Chapter 14

Retrieval Models and Q and A Learning With FAQ Files

Noriko Tomuro and Steven L. Lytinen

14.1 Introduction

The idea of compiling knowledge into FAQ files has existed for some time. The Usenet became an early repository of on-line FAQ files, and currently the Internet FAQ Archives web site (http://www.faqs.org) has 2490 "popular" FAQs archived. There are many other sources of FAQ files. Call-center manuals are also often structured as FAQ files. As the world wide web has become widely accessible, FAQ files have also become a popular way for web sites to store knowledge and convey answers to customers/site users about questions that these users would commonly ask. Thus, answers to a very wide variety of questions can be found in FAQ files, and developing applications and techniques for question answering tailored to FAQ files is an important thrust in question answering.

An FAQ file typically contains several question-and-answer (Q and A) pairs where questions are pre-answered and compiled by domain experts. Thus, finding answers from FAQ files essentially is to reuse previously answered questions instead of finding answers from scratch every time—an economical solution. Due to this semistructured format, retrieval models for FAQ files place emphasis on different techniques than retrieval models for unstructured documents. First, the primary focus is on finding an FAQ question (not answer) which is similar to the user query/question, that is, a Q-to-Q match. In essence, this is the task of recognizing question paraphrases-two or more questions which ask the same thing(s) but are formulated in different ways. Recently, the issue of paraphrase recognition has been receiving attention in question-answering research as a way to fill the gap between words in a question and those in an answer. Most approaches try to enumerate paraphrase patterns, for instance "How much does *X* cost?" $\Rightarrow$ "*X* costs *Y*" $\equiv$ "the price of *X* is *Y*" (Lin and Pantel, 2001; Hermjakob, Abdessamad, and Marcu, 2002).

Type	FAQ files	FAQFinder	AskJeeves
Open ended	8079 (62%)	435 (70%)	2131 (62%)
Specific	4859 (38%)	185 (30%)	1322 (38%)
Total	12938 (100%)	620 (100%)	3453 (100%)

Table 14.1. Proportions of Open-Ended Questions

However, these patterns express variations of potential answer sentences (to be matched with a given question), not of questions. To do Q-to-Q match, systems must be able to account for variations of interrogative words/phrases, such as "How do I make *X*?" ⇒ "What is involved in making *X*?"

Second, many questions answered in FAQ files are general and open-ended, such as "how," "why," and "yes/no" questions,[1] in contrast to narrow, specific questions which ask for simple facts, such as those asked in TREC QA tasks (Voorhees 2000). Although FAQ answers are prepackaged "canned answers," it is not necessary for the retrieval models to pinpoint (or generate) exact answers, those types of general questions must be properly analyzed and comprehensively covered in FAQ-based systems. An interesting observation is that many questions asked by users of real-world QA systems, including FAQFinder (Burke et al. 1997) and AskJeeves (http://www.askjeeves.com), are indeed such questions as well. This implies that in practice, people often seek substantial information from real QA systems. Table 14.1 shows the proportions of open-ended questions ("how"" "why," "yes/no") observed in sample data randomly selected from FAQ files, FAQFinder user logs and AskJeeves user logs.

In addition to Q-to-Q matching, retrieval models for FAQ files may also include an effort to match questions directly with answers, that is, a Q-to-A match. Since both question and answer are available in an FAQ file, it may be useful at times to find answers from the answer part of the Q and A pairs, either when no similar FAQ questions are found, or in addition to the Q-to-Q match.

This chapter describes on-going research on a real-world FAQ-based QA system called FAQFinder (Burke et al. 1997, Lytinen and Tomuro 2002). The focus of our research has been primarily on Q-to-Q matching. In this chapter, we first describe the system's current Q-to-Q matching strategy and report its retrieval performance to date. Then we present our latest work on Q-to-A matching. Finally we discuss future research directions, including Q and A learning, that apply to FAQ-based question-answering systems in general.

