User Behaviors Lend a Helping Hand: Learning Paraphrase Query Patterns from Search Log Sessions

Shiqi Zhao[1,2] *Haifeng Wang*[1] *Ting Liu*[2*]
(1) Baidu, Beijing, China
(2) School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
zhaoshiqi@baidu.com, wanghaifeng@baidu.com, tliu@ir.hit.edu.cn

ABSTRACT
Search log sessions contain a large number of paraphrases contributed by users during query rewriting. However, it is a big challenge to distinguish paraphrases from the simply related queries in the sessions. This paper addresses this problem by making innovative use of user behavior information embodied in query sessions. Specifically, we learn paraphrase patterns from the search log sessions with a classification framework, in which three types of user behavior features are exploited besides the conventional features. We evaluate the method using a query log of a commercial search engine. Experimental results demonstrate the effectiveness of our method, especially the significant contribution of the user behavior features. We extract over 250,000 pairs of paraphrase patterns from the used search log, with a precision over 76%.

TITLE AND ABSTRACT IN ANOTHER LANGUAGE (CHINESE)

基于用户行为特征从搜索会话中挖掘复述查询模板

搜索引擎用户在搜索时经常会对查询进行同义改写，以获得更好的搜索结果，因此搜索日志的查询会话中包含大量的复述资源。然而一个困难的问题是如何区分查询会话中哪些查询是复述关系，哪些则仅仅是语义相关而已。本文通过更好地利用查询会话中隐含的用户行为特征来解决这一问题。具体地，我们基于分类的方法从查询会话资源中挖掘复述查询模板。在分类模型中，我们不仅使用了传统的复述识别特征，还尝试了三种用户行为特征。我们使用一个商业搜索引擎的用户搜索日志来评估本文提出的方法。评估结果证明了本文方法的有效性，尤其是用户行为特征起到的明显作用。我们从所使用的用户日志中共抽取出了超过250,000对的复述模板，其准确率超过76%。

KEYWORDS: paraphrase, pattern, query, search log.

KEYWORDS IN CHINESE: 复述, 模板, 查询, 搜索日志.

* Corresponding author

Proceedings of COLING 2012: Technical Papers, pages 3137–3152,
COLING 2012, Mumbai, December 2012.

1 Introduction

Paraphrases have been shown to be useful in plenty of areas, such as machine translation (MT) (Callison-Burch et al., 2006; Madnani et al., 2007; Marton et al., 2009), question answering (QA) (Lin and Pantel, 2001; Ravichandran and Hovy, 2002; Duboue and Chu-Carroll, 2006; Riezler et al., 2007), and web search (Zukerman and Raskutti, 2002). In particular, the capability of paraphrasing is essential in web search, since in many cases the user queries need to be paraphrased so as to improve the quality of the search results.

This paper focuses on learning paraphrase query patterns from search log sessions, which could be useful in various applications, especially in query paraphrasing. Although search log sessions have been extensively exploited for mining related queries, this is the first work, as far as we know, to learn paraphrases from this data source. Mining paraphrase query patterns and related queries are both useful for search engines, but in different aspects. On the one hand, related queries can be used for query suggestion and recommendation, using which the users can extend their search interest and get some related information. However, the related queries are not suitable for direct query rewriting with the purpose of retrieving more and better results exactly reflecting the user's requirement, since the related queries often have different meanings from the original user query. On the other hand, the paraphrase query patterns are mined for query paraphrasing, which is to directly rewrite user queries during search. This is useful especially in the cases where the original user queries contain some uncommon words that need to be rewritten into more common expressions with the same meaning. The following examples show the difference between them:

Related queries:
q_1: 凯文 杜兰特 的 身高 是 多少 *(what is the height of Kevin Durant)*
q_2: 凯文 杜兰特 的 体重 是 多少 *(what is the weight of Kevin Durant)*
Paraphrase queries:
q_1: 凯文 杜兰特 的 身高 是 多少 *(what is the height of Kevin Durant)*
q_2: 凯文 杜兰特 的 真实 身高 *(true height of Kevin Durant)*

In a nutshell, our method involves two steps. In the first step, we induce candidate pattern pairs from query pairs co-occurring within the same query sessions, while in the second step, we recognize paraphrase patterns from the candidates using a classifier. We investigate comprehensive features in paraphrase recognition, including not only conventional features based on string similarities, but also novel features based on user behaviors. In detail, we design three types of user behavior features, namely, (1) the frequency based features for measuring the co-occurrence frequency between two patterns within sessions, (2) the lexical score features for estimating the lexical level paraphrasing likelihood, and (3) the distance based features for measuring the separation distance between queries in sessions.

