Interrogative Reformulation Patterns and Acquisition of Question Paraphrases

Noriko Tomuro
DePaul University
School of Computer Science, Telecommunications and Information Systems
243 S. Wabash Ave.
Chicago, IL 60604 U.S.A.
tomuro@cs.depaul.edu

Abstract

We describe a set of paraphrase patterns for questions which we derived from a corpus of questions, and report the result of using them in the automatic recognition of question paraphrases. The aim of our paraphrase patterns is to factor out different syntactic variations of interrogative words, since the interrogative part of a question adds a syntactic superstructure on the sentence part (i.e., the rest of the question), thereby making it difficult for an automatic system to analyze the question. The patterns we derived are rules which map surface syntactic structures to semantic case frames, which serve as the canonical representation of questions. We also describe the process in which we acquired question paraphrases, which we used as the test data. The results obtained by using the patterns in paraphrase recognition were quite promising.

1 Introduction

The phenomenon of paraphrase in human languages is essentially the inverse of ambiguity – a given sentence could ambiguously have several meanings, while any given meaning could be formulated into several paraphrases using various words and syntactic constructions. For this reason, paraphrase poses a great challenge for many Natural Language Processing (NLP) tasks, just as ambiguity does, notably in text summarization and NL generation (Barzilay and Lee, 2003; Pang et al., 2003).

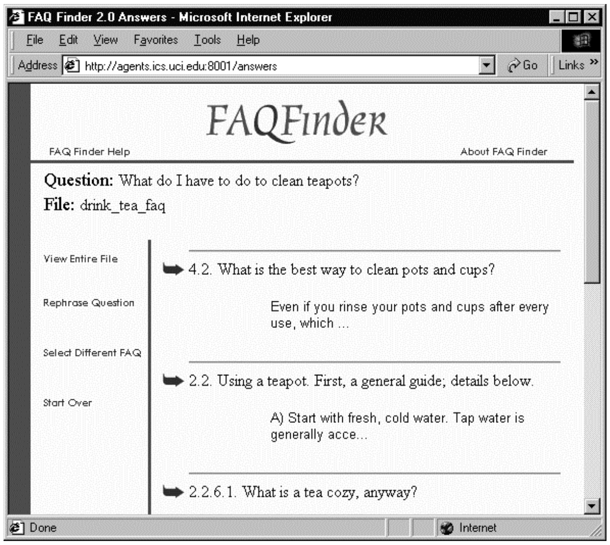
The problem of paraphrase is important in Question-Answering systems as well, because the systems must return the same answer to questions which ask for the same thing but are expressed in different ways. Recently there have been several work which utilized reformulations of questions as a way to fill the chasm between words in a question and those in a potential answer sentence (Hermjakob et al., 2002; Murata and Isahara, 2001; Agichtei et al., 2001). In general, paraphrasing a question, be it for recognition or generation, is more difficult than a declarative sentence, because interrogative words carry a meaning of their own, which is subject to reformulation, in addition to the rest (or the sentence part) of the question. Reformulations of the interrogative part of questions have some interesting characteristics which are distinct from reformulations of the sentence part or declarative sentences. First, paraphrases of interrogatives are strongly lexical and *idiosyncratic*, containing many keywords, idioms or fixed expressions. For example, for a question “How can I clean teapots?” one can easily think of some variations of the ’how’ part while fixing the sentence part:

- “In what way should I clean teapots?”
- “What do I have to do to clean teapots?”
- “What is the best way to clean teapots?”
- “What method is used for cleaning teapots?”
- “How do I go about cleaning teapots?”
- “What is involved in cleaning teapots?
- “What should I do if I want to clean teapots?

Second, reformulation patterns of interrogatives seem to be governed by *question types*. For example, the variation patterns above apply to almost all ’how-to’ questions, while ’why’ questions undergo a different set of transformations (e.g. “Why ..”, “For what reason ..”, “What was the reason why ..” etc.). Also, further observations suggest that questions of the same question type have the same semantic *empty category*: something (or some things) which a question is asking.

