Learning Question Paraphrases for QA from Encarta Logs*

Shiqi Zhao[1], **Ming Zhou**[2], **Ting Liu**[1]
[1] Information Retrieval Laboratory, Harbin Institute of Technology
No. 27 Jiaohua Street, Nangang, Harbin, China, 150006
{zhaosq, tliu}@ir-lab.org
[2] Microsoft Research Asia
No. 49 Zhichun Road, Haidian, Beijing, China, 100080
mingzhou@microsoft.com

Abstract

Question paraphrasing is critical in many Natural Language Processing (NLP) applications, especially for question reformulation in question answering (QA). However, choosing an appropriate data source and developing effective methods are challenging tasks. In this paper, we propose a method that exploits Encarta logs to automatically identify question paraphrases and extract templates. Questions from Encarta logs are partitioned into small clusters, within which a perceptron classier is used for identifying question paraphrases. Experiments are conducted and the results have shown: (1) Encarta log data is an eligible data source for question paraphrasing and the user clicks in the data are indicative clues for recognizing paraphrases; (2) the supervised method we present is effective, which can evidently outperform the unsupervised method. Besides, the features introduced to identify paraphrases are sound; (3) the obtained question paraphrase templates are quite effective in question reformulation, enhancing the MRR from 0.2761 to 0.4939 with the questions of TREC QA 2003.

1 Introduction

Paraphrases are alternative ways of conveying the same information. In recent years, there has been growing research interest on paraphrasing since it is important in many NLP applications, including multi-document summarization, QA, text generation, and machine translation.

Question paraphrases, as a sub-class of paraphrases, are formally distinct questions that actually mean the same thing and have the same answer. Question paraphrasing is crucial in the question reformulation phase of a QA system. If an input question can be expanded with its various paraphrases, the recall of answers can be improved.

Compared with declarative sentences, questions contain some additional information, such as question words, question types and focuses. These are all indicative features when identifying question paraphrases. Thus, a method specially designed for question paraphrasing is worthy of study.

Question reformulation in QA has been widely researched. Some researchers have written reformulation templates manually [Brill et al., 2002]. Others have expanded questions using dictionaries such as WordNet [Hovy et al., 2001]. Some researchers have used the web as a resource for question reformulation [Yang et al., 2003]. Although employing various reformulation methods, all the above researches have verified the effectiveness of question reformulation.

This paper exploits a new resource, the Encarta logs, for question paraphrasing. An automatic method is designed to process the logs and identify paraphrases, including question classification, question partition, and paraphrase identification using a classifier. In recognizing paraphrases, some novel features are presented. Especially, user click information is used, which proves effective in experiments.

Templates are extracted from the derived question paraphrases and applied in question reformulation. Experiments show that the templates achieve wide coverage when tested on a TREC-QA question corpus, demonstrating that the Encarta log data is a good resource to learn question paraphrase templates. Experiments also show that the extracted templates are quite effective in question reformulation.

The remainder of the paper is organized in this way: Section 2 introduces related work. Our method of question paraphrasing is presented in Section 3. Experiments and results are described in Section 4. Section 5 is the conclusion and discusses future work.

2 Related Work

Various resources have been employed for paraphrase extraction. One resource is parallel monolingual corpus, such as multiple translations of literary works [Barzilay and McKeown, 2001]. While the translation-based methods facilitate the identification of paraphrases, such corpora are of limited availability since multiple translations on a large scale are not readily available in non-literary domains.

Other researchers exploit nonparallel monolingual corpora. Lin and Pantel (2001) discovered paraphrases by parsing a large unlabeled monolingual corpus and extracting semantically similar paths from dependency trees. The dis-

* This work was finished while the first author was visiting Microsoft Research Asia as a member of the project of AskBill Chatbot led by Dr. Ming Zhou.

advantage is that, only the templates with two arguments are considered. Another kind of nonparallel monolingual resource is the comparable news articles that report the same events [Shinyama et al., 2002; Barzilay and Lee, 2003]. The assumption behind it is that articles derived from different newspapers can contain paraphrases if they report the same event on the same day [Shinyama et al., 2002]. However, these methods seem to be of limited generality and difficult to be extended to other domains.

