Automatically Mining Question Reformulation Patterns from Search Log Data

Xiaobing Xue*
Univ. of Massachusetts, Amherst
xuexb@cs.umass.edu

Yu Tao*
Univ. of Science and Technology of China
v-yutao@microsoft.com

Daxin Jiang **Hang Li**
Microsoft Research Asia
{djiang,hangli}@microsoft.com

Abstract

Natural language questions have become popular in web search. However, various questions can be formulated to convey the same information need, which poses a great challenge to search systems. In this paper, we automatically mined *5w1h question reformulation patterns* from large scale search log data. The question reformulations generated from these patterns are further incorporated into the retrieval model. Experiments show that using question reformulation patterns can significantly improve the search performance of natural language questions.

1 Introduction

More and more web users tend to use natural language questions as queries for web search. Some commercial natural language search engines such as InQuira and Ask have also been developed to answer this type of queries. One major challenge is that various questions can be formulated for the same information need. Table 1 shows some alternative expressions for the question "how far is it from Boston to Seattle". It is difficult for search systems to achieve satisfactory retrieval performance without considering these alternative expressions.

In this paper, we propose a method of automatically mining *5w1h question*[1] *reformulation patterns* to improve the search relevance of 5w1h questions. *Question reformulations* represent the alternative expressions for 5w1h questions. *A question reformulation pattern* generalizes a set of similar question reformulations that share the same structure. For example, users may ask similar questions "how far is it from X_1 to X_2" where X_1 and X_2 represent some other cities besides Boston and Seattle. Then, similar question reformulations as in Table 1 will be generated with the city names changed. These patterns increase the coverage of the system by handling the queries that did not appear before but share similar structures as previous queries.

Table 1: Alternative expressions for the original question

Original Question:
how far is it from Boston to Seattle
Alternative Expressions:
how many miles is it from Boston to Seattle
distance from Boston to Seattle
Boston to Seattle
how long does it take to drive from Boston to Seattle

Using reformulation patterns as the key concept, we propose a question reformulation framework. First, we mine the question reformulation patterns from search logs that record users' reformulation behavior. Second, given a new question, we use the most relevant reformulation patterns to generate question reformulations and each of the reformulations is associated with its probability. Third, the original question and these question reformulations are then combined together for retrieval.

The contributions of this paper are summarized as two folds. First, we propose a simple yet effective approach to automatically mine 5w1h question reformulation patterns. Second, we conduct comprehensive studies in improving the search performance of 5w1h questions using the mined patterns.

*Contribution during internship at Microsoft Research Asia

[1]5w1h questions start with "Who", "What", "Where", "When", "Why" and "How".

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 187–192,
Jeju, Republic of Korea, 8-14 July 2012.

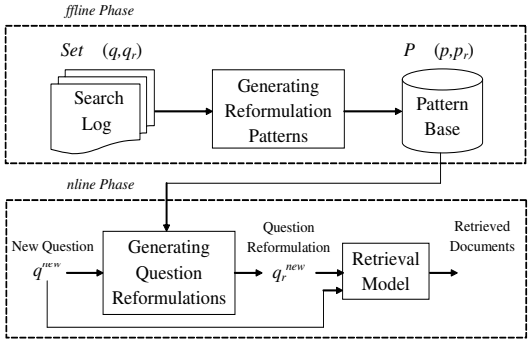

Figure 1: The framework of reformulating questions.

2 Related Work

In the Natural Language Processing (NLP) area, different expressions that convey the same meaning are referred as *paraphrases* (Lin and Pantel, 2001; Barzilay and McKeown, 2001; Pang et al., 2003; Paşca and Dienes, 2005; Bannard and Callison-Burch, 2005; Bhagat and Ravichandran, 2008; Callison-Burch, 2008; Zhao et al., 2008). Paraphrases have been studied in a variety of NLP applications such as machine translation (Kauchak and Barzilay, 2006; Callison-Burch et al., 2006), question answering (Ravichandran and Hovy, 2002) and document summarization (McKeown et al., 2002). Yet, little research has considered improving web search performance using paraphrases.

Query logs have become an important resource for many NLP applications such as class and attribute extraction (Paşca and Van Durme, 2008), paraphrasing (Zhao et al., 2010) and language modeling (Huang et al., 2010). Little research has been conducted to automatically mine 5w1h question reformulation patterns from query logs.