In this paper, we describe the set of paraphrase/reformulation patterns we derived from a corpus of questions, and report the result of using them in the automatic recognition of question paraphrases. We also describe the process in which we acquired paraphrases, which we used as the test data. Our approaches to constructing those resources were manual – the transformation patterns were derived by inspecting an existing large corpus of questions, and the paraphrases were collected by asking web users to type in reformulations of sample questions. Our work here is focused on the reformulations of the interrogative part of questions in contrast to other work in question-answering where major emphases are placed on the reformulations of phrases or words in the sentence part (Lin and Pantel, 2001; Hermjakob et al., 2002). The patterns we derived are essentially rules which map surface syntactic structures to semantic *case frame* representations. We use those case frame representations when we compare questions for similarity. The results obtained by the use of the patterns in paraphrase recognition were quite promising.

The motivation behind the work we present here is to improve the retrieval accuracy of our system called FAQFinder (Burke et al., 1997). FAQFinder is a web-based, natural language question-answering system which uses Usenet Frequently Asked Questions (FAQ) files to answer users’ questions. Each FAQ file contains a list of question-and-answer (Q&A) pairs on a particular subject. Given a user’s question as a query, FAQFinder tries to find an answer by matching the user’s question against the question part of each Q&A pair, and displays 5 FAQ questions which are ranked the highest by the system’s similarity measure. Thus, FAQFinder’s task is to identify FAQ questions which are the best paraphrases of the user’s question. Figure 1 shows a screen snapshot of FAQFinder where a user’s query “What do I have to do to clean teapots?” is matched against the Q&A pairs in ’drink_tea_faq’. The current similarity measure used in the system is a combination of four independent metrics: term vector similarity, coverage, semantic similarity, and question type similarity (Lytinen and Tomuro, 2002). Although those metrics are additive and complemental to each other, they cannot capture the relations and interactions between them. The idea of paraphrase patterns proposed in this paper is a first step in developing an alternative, integrated similarity measure for question sentences.



Question: What do I have to do to clean teapots?
File: drink_tea_faq

4.2. What is the best way to clean pots and cups?

Even if you rinse your pots and cups after every use, which ...

2.2. Using a teapot. First, a general guide; details below.

A) Start with fresh, cold water. Tap water is generally acce...

2.2.6.1. What is a tea cozy, anyway?

Figure 1: The 5 best-matching FAQ questions returned by FAQFinder

2 Paraphrasing Patterns for Questions

2.1 Training Data

Paraphrasing patterns were extracted from a large corpus of question sentences which we had used in our previous work (Tomuro and Lytinen, 2001; Lytinen and Tomuro, 2002). It consisted of 12938 example questions taken from 485 Usenet FAQ files. In the current work, we used a subset of that corpus consisting of examples whose question types were PRC (procedure), RSN (reason) or ATR (atrans). Those question types are members of the 12 question types we had defined in our previous work (Tomuro and Lytinen, 2001). As described in that paper, PRC questions are typical ’how-to’ questions and RSN questions are ’why’ questions. The type ATR

```
;(1) how can/do .. anyVerb
(defpattern prc-how 1
  (:WH how) (:S <NPS>) (:V <V>) (:O <NPO>)
  =>
  (:proc ?) (:actor <NPS>) (:verb <V>) (:theme <NPO>))

;(2) how can/do .. obtain
(defpattern atr-1-how-obtainV 3
  (:WH how) (:S <NPS>) (:V <obtainV>) (:O <NPO>)
  =>
  (:source ?) (:proc ?) (:actor <NPS>) (:verb <obtainV>) (:theme <NPO>))

;(3) what is the .. method for obtaining
(defpattern atr-1-what-is-method 4
  (:WH what) (:S NIL) (:V <beV>) (:O <methodN>) (:VG <obtainV>) (:NP <NPO>)
  =>
  (:source ?) (:proc ?) (:actor I) (:verb <obtainV>) (:theme <NPO>))

;(4) who sells
(defpattern atr-who-sourceNP 4
  (:WH who) (:S NIL) (:V <sellV>) (:O <NPO>)
  =>
  (:source ?) (:proc ?) (:actor I) (:verb obtain) (:theme <NPO>))
```