We conduct experiments using a Chinese query log from Baidu[1], a commercial search engine. The results show that the classification based approach is effective in paraphrase recognition. Particularly, the user behavior features can significantly enhance the classification performance. More than 250,000 pairs of paraphrase patterns are learned from the used search log with a precision over 76%, which suggests that the search log sessions are rich in high-quality paraphrases. Furthermore, we find that our method is complementary to a previous method learning paraphrases from query-click pairs of query logs, which inspires us to integrate them in our future research.

[1] www.baidu.com

In what follows, we first review related studies in Section 2. We then introduce the pattern pair induction method in Section 3. The paraphrase pattern recognition method is proposed in Section 4, in which the user behavior features are described in detail. We present the experiment results in Section 5 and conclude the paper in Section 6.

2 Related Work

2.1 Paraphrase Learning

Plenty of methods have been proposed to extract paraphrases from various data sources. In (Barzilay and McKeown, 2001), the authors viewed multiple translation versions of the same literary works as monolingual parallel corpora and extracted paraphrases with a co-training algorithm. In (Barzilay and Lee, 2003; Dolan et al., 2004), researchers collected comparable news articles reporting on the same event, and further extracted parallel sentences for learning paraphrase phrases and patterns. There are also studies focusing on extracting paraphrases from large-scale monolingual corpora based on distributional hypothesis (Lin and Pantel, 2001; Bhagat and Ravichandran, 2008). The basic idea is that phrases or patterns appearing in similar contexts tend to have the same meaning.

Besides monolingual corpora, bilingual corpora have also been exploited for paraphrase extraction. Bannard and Callison-Burch (2005) first presented the method to learn paraphrase phrases from a bilingual phrase table. The key idea is that phrases aligned with the same foreign phrase could be paraphrases. Callison-Burch (2008) then improved the method by imposing syntax constraints to filter paraphrases with different syntactic structures. In addition, Zhao et al. (2008) extended this method to paraphrase pattern extraction.

To our knowledge, few studies have been conducted on learning paraphrases from query logs. Zhao et al. (2010)'s study might be the closest to our work. Their method is motivated by the assumption that user queries and the clicked titles are potential paraphrases. Accordingly, they train a classifier to recognize paraphrases from query-title pairs. They further extract query-query and title-title paraphrases from the query-title paraphrases based on the assumption that queries clicking on the same title and titles clicked on for the same query are also likely to be paraphrases. Additionally, they induce paraphrase patterns from the mined paraphrases. Our work differs from Zhao et al.'s mainly in that we learn paraphrase patterns from query sessions instead of query-click pairs. We compare these two methods in the experiments (Section 5.3).

2.2 Search Log Mining

Search engine query logs have been extensively exploited. Especially, there is a large body of research focusing on mining related queries from search logs and applying them in query rewriting and recommendation. Such research can be classified into three groups. The first group of methods utilizes user clicks when computing query similarity. The underlying assumption is that if users tend to click on similar documents for two queries, then the meanings of the queries should be similar (Wen et al., 2002; Baeza-Yates and Tiberi, 2007). In the second group of methods, researchers mine query rewriting terms directly from user clicked documents. Their basic idea is that terms from queries and user clicked documents are related (Cui et al., 2002; Riezler et al., 2008). The third group of methods learns related queries from query sessions. The assumption is that queries submitted by the same user within a short time might be related in meaning (Fonseca et al., 2005; Jones et al., 2006; Zhang and Nasraoui, 2006; Szpektor et al., 2011). Our work is close to the third group. However, what we learn are

paraphrase patterns rather than related queries or patterns.

3 Pattern Pair Induction

3.1 Concepts

Query. In this work, we collect user queries and other useful information from the used search logs and represent a query q as a triplet:

$$q = \langle qc, qt, cn \rangle,$$

where qc denotes the query content, qt is the time when qc is searched, and cn ($\geq$0) denotes the number of results clicked by the user. All queries in the search log are first preprocessed, including word segmentation, Part-of-Speech (POS) tagging, and Named Entity (NE) recognition.

Session. A query session (QS) is a sequence of queries submitted by the same user within a short time. We represent a query session with n queries as:

$$QS = q_1 ... q_i ... q_n,$$

where q_i is a query described above. Note that the order of queries in a session does matter, which records the sequence of user actions.

Pattern. A pattern is composed of two parts, i.e., pattern words and slots. Pattern slots can be instantiated with different words that meet certain constraints. We represent pattern slots as POS tags, which means that each slot can be instantiated by words with the specified POS. In our experiments, words with five kinds of POSes are allowed to form slots, including *noun* (n), *verb* (v), *adjective* (a), *numeral* (m), and *time* (t).

