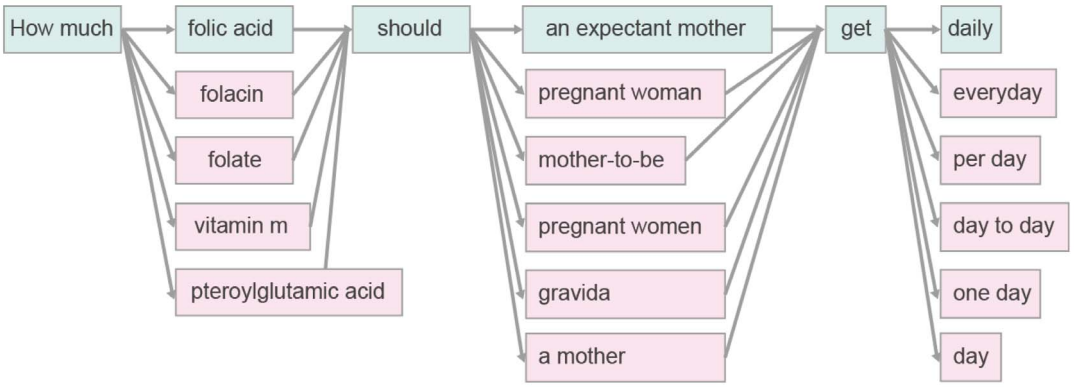

Figure 4. Illustration of question reformulations in the form of Viterbi decoding structure.
doi:10.1371/journal.pone.0064601.g004

Experimental Results

Data Set

We collected a large real world data set from Yahoo! Answers, which contains 1,123,134 unique questions as our searching corpora and covering a wide range of topics, including health, internet, etc. We randomly selected 200 questions from this collection as our searching queries. We processed these queries by manually filtering ill-formed questions, ungrammatical or incomplete ones. After preprocessing, we obtained 168 questions. From the remaining questions, we randomly chose 140 questions for testing, and the rest 28 questions were used for development. The experimental data is available at http://pan.baidu.com/share/link?shareid = 343582&uk = 2903372971.

On Paraphrase Extraction

We employ manual judgement, which is used in [1,39], to evaluate the phrasal paraphrase extraction method. Two native English speakers are involved in the processing to produce their judgements as to whether the extracted paraphrases are in the same meaning with the original phrases and grammatically correct. We set 20% of overlapped data to compute their agreements and get $\kappa = 0.617$, which is interpreted as "good" agreement. Here the kappa coefficient is a statistical measure of inter-annotator agreement for categorical items.

For performance comparison, we introduce the state-of-the-art method on phrasal paraphrase extraction, which is proposed in [39]. The toolkit which is published by Callison-Burch et al. can be found in http://www.cs.jhu.edu/~ccb/howto-extract-paraphrases.html, as our baseline on paraphrase extraction. In this method, the authors improved the classical pivot language translation approach to extract phrasal paraphrases from bilingual parallel corpora [1] by using syntactic constraints.

After phrase extraction from the 140 original testing queries, we obtain a total number of 1100 phrases. For the baseline method, we empirically set a threshold of paraphrase probabilities, which are computed using the approach in [39], to filter the paraphrases of lower quality. Finally, we obtain 7752 paraphrases for the original phrases.

The experimental results are shown in Table 3 with the evaluation of correct meaning (CM), correct grammar (CG) and both correct (BC) in *accuracy*. Meanwhile, we also check the macro-average and micro-average of CM, CG and BC. Here, macro-average means that we compute the average score of the paraphrase generation results for all the phrases. And micro-average represents that we compute the paraphrase generation results for each phrase respectively, and then calculate the average score of these results. All of the results are statistically significant over the respective baseline at 0.95 confidence interval using the *t*-test. Here, the *t*-test works for testing the statistical significance on paraphrase generation result which contains a large number of 7752 paraphrases. As the *t*-test works for the non-normal data only if the sample size is large, the *t*-test used in our experimental data set is rational. %chg denotes the performance improvement in percent of our method over the corresponding baseline.