Figure 2: Example Paraphrase Patterns

(for ATRANS in Conceptual Dependency (Schank, 1973)) is essentially a special case of PRC, where the (desire for the) transfer of possession is strongly implied. An example question of this type would be “How can I get tickets for the Indy 500?”. Not only do ATR questions undergo the paraphrasing patterns of PRC questions, they also allow reformulations which ask for the (source or destination) location or entity of the thing(s) being sought, for instance, “Where can I get tickets for the Indy 500?” and “Who sells tickets for the Indy 500?”. We had observed that such ATR questions were in fact asked quite frequently in question-answering systems.[1] Also those question types seem to have a richer set of paraphrasing patterns than other types (such as definition or simple reference questions given in TREC competitions (Voorhees, 2000)) with regard to the interrogative reformulation. In the corpus, there were 2417, 1022 and 968 questions of type PRC, RSN, ATR respectively, and they constituted the training data in the current work.

2.2 Paraphrase Patterns

The aim of our paraphrasing patterns is to account for different syntactic variations of interrogative words. As we showed examples in section 1, the interrogative part of a question adds a syntactic superstructure to the sentence part, thereby making it difficult for an automatic system to get to the core of the question. By removing this syntactic overhead, we can derive the canonical representations of questions, and by using them we can perform a many-to-one matching instead of many-to-many when we compare questions for similarity.

In the pre-processing stage, we first applied a shallow parser to each question in the training data and extracted its phrase structure. The parser we used is customized for interrogative sentences, and its complexity is equivalent to a finite-state machine. The output of the parser is a list of phrases in which each phrase is labeled with its syntactic function in the question (subject, verb, object etc.). Passive questions are converted to active voice in the last step of the parser by inverting the subject and object noun phrases. Then using the pre-processed data, we manually inspected all questions and defined patterns which seemed to apply to more than two instances. By this enumeration process, we derived a total of 127 patterns, consisting of 18, 23 and 86

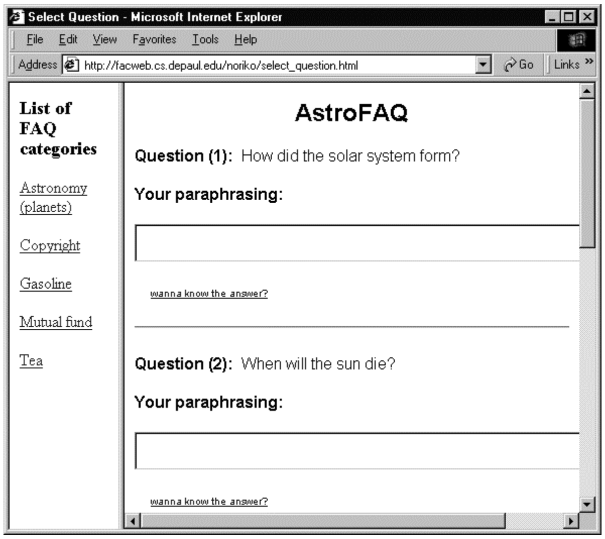
[1] Although we did not use it in the current work, we also had access to the user log of AskJeeves system (`http://www.askjeeves.com`). We observed that a large portion of the user questions were ATR questions.

patterns for PRC, RSN and ATR respectively.

Each pattern is expressed in the form of a rule, where the left-hand side (LHS) expresses the phrase structure of a question, and the right-hand side (RHS) expresses the semantic case frame representation of the question. When a rule is matched against a question, the LHS of the rule is compared with the question first, and if they match, the RHS is generated using the variable binding obtained from the LHS. Figure 2 shows some example patterns.