3.2 Induce Pattern Pairs

The reason why we learn paraphrase query patterns rather than directly extracting paraphrase queries is that paraphrase patterns usually achieve a higher coverage in applications than paraphrase instances. In addition, query patterns are much less sparse than queries. This work induces pattern pairs from query pairs co-occurring within the same query sessions. In detail, from a session $QS = q_1 ... q_i ... q_n$, we first extract all query pairs $\langle q_i, q_j \rangle$ ($1 \leq i < j \leq n$) in which q_i and q_j share at least one identical word. We then replace one or more shared identical word with their POS tags (slots), and thereby generate pattern pairs. It is obvious that we may induce more than one pattern pair from a query pair by selecting different slots. In addition, we assign a unique number to each pair of aligned slots in a pattern pair to distinguish slots with identical POS tags.

Figure 1 shows an example of pattern pair induced from a query pair. The slot $[n-1]$ denotes that the slot is the first slot in the pattern and the POS of fillers should be noun (n). In practice, we constrain that each pattern contains at least one content word besides slots, so as to filter meaningless patterns. In addition, since queries are mostly short, we constrain that each pattern contains at most two slots. We aggregate pattern pairs induced from all sessions in the search log and sum up the frequencies for each pair. Pattern pairs satisfying the following requirement are retained: (1) the frequency of the pattern pair exceeds a threshold T_1, (2) the number of unique fillers for each slot exceeds a threshold T_2.

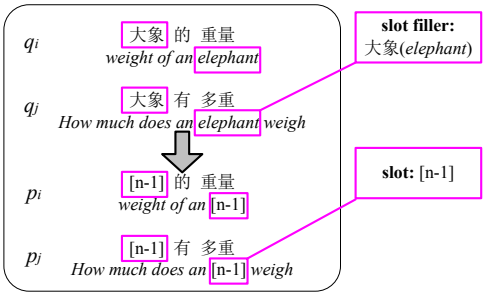

Figure 1: Example of pattern pair induction.

4 Paraphrase Pattern Recognition

Following the previous studies (Brockett and Dolan, 2005; Finch et al., 2005; Malakasiotis, 2009), we recast paraphrase pattern recognition as a classification problem. Each induced pattern pair is classified into one of the two classes, i.e., *paraphrase* and *non-paraphrase*. A Support Vector Machines (SVM) classifier is used in our experiments, since it has proven effective in this task (Brockett and Dolan, 2005; Finch et al., 2005). Our classification features can be divided into two groups: the baseline features examined in previous studies (Section 4.1) and user behavior based features proposed in this work (Section 4.2).

4.1 Baseline Features (F_{BL})

Conventional features for paraphrase recognition include three classes, i.e., lexical features, syntactic features, and semantic features. The lexical features measure the surface similarity between word sequences. Syntactic features compute the structural similarity between parse trees. Semantic features measure deep semantic relatedness based on some external knowledge base, such as WordNet in English. Our baseline features are mostly lexical features. In detail, given a pair of candidate patterns p_1 and p_2, our baseline features include: (1) length ratio feature, computed as the length of the shorter pattern divided by that of the longer one, (2) edit distance feature, computed as described in (Zhao et al., 2010), (3) cosine similarity feature, in which the words are weighted based on tf-idf, (4) word overlap rate feature, (5) character overlap rate feature[2], and (6) named entity feature, a boolean feature indicating whether p_1 and p_2 contain identical named entities.

We do not use syntactic features because most of user queries are not well-formed sentences but short n-grams, which cannot be parsed. We do not employ semantic features, either, since the underlying semantic knowledge bases are language-dependent.

4.2 User Behavior Features (F_{UB})

The most distinguishing characteristic of the query log, compared with other data sources, is that it contains rich information about users' searching and browsing behaviors, which could be

[2]Chinese words are composed of characters and words with the same meaning often contain similar characters.

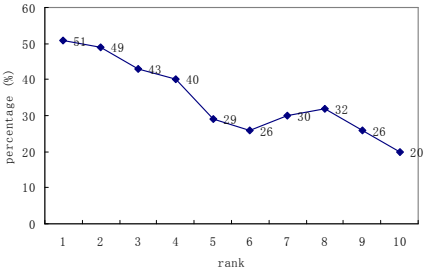

Figure 2: Percentage of paraphrases at each rank.

useful features for recognizing paraphrases. In this work, we design three types of user behavior features based on the observation and analysis of user behaviors from different aspects, which are detailed as follows.

Frequency based Features (F_{fr}). According to our observation, pattern pairs that frequently co-occur within sessions are more likely to be paraphrases. We substantiate our observation with an experiment, in which we randomly sampled 100 patterns with at least 10 candidate paraphrase patterns. We ranked all candidates according to their co-occurrence frequency with the target pattern and kept top-10. The 1000 pattern pairs are manually evaluated and the percentage of paraphrases at each rank is depicted in Figure 2. As can be seen, the percentage of paraphrases decreases as the rank gets lower. We therefore design the frequency based feature as:

$$F_{fr}(p_1, p_2) = \frac{freq(p_1, p_2)}{\sum_p freq(p_1, p) + C_1}, \tag{1}$$

where $freq(p_1, p_2)$ is the frequency of $\langle p_1, p_2 \rangle$ on the whole set of pattern pairs, C_1 is a constant parameter used to avoid overestimating the feature value when p_1 is too infrequent. We also compute the frequency based feature in the other direction, i.e., $F_{fr}(p_2, p_1)$ in the same way.