14.2 Overview of FAQFinder: Q-to-Q Match

FAQFinder is a web_based natural language QA system. Currently the system uses a library of approximately 600 Usenet FAQ files as the knowledge base, and tries to find an answer to a user's question by matching it against the ques-

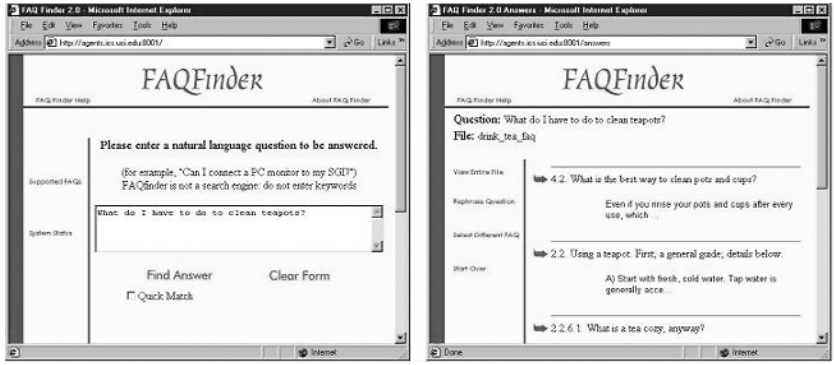

Figure 14.1. FAQFinder Initial Screen (Left Panel) and Five Best-Matching FAQ Questions (Right Panel)

tion part of the Q and A pairs in the FAQ files (i.e., FAQ questions). Figure 14.1 shows an example session with FAQFinder. After the user has typed a question (left panel), FAQFinder matches it with an FAQ question in two stages. In the first stage, the system displays FAQ files that are judged most likely to be relevant to the user's question. The SMART information retrieval system (Salton 1971) is used to select these files. Then in the second stage, after the user chooses one of the FAQ files from this list, (up to) 5 best-matching FAQ questions from that file are displayed, with their answers (right panel).

The second stage of processing uses a combination of four metrics to judge the similarity between a user question and an FAQ question. These metrics measure different aspects/dimensions of the similarity of the two questions; thus, they are complementary to each other. The metrics are (1) term vector similarity, (2) coverage, (3) semantic similarity, and (4) question type similarity. Each metric is normalized to produce a value of between 0 and 1, where 1 indicates the strongest similarity. In the current version of FAQFinder, overall similarity is computed by averaging the four metrics together. Below we describe the four metrics in detail.

14.2.1 Vector, Coverage, and Semantic Similarity

The first metric, term vector similarity, is computed using tf·idf (Salton and McGill 1983), a standard IR measure. In the context of Q_to_Q match, each (user or FAQ) question is considered a "document," and represented by a vector of word/term frequencies (after some closed_class terms are discarded using a stop list, as is standard in IR). The similarity score is computed as the cosine of two term vectors.

The second metric, coverage, is computed as the percentage of words in the user question that also appear in the FAQ question. This metric is intended to

ensure that a matched FAQ question is covering the important concepts/keywords mentioned in the user question.

The third metric, semantic similarity, is computed by using the semantic distances between all terms in the user question and all terms in the FAQ question. To obtain a semantic distance between two terms, FAQFinder utilizes the WordNet (Miller 1990) hierarchy, and finds the shortest path between WordNet concepts (called synsets) referred to by the terms through hypernym/hyponym links. Detailed description of the computation of this metric is found in Lytinen, Tomuro, and Repede (2000).

14.2.2 Question Type Similarity

The fourth metric used in FAQFinder is question type similarity. This metric is based on a set of 12 question types (shown below) which we had developed in our previous work (Tomuro and Lytinen 2001). Notice our question types are quite general — since FAQFinder is a general Q and A system, we need a comprehensive set of question types that cover a more general class of questions.

1. DEF (definition)	2. REF (refernce)
3. TME (time)	4. LOC (location)
5. ENT (entity)	6. RSN (reason)
7. PRC (rpcedure)	8. MNR (manner
9. DEG (degree)	10. ART (atrans)
11. INT (interval)	12. YNQ (yes-no)

Our question types were defined based on the paraphrasing patterns of interrogatives. For instance, types PRC and MNR both include "how" questions, such as "How should I store beer?" (PRC) and "How did the solar system form?" (MNR). Even the meanings of "how" in these sentences are the same: "In what manner or way" (*Webster's Collegiate Dictionary,* sense 1 of "how"). However, some of the paraphrasing patterns for PRC questions do not apply to MNR questions. For example, "What do I have to do to store beer?" * "What did the solar system have to do to form?"