Bannard and Callison-Burch (2005) have sought to derive paraphrases from parallel bilingual corpora. They equated different English phrases aligned with the same phrase in another language based on the assumption that phrases mapped onto a single foreign language phrase tend to mean the same thing. Though this is a promising method, its performance depends greatly on word alignments.

None of the resources above are suitable in question paraphrasing because of domain limitation as well as the sparseness of question sentences. In contrast, Encarta logs are not domain limited, the queries in which can be about any topic. In addition, a sizable corpus of questions can be easily constructed from Encarta logs.

There is very limited work reported on question paraphrasing. Tomuro (2003) employed an FAQ corpus and defined patterns manually for question paraphrasing. Compared with FAQ corpora, Encarta logs supply additional information, i.e. the user click information, which proves a good indicator of paraphrases in our experiments (described in Section 4.1.3). Another difference from Tomuro's method is that our method identifies question paraphrases and extracts templates automatically.

3 Our Approach

The method comprises five steps: (1) extracts questions from Encarta logs; (2) classifies the extracted questions according to question types; (3) partitions the classified questions into fine-grained clusters; (4) identifies paraphrases from all question pairs within each cluster; (5) extracts templates from the identified paraphrases.

3.1 Question Extraction

Encarta is an online encyclopedia (http://encarta.msn.com). Encarta logs are user logs containing queries and documents that users clicked on for review. A small segment of Encarta logs is shown in Figure 1. For each line (a query session), the first half is a query. The codes following the query, separated by "*#*", are IDs of clicked documents.

```
……
Plant Cells: #761568511
Malaysia: #761558542 #761558542
rainforests: #761552810 #761552810 #761552810
what is the role of a midwife: #761565842
……
```

Figure 1. Encarta logs

The Encarta logs have been used for query clustering [Wen et al., 2002], in which the user click information proved a helpful feature. Encarta log data used in our experiments contains 4,946,932 query sessions. Though the number of logs is substantial, most of them are keywords or phrases rather than well-formed questions. Therefore, we need to filter the query logs and only retain questions. Here, simple heuristic rules are used: a query is recognized as a question if it contains three words or more, and a question word (i.e. *who*, *what*, *when*, *where*, *why*, and *how*).

Note that, in fact not all questions contain question words. For example, "*Name a stimulant*." is a question from TREC-QA which contains no question word. Currently, we do not process this kind of questions since it is difficult to differentiate them from declarative sentences. Future work may apply our method to these questions.

After extracting questions using the above method, a corpus containing 127,679 questions is constructed. In what follows, this corpus is called "question corpus".

3.2 Question Type Classification

In principle, any pair of questions in the question corpus should be considered when identifying paraphrases. However, the corpus contains over 120,000 questions, it is infeasible to identify paraphrases for each pair of questions. Therefore, a two-step process, involving question type classification (described in this section) and question partition (in Section 3.3), is employed to divide the whole corpus into thousands of small clusters and the identification of paraphrases is performed within each cluster.

The question type is an important attribute of a question, which usually indicates the category of its answer. In QA, question type classification is a necessary preprocessing stage. Table 1 shows a widely accepted question type taxonomy in QA [Li and Roth, 2002].

abbreviation, explanation
animal, body, color, creative, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
definition, description, manner, reason
group, individual, title, human-description
city, country, mountain, other, state
code, count, date, distance, money, order, other, period, percent, speed, temperature, size, weight

Table 1. Question type taxonomy

Based on the observation that two questions with different question types can hardly be paraphrases, questions in the corpus are first classified into 50 different types (Table 1). Our question classification method is similar to that introduced by Metzler and Croft (2005). We also build a two-level classifier. At the first level, questions are divided into six sets. Each set corresponds to a type of question word (i.e. *who, what, when, where, why*, and *how*). At the second level, a Support Vector Machine (SVM) classifier based on the taxonomy in Table 1 is trained for each set using the words as features. When classifying new questions, the process closely mimics the training steps. Given a new question, its question word is first identified. A feature vec-

tor is then created using the same features as in the training. Finally, the SVM corresponding to the question word is used for classification [Metzler and Croft, 2005].