Lexical Score Features (F_{ls}). Inspired by lexical weight features used to measure phrase pair quality in machine translation (Koehn et al., 2003), we introduce lexical score features to measure the lexical level paraphrase likelihood of each pattern pair. We design a lexical scoring approach based on the observation that many words keep unchanged when users rewrite their queries within sessions. It is reasonable to assume that those unchanged words across queries should exclusively align with themselves, while the changed words may likely form paraphrase word pairs. Accordingly, given a pair of related queries $q_1 = w_{11}...w_{1m}$ and $q_2 = w_{21}...w_{2n}$ extracted from the same session, we compute two scores for any word pair $\langle w_{1i}, w_{2j} \rangle$ $(1 \le i \le m, 1 \le j \le n, w_{1i} \ne w_{2j})$ from them, namely, a *positive* score $a_+(w_{1i}, w_{2j})$ and a *negative* score $a\_(w_{1i}, w_{2j})$. Suppose W is the set of identical words shared by q_1 and q_2, and $l = |W|$, we compute $a_+(w_{1i}, w_{2j})$ and $a\_(w_{1i}, w_{2j})$ as:

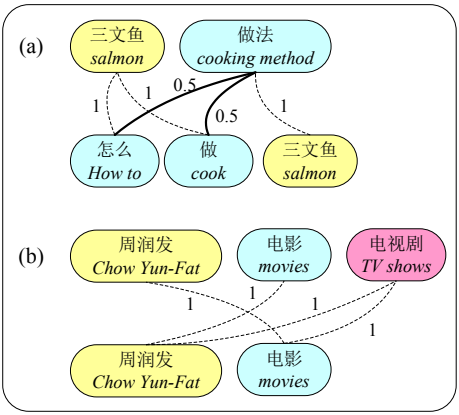

Figure 3: Examples of lexical scoring.

$$a_+(w_{1i}, w_{2j}) = \begin{cases} 0 & if\ w_{1i} \in W \vee w_{2j} \in W \\ \frac{1}{(m-l)*(n-l)} & otherwise \end{cases}$$

$$a_-(w_{1i}, w_{2j}) = \begin{cases} 1 & if\ w_{1i} \in W \vee w_{2j} \in W \\ 0 & otherwise \end{cases}$$

It can be interpreted that, if a word w in q_1 also appears in the other q_2, then w cannot *align* with other words in q_2 (i.e., such alignment gets a negative score). Otherwise, w will get an equal likelihood to *align* with each word in q_2 (i.e., gets a positive score). Examples in Figure 3 illustrate the lexical scoring process. A solid line denotes a positive alignment (a_+), whereas a dashed line denotes a negative alignment (a_-). The a_+ and a_- scores are also given in the figure. As can be seen, the lexical scoring approach assigns a_+ scores to the potential paraphrases (e.g., *cooking method* and *how to cook*) and a_- scores to the incorrectly aligned pairs (e.g., *movies* and *TV shows*).

We sum up a_+ and a_- scores for each word pair w_{1i} and w_{2j} over all the extracted query pairs and compute the lexical score $LS(w_{1i}, w_{2j})$ as follows:

$$LS(w_{1i}, w_{2j}) = \frac{\sum a_+(w_{1i}, w_{2j})}{\sum a_+(w_{1i}, w_{2j}) + \sum a_-(w_{1i}, w_{2j}) + C_2}, \quad (2)$$

where C_2 is a smoothing parameter. At run time, for a pattern pair $\langle p_1, p_2 \rangle$, we ignore slots, stop words, and the shared identical words from two patterns. Suppose the left words are

$p_1' = w_{11}...w_{1m}$ and $p_2' = w_{21}...w_{2n}$, we define the lexical score feature $F_{ls}(p_1, p_2)$ as follows:

$$F_{ls}(p_1, p_2) = \frac{1}{n}\sum_{j=1}^{n} \max_{1 \le i \le m} LS(w_{1i}, w_{2j}). \quad (3)$$

We compute feature $F_{ls}(p_2, p_1)$ in the same way.