Also, we defined a type ATR (ATRANS in conceptual dependency [Schank 1973]) as a special case of PRC. 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 rephrasings which ask for the 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?"

The question type similarity is computed by comparing the question types of user and FAQ questions. Note this metric looks at original questions, before the stop_list is applied. The question type for a question is determined by automatically classifying it using C5.0: the commercial version of the C4.5 decision tree induction system (Quinlan 1994); available at http://www.rulequest.com). In our previous work, we had trained C5.0 with 7637 questions selected randomly from 150 FAQ files and produced a decision tree that clas-

sifies a question according to the 12 question types. The classification accuracy for the training data was 85.3%. We will use this tree in the evaluation of FAQFinder, which we describe in the next section.

14.3 Evaluation of FAQFinder

To test the performance of FAQFinder, and to measure the effect of the four similarity metrics on the system's overall performance, we developed two test sets of user questions. The first test set consisted of a set of paraphrases of some FAQ questions. To gather paraphrases, we randomly chose 35 questions of various question types from the FAQ files and posted them on a Web site. Visitors to the site were shown an arbitrary sampling of those questions and were asked to rephrase them. After leaving the site on the Web for a period of 2 weeks, we had gathered a set of 679 example questions. The second test set consisted of a set of 153 questions typed by FAQFinder users, arbitrarily chosen from the system server logs.

We ran FAQFinder on both test sets. For the paraphrase test set, since the 679 questions in the test set were all generated from the original 35 FAQ questions, the correct match for each paraphrase was the original FAQ question from which it was generated. For the FAQFinder log test set, we manually inspected the FAQ files and determined which (if any) was the best matching question.

We evaluated the system performance by examining the trade_off between recall and rejection for varying threshold values. Recall is defined as the percentage of answerable user questions (i.e., questions in the test sets whose answers exist in some FAQ file) for which the system returned their matching/correct FAQ questions in the top 5 matches (after the threshold cut_off). Rejection is defined as the percentage of unanswerable user questions (i.e., questions in the test sets for which no answer exists in any FAQ file) for which the system displays no matches to the user (again after the threshold cut_off). We used rejection rather than the standard precision metric because it is a more pure measure of system performance on unanswerable questions (precision also measures the number of irrelevant answers (up to 5) returned for answerable questions). The FAQFinder log test set contained 91 answerable and 62 unanswerable questions. As for the paraphrase test set, since there were no unanswerable questions, we measured rejection by running the system on the same set of 679 test questions with the correct FAQ question removed.

Figure 14.2 shows FAQFinder's performance, as well as the performance of each individual similarity metric, on the two test sets. As the figure illustrates, the combination of the four similarity metrics produces better performance than do the individual metrics. In the case of the paraphrase test set, the dif-

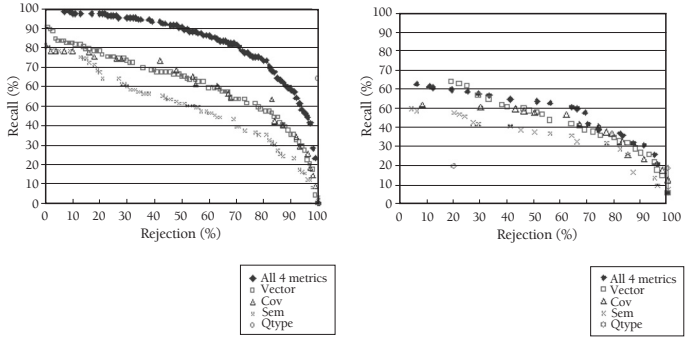

Figure 14.2. Ablation Study for Paraphrase Test Set (Left Panel); Ablation Study FAQ Finder Log Test Set (Right Panel)

ference is dramatic, showing a synergistic effect of combining complimentary metrics. Also, overall recall is quite high, remaining above 90% for rejection rates as high as 50%. However in the case of the FAQFinder log test set, the effect of combining measures is rather marginal. An immediate reason for this would be the way in which the paraphrase test set was originally generated; when asked to write paraphrases of FAQ questions, our web site users often used many of the same or similar words as the original questions, thus the paraphrases were naturally more similar than questions typed in by actual FAQFinder users.[2] This fact is also reflected on the difference of the general system performance: for instance, at 0% rejection recall is almost perfect (99%) for the paraphrase test set, whereas it is only 62% for the FAQFinder log test set.