Recently, query reformulation (Boldi et al., 2009; Jansen et al., 2009) has been studied in web search. Different techniques have been developed for query segmentation (Bergsma and Wang, 2007; Tan and Peng, 2008) and query substitution (Jones et al., 2006; Wang and Zhai, 2008). Yet, most previous research focused on keyword queries without considering 5w1h questions.

3 Mining Question Reformulation Patterns for Web Search

Our framework consists of three major components, which is illustrated in Fig. 1.

Table 2: Question reformulation patterns generated for the query pair ("how far is it from Boston to Seattle" ,"distance from Boston to Seattle").

S_1 = {Boston}:("how far is it from X_1 to Seattle" ,"distance from X_1 to Seattle")
S_2 = {Seattle}:("how far is it from Boston to X_1" ,"distance from Boston to X_1")
S_3 = {Boston, Seattle}:("how far is it from X_1 to X_2" ,"distance from X_1 to X_2")

3.1 Generating Reformulation Patterns

From the search log, we extract all successive query pairs issued by the same user within a certain time period where the first query is a 5w1h question. In such query pair, the second query is considered as a question reformulation. Our method takes these query pairs, i.e. $Set = \{(q, q_r)\}$, as the input and outputs a pattern base consisting of 5w1h question reformulation patterns, i.e. $P = \{(p, p_r)\}$). Specifically, for each query pair (q, q_r), we first collect all common words between q and q_r except for stopwords ST[2], where $CW = \{w | w \in q, w \in q', w \notin ST\}$. For any non-empty subset S_i of CW, the words in S_i are replaced as slots in q and q_r to construct a reformulation pattern. Table 2 shows examples of question reformulation patterns. Finally, the patterns observed in many different query pairs are kept. In other words, we rely on the frequency of a pattern to filter noisy patterns. Generating patterns using more NLP features such as the parsing information will be studied in the future work.

3.2 Generating Question Reformulations

We describe how to generate a set of question reformulations $\{q_r^{new}\}$ for an unseen question q^{new}.

First, we search $P = \{(p, p_r)\}$ to find all question reformulation patterns where p matches q^{new}. Then, we pick the best question pattern $p^\star$ according to the number of prefix words and the total number of words in a pattern. We select the pattern that has the most prefix words, since this pattern is more likely to have the same information as q^{new}. If several patterns have the same number of prefix words, we use the total number of words to break the tie.

After picking the best question pattern $p^\star$, we further rank all question reformulation patterns containing $p^\star$, i.e. $(p^\star, p_r)$, according to Eq. 1.

[2]Stopwords refer to the function words that have little meaning by themselves, such as "the", "a", "an", "that" and "those".

Table 3: Examples of the question reformulations and their corresponding reformulation patterns

q^{new}: how good is the eden pure air system $p^\star$: how good is the X		q^{new}: how to market a restaurant $p^\star$: how to market a X	
q_r^{new}	p_r	q_r^{new}	p_r
eden pure air system	X	marketing a restaurant	marketing a X
eden pure air system review	X review	how to promote a restaurant	how to promote a X
eden pure air system reviews	X reviews	how to sell a restaurant	how to sell a X
rate the eden pure air system	rate the X	how to advertise a restaurant	how to advertise a X
reviews on the eden pure air system	reviews on the X	restaurant marketing	X marketing

$$P(p_r|p^\star) = \frac{f(p^\star, p_r)}{\sum_{p'_r} f(p^\star, p'_r)} \quad (1)$$

Finally, we generate k question reformulations q_r^{new} by applying the top k question reformulation patterns containing $p^\star$. The probability $P(p_r|p^\star)$ associated with the pattern $(p^\star, p_r)$ is assigned to the corresponding question reformulation q_r^{new}.

3.3 Retrieval Model

Given the original question q^{new} and k question reformulations $\{q_r^{new}\}$, the query distribution model (Xue and Croft, 2010) (denoted as QDist) is adopted to combine q^{new} and $\{q_r^{new}\}$ using their associated probabilities. The retrieval score of the document D, i.e. $score(q^{new}, D)$, is calculated as follows:

$$score(q^{new}, D) = \lambda \log P(q^{new}|D) + (1-\lambda) \sum_{i=1}^{k} P(p_{r_i}|p^\star) \log P(q_{r_i}^{new}|D) \quad (2)$$

In Eq. 2, λ is a parameter that indicates the probability assigned to the original query. $P(p_{r_i}|p^\star)$ is the probability assigned to $q_{r_i}^{new}$. $P(q_{new}|D)$ and $P(q'|D)$ are calculated using the language model (Ponte and Croft, 1998; Zhai and Lafferty, 2001).