The reason why a two-level classifier is employed is that the question words are prior knowledge and imply a great deal of information about the question types. A two-level classifier can make better use of this knowledge than a flat classifier that uses the question words simply as classification features. The training and testing data are the UIUC corpus [Li and Roth, 2002] and the TREC-10 questions. The experimental result shows that the classifier can achieve a classification accuracy of 84.2%.

3.3 Question Partition

In this stage, questions in each of the 50 classes are further partitioned into more fine-grained clusters. In this work, the paraphrases that have no common words are not considered.

Formally, given a content word (non-stopword) *w*, all questions within each question class that contain *w* are put into the same cluster (we may take it that this cluster is "indexed" by *w*). Apparently, if a question contains *n* different content words, it will be put into *n* clusters. After this process, the 50 classes obtained in the last step are further partitioned into about 37,000 clusters.

The question partition approach will be improved in future, in which two questions will be put into a same cluster if at least one pair of their content words are identical or synonymous.

3.4 Question Paraphrase Identification

3.4.1 Feature Selection

At this step, a classifier is used to identify paraphrases within the clusters obtained at the last step. If a cluster has *n* questions, *n*(n-1)/2* question pairs are generated by pairing any two questions in the cluster. For each pair, the classifier learns whether they are paraphrases (classifier outputs 1) or not (classifier outputs -1).

There are other researchers taking paraphrase identification as a problem of classification [Brockett and Dolan, 2005]. However, different features are used. The following are the features used in our work.

Cosine Similarity Feature (CSF): The cosine similarity of two questions is calculated after stemming and removing stopwords. Suppose q_1 and q_2 are two questions, V_{q1} and V_{q2} are the vectors of their content words. Then the similarity of q_1 and q_2 is calculated as in Equation (1).

$$Sim(q_1, q_2) = \cos(V_{q1}, V_{q2}) = \frac{\langle V_{q1}, V_{q2} \rangle}{\|V_{q1}\| \|V_{q2}\|} \quad (1)$$

where $\langle .,. \rangle$ denotes the inner product of two vectors and $\|.\|$ denotes the length of a vector.

Named Entity Overlapping Feature (NEF): Since named entities (e.g. person names, locations, time…) should be preserved across paraphrases [Shinyama et al., 2002], the overlapping rate of named entities in two questions is selected as a feature. The overlapping rate of two sets can be computed as in Equation (2):

$$OR(S_1, S_2) = \frac{|S_1 \cap S_2|}{\max(|S_1|, |S_2|)} \quad (2)$$

where S_1 and S_2 are two sets. $|.|$ is the cardinality of a set.

User Click Feature (UCF): It is easy to understand that if two questions often lead to the same document clicks, then these two questions tend to be similar [Wen et al., 2002]. The feature of user click similarity of two questions is calculated using Equation (3):

$$Sim_{user\_click}(q_1, q_2) = \frac{RD(q_1, q_2)}{\max(rd(q_1), rd(q_2))} \quad (3)$$

where $rd(.)$ is the number of clicked documents for a question and $RD(q_1, q_2)$ is the number of document clicks in common.

WordNet Synonyms Feature (WSF): The pair of questions is expanded with the synonyms extracted from WordNet synset entries. Specifically, a question *q* can be expanded to *q'*, which contains the content words in *q* along with their synonyms. Then for the expanded questions, the overlapping rate is calculated and selected as a feature.

Unmatched Word Feature (UWF): The above features measure the similarity of two questions while the unmatched word feature is designed to measure the divergence of two questions. Given questions q_1, q_2 and q_1's content word w_1, if neither w_1 nor its synonyms can be found in q_2, w_1 is defined as an unmatched word of q_1. We calculate the unmatched rate as in Equation (4) and use it as a feature.

$$UR(q_1, q_2) = \max(ur(q_1), ur(q_2)) \quad (4)$$

where *ur*(.) denotes the percentage of unmatched words in a question.