In a pattern, both LHS and RHS are a set of slot-value tuples. In each tuple, the first element, which is always prefixed with :, is the slot name and the remaining elements are the values. Slots names which appear on the LHS (:S, :V, :O, etc.) relate to syntactic phrases, while those on the RHS (:actor, :theme, :source etc.) indicate semantic cases. A slot value could be either a variable, indicated by a symbol enclosed in <..> (e.g. <NPS>), or a constant (e.g. how). A variable could be either constrained (e.g. <obtainV>) or unconstrained (e.g. <NPS>, <NPO>). Constrained variables are defined separately, and they specify that a phrase to be matched must satisfy certain conditions. Most of the conditions are lexical constraints – a phrase must contain a word of a certain class. For instance, <obtainV> denotes a word class ’obtainV’ and it includes words such as “obtain”, “get”, “buy” and “purchase”. Word classes are groupings of words appeared in the training data which have similar meanings (i.e., synonyms), and they were developed in tandem with the paraphrase patterns. Whether constrained or unconstrained, a variable gets bound with one or more words in the matched question (if possible for constrained variables). A constant indicates a word and requires the word to exist in the tuple. ’NIL’ and ’?’ are special constants where ’NIL’ requires the tuple (phrase in the matched question) to be empty, and ’?’ indicates that the slot is an empty category. Each rule is also given a priority level (e.g. 3 in pattern (2)), with a large number indicating a high priority.

In the example patterns shown in Figure 2, pattern (1) matches a typical ’how-to’ question such as “How do I make beer?”. Its meaning, according to the case frame generated by the RHS, would be “I” for the actor, “make” for the verb, “beer” for the theme, and the empty category is :proc (for procedure). Patterns (2) through (4) are rules for ATR questions. Notice they all have two empty categories – :proc and :source – as consistent with our definition of type ATR. Also notice the semantic case roles are taken from various syntactic phrases: pattern (2) takes the actor and theme from syntactic subject and object straight-forwardly, while pattern (3), which matches a question such as “What is a good way to buy tickets for the Indy 500”, takes the theme from the object in the infinitival phrase (:NP) and fills the actor with “I” which is implicit in the question. Pattern (4), which matches a question such as “Who sells tickets for the Indy 500”, changes the verb to “obtain” as well as filling the implicit actor with “I”. This way, ATR paraphrases are mapped to identical case frames (modulo variable binding).



http://facweb.cs.depaul.edu/noriko/select_question.html

List of FAQ categories

Astronomy (planets)

Copyright

Gasoline

Mutual fund

Tea

AstroFAQ

Question (1): How did the solar system form?

Your paraphrasing:

wanna know the answer?

Question (2): When will the sun die?

Your paraphrasing:

wanna know the answer?

Figure 3: Paraphrase Entry Site

3 Acquisition of Question Paraphrases

To evaluate the question paraphrase patterns, we used the set of question paraphrases which we had acquired in our previous work (Tomuro and Lytinen, 2001) for the test data. In that work, we obtained question paraphrases in the following way. First we selected a total of 35 questions from 5 FAQ categories: `astronomy`, `copyright`, `gasoline`, `mutual-fund` and `tea`. Then we created a web site where users could enter paraphrases for any of the 35 questions. Figure 3 shows a snapshot of the site when the astronomy FAQ is displayed.[2] After keeping the site public for two weeks, a total of 1000 paraphrases were entered. Then we inspected each entry and discarded ill-formed ones (such as keywords or boolean queries) and incorrect paraphrases. This process left us with 714 correct paraphrases (including the original 35 questions).

Figure 4 shows two sets of example paraphrases entered by the site visitors. In each set, the first sentence in bold-face is the original question (and its question type). In the paraphrases of the first question, we see more variations of the interrogative part of ATR questions. For instance, 1c explicitly refers to the source location/entity as "store" and 1d uses "place". Those words are essentially hyponyms/specializations of the concept 'location'. Paraphrases of the second question, on the other hand, show variations in the sentence part of the questions. The expression "same face" in the original question is rephrased as "one side" (2a), "same side" (2b), "not .. other side" (2c) and "dark side" (2f). The verb is changed from "show" to "face" (2b), "see" (2c, 2d) and "look" (2e). Those rephrasings are rather subtle, requiring deep semantic knowledge and inference beyond lexical semantics, that is, the common-sense knowledge.