From Table 3, we can see that our method outperforms the baseline method, which is the state-of-the-art method on paraphrase extraction using bilingual parallel corpora. For the experimental results, we have the following analysis.

First, the effect of baseline method for paraphrase extraction is mainly depended on the quality of word alignment in corpora. Hence, bad quality of word alignments may cause worse results in paraphrase extraction, especially on fixing the boundary of phrases.

Second, although the baseline method used syntactic match as a constraint for paraphrase extraction, there are many phrases or fragments, which have different POS and tense with original

How much folic acid should an expectant mother get daily?	1.0
How much folate should an expectant mother get daily?	0.375
How much folic acid should a pregnant woman get daily?	0.161
How much folic acid should pregnant women get daily?	0.049
How much folate should pregnant women get daily?	0.018
How much folate should a mother get daily?	0.008

Figure 5. Top 5 reformulated questions with their generation probabilities.
doi:10.1371/journal.pone.0064601.g005

Table 3. Experimental results of phrasal paraphrase extraction both on percentage of correct meaning and grammar.

	macro-CM	micro-CM	macro-CG	micro-CG	macro-BC	micro-BC
Baseline	0.4724	0.5575	0.7825	0.8090	0.4617	0.5375
%chg	+73.05%	+65.33%	+7.13%	+18.89%	+76.87%	+68.02%
Our Method	0.8175	0.9217	0.8383	0.9618	0.8166	0.9031

doi:10.1371/journal.pone.0064601.t003

phrases, are extracted as paraphrases. It usually causes the grammar error on paraphrase extraction.

Different from the state-of-the-art method, the proposed method uses online translation engines to translate the original phrases into pivot language, which is Chinese, and then translates these pivot phrases back to the original language. The online translation engines work better on boundary control for phrases and contain the information from high quality semantic dictionaries, meanwhile, they automatically transfer phrases in different forms and tense into uniform expressions, therefore, the proposed method works better on paraphrase extraction.

One limitation is that the quantity of extracted paraphrases is less than the baseline method. For the 1100 phrases, we only obtain 4750 candidate paraphrases. However, the quality of paraphrases extracted by our method is higher than baseline. In future work, we plan to combine the two methods to get better performance on paraphrase extraction.

On Question Reformulation

As the previous description, this section we will check the effectiveness of our question reformulation method for question retrieval task.

For performance comparison, we introduce two methods as the baselines for question reformulation. The first one is synonym based question reformulation which is selected as the baseline-1 in our experiments. Here, we use WordNet as the lexical resource and distance based word similarities are used for synonym selection. The second one is the state-of-the-art method on sentence-level paraphrase generation, namely statistical paraphrase generation (SPG) [43]. Here, we consider that the generated question paraphrases can be seen as the reformulated questions. Therefore, we run SPG under the setting of baseline-2 as described in [43].

Then, we utilize the three kinds of reformulated questions as queries for the question retrieval model. Hence, we can get the performance comparisons among these methods. Here, we use the state-of-the-art question retrieval model which is proposed by Xue et al. [44], namely translation-based language model (TLM), as our basic question retrieval unit. Thus, we get three question retrieval systems, the first is WordNet-based question reformulation TLM (WN-TLM), the second is SPG-based question reformulation TLM (SPG-TLM) and the last is our proposed Viterbi decoding-based question reformulation TLM (VD-TLM).

As the reformulated questions can be seen as the rewriting of original question queries, we design the performance comparisons between question retrieval systems which only use the reformulated questions as queries and the systems that use both original and reformulated questions as queries. For the latter one, we need to combine and re-rank the question retrieval results. Here, we use the blending model which is proposed by Xu et al. [45], to complete the retrieval results combination and re-ranking. It considers the reformulated questions (rq) as the similar queries to the original questions (oq), and then we capture the four kinds of similarities which are the similarity between oq and rq, the similarity between oq and rq retrieval result, the similarity between rq and oq retrieval result, the last similarity between oq retrieval result and rq retrieval result. Finally, we use linear combination to joint these similarities for question retrieval results re-ranking.