To gain insights on how we could improve the retrieval performance, we inspected the failed matches (i.e., test questions to which the correct FAQ questions were not ranked within the top 5 matches). There were 10 and 35 such instances in the two test sets respectively. Upon inspection, we discovered that there were roughly three types of errors: (1) difficult paraphrasing, (2) unexpected match, and (3) other errors. Table 14.2 shows the breakdown.

Difficult paraphrasing includes cases where the paraphrasing pattern/relation between a user question and its correct FAQ question is difficult to identify. For example, the correct FAQ question to a user question "Is CF fatal?" is "Do people die from CF?" (where CF stands for "chronic fatigue"). Those two questions are obviously paraphrases of each other, but the system failed to recognize them because it currently does not encode phrasal paraphrases (e.g., "*X* is fatal" ("people die from *X*"). Even if we try to match single words "fatal" and "die," our current WordNet_based semantic similarity does not facilitate inferences between words of different part_of_speech categories. Other paraphrase patterns are more difficult and subtle, and require inferences based on

Error type	Paraphrase	FAQFinder log
A. Difficult paraphrasing	10 (100%)	6 (17%)
B. Unexpected match	0 (0%)	21 (60%)
C. Other errors	0 (0%)	8 (23%)
Total	10 (100%)	35 (100%)

Table 14.2. Types of Errors in Failed Matches

deep semantic or common_sense knowledge, such as "How can I separate my credit history from my ex_husband's?" vs. "How do I take my ex_husband's debts off of my credit report?" Although the meanings of those questions may not be exactly identical, they are close enough (i.e., near_paraphrases) such that we wish to be able to recognize them by Q_to_Q matching (without resorting to a search through the entire FAQ file for the actual answer). To do so, we must enhance our current Q_to_Q matching scheme, in particular the semantic inference capability. In future work, we are planning to incorporate a large database of paraphrase patterns described in Lin and Pantel (2001) and examine its applicability to our data.

Unexpected matches, on the other hand, include cases where the user question and FAQ question are dissimilar or even unrelated (thus they are not paraphrases of each other) but the answer happens to be contained in the FAQ answer part. A good example would be a user question "Is the picture quality of satellite TV identical to cable or broadcast TV?" In the Satellite_TV_FAQ, the answer to this question is found in the answer part of the question "Who should own a satellite system?" But the system retrieved other FAQ questions which were more (lexically) similar. Another example is "Is caffeine linked to high blood pressure or asthma?"-answered in the caffeine_faq, under the question "What happens when you overdose?" In those cases, it is not possible to tell if the user's questions are really answered by those FAQ questions by looking at the questions alone (even with some semantic inferences). In other words, the problem of unexpected match is a limitation of Q_to_Q matching.

Lastly, other errors included preprocessing errors, such as those by tokenizer and part_of_speech tagger, and incorrect answer keys assigned by the human coder.

14.4 Q-to-A Match

In addition to Q_to_Q match, we are currently investigating the possibility of matching a user question directly with the answer part of the Q and A pairs. Since the answer part of a Q and A pair most often contains more information/words than the question part, it is more indicative of what the Q and A is about and may allow us to do more precise matching. Q_to_A matching

would be also useful when Q_to_Q match fails-in the case of unexpected matches discussed in the previous section.