To see the kinds of rephrasing the web users entered, we categorized the 679 (= 714 - 35) paraphrased questions roughly into the following 6 categories.[3]

(1) Lexical substitution – synonyms; involves no or minimal sentence transformation
(2) Passivization
(3) Verb denominalization – e.g. "destroy" vs. "destruction"
(4) Lexical semantics & inference – e.g. "show" vs. "see"
(5) Interrogative reformation – variations in the interrogative part
(6) Common-sense – e.g. "dark side of the Moon"

Table 1 shows the breakdown by those categories. As you see, interrogative transformation had the largest proportion. This was partly because all transformations to questions that start with "What" were classified as this category. But the data indeed contained many instances of transformation between different interrogatives (why ↔ how ↔ where ↔ who etc.). From the statistics above, we can thus see the importance of understanding the reformulations of the interrogatives. As for other categories, lexical substitution had the next largest proportion. This means a fair number of users entered relatively simple transformations. On this, (Lin and Pantel, 2001) makes a comment on manually generated paraphrases (as versus automatically extracted paraphrases): "It is difficult for humans to generate a diverse list of paraphrases, given a starting formulation and no context". Our data is in agreement with their observations indeed.

Table 1: Breakdown of the paraphrases by paraphrase category

Category	# of paraphrases	
(1) Lexical substitution	168	(25 %)
(2) Passivization	37	(5 %)
(3) Verb denominalization	18	(3 %)
(4) Lexical semantics & inference	107	(16 %)
(5) Interrogative reformation	339	(50 %)
(6) Common-sense	10	(1 %)
Total	679	(100 %)

4 Evaluation

Using the paraphrase data described in the previous section, we evaluated our question reformulation patterns on coverage and in the paraphrase recognition task. From the data, we selected all paraphrases derived from the original questions of type PRC, RSN and ATR. There were 306 such examples, and they constituted the testset for the evaluation.

4.1 Coverage

We first applied the transformation patterns to all examples in the testset and generated their case frame representations. In the 306 examples, 289 of them found at least one pattern. If an example matched with two or more patterns, the one with the highest priority was selected. Thus the coverage was 94%.

However after inspecting the results, we observed that in some successful matches, the syntactic structure of the question did not exactly correspond to

[2] In order to give a context to a question, we put a link ("wanna know the answer?") to the actual Q&A pair in the FAQ file for each sample question.

[3] If a paraphrase fell under two or more categories, the one with the highest number was chosen.

1. Where can I get British tea in the United States? [ATR]

a. How can I locate some British tea in the United States?
b. Who sells English tea in the U.S.?
c. What stores carry British tea in the United States?
d. Where is the best place to find English tea in the U.S.?
e. Where exactly should I go to buy British tea in the U.S.?
f. How can an American find British tea?

2. Why does the Moon always show the same face to the Earth? [RSN]

a. What is the reason why the Moon show only one side to the Earth?
b. Why is the same side of the Moon facing the Earth all the time?
c. How come we do not see the other side of the Moon from Earth?
d. Why do we always see the same side of the Moon?
e. Why do the Moon always look the same from here?
f. Why is there the dark side of Moon?

Figure 4: Examples of question paraphrases entered by the web users

the pattern as intended. For example, “How can I learn to drink less tea and coffee?”[4] matched the pattern (1) shown in Figure 2 and produced a frame where “I” was the actor, “learn” was the verb and the theme was null (because the shallow parser analyzed “to drink less tea and coffee” to be a verb modifier). Although the difficulty with this example was incurred by inadequate pre-processing or inherent difficulty in shallow parsing, the end result was a spurious match nonetheless. In the 289 matches, 15 of them were such false matches.

As for the 17 examples which failed to match with any patterns, one example is “What internet resources exist regarding copyright?”[5] – there were patterns that matched the interrogative part (“What internet resources”), but all of them had constrained variables for the verb which did not match “exist”. Other failed matches were because of elusive paraphrasing. For example, for an original question “Why is evaporative emissions a problem?”, web users entered “What’s up with evaporative emissions?” and “What is wrong with evaporative emissions?”. Those paraphrases seem to be keyed off from “problem” rather than “why”.