For evaluation of question retrieval, we use precision at position 1 (p@1), mean average precision (MAP) and mean reciprocal rank (MRR). We pool the top 20 results from various methods, such as vector space model, okapi BM25 model, language model and our proposed methods, to establish our ground truth with manually checking. We asked two annotators serve their judgements on whether each of the retrieved question is similar or relevant (score

Table 4. Overall performance comparison of MRR, MAP and p1. All improvements obtained by VD-TLM are statistically significant over other methods within 0.95 confidence interval using the t-test.

Question Retrieval Models	TLM	WN-TLM	SPG-TLM	VD-TLM	WN-TLM	SPG-TLM	VD-TLM
	(oq)	(rq)	(rq)	(rq)	(oq+rq)	(oq+rq)	(oq+rq)
MRR	0.1889	0.1875	0.2024	0.2157	0.2206	0.2301	0.2583
% MRR improvement over TLM	N/A	N/A	+7.15	+14.19	+16.78	+21.81	+36.74
WN-TLM (*rq*)	+0.75	N/A	+7.95	+15.04	+17.65	+22.72	+37.76
SPG-TLM (*rq*)	N/A	N/A	N/A	+6.57	+8.99	+13.69	+27.62
WN-TLM (*oq+rq*)	N/A	N/A	N/A	N/A	N/A	+4.31	+17.09
SPG-TLM (*oq+rq*)	N/A	N/A	N/A	N/A	N/A	N/A	+12.26
MAP	0.2889	0.2870	0.3037	0.3269	0.3384	0.3664	0.4188
p@1	0.1928	0.1967	0.2214	0.2357	0.2429	0.2643	0.2786

doi:10.1371/journal.pone.0064601.t004

1) with the original question or not (score 0). When conflicts occur, a third annotator was involved in making the final decision.

The parameters in TLM are well tuned on the development set using grid search. The experimental results on question retrieval are shown in Table 4.

From Table 4, we draw the following observations:

First, comparing to the state-of-the-art question retrieval model, TLM, all of the reformulation-based methods outperform TLM except WN-TLM on MRR, which indicates that question reformulation is necessary and effective for question retrieval. The reason may be that word mismatch problem is widely existing between the queries and the candidate questions. Hence, effective question reformulation methods lead to better retrieval performance as they can bridge the lexical gap. The performance of WN-TLM on MRR is a bit lower than TLM, which indicates that the synonym expanded by WordNet is not always available to real world data, some expanded words may cause chaos in meaning.

Second, both the experimental results on rq and oq+rq indicate that our proposed VD-based question reformulation methods outperform the WN-based and SPG-based methods under the results of question retrieval. It proves that VD-based question reformulation methods are more effective on question reformulation. For the WN-based question reformulation methods, they only capture word level synonyms and overlook the importance of context information. The SPG-based question reformulation methods only focus on generating high quality question paraphrases and overlook the statistical distribution of the phrases in the whole corpus. Hence the generated paraphrases may not well adapt to the archived question retrieval.

Third, the oq+rq-based methods correspondingly outperform the rq-based methods which indicates that the retrieval results which only use rq as queries could lead improvements over TLM method. However, oq cannot be overlooked, as the strict word match is also important for question retrieval. It means that question retrieval results, which use oq as queries, can be enhanced by combining rq results, through the using of the blending model for re-ranking.

Performance Variation to the Number of Reformulated Questions

In this section, we will check how the number of reformulated questions adding to the question retrieval model, influences the performance of the question retrieval results. Hence, we generate the top 5 question reformulation results to observe the performance variation. Here, we use the VD-TLM model for the question retrieval task. Each time, we add the retrieval results of one reformulated question into the blending model and obtain the results in p@1, MAP and MRR. Finally, we draw the performance variation in Figure 6.

