Paraphrasing Adaptation for Web Search Ranking

Chenguang Wang[*]
School of EECS
Peking University
wangchenguang@pku.edu.cn

Nan Duan
Microsoft Research Asia
nanduan@microsoft.com

Ming Zhou
Microsoft Research Asia
mingzhou@microsoft.com

Ming Zhang
School of EECS
Peking University
mzhang@net.pku.edu.cn

Abstract

Mismatch between queries and documents is a key issue for the web search task. In order to narrow down such mismatch, in this paper, we present an in-depth investigation on adapting a paraphrasing technique to web search from three aspects: a search-oriented paraphrasing model; an NDCG-based parameter optimization algorithm; an enhanced ranking model leveraging augmented features computed on paraphrases of original queries. Experiments performed on the large scale query-document data set show that, the search performance can be significantly improved, with $+3.28\%$ and $+1.14\%$ NDCG gains on dev and test sets respectively.

1 Introduction

Paraphrasing is an NLP technique that generates alternative expressions to convey the same meaning of the input text in different ways. Researchers have made great efforts to improve paraphrasing from different perspectives, such as paraphrase extraction (Zhao et al., 2007), paraphrase generation (Quirk et al., 2004), model optimization (Zhao et al., 2009) and etc. But as far as we know, none of previous work has explored the impact of using a well designed paraphrasing engine for web search ranking task specifically.

In web search, mismatches between queries and their relevant documents are usually caused by expressing the same meaning in different natural language ways. E.g., *X is the author of Y* and *Y was written by X* have identical meaning in most cases, but they are quite different in literal sense. The capability of paraphrasing is just right to alleviate such issues. Motivated by this, this paper presents an in-depth study on adapting paraphrasing to web search. First, we propose a search-oriented paraphrasing model, which includes specifically designed features for web queries that can enable a paraphrasing engine to learn preferences on different paraphrasing strategies. Second, we optimize the parameters of the paraphrasing model according to the Normalized Discounted Cumulative Gain (NDCG) score, by leveraging the minimum error rate training (MERT) algorithm (Och, 2003). Third, we propose an enhanced ranking model by using augmented features computed on paraphrases of original queries.

Many query reformulation approaches have been proposed to tackle the query-document mismatch issue, which can be generally summarized as query expansion and query substitution. Query expansion (Baeza-Yates, 1992; Jing and Croft, 1994; Lavrenko and Croft, 2001; Cui et al., 2002; Yu et al., 2003; Zhang and Yu, 2006; Craswell and Szummer, 2007; Elsas et al., 2008; Xu et al., 2009) adds new terms extracted from different sources to the original query directly; while query substitution (Brill and Moore, 2000; Jones et al., 2006; Guo et al., 2008; Wang and Zhai, 2008; Dang and Croft, 2010) uses probabilistic models, such as graphical models, to predict the sequence of rewritten query words to form a new query. Comparing to these works, our paraphrasing engine alters queries in a similar way to statistical machine translation, with systematic tuning and decoding components. Zhao et al. (2009) proposes an unified paraphrasing framework that can be adapted to different applications using different usability models. Our work can be seen as an extension along this line of research, by carrying out in-depth study on adapting paraphrasing to web search.

Experiments performed on the large scale data set show that, by leveraging additional matching features computed on query paraphrases, significant NDCG gains can be achieved on both dev

[*] This work has been done while the author was visiting Microsoft Research Asia.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 41–46,
Sofia, Bulgaria, August 4-9 2013.

($+3.28\%$) and test ($+1.14\%$) sets.

2 Paraphrasing for Web Search

In this section, we first summarize our paraphrase extraction approaches, and then describe our paraphrasing engine for the web search task from three aspects, including: 1) a search-oriented paraphrasing model; 2) an NDCG-based parameter optimization algorithm; 3) an enhanced ranking model with augmented features that are computed based on the extra knowledge provided by the paraphrase candidates of the original queries.