4.2 Paraphrase Recognition

Using the case frame representations derived from the first experiment, we applied a *frame similarity* measure for all pairs of frames. This measure is rather rudimentary, and we are planning to fine-tune it in the future work. This measure focuses on the effect of paraphrase patterns – how much the canonical representations, after the variations of interrogatives are factored out, can bring closer the (true) paraphrases (i.e., questions generated from the same original question), thereby possibly improving the recognition of paraphrases.

The frame similarity between a pair of frames is defined as a weighted sum of two similarity scores: one for the interrogative part (which we call *interrogative similarity*) and another for the sentence part (which we call *case role similarity*). The interrogative similarity is obtained by computing the average slot-wise correspondence of the empty categories (slots whose value is ’?’), where the correspondence value of a slot is 1 if both frames have ’?’ for the slot or 0 otherwise. The case role similarity, on the other hand, is obtained by computing the distance between two term vectors, where terms are the union of words that appeared in the remaining slots (i.e., non-empty category slots) of the two frames. Those terms/words are considered as a bag of words (as in Information Retrieval), irrespective of the order or the slots in which they appeared. We

[4] The original question for this paraphrase was “How can I get rid of a caffeine habit?”.

[5] This question can be paraphrased as “Where can I find information about copyright on the internet?”

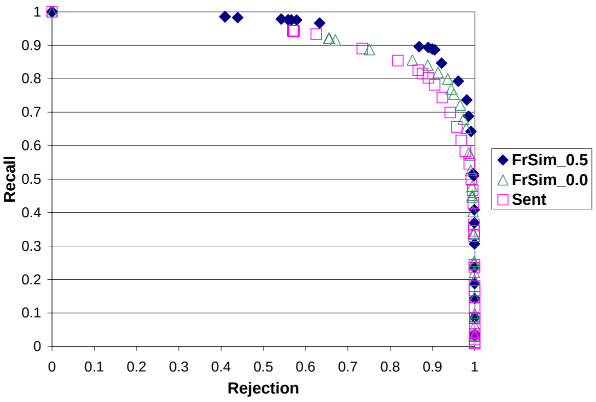


Figure 5: Recall vs. Rejection

chose this scheme for the non-empty category slots because our current work does not address the issue of paraphrases in the sentence part of the questions (as we mentioned earlier). Value of each term in a frame is either 1 if the word is present in the frame or 0 otherwise, and the cosine of the two vectors is returned as the distance. The final frame similarity value, after applying weights which sum to 1, would be between 0 and 1, where 1 indicates the strongest similarity.[6]

Using the frame similarity measure, we computed two versions – one with 0.5 for the weight of the interrogative similarity and another with 0.0. In addition, we also computed a baseline metric, *sentence similarity*. It was computed as the term vector similarity where terms in the vectors were taken from the phrase representation of the questions (i.e., syntactic phrases generated by the shallow parser). Thus the terms here included various wh-interrogative words as well as words that were dropped or changed in the paraphrase patterns (e.g. words instantiated with <methodN> in pattern (3) in Figure 2). This metric produces a value between 0 and 1, thus it is comparable to the frame similarity.

The determination of whether or not two frames (or questions) are paraphrase of each other depends on the threshold value – if the similarity value is above a certain threshold, the two frames/questions are determined to be paraphrases. With the 306 case frames in the testset, there were a total of 46665 (= $\frac{306*305}{2}$) distinct combinations of frames, and 3811 of them were (true) paraphrases. After computing the three metrics (two versions of frame similarity, plus sentence similarity) for all pairs, we evaluated their performance by examining the trade-off between recall and *rejection* for varying threshold values. Recall is defined in the usual way, as the ratio of true positives $(=\frac{\text{\# classified as paraphrase}}{\text{\# true paraphrases}})$, and rejection is defined as the ratio of true negatives $(=\frac{\text{\# classified as non-paraphrase}}{\text{\# true non-paraphrases}})$. We chose to use rejection instead of precision or accuracy because those measures are not normalized for the number of instances in the classification category (# true paraphrases vs. # true non-paraphrases); since our testset had a skewed distribution (8% paraphrases, 92% non-paraphrases), those measures would have only given scores in which the results for paraphrases was overshadowed by those for non-paraphrases.