From Figure 6, we can see that the best performance can be achieved when the number of reformulated questions equals to 1. And then the performance decreases with the number growing. This indicates that better performance can be achieved by only choosing 1 reformulated question for the retrieval results blending. This is because more reformulated questions may introduce more noise for question retrieval. Meanwhile, the query intent may be shifted by adopting more than 1 reformulated questions. Hence, in the question retrieval systems, more reformulation queries may lead to more noise.

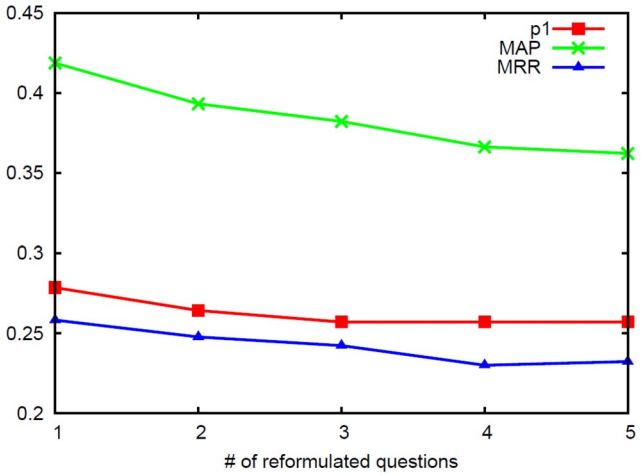

Figure 6. Illustration of performance variation when different number of reformulated questions are added in blending model for question retrieval.
doi:10.1371/journal.pone.0064601.g006

Conclusion and Future Work

In this paper, we presented a novel question reformulation method for archived question retrieval. Given a question, we first automatically extracted key-phrases and employed three online translation engines to obtain their translations. Further we utilized the pivot language approach to extracting phrasal paraphrases. Finally, the Viterbi decoding method was involved to generate question reformulations. The experimental results indicated that both the proposed phrasal paraphrase extraction and the question reformulation-based question retrieval methods outperformed the state-of-the-art methods significantly. It demonstrated the effectiveness of our question reformulation method.

Inspired by [46–48], in the future, we expect to combine multiple methods on paraphrase extraction in order to optimize the estimation of paraphrase probability and get better results. We plan to combine the monolingual and bilingual based paraphrase extraction methods by integrating the intermediate results of the above two methods. And then we can estimate the paraphrase probabilities by linearly combining the two generated paraphrase probabilities with different integration weights. Finally, we can obtain a new paraphrase ranking list through the jointly estimating of the candidate paraphrases generated by the above two methods.

Author Contributions

Conceived and designed the experiments: YZ TL. Performed the experiments: WZ FW. Analyzed the data: KL RJ. Contributed reagents/materials/analysis tools: WZ. Wrote the paper: WZ FW.

References

1. Bannard C, Callison-Burch C (2005) Paraphrasing with bilingual parallel corpora. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, , USA: Association for Computational Linguistics, ACL '05, pp. 597–604. doi:10.3115/1219840.1219914. URL http://dx.doi.org/10.3115/1219840.1219914.
2. Hovy E, Hermjakob U, Lin CY (2001) The use of external knowledge in factoid QA. In: Proceedings of the Tenth Text REtrieval Conference. TREC '01. URL http://trec.nist.gov/pubs/trec10/papers/TREC10-webclopedia.pdf.
3. Buscaldi D, Rosso P, Arnal ES (2005) A wordnet-based query expansion method for geographical information retrieval. In: Working notes for the Cross-Language Evaluation Forum (CLEF) workshop. Citeseer, CLEF '05. URL http://clef.isti.cnr.it/2005/working _notes/workingnotes2005/buscaldi05.pdf.