Distance based Features (F_{dis}). It is obvious that we should not treat all query pairs from a session equally. Our observation of the user behaviors reveals that queries close to each other in a session are more likely to be paraphrases than those far apart. Therefore, given a pair of related queries $\langle q_i, q_j \rangle$ from a session QS, we measure the distance between two queries from three aspects:

- Query based distance dq. dq is defined as the number of queries between q_i and q_j in the session QS:
$$dq(q_i, q_j) = j - i - 1. \quad (4)$$
- Click based distance dc. The motivation here is that if a user clicks on a few retrieved results, it is likely that the user finds related information from the current results, and it is therefore less likely for her to further paraphrase the query. Given the number of clicks cn_k for query q_k, dc is defined as the sum of clicks between q_i and q_j:
$$dc(q_i, q_j) = \sum_{k=i}^{j-1} cn_k. \quad (5)$$
- Search time based distance dt. Given the timestamps qt_i and qt_j for queries q_i and q_j, dt is defined as the search time interval between q_i and q_j in the session QS:
$$dt(q_i, q_j) = qt_j - qt_i. \quad (6)$$

The distance of a pattern pair $\langle p_1, p_2 \rangle$ is defined as the average distance between query pairs from which $\langle p_1, p_2 \rangle$ is induced. Let $dx(p_1, p_2)$ be the distance between two patterns, we define the distance based feature as: $F_{dx}(p_1, p_2) = exp\{-dx(p_1, p_2)\}$, which guarantees that: (1) the smaller the distance, the larger the feature value, and (2) the feature values vary in the range (0,1].

5 Experiments

In our experiments, we used a query log from Baidu, a Chinese commercial search engine for extracting paraphrase query patterns. The queries were first segmented into sessions using an algorithm based on both time interval and content relatedness. A total of 87,744,130 sessions, containing 362,994,092 queries, were collected from the used query log after removing sessions with only one query. Query preprocessing, including word segmentation, POS tagging, and NE recognition[3], was performed using toolkits implemented based on the state-of-the-art models.

[3]Our NE classes include not only conventional classes like *person*, *location*, *organization*, *numeral*, and *time*, but also some classes frequently occur in user queries, including *movie*, *tv show*, *song*, *novel*, *brand*, *video game*, and *software*.

	P (%)	R (%)	F (%)
F_{BL}	73.09	57.88	64.45
$F_{BL}+F_{fr}$	73.84	63.01*	67.90*
$F_{BL}+F_{ls}$	75.19*	61.87*	67.82*
$F_{BL}+F_{dis}$	73.96	63.90*	68.48*
$F_{BL}+F_{fr}+F_{ls}$	75.85*	64.32*	69.54*
$F_{BL}+F_{fr}+F_{ls}+F_{dis}$	**76.39***	**67.36***	**71.54***

Table 1: P/R/F under different feature combinations. "*" indicates that the improvement is significant ($p < 0.05$) compared with the classifier using only baseline features.

We induced 868,243 pattern pairs from the sessions as described in Section 3. Note that, in practice, we eliminated pattern pairs in which one pattern subsumes the other, i.e., the case of expansion or reduction, as well as the pairs in which two patterns only have some trivial differences, such as inserting or deleting a stop word. The libsvm toolkit[4] was used as the classifier, with its default parameter settings. Some other parameters used in our method were set empirically: $T_1 = 5$, $T_2 = 3$, $C_1 = 20$, $C_2 = 10$.

5.1 Evaluation of the Classifier

We randomly sampled 5115 candidate pattern pairs to form the experimental data set. Two Chinese native speakers were asked to annotate the pattern pairs separately. A pattern pair should be annotated as *positive* (correct paraphrase patterns) or *negative* (otherwise). We follow the instance-based evaluation approach proposed by Szpektor et al. (2007). Particularly, we provide pattern slot fillers to the annotators along with the pattern pairs. A pattern pair is judged as paraphrase only when most of the instances generated by filling the slots with the provided fillers are paraphrases. We calculated the annotation agreement between two annotators. The result shows that the observed agreement is 0.96 and the Kappa value is 0.90. We believe that the high annotation agreement is due to the careful training of the annotators and the instance-based evaluation approach. A third annotator was asked to decide the final annotation for the disagreed pattern pairs.

To evaluate the classifier, we ran 5-fold cross validation with the human annotated data, in which we used 4/5 of the data for training and the rest 1/5 for testing in each run. The evaluation criteria are precision (P), recall (R), and f-measure (F) with regard to the positive class. The average P, R, and F of the classifier under different feature combinations over five runs are reported in Table 1.

The first line of table 1 shows the classification performance when we only use the baseline features. Lines 2-4 summarize the performance when we add each type of user behavior features separately. As can be seen, all the user behavior features can significantly improve the recall and f-measure. This is not surprising, since many paraphrase patterns are not similar enough at the surface level. The user behavior features supply additional evidences for measuring the semantic closeness between patterns, which help to recognize more paraphrase patterns with larger surface difference. Furthermore, lines 5-6 show that the classification performance keeps improving when the user behavior features are added one by one. We achieve the highest P, R, and F when all three types of user behavior features are used. This result indicates that the

[4] downloaded from: www.csie.ntu.edu.tw/∼cjlin/libsvm/.