Figure 5 shows the recall vs. rejection curves for the three metrics. As you see, both versions of the frame similarity (`FrSim_0.5` and `FrSim_0.0` in the figure) outperformed the sentence similarity (`Sent`), suggesting that the use of semantic representation was very effective in recognizing paraphrases compared to syntactic representation. For example, `FrSim_0.5` correctly recognized 90% of the true paraphrases while making only a 10% error in recognizing false positives, whereas `Sent` made a slightly over 20% error in achieving the same 90% recall level. This is a quite encouraging result.

The figure also shows that `FrSim_0.5` performed much better than `FrSim_0.0`. This means that explicit representation of empty categories (or question types) contributed significantly to the paraphrase recognition. This also underscores the importance of considering the formulations of interrogatives in analyzing question sentences.

5 Conclusions and Future Work

In this paper, we showed that automatic recognition of question paraphrases can benefit from understanding the various formulations of the interrogative part. Our paraphrase patterns remove those variations and produce canonical forms which reflect the meaning of the questions (i.e., case frames). Not only does this semantic representation facilitates simple and straight-forward ways to compute

[6] If either one of the frames is null (for which the pattern-matching failed), the frame similarity is 0.

the similarity of questions, it also produces more accurate results than syntactic phrase representation.

Our immediate future work is to define paraphrase patterns for other question types. While doing so, we would also like to look into ways to automatically extract patterns. A good starting point would be (Agichtei et al., 2001), which looked for common n-grams anchored at the beginning of questions.

Once the syntactic superstructure of the interrogative part is factored out, the next task is to tackle reformulations of the sentence part of questions. Lately several interesting efforts have been made to extract paraphrase expressions automatically, for instance (Lin and Pantel, 2001; Shinyama et al., 2002). We would like to experiment doing the same with the web as the resource.

Finally, we would like to synthesize the reformulation patterns of the two parts of questions and develop unified paraphrase patterns. Then we will incorporate this new approach in FAQFinder and conduct end-to-end question-answering experiments in order to see how much the use of paraphrase patterns can improve the performance of the system.

References

E. Agichtei, S. Lawrence, and L. Gravano. 2001. Learning search engine specific query transformations for question answering. In *Proceedings of the 10th International World Wide Web Conference (WWW10)*, Hong Kong.

R. Barzilay and L. Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the DARPA Human Language Technologies (HLT-2003)*.

R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. 1997. Question answering from frequently asked question files: Experiences with the faqfinder system. *AI Magazine*, 18(2).

U. Hermjakob, E. Abdessamad, and D. Marcu. 2002. Natural language based reformulation resource and web exploitation for question answering. In *Proceedings of TREC-2002*.

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

S. Lytinen and N. Tomuro. 2002. The use of question types to match questions in faqfinder. In *Papers from the 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*.

M. Murata and H. Isahara. 2001. Universal model for paraphrasing using transformation based on a defined criteria. In *Proceedings of the workshop on Automatic Paraphrasing at NLP Pacific Rim (NLPRS-2001)*, Tokyo, Japan.

B. Pang, K. Knight, and D. Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings of the DARPA Human Language Technologies (HLT-2003)*.

R. Schank. 1973. Identification of conceptualizations underlying natural language. In R. Schank and K. Colby, editors, *Computer Models of Thought and Language*. Freeman.

Y. Shinyama, S. Sekine, K. Sudo, and R. Grishman. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of Human Language Technology Conference (HLT-2002)*.

N. Tomuro and S. Lytinen. 2001. Selecting features for paraphrasing question sentences. In *Proceedings of the workshop on Automatic Paraphrasing at NLP Pacific Rim (NLPRS-2001)*, Tokyo, Japan.

E. Voorhees. 2000. The trec-9 question answering track report. In *Proceedings of TREC-9*.