4 Experiments

A large scale search log from a commercial search engine (2011.1-2011.6) is used in experiments. From the search log, we extract all successive query pairs issued by the same user within 30 minutes (Boldi et al., 2008)[3] where the first query is a 5w1h question. Finally, we extracted 6,680,278 question reformulation patterns.

For the retrieval experiments, we randomly sample 10,000 natural language questions as queries from the search log before 2011. For each question, we generate the top ten questions reformulations. The Indri toolkit[4] is used to implement the language model. A web collection from a commercial search engine is used for retrieval experiments. For each question, the relevance judgments are provided by human annotators. The standard NDCG@k is used to measure performance.

Table 4: Retrieval Performance of using question reformulations. * denotes significantly different with Orig.

	NDCG@1	NDCG@3	NDCG@5
Orig	0.2946	0.2923	0.2991
QDist	0.3032*	0.2991*	0.3067*

4.1 Examples and Performance

Table 3 shows examples of the generated questions reformulations. Several interesting expressions are generated to reformulate the original question.

We compare the retrieval performance of using the question reformulations (QDist) with the performance of using the original question (Orig) in Table 4. The parameter λ of QDist is decided using ten-fold cross validation. Two sided t-test are conducted to measure significance.

Table 4 shows that using the question reformulations can significantly improve the retrieval performance of natural language questions. Note that, considering the scale of experiments (10,000 queries), around 3% improvement with respect to NDCG is a very interesting result for web search.

4.2 Analysis

In this subsection, we analyze the results to better understand the effect of question reformulations.

First, we report the performance of always picking the best question reformulation for each query (denoted as Upper) in Table 5, which provides an

[3] In web search, queries issued within 30 minutes are usually considered having the same information need.

[4] www.lemurproject.org/

Table 5: Performance of the upper bound.

	NDCG@1	NDCG@3	NDCG@5
Orig	0.2946	0.2923	0.2991
QDist	0.3032	0.2991	0.3067
Upper	0.3826	0.3588	0.3584

Table 6: Best reformulation within different positions.

top 1	within top 2	within top 3
49.2%	64.7%	75.4%

upper bound for the performance of the question reformulation. Table 5 shows that if we were able to always picking the best question reformulation, the performance of Orig could be improved by around 30% (from 0.2926 to 0.3826 with respect to NDCG@1). It indicates that we do generate some high quality question reformulations.

Table 6 further reports the percent of those 10,000 queries where the best question reformulation can be observed in the top 1 position, within the top 2 positions and within the top 3 positions, respectively.

Table 6 shows that for most queries, our method successfully ranks the best reformulation within the top 3 positions.

Second, we study the effect of different types of question reformulations. We roughly divide the question reformulations generated by our method into five categories as shown in Table 7. For each category, we report the percent of reformulations which performance is bigger/smaller/equal with respect to the original question.

Table 7 shows that the "more specific" reformulations and the "equivalent" reformulations are more likely to improve the original question. Reformulations that make "morphological change" do not have much effect on improving the original question. "More general" and "not relevant" reformulations usually decrease the performance.

Third, we conduct the error analysis on the question reformulations that decrease the performance of the original question. Three typical types of errors are observed. First, some important words are removed from the original question. For example, "what is the role of corporate executives" is reformulated as "corporate executives". Second, the reformulation is too specific. For example, "how to effectively organize your classroom" is reformulated as "how to effectively organize your elementary classroom". Third, some reformulations entirely change the meaning of the original question. For example, "what is the adjective of anxiously" is reformulated as "what is the noun of anxiously".

Table 7: Analysis of different types of reformulations.

Type	increase	decrease	same
Morphological change	11%	10%	79%
Equivalent meaning	32%	30%	38%
More specific/Add words	45%	39%	16%
More general/Remove words	38%	48%	14%
Not relevant	14%	72%	14%

Table 8: Retrieval Performance of other query processing techniques.

	NDCG@1	NDCG@3	NDCG@5
ORIG	0.2720	0.2937	0.3151
NoStop	0.2697	0.2893	0.3112
DropOne	0.2630	0.2888	0.3102
QDist	0.2978	0.3052	0.3250

Fourth, we compare our question reformulation method with two long query processing techniques, i.e. NoStop (Huston and Croft, 2010) and DropOne (Balasubramanian et al., 2010). NoStop removes all stopwords in the query and DropOne learns to drop a single word from the query. The same query set as Balasubramanian et al. (2010) is used. Table 8 reports the retrieval performance of different methods.