	P (%)	R (%)	F (%)
all features	76.39	67.36	71.54
w/o F_{dq}	76.49	67.36	71.58
w/o F_{dc}	76.53	67.24	71.54
w/o F_{dt}	76.60	67.36	71.64
w/o F_{dq}+F_{dc}	76.74	67.42	71.71
w/o F_{dq}+F_{dt}	76.45	67.24	71.50
w/o F_{dc}+F_{dt}	75.42	64.14*	69.26*

Table 2: Analysis of the distance based features. "*" indicates that the degradation is significant ($p < 0.05$) compared with the classifier using all features.

user behavior features improve the performance from different aspects, it is thus necessary to combine them together.

We can find from line 3 of Table 1 that the lexical score features also significantly improve the precision of the classifier, which implies that this type of features is useful for filtering noise. We find after analyzing the data that, query pairs like "贾斯汀比伯 身高 (*Justin Bieber height*)" and "贾斯汀比伯 身高 体重 (*Justin Bieber height and weight*)" are quite common in users' search sessions, in which a query is expanded by adding a word closely related to the original query words. As a result, many closely related non-paraphrase word pairs, like 身高 (*height*) and 体重 (*weight*), get large $a\_$ scores and thereby penalized by the lexical score features. That's the main reason why the precision can be enhanced using this type of features.

People may wonder if our lexical score features can outperform the lexical weight features used in MT. For comparison, we implemented the latter on our data set. In detail, we conducted word alignment on the candidate pattern pairs and computed lexical weights in two directions as proposed in (Koehn et al., 2003). We then replaced our lexical score features with the lexical weights and evaluated the classifier via 5-fold cross validation. The average performance is: P=74.78%, R=67.30%, F=70.76%. As can be seen, the performance is lower than the current result (See the last line of Table 1), especially in precision. This result suggests that the lexical weighting approach in MT is unsuitable in paraphrase recognition. The main reason, we believe, is that it is unable to penalize the closely related non-paraphrases as our approach does.

In addition, since the distance based features are defined from three different aspects, we need to evaluate the individual contributions of these features. To this end, we omitted one or two distance based features from several runs and analyzed the influence. The results are given in Table 2. It is interesting to find from lines 2-4 of the table that, there is no degradation in performance when we omitted each distance based feature separately, which suggests that the features may not be independent of each other. Further results can be found from lines 5-7, where we omitted two features together each time. We can see that eliminating the click based and search time based distance features together (F_{dc}+F_{dt}) leads to an obvious degradation in recall and f-measure, while the query based distance feature (F_{dq}) seems helpless. It also implies that the F_{dc} and F_{dt} features follow the same trend. Actually, it is likely that the user's requirement has already been satisfied by the current search results if she spends a long time clicking on and browsing the results. It is therefore less possible for her to paraphrase the query any more. In other words, users' clicking and browsing behaviors are good indicators for recognizing paraphrases.

Phrase Substitution (42.80%)	
p_1: [n-1] 词汇 (*[n-1] vocabulary*)	p_2: [n-1] 词语 (*[n-1] words*)
p_1: [n-1] 为什么 坐牢 (*why was [n-1] imprisoned*)	p_2: [n-1] 为什么 入狱 (*why was [n-1] jailed*)
Information +/- (38.22%)	
p_1: 诺基亚手机 [n-1] (*Nokia mobile phone [n-1]*)	p_2: 诺基亚 的 [n-1] (*Nokia [n-1]*)
p_1: [n-1] [n-2] 考试成绩 (*[n-1] [n-2] exam results*)	p_2: [n-1] [n-2] 的 成绩 (*[n-1] [n-2] results*)
Spelling Correction (12.04%)	
p_1: [n-1] [v-2] 那里 (*where is [n-1] [v-2]*)	p_2: [n-1] [v-2] 哪里 (*where is [n-1] [v-2]*)
p_1: [n-1] tongyici (*[n-1] synonyms*)	p_2: [n-1] 同义词 (*[n-1] synonyms*)
Complex Paraphrase (6.94%)	
p_1: [n-1] 重要 吗 (*Is [n-1] important?*)	p_2: [n-1] 的 重要性 (*importance of [n-1]*)
p_1: 如何 治疗 [n-1] (*how to cure [n-1]*)	p_2: [n-1] 的 治疗方法 (*treatment of [n-1]*)

Table 3: Examples of the extracted paraphrase patterns under different types.

5.2 Evaluation of the Paraphrase Patterns

We used all the 5115 pairs of human annotated patterns to train a classifier, which was then applied to recognize paraphrase patterns from the candidate pattern pairs. A total of 252,963 pairs of paraphrase patterns were extracted in this way. Our statistics show that the average length of the patterns is 2.78 (words), which is mainly because the user queries are mostly short. One may argue that the short paraphrase patterns cannot cover long queries in applications such as query paraphrasing. We believe this problem can be alleviated by allowing partial match of patterns when applying them on long queries. Further statistics show that, over 77% of the paraphrase patterns contain only one slot, and over 90% slots are noun slots.