2.1 Paraphrase Extraction

Paraphrases can be mined from various resources. Given a bilingual corpus, we use Bannard and Callison-Burch (2005)'s pivot-based approach to extract paraphrases. Given a monolingual corpus, Lin and Pantel (2001)'s method is used to extract paraphrases based on distributional hypothesis. Additionally, human annotated data can also be used as high-quality paraphrases. We use Miller (1995)'s approach to extract paraphrases from the synonym dictionary of WordNet. Word alignments within each paraphrase pair are generated using GIZA++ (Och and Ney, 2000).

2.2 Search-Oriented Paraphrasing Model

Similar to statistical machine translation (SMT), given an input query Q, our paraphrasing engine generates paraphrase candidates[1] based on a linear model.

$$\begin{aligned}\hat{Q} &= \arg\max_{Q' \in \mathcal{H}(Q)} P(Q'|Q) \\ &= \arg\max_{Q' \in \mathcal{H}(Q)} \sum_{m=1}^{M} \lambda_m h_m(Q, Q')\end{aligned}$$

$\mathcal{H}(Q)$ is the hypothesis space containing all paraphrase candidates of Q, h_m is the m^{th} feature function with weight λ_m, Q' denotes one candidate. In order to enable our paraphrasing model to learn the preferences on different paraphrasing strategies according to the characteristics of web queries, we design search-oriented features[2] based on word alignments within Q and Q', which can be described as follows:

- Word Addition feature $h_{WADD}(Q, Q')$, which is defined as the number of words in the paraphrase candidate Q' without being aligned to any word in the original query Q.
- Word Deletion feature $h_{WDEL}(Q, Q')$, which is defined as the number of words in the original query Q without being aligned to any word in the paraphrase candidate Q'.
- Word Overlap feature $h_{WO}(Q, Q')$, which is defined as the number of word pairs that align identical words between Q and Q'.
- Word Alteration feature $h_{WA}(Q, Q')$, which is defined as the number of word pairs that align different words between Q and Q'.
- Word Reorder feature $h_{WR}(Q, Q')$, which is modeled by a relative distortion probability distribution, similar to the distortion model in (Koehn et al., 2003).
- Length Difference feature $h_{LD}(Q, Q')$, which is defined as $|Q'| - |Q|$.
- Edit Distance feature $h_{ED}(Q, Q')$, which is defined as the character-level edit distance between Q and Q'.

Besides, a set of traditional SMT features (Koehn et al., 2003) are also used in our paraphrasing model, including translation probability, lexical weight, word count, paraphrase rule count[3], and language model feature.

2.3 NDCG-based Parameter Optimization

We utilize minimum error rate training (MERT) (Och, 2003) to optimize feature weights of the paraphrasing model according to NDCG. We define $\mathcal{D}$ as the entire document set. $\mathcal{R}$ is a ranking model[4] that can rank documents in $\mathcal{D}$ based on each input query. $\{Q_i, \mathcal{D}_i^{Label}\}_{i=1}^{S}$ is a human-labeled development set. Q_i is the i^{th} query and $\mathcal{D}_i^{Label} \subset \mathcal{D}$ is a subset of documents, in which the relevance between Q_i and each document is labeled by human annotators.

MERT is used to optimize feature weights of our linear-formed paraphrasing model. For

[1] We apply CYK algorithm (Chappelier and Rajman, 1998), which is most commonly used in SMT (Chiang, 2005), to generating paraphrase candidates.

[2] Similar features have been demonstrated effective in (Jones et al., 2006). But we use SMT-like model to generate query reformulations.

[3] Paraphrase rule count is the number of rules that are used to generate paraphrase candidates.

[4] The ranking model $\mathcal{R}$ (Liu et al., 2007) uses matching features computed based on original queries and documents.

each query Q_i in $\{Q_i\}_{i=1}^S$, we first generate N-best paraphrase candidates $\{Q_i^j\}_{j=1}^N$, and compute NDCG score for each paraphrase based on documents ranked by the ranker $\mathcal{R}$ and labeled documents $\mathcal{D}_i^{Label}$. We then optimize the feature weights according to the following criterion:

$$\hat{\lambda}_1^M = \arg\min_{\lambda_1^M}\{\sum_{i=1}^S Err(\mathcal{D}_i^{Label}, \hat{Q}_i; \lambda_1^M, \mathcal{R})\}$$

The objective of MERT is to find the optimal feature weight vector $\hat{\lambda}_1^M$ that minimizes the error criterion Err according to the NDCG scores of top-1 paraphrase candidates.