The Q_to_A match in FAQFinder is fundamentally the same as finding answers from unstructured documents-by looking at each FAQ answer block as a document, except that, since FAQ answers are "canned answers," the system does not in principle attempt to pinpoint exact answers. So for narrow specific questions, most of the answer_finding techniques developed so far in question_answering research such as that done by some TREC QA track systems (Harabagiu, Pasca, and Maiorano, 2000, Hovy et al. 2001) would apply directly. However, large proportions of questions entered by the FAQFinder users are complex and open_ended (as shown earlier in table 14.1), therefore we must develop different novel techniques to deal with long answers. While finding answers to specific questions often relies heavily on the semantic categories of the words in user questions and potential answers (e.g., MONEY, TIME), it seems indicative that features for complex questions include meta_linguistic features (e.g., the number and style of the answer paragraph(s)) and domain_independent words (e.g., "step", "because"). To test our hypothesis, we conducted a preliminary experiment where we selected several such features (which hereafter we call "meta_linguistic features") and observed the retrieval performance for "how_to" questions. The result we obtained, as described in section 14.4.1, showed significant improvements by the use of meta_linguistic features.

14.4.1 Meta_Linguistic Features

To develop our intuitions about "how_to" answers, we first randomly selected a set of 253 Q and A pairs from various FAQ files, consisting of 123 Q and A's of type PRC (procedure) and 130 non_PRC Q and A's. Note that, in the current experiment, we tested only PRC questions to examine the potential of our approach; we plan to examine questions of other types in future work.

After inspecting the data, we (manually) decided on the following six features as good indicators of a PRC answer block: (1) Number of paragraphs, (2) Number of sentences, (3) Number of imperative sentences, (4) Number of occurrences of PRC keywords (e.g., "step," "way," "first"), (5) Number of occurrences of non_PRC keywords (e.g., "because," "yes," "no"), (6) Number of occurrences of "if."

Then we constructed a decision tree using C5.0 from those 253 PRC answers. All six features appeared in the tree, and the classification accuracy was 90.1%.

14.4.2 The Experiment

Using the six meta_linguistic features, we conducted an experiment to see how much those features could help improve the retrieval of "how_to" questions. We chose term vector similarity (tf·idf) as the baseline measure for

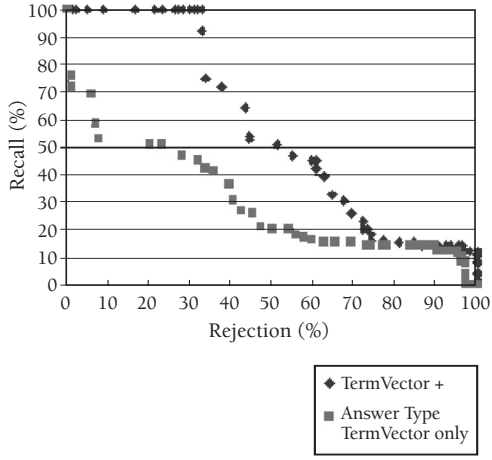

Figure 14.3. Recall vs. Rejection for Paraphrase Dataset

matching a user question with an FAQ answer, and compared the system's performance with and without using the additional measure of automatic answer type classification (how_to vs. non_how_to). Note that, here in the context of Q_to_A match, the two term vectors compared are a paraphrased test question and an FAQ answer instead of an FAQ question.

For the test set, we used a subset of the paraphrase dataset we used in the Q_to_Q match experiment, consisting of 117 questions of type PRC. This set is essentially the held_out test set (thus, the system was trained on one set of Q and A's and tested with another set of totally unseen Q and A's). The answer type classification is determined by running the C5.0 decision tree classifier obtained in the last section. The classifier returns a value of either 1 (for a how_to answer) or 0 (for a non_how_to answer). Then, the overall similarity between a test question and an FAQ answer is computed as a weighted sum of the two measures, where in the current experiment, we used the distribution of 90% on term vector similarity and 10% on answer type classification. The combined score would be between 0 and 1, where 1 indicates the strongest similarity.

Figure 14.3 shows the recall vs. rejection curves. Rejection was computed in the same way as Q_to_Q match, by removing the correct answers. As the figure shows, the additional use of answer type classification showed a dramatic improvement. Although the current experiment is still preliminary and only tests for how_to questions/answers, we consider this result to be an excellent indicator for the potential usefulness of meta_linguistic features in retrieving long answers. In future work, we are planning to identify features for other question types that require long answers, and conduct a comprehensive end_to_end retrieval experiment using both Q_to_Q and Q_to_A matches.