We randomly sampled 1000 pairs of paraphrase patterns for human assessment. The result shows that the precision is 76.4%. Typological analysis shows that the correct paraphrase patterns can be classified into four groups, including (1) phrase substitution, (2) adding or removing (+/-) information that does not change the meaning, (3) spelling correction, and (4) complex paraphrases, involving both word replacement and structural transformation. The distributions and examples are depicted in Table 3.

As can be seen, phrase substitution is the most represented class, with over 42% of all paraphrase patterns. This is consistent with the statistics reported in (Zhao et al., 2008). There are also quite a few paraphrase patterns in the class "information +/-", reflecting that many users tend to add or remove information to refine their queries while preserving the meaning. The third class, i.e., spelling correction was not conventionally regarded as a type of paraphrase. However, pattern pairs of this class actually convey the same requirement of users and are useful in applications such as query correction. Complex paraphrases are scarce compared with the other

three classes, suggesting that users do not often dramatically transform their queries if their requirement does not change. Note that, paraphrase patterns of the type *spelling correction* can only be applied in one direction, namely, to paraphrase the incorrect queries to the correct forms, while the other three types are not sensitive to the direction. Our method is not likely to learn paraphrase patterns with wrong direction, as people seldom paraphrase a correct query to an incorrect one in the search sessions.

It is interesting to find out to what extent the paraphrase patterns are dependent on the slot fillers. Our analysis shows that more than 76% of the paraphrase patterns are context-independent, which means that the pattern pairs convey the same meaning no matter what words instantiate the slots. For the other 24% of paraphrase patterns, the paraphrase relationship holds only under certain contexts. For example, the pattern pair "什么 [n-1] 好 (*what [n-1] is good*)" and "什么 [n-1] 好吃 (*what [n-1] is delicious*)" can be viewed as paraphrases only when the slot is filled with a food name. Judging in what context the paraphrase patterns can be applied is important in applications, which will be left in our future work.

5.3 Comparison Experiments

In this section, we compare our method with the method proposed in (Zhao et al., 2010), which is referred to as Zhao-10 hereafter. As mentioned above, Zhao-10 makes use of the click-through relationship between queries and clicked document titles from query logs. In particular, Zhao-10 learns paraphrase patterns in two steps. In the first one, it extracts candidate paraphrases from three sources, namely, pairs of queries and clicked titles, pairs of queries clicking on the same title, and pairs of titles clicked on for the same query. A classifier is employed to filter the candidates from each source. Then in the second step, it induces paraphrase patterns from each pair of paraphrases $\langle p_1, p_2 \rangle$ by abstracting one word shared by p_1 and p_2 as slot [X].

We ran Zhao-10 on our search log data and extracted 53,198 pairs of paraphrase patterns after removing the ones with only trivial differences between each other. We randomly selected 1000 pairs for manual annotation, and the result shows that the precision is 82.6%. Comparing this result with that reported in Section 5.2, we can find that the paraphrase patterns extracted with Zhao-10 are more precise than those extracted with the method proposed in this paper, but the quantity is smaller. We also analyze the types of the extracted correct paraphrase patterns as we do in section 5.2. The analysis result suggests that the paraphrase patterns extracted with Zhao-10 can also be classified into the four types as mentioned above, but the distribution is quite different: (1) phrase substitution: 82.13%, (2) information +/-: 9.79%, (3) spelling correction: 2.20%, (4) complex paraphrase: 5.88%.

Furthermore, we examined the intersection between paraphrase patterns extracted with our method and Zhao-10[5]. The result shows that the intersection is extremely small. Only 707 pairs of paraphrase patterns were extracted with both methods. This result implies that our method and Zhao-10 are quite complementary. It is promising to integrate these two methods for paraphrase extraction, whereby we can make full use of the search log information, including queries, clicks, and sessions.

[5]Since the pattern slots are not specified with POS tags in Zhao-10, we did not consider POS mismatch when counting the intersection of two sets of paraphrase patterns.

5.4 Analysis of Portability

People may wonder whether the proposed method can be extended to other languages or other applications. Here we analyze the portability of the method from two aspects:

Language portability. In our experiments, we used preprocessing tools to process Chinese query logs, which include word segmentation, POS tagging, and NE recognition (NER). Although these modules were implemented to process Chinese, the underlying algorithms and models can be language independent. However, please note that the models should be trained with annotated query data, since models trained with normal sentences or texts usually do not work well if they are directly applied on query corpora. Especially, in the NER module, we automatically mine new candidate NEs from the query logs everyday and update the NE dictionary, so as to handle the emerging NEs in user queries. To summarize, the preprocessing modules can be implemented based on language-independent models, but they should be specially trained and adapted for query data.