Table 8 shows that both NoStop and DropOne perform worse than using the original question, which indicates that the general techniques developed for long queries are not appropriate for natural language questions. On the other hand, our proposed method outperforms all the baselines.

5 Conclusion

Improving the search relevance of natural language questions poses a great challenge for search systems. We propose to automatically mine 5w1h question reformulation patterns from search log data. The effectiveness of the extracted patterns has been shown on web search. These patterns are potentially useful for many other applications, which will be studied in the future work. How to automatically classify the extracted patterns is also an interesting future issue.

Acknowledgments

We would like to thank W. Bruce Croft for his suggestions and discussions.

References

N. Balasubramanian, G. Kumaran, and V.R. Carvalho. 2010. Exploring reductions for long web queries. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 571–578. ACM.

C. Bannard and C. Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604. Association for Computational Linguistics.

R. Barzilay and K.R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 50–57. Association for Computational Linguistics.

S. Bergsma and Q. I. Wang. 2007. Learning noun phrase query segmentation. In *EMNLP-CoNLL07*, pages 819–826, Prague.

R. Bhagat and D. Ravichandran. 2008. Large scale acquisition of paraphrases for learning surface patterns. *Proceedings of ACL-08: HLT*, pages 674–682.

Paolo Boldi, Francesco Bonchi, Carlos Castillo, Debora Donato, Aristides Gionis, and Sebastiano Vigna. 2008. The query-flow graph: model and applications. In *CIKM08*, pages 609–618.

P. Boldi, F. Bonchi, C. Castillo, and S. Vigna. 2009. From "Dango" to "Japanese Cakes": Query reformulation models and patterns. In *Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 183–190. IEEE.

C. Callison-Burch, P. Koehn, and M. Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24. Association for Computational Linguistics.

C. Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 196–205. Association for Computational Linguistics.

Jian Huang, Jianfeng Gao, Jiangbo Miao, Xiaolong Li, Kuansan Wang, Fritz Behr, and C. Lee Giles. 2010. Exploring web scale language models for search query processing. In *WWW10*, pages 451–460, New York, NY, USA. ACM.

S. Huston and W.B. Croft. 2010. Evaluating verbose query processing techniques. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298. ACM.

B.J. Jansen, D.L. Booth, and A. Spink. 2009. Patterns of query reformulation during web searching. *Journal of the American Society for Information Science and Technology*, 60(7):1358–1371.

R. Jones, B. Rey, O. Madani, and W. Greiner. 2006. Generating query substitutions. In *WWW06*, pages 387–396, Ediburgh, Scotland.

D. Kauchak and R. Barzilay. 2006. Paraphrasing for automatic evaluation.

D.-K. Lin and P. Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Processing*, 7(4):343–360.

K.R. McKeown, R. Barzilay, D. Evans, V. Hatzivassiloglou, J.L. Klavans, A. Nenkova, C. Sable, B. Schiffman, and S. Sigelman. 2002. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the second international conference on Human Language Technology Research*, pages 280–285. Morgan Kaufmann Publishers Inc.

B. Pang, K. Knight, and D. Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 102–109. Association for Computational Linguistics.

M. Paşca and P. Dienes. 2005. Aligning needles in a haystack: Paraphrase acquisition across the web. *Natural Language Processing–IJCNLP 2005*, pages 119–130.

M. Paşca and B. Van Durme. 2008. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 19–27.

J. M. Ponte and W. B. Croft. 1998. A language modeling approach to information retrieval. In *SIGIR98*, pages 275–281, Melbourne, Australia.

D. Ravichandran and E. Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL02*, pages 41–47.

B. Tan and F. Peng. 2008. Unsupervised query segmentation using generative language models and Wikipedia. In *WWW08*, pages 347–356, Beijing,China.

X. Wang and C. Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In *CIKM08*, pages 479–488, Napa Valley, CA.

X. Xue and W. B. Croft. 2010. Representing queries as distributions. In *SIGIR10 Workshop on Query Rep-*

resentation and Understanding, pages 9–12, Geneva, Switzerland.

C. Zhai and J. Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR01*, pages 334–342, New Orleans, LA.

S. Zhao, H. Wang, T. Liu, and S. Li. 2008. Pivot approach for extracting paraphrase patterns from bilingual corpora. *Proceedings of ACL-08: HLT*, pages 780–788.

S. Zhao, H. Wang, and T. Liu. 2010. Paraphrasing with search engine query logs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1317–1325. Association for Computational Linguistics.