4. Riezler S, Vasserman A, Tsochantaridis I, Mittal V, Liu Y (2007) Statistical machine translation for query expansion in answer retrieval. In: Proceedings of the 45rd Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA, , USA: Association for Computational Linguistics, volume 45 of *ACL '07*, pp. 464–471. URL http://acl.ldc.upenn.edu/P/P07/P07-1059.pdf.
5. Zhao S, Zhou M, Liu T (2007) Learning question paraphrases for QA from Encarta logs. In: Proceedings of the 20th international joint conference on Artifical intelligence. San Francisco, CA, , USA: Morgan Kaufmann Publishers Inc., IJCAI'07, pp. 1795–1800. URL http://dl.acm.org/citation.cfm?id = 1625275.1625566.
6. Duclaye F, Yvon F, Collin O (2003) Learning paraphrases to improve a question-answering system. In: Proceedings of the European Chapter of Association for Computational Linguistics (EACL) Workshop on Natural Language Processing for Question Answering Systems. Stroudsburg, PA, , USA: Association for Computational Linguistics, EACL '03, pp. 35–41. URL http://acl.ldc.upenn.edu/eacl2003/papers/workshop/w11.pdf#page = 44.
7. Gao Y, Wang M, Luan H, Shen J, Yan S, et al. (2011) Tag-based social image search with visualtext joint hypergraph learning. In: Proceedings of the 19th ACM international conference on Multimedia. ACM, pp. 1517–1520.
8. Gao Y, Wang M, Zha Z, Shen J, Li X, et al. (2013) Visual-textual joint relevance learning for tag-based social image search. IEEE Transactions on Image Processing 22: 363–376.
9. Chakrabarti D, Punera K (2011) Event summarization using tweets. In: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media. pp. 66–73.
10. Sang J, Xu C (2011) Browse by chunks: Topic mining and organizing on web-scale social media. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 7: 30.
11. Gao Y, Tang J, Hong R, Dai Q, Chua TS, et al. (2010) W2go: a travel guidance system by automatic landmark ranking. In: Proceedings of the international conference on Multimedia. ACM, pp. 123–132.
12. Sang J, Xu C (2012) Right buddy makes the difference: An early exploration of social relation analysis in multimedia applications. In: Proceedings of the 20th ACM international conference on Multimedia. ACM, pp. 19–28.
13. Wang K, Ming Z, Chua TS (2009) A syntactic tree matching approach to finding similar questions in community-based QA services. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '09, pp. 187–194. doi:10.1145/1571941.1571975. URL http://doi.acm.org/10.1145/1571941.1571975.
14. Wang K, Chua TS (2010) Exploiting salient patterns for question detection and question retrieval in community-based question answering. In: Proceedings of the 23rd International Conference on Computational Linguistics. Stroudsburg, PA, , USA: Association for Computational Linguistics, COLING '10, pp. 1155–1163. URL http://dl.acm.org/citation.cfm?id = 1873781.1873911.
15. Bendersky M, Croft WB (2008) Discovering key concepts in verbose queries. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '08, pp. 491–498. doi:10.1145/1390334.1390419. URL http://doi.acm.org/10.1145/1390334.1390419.
16. Bendersky M, Metzler D, Croft WB (2010) Learning concept importance using a weighted dependence model. In: Proceedings of the third ACM international conference on Web search and data mining. New York, NY, , USA: ACM, WSDM '10, pp. 31–40. doi:10.1145/1718487.1718492. URL http://doi.acm.org/10.1145/1718487.1718492.
17. Bendersky M, Metzler D, Croft WB (2011) Parameterized concept weighting in verbose queries. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. New York, NY, , USA: ACM, SIGIR '11, pp. 605–614. doi: 10.1145/2009916.2009998. URL http://doi.acm.org/10.1145/2009916.2009998.
18. Bendersky M, Croft WB (2012) Modeling higher-order term dependencies in information retrieval using query hypergraphs. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '12, pp. 941–950. doi:10.1145/2348283.2348408. URL http://doi.acm.org/10.1145/2348283.2348408.
19. Song F, Croft WB (1999) A general language model for information retrieval (poster abstract). In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '99, pp. 279–280. doi: 10.1145/312624.312698. URL http://doi.acm.org/10.1145/312624.312698.
20. Srikanth M, Srihari R (2002) Biterm language models for document retrieval. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '02, pp. 425–426. doi:10.1145/564376.564476. URL http://doi.acm.org/10.1145/564376.564476.
21. Gao J, Nie JY, Wu G, Cao G (2004) Dependence language model for information retrieval. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '04, pp. 170–177. doi: 10.1145/1008992.1009024. URL http://doi.acm.org/10.1145/1008992.1009024.