The error function Err is defined as:

$$Err(\mathcal{D}_i^{Label}, \hat{Q}_i; \lambda_1^M, \mathcal{R}) = 1 - \mathcal{N}(\mathcal{D}_i^{Label}, \hat{Q}_i, \mathcal{R})$$

where $\hat{Q}_i$ is the best paraphrase candidate according to the paraphrasing model based on the weight vector λ_1^M, $\mathcal{N}(\mathcal{D}_i^{Label}, \hat{Q}_i, \mathcal{R})$ is the NDCG score of $\hat{Q}_i$ computed on the documents ranked by $\mathcal{R}$ of $\hat{Q}_i$ and labeled document set $\mathcal{D}_i^{Label}$ of Q_i. The relevance rating labeled by human annotators can be represented by five levels: "Perfect", "Excellent", "Good", "Fair", and "Bad". When computing NDCG scores, these five levels are commonly mapped to the numerical scores 31, 15, 7, 3, 0 respectively.

2.4 Enhanced Ranking Model

In web search, the key objective of the ranking model is to rank the retrieved documents based on their relevance to a given query.

Given a query Q and its retrieved document set $\mathbf{D} = \{D_Q\}$, for each $D_Q \in \mathbf{D}$, we use the following ranking model to compute their relevance, which is formulated as a weighted combination of matching features:

$$\mathcal{R}(Q, D_Q) = \sum_{k=1}^K \lambda_k F_k(Q, D_Q)$$

$\mathbf{F} = \{F_1, ..., F_K\}$ denotes a set of matching features that measure the matching degrees between Q and D_Q, $F_k(Q, D_Q) \in \mathbf{F}$ is the k^{th} matching feature, λ_k is its corresponding feature weight.

How to learn the weight vector $\{\lambda_k\}_{k=1}^K$ is a standard learning-to-rank task. The goal of learning is to find an optimal weight vector $\{\hat{\lambda}_k\}_{k=1}^K$, such that for any two documents $D_Q^i \in \mathbf{D}$ and $D_Q^j \in \mathbf{D}$, the following condition holds:

$$\mathcal{R}(Q, D_Q^i) > \mathcal{R}(Q, D_Q^j) \Leftrightarrow r_{D_Q^i} > r_{D_Q^j}$$

where r_{D_Q} denotes a numerical relevance rating labeled by human annotators denoting the relevance between Q and D_Q.

As the ultimate goal of improving paraphrasing is to help the search task, we present a straightforward but effective method to enhance the ranking model $\mathcal{R}$ described above, by leveraging paraphrase candidates of the original query as the extra knowledge to compute matching features.

Formally, given a query Q and its N-best paraphrase candidates $\{Q'_1, ..., Q'_N\}$, we enrich the original feature vector $\mathbf{F}$ to $\{\mathbf{F}, \mathbf{F}_1, ..., \mathbf{F}_N\}$ for Q and D_Q, where all features in $\mathbf{F}_n$ have the same meanings as they are in $\mathbf{F}$, however, their feature values are computed based on Q'_n and D_Q, instead of Q and D_Q. In this way, the paraphrase candidates act as hidden variables and expanded matching features between queries and documents, making our ranking model more tunable and flexible for web search.

3 Experiment

3.1 Data and Metric

Paraphrase pairs are extracted as we described in Section 2.1. The bilingual corpus includes 5.1M sentence pairs from the NIST 2008 constrained track of Chinese-to-English machine translation task. The monolingual corpus includes 16.7M queries from the log of a commercial search engine. Human annotated data contains 0.3M synonym pairs from WordNet dictionary. Word alignments of each paraphrase pair are trained by GIZA++. The language model is trained based on a portion of queries, in which the frequency of each query is higher than a predefined threshold, 5. The number of paraphrase pairs is 58M. The minimum length of paraphrase rule is 1, while the maximum length of paraphrase rule is 5.