Application portability. The method is designed for mining paraphrase queries, which could then be used in query rewriting. However, the mined paraphrase patterns can also be used in other applications, especially the paraphrases with the type "phrase substitution", which we believe can be used in sentence rewriting and sentence similarity computation (i.e., matching paraphrases from two sentences when computing their similarity). We will examine the usefulness of the mined paraphrases in other applications in our future experiments.

6 Conclusions and Future Work

This paper proposes a classification-based method for learning paraphrase query patterns from search log sessions. The following conclusions can be drawn from the experiment results. Firstly, we for the first time demonstrate that search log session data is a rich source for extracting paraphrase patterns. Secondly, the classification-based method is effective in paraphrase pattern extraction. Especially, the proposed user behavior features evidently improve the classification performance. Thirdly, our method and the click-through based method (Zhao et al., 2010) are complementary. In our future work, we will improve our paraphrase extraction model by taking advantages of both the query session and click-through knowledge from search logs. In addition, we will try to automatically classify the extracted paraphrase patterns into different types (see Table 3), so as to suit different applications.

Acknowledgments

This work was supported by 863 State Key Project (Grant No. 2011AA01A207), National Natural Science Foundation of China (Grant No. 61073126, 61133012).

References

Baeza-Yates, R. and Tiberi, A. (2007). Extracting semantic relations from query logs. In *Proceedings of KDD*, pages 76–85.

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

Barzilay, R. and Lee, L. (2003). Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of HLT-NAACL*, pages 16–23.

Barzilay, R. and McKeown, K. R. (2001). Extracting paraphrases from a parallel corpus. In *Proceedings of ACL/EACL*, pages 50–57.

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

Brockett, C. and Dolan, W. B. (2005). Support vector machines for paraphrase identification and corpus construction. In *Proceedings of IWP*, pages 1–8.

Callison-Burch, C. (2008). Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP*, pages 196–205.

Callison-Burch, C., Koehn, P., and Osborne, M. (2006). Improved statistical machine translation using paraphrases. In *Proceedings of HLT-NAACL*, pages 17–24.

Cui, H., Wen, J.-R., Nie, J.-Y., and Ma, W.-Y. (2002). Probabilistic query expansion using query logs. In *Proceedings of WWW*, pages 325–332.

Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of COLING*, pages 350–356.

Duboue, P. A. and Chu-Carroll, J. (2006). Answering the question you wish they had asked: The impact of paraphrasing for question answering. In *Proceedings of HLT-NAACL*, pages 33–36.

Finch, A., Hwang, Y.-S., and Sumita, E. (2005). Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In *Proceedings of IWP*, pages 17–24.

Fonseca, B. M., Golgher, P., Pôssas, B., Ribeiro-Neto, B., and Ziviani, N. (2005). Concept-based interactive query expansion. In *Proceedings of CIKM*, pages 696–703.

Jones, R., Rey, B., Madani, O., and Greiner, W. (2006). Generating query substitutions. In *Proceedings of WWW*, pages 387–396.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of HLT-NAACL*, pages 127–133.

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

Madnani, N., Ayan, N. F., Resnik, P., and Dorr, B. J. (2007). Using paraphrases for parameter tuning in statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 120–127.

Malakasiotis, P. (2009). Paraphrase recognition using machine learning to combine similarity measures. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 27–35.

Marton, Y., Callison-Burch, C., and Resnik, P. (2009). Improved statistical machine translation using monolingually-dervied paraphrases. In *Proceedings of EMNLP*, pages 381–390.

Ravichandran, D. and Hovy, E. (2002). Learning surface text patterns for a question answering system. In *Proceedings of ACL*, pages 41–47.

Riezler, S., Liu, Y., and Vasserman, A. (2008). Translating queries into snippets for improved query expansion. In *Proceedings of COLING*, pages 737–744.

Riezler, S., Vasserman, A., Tsochantaridis, I., Mittal, V., and Liu, Y. (2007). Statistical machine translation for query expansion in answer retrieval. In *Proceedings of ACL*, pages 464–471.

Szpektor, I., Gionis, A., and Maarek, Y. (2011). Improving recommendation for long-tail queries via templates. In *Proceedings of WWW*, pages 47–56.

Szpektor, I., Shnarch, E., and Dagan, I. (2007). Instance-based evaluation of entailment rule acquisition. In *Proceedings of ACL*, pages 456–463.

Wen, J.-R., Nie, J.-Y., and Zhang, H.-J. (2002). Query clustering using user logs. *ACM Trans. Information. Systems*, 20(1):59–81.

Zhang, Z. and Nasraoui, O. (2006). Mining search engine query logs for query recommendation. In *Proceedings of WWW*, pages 1039–1040.

Zhao, S., Wang, H., and Liu, T. (2010). Paraphrasing with search engine query logs. In *Proceedings of COLING*, pages 1317–1325.

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

Zukerman, I. and Raskutti, B. (2002). Lexical query paraphrasing for document retrieval. In *Proceedings of COLING*, pages 1177–1183.