22. Cui H, Sun R, Li K, Kan MY, Chua TS (2005) Question answering passage retrieval using dependency relations. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '05, pp. 400–407. doi:10.1145/1076034.1076103. URL http://doi.acm.org/10.1145/1076034.1076103.
23. Harabagiu S, Moldovan D, Pasca M, Surdeanu M, Mihalcea R, et al. (2001) Answering complex, list and context questions with LCC's Question-Answering Server. In: Proceedings of the TExt Retrieval Conference for Question Answering. TREC '01. URL http://trec.nist.gov/pubs/trec10/papers/lcc-trec10.pdf.
24. Harman D (1988) Towards interactive query expansion. In: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '88, pp. 321–331. doi:10.1145/62437.62469. URL http://doi.acm.org/10.1145/62437.62469.
25. Qiu Y, Frei HP (1993) Concept based query expansion. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '93, pp. 160–169. doi:10.1145/160688.160713. URL http://doi.acm.org/10.1145/160688.160713.
26. Robertson SE, Walker S, Jones S, Hancock-Beaulieu MM, Gatford M, et al. (1995) Okapi at TREC-3. In: NIST Special Publication SP. National Institute of Standards & Technology, TREC '95, pp. 109–127. URL http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.
27. Ponte JM, Croft WB (1998) A language modeling approach to information retrieval. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '98, pp. 275–281. doi:10.1145/290941.291008. URL http://doi.acm.org/10.1145/290941.291008.
28. De Marneffe MC, MacCartney B, Manning CD (2006) Generating typed dependency parses from phrase structure parses. In: Proceedings of the International Conference on Language Resources and Evaluation. volume 6 of *LREC '06*, pp. 449–454. URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/440 pdf.pdf.
29. Xu J, Croft WB (1996) Query expansion using local and global document analysis. In: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '96, pp. 4–11. doi:10.1145/243199.243202. URL http://doi.acm.org/10.1145/243199.243202.
30. Callan JP, Croft WB, Broglio J (1995) TREC and TIPSTER experiments with INQUERY. In: Proceedings of the second conference on Text REtrieval Conference. Elmsford, NY, , USA: Pergamon Press, Inc., TREC-2, pp. 327–343. URL http://dl.acm.org/citation.cfm?id = 208211.204457.
31. Robertson SE, Walker S (1999) Okapi/keenbow at TREC-8. In: Proceedings of the Eighth Text REtrieval Conference. volume 8 of T*REC'99*. URL http://trec.nist.gov/pubs/trec8/papers/okapi.pdf.
32. Robertson S (2002) Threshold setting and performance optimization in adaptive filtering. In: Information Retrieval. Hingham, MA, , USA: Kluwer Academic Publishers, volume 5, pp. 239–256. doi:10.1023/A:1015702129514. URL http://dx.doi.org/10.1023/A:1015702129514.
33. Lam-Adesina AM, Jones GJF (2001) Applying summarization techniques for term selection in relevance feedback. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '01, pp. 1–9. doi:10.1145/383952.383953. URL http://doi.acm.org/10.1145/383952.383953.
34. Dinh D, Tamine L (2011) Combining global and local semantic contexts for improving biomedical information retrieval. In: Proceedings of the 33rd European conference on Advances in information retrieval. Berlin, Heidelberg: : Springer-Verlag, ECIR'11, pp. 375–386. URL http://dl.acm.org/citation.cfm?id = 1996889.1996939.
35. Grineva M, Grinev M, Lizorkin D (2009) Extracting key terms from noisy and multitheme documents. In: Proceedings of the 18th international conference on World Wide Web. New York, NY, , USA: ACM, WWW '09, pp. 661–670. doi:10.1145/1526709.1526798. URL http://doi.acm.org/10.1145/1526709.1526798.
36. Mihalcea R, Tarau P (2004) Textrank: Bringing order into texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, , USA: Association for Computational Linguistics, volume 4 of *EMNLP'04*, pp. 404–411. URL http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mihalcea.pdf.
37. Regneri M, Wang R (2012) Using discourse information for paraphrase extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Stroudsburg, PA, , USA: Association for Computational Linguistics, EMNLP-CoNLL '12, pp. 916–927. URL http://dl.acm.org/citation.cfm?id = 2390948.2391048.
38. Barzilay R, Lee L (2003) Learning to paraphrase: an unsupervised approach using multiplesequence alignment. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. Stroudsburg, PA, , USA: Association for Computational Linguistics, NAACL '03, pp. 16–23. doi: 10.3115/1073445.1073448. URL http://dx.doi.org/10.3115/1073445.1073448.