14.5 Future Directions

There are several interesting future research directions for question answering with FAQ files. We list some of them below.

14.5.1 Integrated Measure for Question Paraphrase Recognition

The current Q_to_Q similarity measure used in FAQFinder is a combination of four independent metrics. Although the metrics are additive and complementary to each other, they cannot capture the relations and interactions between them. For instance, when a question "What is the best way to clean teapots?" is pre_processed, the word "way" is used in the computation of vector, semantic, and coverage similarities, and thus is considered a content-word. But with respect to question type, this word serves as a predictor for type PRC (and ATR), and is considered a non_content word.

In an effort to develop an alternative, integrated similarity measure, we have recently defined a set of question paraphrase patterns (Tomuro 2003) as the first attempt. Those patterns are intended to specify and remove syntactic superstructures imposed by interrogative words/phrases (such as "How do I .." and "What is the best way to .."), and represent questions canonically using semantic case frames. A preliminary investigation, although with a small sample data, showed a quite promising result. We are currently investigating ways to combine them with paraphrase patterns of content words (along the lines of (Lin and Pantel 2001) which we mentioned in the evaluation of Q_to_Q match). An interesting alternative would be to take a less knowledge_intensive approach, for instance by keyword expansion and prioritized matching (Sneider 1999).

14.5.2 Passage_Based Answer Retrieval

Another important direction is to investigate how to retrieve a part of an FAQ answer for long answers. In our Q-to-A experiment, a whole FAQ answer was taken as one answer unit. But typically FAQ answers give long explanations and discuss surrounding issues. Each of those parts can be an answer to a different question by itself. This issue is also addressed in Chapter 13 in this book, as well as Light et al. (2003).

To retrieve long segments in FAQ answers, the system must incorporate a passage_based retrieval scheme. To that end, we anticipate cue/focus phrases as well as meta_linguistic features will be effective. For baseline measures, statistical approaches used in Berger and Mittal (2000) (in which a passage in an FAQ file is viewed as a summary of the file relative to a given query) seem a good starting point.

14.5.3 Learning of Answer Formulation

Aside from FAQFinder, FAQ files are also an excellent resource for learning

models for query_relevant answer formulation. Since FAQ files are written by domain experts, FAQ answers can be used as model answers from which we can develop criteria for the level and extent of expertise to be found in automatically extracted or generated answers/summaries. The structures of answer content/argument can also be learned from FAQ answers. For instance, an answer to a why question generally starts out with explanations of the reason(s), then discusses ways to fix the problem(s), and finally concludes with tips and advice. We anticipate the techniques developed for the passage_based answer retrieval can be utilized to automatically identify such structures.

14.6 Conclusions

FAQ files can be a rich source of information for question answering. The number and range of FAQ files is large and growing. While an FAQ_based QA system such as FAQFinder is limited to providing answers which have already been written by human experts, we can capitalize on the semi_structured nature of FAQ files to retrieve answers efficiently by answer re_use, and to find potential answers that may not be otherwise found by narrowly_scoped question_answering techniques. For example, the "how_to" and "why" types of questions that FAQFinder is able to answer are beyond the scope of the sorts of questions which are included in the TREC QA track.

The majority of our work on FAQFinder has been on refining the Q_to_Q matching techniques. Our preliminary work on Q_to_A matching shows good promise, especially for the types of questions which tend to have long, complex answers. We are hopeful that the full addition of Q_to_A matching will effectively complement Q_to_Q matching and help improve the system's performance.

Notes

1. Yes/no questions are often indirect requests for other types of information, and a good answer is rarely a simple yes/no response (Searle 1975).

2. On this, Lin and Pantel (2001) make a comment on manually generated paraphrases (vs. 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. Our paraphrase data is described in detail in Tomuro (2003).

Noriko Tomuro (tomuro@cs.depaul.edu) is an assistant professor at DePaul University's School of Computer Science, Telecommunications, and Information Systems. Her research interests include natural language processing and question-answering systems.

Steven Lytinen (lytinen@cs.depaul.edu) is a professor at DePaul University's School of Computer Science, Telecommunications, and Information Systems. His research areas are natural language processing and intelligent information retrieval.