Syntactic Similarity Feature (SSF): In order to extract the syntactic similarity feature, the question pairs are parsed by a shallow parser whereby the key dependency relations can be extracted from a sentence. Four types of key dependency relations are defined: *SUB, OBJ, ATTR,* and *ADV*. For example, for the question "*What is the largest country*," the shallow parser will generate *(What, is, SUB), (is, country, OBJ), (largest, country, ATTR)* as the parsing result. As can be seen, the parsing result of each question is represented as a set of triples, where a triple comprises two words and their syntactic relation. The overlapping rate of two questions' syntactic relation triples is computed and used as their syntactic similarity.

Question Focus Feature (QFF): The question focus can be viewed as the target of a question. For example, in the question "*What is the capital of China?*" the question focus is "*capital*". Obviously, two questions are more likely to be paraphrases if they have identical question focus. Currently, the question focuses are extracted using predefined rules. The QFF feature has a binary value, namely, 1 (two questions have identical question focus) or 0 (otherwise).

Translation Similarity Feature (TSF): Translation information proves useful in paraphrase identification [Wu and Zhou, 2003]. In our experiments, Google online translation (http://translate.google.com/translate_t) is called to translate each English question into Chinese. Then the cosine similarity of the translations of two questions is calculated.

3.4.2 PAUM Classifier

It is found in the experiments that the input data for the paraphrase identifier is rather unbalanced, in which, only a very small proportion of the question pairs are paraphrases. The methods dealing with classification with unbalanced data include the Positive Example Based Learning (PEBL), one-class SVMs and Perceptron Algorithm with Uneven Margins (PAUM). Among these methods we use PAUM in our experiments [Li et al., 2002]. PAUM is an extension of the perceptron algorithm, which is specially designed to cope with two class problems where positive examples are very rare compared with negative ones, as is the case in the paraphrase identification task. PAUM considers the positive and negative margins separately. The positive (negative) margin $\gamma_{\pm 1}(\mathrm{w}, b, z)$ is defined as:

$$\gamma_{\pm 1}(\mathrm{w}, b, z) = \min_{(x_i, \pm 1) \in z} \frac{\pm(\langle \mathrm{w}, \mathrm{x}_i \rangle + b)}{\|\mathrm{w}\|} \quad (5)$$

where $z = ((x_1, y_1), ..., (x_m, y_m)) \in (\chi \times \{-1, +1\})^m$ is a training sample. $\phi : \chi \to \kappa \in R^n$ is a feature mapping into an n-dimension vector space κ. $\mathrm{x}_i = \phi(x_i)$. $\mathrm{w} \in \kappa, b \in R$ are parameters. $\langle .,. \rangle$ denotes the inner product in κ

3.5 Template Extraction

Templates are extracted from the derived question paraphrases. There is some research work on paraphrase template generation [Barzilay and Lee, 2003]. In our work, we use a simple method to extract templates. A better method will be presented in future.

As mentioned above, paraphrases are identified from each cluster in which a common content word *w* is shared by all questions. Hence, the paraphrase templates are formalized by simply replacing the index word *w* with wildcard "*". For example, the questions "*What is the length of Nile?*" and "*How long is Nile?*" are recognized as paraphrases from the cluster indexed by "*Nile*." Then the paraphrase template "*What is the length of* * ⇔ *How long is* *" is induced by replacing "*Nile*" with "*."

4 Evaluation

To evaluate the effectiveness of our method, three experiments are carried out. The first one is designed to evaluate the paraphrase identifier, especially feature selection. The second experiment evaluates the performance of the whole paraphrase acquisition process, which includes question type classification, question partition, and paraphrase identification. The third is designed to verify the usefulness of the generated templates in question reformulation.