39. Callison-Burch C (2008) Syntactic constraints on paraphrases extracted from parallel corpora. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, , USA: Association for Computational Linguistics, EMNLP '08, pp. 196–205. URL http://dl.acm.org/citation.cfm?id = 1613715.1613743.
40. Zhao S, Niu C, Zhou M, Liu T, Li S (2008) Combining multiple resources to improve SMT-based paraphrasing model. In: Proceedings of the 46rd Annual Meeting on Association for Computational Linguistics: Human Language Technique. Stroudsburg, PA, , USA: Association for Computational Linguistics, ACL '09 : HLT, pp. 1021–1029. URL https://www.aclweb.org/anthology-new/P/P08/P08-1116.pdf.
41. Zhao S, Wang H, Lan X, Liu T (2010) Leveraging multiple MT engines for paraphrase generation. In: Proceedings of the 23rd International Conference on Computational Linguistics. Stroudsburg, PA, , USA: Association for Computational Linguistics, COLING '10, pp. 1326–1334. URL http://dl.acm.org/citation.cfm?id = 1873781.1873930.
42. Viterbi A (2006) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. In: IEEE Transactions on Information Theory. Piscataway, NJ, , USA: IEEE Press, volume 13, pp. 260–269. doi:10.1109/TIT.1967.1054010. URL http://dx.doi.org/10.1109/TIT.1967.1054010.
43. Zhao S, Lan X, Liu T, Li S (2009) Application-driven statistical paraphrase generation. 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 2 - Volume 2. Stroudsburg, PA, , USA: Association for Computational Linguistics, ACL '09, pp. 834–842. URL http://dl.acm.org/citation.cfm?id = 1690219.1690263.
44. Xue X, Jeon J, Croft WB (2008) Retrieval models for question and answer archives. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, , USA: ACM, SIGIR '08, pp. 475–482. doi:10.1145/1390334.1390416. URL http://doi.acm.org/10.1145/1390334.1390416.
45. Xu J, Wu W, Li H, Xu G (2011) A kernel approach to addressing term mismatch. In: Proceedings of the 20th international conference companion on World Wide Web. New York, NY, , USA: ACM, WWW '11, pp. 153–154. doi:10.1145/1963192.1963270. URL http://doi.acm.org/10.1145/1963192.1963270.
46. Zhang S, Huang J, Li H, Metaxas DN (2012) Automatic image annotation and retrieval using group sparsity. In: IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics. IEEE, volume 42, pp. 838–849. URL http://ieeexplore.ieee.org/xpl/login.jsp?tp = &arnumber = 6127919.
47. Gehler P, Nowozin S (2009) On feature combination for multiclass object classification. In: 2009 IEEE 12th International Conference on Computer Vision. IEEE, ICCV '09, pp. 221–228. URL http://ieeexplore.ieee.org/xpl/login.jsp?tp = &arnumber = 5459169.
48. Zhang S, Yang M, Cour T, Yu K, Metaxas DN (2012) Query specific fusion for image retrieval. In: European Conference on Computer Vision. ECCV, ECCV '12, pp. 660–673. URL http://www.research.rutgers.edu/shaoting/paper/ECCV12-retrieval.pdf.