We randomly select $2,838$ queries from the log of a commercial search engine, each of which attached with a set of documents that are annotated with relevance ratings described in Section 2.3. We use the first $1,419$ queries together with their annotated documents as the development set to tune paraphrasing parameters (as we discussed in Section 2.3), and use the rest as the test set. The ranking model is trained based on the development set. NDCG is used as the evaluation metric of the web search task.

3.2 Baseline Systems

The baselines of the paraphrasing and the ranking model are described as follows:

The paraphrasing baseline is denoted as **BL-Para**, which only uses traditional SMT features described at the end of Section 2.2. Weights are optimized by MERT using BLEU (Papineni et al., 2002) as the error criterion. Development data are generated based on the English references of NIST 2008 constrained track of Chinese-to-English machine translation task. We use the first reference as the source, and the rest as its paraphrases.

The ranking model baseline (Liu et al., 2007) is denoted as **BL-Rank**, which only uses matching features computed based on original queries and different meta-streams of web pages, including URL, page title, page body, meta-keywords, meta-description and anchor texts. The feature functions we use include unigram/bigram/trigram BM25 and original/normalized Perfect-Match. The ranking model is learned based on SVM^{rank} toolkit (Joachims, 2006) with default parameter setting.

3.3 Impacts of Search-Oriented Features

We first evaluate the effectiveness of the search-oriented features. To do so, we add these features into the paraphrasing model baseline, and denote it as **BL-Para+SF**, whose weights are optimized in the same way with BL-Para. The ranking model baseline BL-Rank is used to rank the documents. We then compare the NDCG@1 scores of the best documents retrieved using either original query, or query paraphrases generated by BL-Para and BL-Para+SF respectively, and list comparison results in Table 1, where Cand@1 denotes the best paraphrase candidate generated by each paraphrasing model.

Test Set		
	BL-Para	**BL-Para+SF**
Original Query	Cand@1	Cand@1
27.28%	26.44%	26.53%

Table 1: Impacts of search-oriented features.

From Table 1, we can see, even using the best query paraphrase, its corresponding NDCG score is still lower than the NDCG score of the original query. This performance dropping makes sense, as changing user queries brings the risks of query drift. When adding search-oriented features into the baseline, the performance changes little, as these two models are optimized based on BLEU score only, without considering characteristics of mismatches in search.

3.4 Impacts of Optimization Algorithm

We then evaluate the impact of our NDCG-based optimization method. We add the optimization algorithm described in Section 2.3 into BL-Para+SF, and get a paraphrasing model **BL-Para+SF+Opt**. The ranking model baseline BL-Rank is used. Similar to the experiment in Table 1, we compare the NDCG@1 scores of the best documents retrieved using query paraphrases generated by BL-Para+SF and BL-Para+SF+Opt respectively, with results shown in Table 2.

Test Set		
	BL-Para+SF	**BL-Para+SF+Opt**
Original Query	Cand@1	Cand@1
27.28%	26.53%	27.06%(+**0.53**%)

Table 2: Impacts of NDCG-based optimization.

Table 2 indicates that, by leveraging NDCG as the error criterion for MERT, search-oriented features benefit more (+0.53% NDCG) in selecting the best query paraphrase from the whole paraphrasing search space. The improvement is statistically significant ($p < 0.001$) by t-test (Smucker et al., 2007). The quality of the top-1 paraphrase generated by BL-Para+SF+Opt is very close to the original query.

3.5 Impacts of Enhanced Ranking Model

We last evaluate the effectiveness of the enhanced ranking model. The ranking model baseline BL-Rank only uses original queries to compute matching features between queries and documents; while the enhanced ranking model, denoted as **BL-Rank+Para**, uses not only the original query but also its top-1 paraphrase candidate generated by BL-Para+SF+Opt to compute augmented matching features described in Section 2.4.