4.1 Evaluation of Paraphrase Identifier

4.1.1 Data

As mentioned in Section 3.4.2, the paraphrase identifier is a PAUM classifier. In order to train and test the classifier, we extracted 67,379 question pairs from the question corpus and annotated them manually. The resulting data is extremely unbalanced, which contains 1,629 positive (paraphrases) and 65,750 negative (non-paraphrases) examples. In the experiment, 4/5 (1,296 positive and 52,466 negative) of the data is used in training while 1/5 (333 positive and 13,284 negative) is left for testing.

4.1.2 Performance of the Identifier

In this experiment, the precision and recall of the identifier are computed. Given that S_{cp} is the set of paraphrases automatically recognized by the identifier; S_{mp} is the set of paraphrases manually annotated. Then precision and recall are defined as in Equations (6) and (7):

$$precision = |S_{cp} \cap S_{mp}| / |S_{cp}| \quad (6)$$

$$recall = |S_{cp} \cap S_{mp}| / |S_{mp}| \quad (7)$$

The classification margins in a PAUM classifier can be adjusted so as to get different trade-offs between precision and recall. In most applications of the paraphrase templates, precision is more important than recall. For instance, in question reformulation of QA, a false expansion might do more harm than good, since it may bring about noise and lead to incorrect answers. Therefore, when setting the classification margin parameters, many different combinations have been tried on a small development set and ultimately we set positive margin parameter $\tau_{+1} = 6$ and negative margin parameter $\tau_{-1} = 1$, which are deliberately skewed towards precision. Experimental results show that the precision and recall are 77.60% and 71.77%, respectively.

4.1.3 Feature Contributions

To investigate the contributions of different features, we omitted each feature from several runs. The results are shown in Table 2.

	Precision(%)	Recall(%)
All Features	77.60	71.77
No CSF	75.47	72.07
No NEF	76.68	72.07
No UCF	74.76	71.17
No WSF	77.10	71.77
No UWF	78.44	63.36
No SSF	75.40	70.87
No QFF	74.70	73.57
No TSF	75.00	72.97

Table 2. Effect of eliminating each feature

Table 2 shows that eliminating the features CSF, UCF, SSF, QFF, and TSF can all produce a notable degradation in precision while eliminating the UWF feature can have large impact on recall. Of all the features, removal of the feature WSF appears to have the least impact. The reason may be that the WordNet synonyms used in WSF are also used in the feature UWF, which makes the WSF feature redundant. Besides, the effect of NEF feature is also small. In our future work, the NE information will be used in preprocessing. Specifically, only the question pairs with identical NEs are retained and identified by the paraphrase identifier.

In particular, we can see that feature UCF improves the precision evidently, which indicates that the user click information is an effective constraint in paraphrase identification. Figure 2 (a) shows an example of non-paraphrases that only can be identified correctly when the UCF feature is considered. As can be seen, the two questions are differenti-

ated by their user clicks though they are highly similar at string level. Figure 2 (b) shows an example of paraphrases that only can be recognized when the UCF feature is used. Evidently, it is the identical user click that makes it possible to identify these two formally distinct questions. However, it can be seen from Table 2 that the UCF feature only makes a slight improvement in recall. We conclude that the calculation of the user click similarity should be improved so as to enhance the recall. Since documents in Encarta are organized into a hierarchy that contains four levels, the hierarchy of logs will be taken into account in our future work.

(a) Where can I find information on automobiles #761576902
Where can I find information about 1930's automobiles #761563934
(b) When did Florida become a state: #761557601
When did Florida join the United States: #761557601

Figure 2. Examples that benefit from UCF feature

4.1.4 Comparison of Feature Selection Approaches

[Brockett and Dolan, 2005] used different features in their SVM classifier to identify paraphrases from related news sentences. Four feature classes are involved in their work, including string similarity features, morphological features, WordNet lexical mappings, and word association pairs. To compare our feature selection strategy with theirs, we have tested their features on the question corpus in our experiments. The comparison result is in Table 3:

	Precision	Recall
Ours	77.60%	71.77%
B & D	59.35%	21.92%

Table 3. Comparison of feature selection approaches

As can be seen, our feature selection dramatically outperforms that of [Brockett and Dolan, 2005] on the question corpus. This shows that the feature set we have designed is more effective in indicating question paraphrases.