Dev Set		
	NDCG@1	NDCG@5
BL-Rank	25.31%	33.76%
BL-Rank+Para	28.59%(+**3.28**%)	34.25%(+**0.49**%)

Test Set		
	NDCG@1	NDCG@5
BL-Rank	27.28%	34.79%
BL-Rank+Para	28.42%(+**1.14**%)	35.68%(+**0.89**%)

Table 3: Impacts of enhanced ranking model.

From Table 3, we can see that NDCG@k ($k = 1, 5$) scores of BL-Rank+Para outperforms BL-Rank on both dev and test sets. T-test shows that the improvement is statistically significant ($p < 0.001$). Such end-to-end NDCG improvements come from the extra knowledge provided by the hidden paraphrases of original queries. This narrows down the query-document mismatch issue to a certain extent.

4 Conclusion and Future Work

In this paper, we present an in-depth study on using paraphrasing for web search, which pays close attention to various aspects of the application including choice of model and optimization technique. In the future, we will compare and combine paraphrasing with other query reformulation techniques, e.g., pseudo-relevance feedback (Yu et al., 2003) and a conditional random field-based approach (Guo et al., 2008).

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC Grant No. 61272343) as well as the Doctoral Program of Higher Education of China (FSSP Grant No. 20120001110112).

References

Ricardo A Baeza-Yates. 1992. Introduction to data structures and algorithms related to information retrieval.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL*, pages 597–604.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of ACL*, pages 286–293.

Jean-Cédric Chappelier and Martin Rajman. 1998. A generalized cyk algorithm for parsing stochastic cfg. In *Workshop on Tabulation in Parsing and Deduction*, pages 133–137.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*, pages 263–270.

Nick Craswell and Martin Szummer. 2007. Random walks on the click graph. In *Proceedings of SIGIR*, SIGIR '07, pages 239–246.

Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. 2002. Probabilistic query expansion using query logs. In *Proceedings of WWW*, pages 325–332.

Van Dang and Bruce W. Croft. 2010. Query reformulation using anchor text. In *Proceedings of WSDM*, pages 41–50.

Jonathan L. Elsas, Jaime Arguello, Jamie Callan, and Jaime G. Carbonell. 2008. Retrieval and feedback models for blog feed search. In *Proceedings of SIGIR*, pages 347–354.

Jiafeng Guo, Gu Xu, Hang Li, and Xueqi Cheng. 2008. A unified and discriminative model for query refinement. In *Proceedings of SIGIR*, SIGIR '08, pages 379–386.

Yufeng Jing and W. Bruce Croft. 1994. An association thesaurus for information retrieval. In *In RIAO 94 Conference Proceedings*, pages 146–160.

Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of KDD*, pages 217–226.

Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of WWW*, pages 387–396.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*, pages 48–54.

Victor Lavrenko and W. Bruce Croft. 2001. Relevance based language models. In *Proceedings of SIGIR*, pages 120–127.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Natural Language Engineering*, pages 343–360.

Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. 2007. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR workshop*, pages 3–10.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, pages 39–41.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of ACL*, pages 440–447.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP*, pages 142–149.

Mark D Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of CIKM*, pages 623–632.

Xuanhui Wang and ChengXiang Zhai. 2008. Mining term association patterns from search logs for effective query reformulation. In *Proceedings of the 17th ACM conference on Information and knowledge management*, Proceedings of CIKM, pages 479–488.

Yang Xu, Gareth J.F. Jones, and Bin Wang. 2009. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of SIGIR*, pages 59–66.

Shipeng Yu, Deng Cai, Ji-Rong Wen, and Wei-Ying Ma. 2003. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *Proceedings of WWW*, pages 11–18.

Wei Zhang and Clement Yu. 2006. Uic at trec 2006 blog track. In *Proceedings of TREC*.

Shiqi Zhao, Ming Zhou, and Ting Liu. 2007. Learning question paraphrases for qa from encarta logs. In *Proceedings of IJCAI*, pages 1795–1800.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. 2009. Application-driven statistical paraphrase generation. In *Proceedings of ACL*, pages 834–842.