4.2 Evaluation of the Whole Method

In the last section, the paraphrase identifier is evaluated. In this section, the paraphrasing method, including question type classification, question partition, and paraphrase identification, is evaluated as a whole. Question classification and partition divide the large question corpus into many small clusters, which makes it feasible for the identifier to detect paraphrases. However, they also bring about lost. Thus our main purpose is to evaluate the effect of these two stages.

To evaluate the performance of paraphrase acquisition, we randomly selected 660 questions from the question corpus, from which 305 pairs have been manually annotated as paraphrases. The 660 questions are first classified by question types. After that, further partition is done within each class as described in Section 3.3. Finally the paraphrase identifier is employed in each cluster to detect paraphrases.

After the above process, 160 pairs of questions are recognized as paraphrases, of which 107 pairs are true paraphrases (which overlap with the 305 hand-tagged ones). Precision and recall are 66.88% and 36.15%.

Compared with the result shown in Table 2, precision decreases from 77.60% to 66.88% and recall decreases from 71.77% to 36.15%. There are two main reasons for the notable decreases. One is that the question classification and partition bring about mistakes. Especially, some paraphrase pairs are divided into different classes or clusters; the other reason is the irregularity of the Encarta logs. There are many spelling mistakes in the logs. (e.g. "*Egyptian*" was written as "*Egyption*") These mistakes influence the calculation of similarity and the performance of paraphrase recognition. Additionally, the questions from the Encarta logs are quite flexible in expression, some of which are even ungrammatical. E.g. "*Russia what do they wear?*", "*Atomic bomb dropped on Hiroshima why?*" and the like. For these questions, the extraction of question focuses and the recognition of syntactic relations are extremely difficult, which makes the SSF feature and the QFF feature fail to work.

We also compared our method with the unsupervised method presented by Wen et al. [2002]. In their method, they clustered query logs using a density-based clustering method. They combined similarity based on query contents and that based on user clicks in clustering. In our experiment, the minimal density parameter was set to 3, which means that only those clusters containing at least 3 queries were kept. Then we varied the similarity threshold from 0.5 to 0.8. The performance is shown in Table 4.

Threshold	Precision	Recall
0.5	26.67%	8.11%
0.6	24.14%	7.09%
0.7	30.36%	11.49%
0.8	35.48%	7.43%

Table 4. Performance of the clustering method

As can be seen, both the precision (66.88%) and recall (36.15%) of our method are much higher than the clustering method, which indicates that the supervised method presented in this paper is more effective than the unsupervised method in recognizing question paraphrases.

4.3 Evaluation of Templates in QA

To evaluate the templates in question reformulation, 380 factoid questions from TREC-2003 QA Track are used. Of the 380 questions, 134 (35.26%) are matched with the extracted templates and reformulated while the left 246 questions are not reformulated.

Since we have not built a TREC QA system at this point, we evaluate the reformulation templates using a web QA method. Specifically, for each question, top 100 web snippets are retrieved by Google and Mean Reciprocal Rank (MRR) is used in the evaluation of question reformulation [Wang et al., 2005]. MRR is defined as follows:

$$MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{r_i} \quad (8)$$

where n is the number of questions and r_i is the rank of the first correct answer occurring in the top 100 retrieved snippets for the *i-th* question.

We compare the MRRs before and after question reformulation on the 134 template-matched questions as well as the total 380 questions. The comparison result is in Table 5.

	MRR (before reformulation)	MRR (after reformulation)
134 questions	0.2761	0.4939
380 questions	0.3137	0.3905

Table 5. MRRs before and after question reformulation

Table 5 shows that question reformulation using the extracted paraphrase templates enhances MRR dramatically, which suggests that question reformulation based on paraphrase templates can make correct answers rank higher in the retrieved snippet lists. This result demonstrates that our question paraphrasing method is effective in QA.

5 Conclusion

This paper presents a novel method to automatically extract question paraphrases from Encarta logs and generate templates for question reformulation. Our contribution is that, a new data source, namely, the Encarta log data is exploited to learn question paraphrases. The process for extracting paraphrases, including question type classification, question partition, and paraphrase identification, proves effective. Different features from all possible constructs are tested and the most effective combination of features is identified. Specifically, we introduce new features such as the user click feature in question paraphrase identification.

The generated templates also prove effective in question reformulation, leading MRR to grow from 0.2761 to 0.4939 with the questions of TREC QA 2003.

In the future work, we shall make better use of the user clicks from Encarta logs for paraphrase identification. Besides, we will evaluate our method on TREC QA corpus.

Acknowledgments

We are grateful to Yi Chen for his help in the experiments. We also thank Changning Huang, Cheng Niu, Dwight Daniels, and Wanxiang Che for their valuable comments on this paper.

Reference

[Bannard and Callison-Burch, 2005] Colin Bannard and Chris Callison-Burch. Paraphrasing with Bilingual Parallel Corpora. In *Proceedings of ACL-2005*, 2005.

[Barzilay and Lee, 2003] Regina Barzilay and Lillian Lee. Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. In *Proceedings of HLT/NAACL*, 2003.

[Barzilay and McKeown, 2001] Regina Barzilay and Kathleen R. McKeown. Extracting Paraphrases from a Parallel Corpus. In *Proceedings of the ACL/EACL*, 2001.

[Brill et al., 2002] Eric Brill, Susan Dumais, Michele Banko. An Analysis of the AskMSR Question-Answering System. In *Proceedings of EMNLP*, 2002.

[Brockett and Dolan, 2005] Chris Brockett and William B. Dolan. Support Vector Machines for Paraphrase Identification and Corpus Construction. In *Proceedings of IWP2005*, 2005.

[Hovy et al., 2001] Eduard Hovy, Ulf Hermjakob, Chin-Yew Lin. The Use of External Knowledge in Factoid QA. In *Proceedings of TREC-10 Conference*, 2001.

[Li and Roth, 2002] Xin Li and Dan Roth. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

[Li et al., 2002] Yaoyong Li, Hugo Zaragoza, Ralf Herbrich, John Shawe-Taylor, Jaz S. Kandola. The Perceptron Algorithm with Uneven Margins. In *Proceedings of ICML 02*, 2002.

[Lin and Pantel, 2001] Dekang Lin and Patrick Pantel. Discovery of Inference Rules for Question Answering. In *Natural Language Engineering* 7(4):343-360, 2001.

[Metzler and Croft, 2005] Donald Metzler, W. Bruce Croft. Analysis of Statistical Question Classification for Fact-based Questions. *Information Retrieval*, 8(3):481-504, 2005.

[Shinyama et al., 2002] Yusuke Shinyama, Satoshi Sekine, Kiyoshi Sudo and Ralph Grishman. Automatic Paraphrase Acquisition from News Articles. In *Proceedings of HLT*, 2002.

[Tomuro, 2003] Noriko Tomuro. Interrogative Reformulation Patterns and Acquisition of Question Paraphrases. In *Proceedings of IWP 2003*, 2003.

[Wang et al., 2005] Yi-Chia Wang, Jian-Cheng Wu, Tyne Liang, and Jason S. Chang. Web-Based Unsupervised Learning for Query Formulation in Question Answering. In *Proceedings of IJCNLP*, 2005.

[Wen et al., 2002] Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Query Clustering Using User Logs. *ACM Trans. Information. Systems*. 20(1): 59-81, 2002.

[Wu and Zhou, 2003] Hua Wu and Ming Zhou. Synonymous Collocation Extraction Using Translation Information. In *Proc. of ACL-2003*, 2003.

[Yang et al., 2003] Hui Yang and Tat-Seng Chua, Shuguang Wang, and Chun-Keat Koh. Structured Use of External Knowledge for Event-based Open Domain Question Answering. In *Proceedings of the Twenty Sixth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003.