Knowledge-based information retrieval from semi-structured text

Robin D. Burke, Kristian J. Hammond & Edwin Cooper
Artificial Intelligence Laboratory
University of Chicago
1100 E. 58th St., Chicago, IL 60637
{burke, kris, cooper}@cs.uchicago.edu

Abstract

This paper describes FAQ FINDER, a natural language question-answering system that uses files of frequently-asked questions as its knowledge base. Unlike AI question-answering systems that focus on the generation of new answers, FAQ FINDER retrieves existing ones found in frequently-asked question files. Unlike information retrieval approaches that rely on a purely lexical metric of similarity between query and document, FAQ FINDER uses a semantic knowledge base (WordNet) and natural language processing techniques to improve its ability to match question and answer.

We describe an evaluation of the system's performance against a corpus of user questions, and show that a combination of techniques from information retrieval and natural language processing works better than any single approach.

Introduction

In the vast information space of the Internet, individuals and small groups have created small pockets of order, organized around their particular interests and hobbies. For the most part those involved in building these information oases have been happy to make their work freely available to the general public. One of the most outstanding examples of this phenomenon can be found in the vast assortment of frequently-asked question (FAQ) files, many associated with USENET newsgroups.

The idea behind a FAQ file is to record the consensus of opinion among a group on some common question and make that answer available, particularly to newcomers to the group who may otherwise ask the same questions again and again. For this reason, most FAQs are periodically posted on the newsgroups to which they are relevant. This information distribution mechanism works well for individuals who are sufficiently interested in a topic to subscribe to its newsgroup, but not for casual users, who might have a question about table saws, but not want to read dozens of messages a day about woodworking.

The aim of the FAQ FINDER project to construct a question-answering system that extends further the intent of the FAQ file phenomenon. The system is an information service, available on the World-Wide Web, to which users can pose their questions. If the question happens to be one of the frequently-asked ones whose answer has been recorded in a FAQ file, FAQ FINDER will return the appropriate answer. This paper describes the different components of FAQ FINDER and demonstrates the operation of the system. It also shows some preliminary results from an evaluation of FAQ FINDER that we performed with a small set of FAQ files and a corpus of questions gathered from users.

The power of our approach rises out of two features: We are using knowledge sources that have already been designed to "answer" the commonly asked questions in a domain and as such are more highly organized than free text. We do not need our systems to actually comprehend the queries they receive (Lang, et al. 1992) or to generate new text that explain the answer (Souther, et .al. 1989). They only have to identify the files that are relevant to the query and then match against the segments of text that are used to organize the files themselves (e.g., questions, section headings, key words, etc.).

The most natural kind of interface to a database of answers is the question, stated in natural language (Ogden, 1988). While the general problem of understanding questions stated in natural language remains open, we believe that the simpler task of matching questions to corresponding question/answer pairs is feasible and practical.

The FAQ Finder system

The operation of FAQ FINDER is relatively simple for the user. The first step is to narrow the search to a single FAQ file likely to contain an answer to the

user's question. The choice of file is confirmed by the user. The FAQ file is considered to be a set of natural language question/answer pairs. The user inputs a question also in natural language. Once a file has been identified, the second stage is to match each question in the file against the user's question to find the ones that best match it and return those as possible answers.

FAQ FINDER uses standard information retrieval technology, the SMART information retrieval system (Buckley, 1985), to perform the initial step of narrowing the focus to one particular file. The user's question is treated as a query to be matched against the library of FAQ files. SMART stems all of the words in the query and removes those on its stop list. It then forms a term vector from the query, which can be matched against similar vectors already created for the FAQ files in an off-line indexing step. The top-ranked files from this procedure are returned to the user for selection.

For example, suppose the user enters the following question: "Is downshifting a good way to slow down my car?" as shown in Figure 1.

The system will pass the question to the SMART retriever and get back a list of files ranked by their relevance to the question. In this case, FAQ FINDER returns "the Automobile Consumer's FAQ" as the most relevant file.

The heart of FAQ FINDER is in its question matching process. Each question from the FAQ file is matched against the user's question and scored. We use three metrics in combination to arrive at a score for each question/answer pair: a term-vector comparison, a semantic similarity score, and a comparison of question type.

The idea behind using the term-vector metric is to allow the system to judge the overall similarity of the user's question and the question/answer pair, taking into account the frequency of occurrence of different terms within the file as a whole. This metric does not require any understanding of the text, a good thing because the answers in FAQ files are free natural language text, often quite lengthy. To create the term-vector comparison, the system performs an operation similar to what SMART does in retrieving the FAQ file. Each question/answer pair is turned into a vector of terms, weighted by the terms' distinctiveness within the FAQ file, calculated using the standard TFIDF method (Salton & McGill, 1983). A similar process is performed for the user's question. The metric used to compare the term vectors is also standard one in information retrieval: the cosine of the angle between the vectors.

This method works surprisingly well (see evaluation discussion below) in spite of the fact that TFIDF is not considered to be a useful technique for small collections and small queries. The problem with the term-vector comparison is its narrowness. It does not take into account the meaning of words, relying instead on the global statistical properties of large documents and large queries to ensure that relevant terms will appear. FAQ FINDER on the other hand deals with small queries and small "documents" – the individual question-answer pairs in each file. If the user asks "How do I get my ex-wife's name off of my credit history?" and there are two similar questions in the file: "How do I get my ex-husband off of my credit history?", and "How do I get my bad debts off of my credit history?", a term-vector comparison will rate the two questions as the same. It is unaware of the semantic relationship between "ex-husband" and "ex-wife" namely that they are both "ex-spouses." To enable this type of match to succeed, FAQ FINDER uses a semantic network of words, WordNet (Miller, 1995). Words are associated by related meanings, so "ex-husband" and "ex-wife" are both related to "ex-spouse."

However, we do not use WordNet as a thesaurus for undirected query expansion: augmenting the term-vector with all known synonyms for each term. Even a simple noun such as "name" in the above example can be either a noun and a verb and has different meanings in these cases. Simple expansion created too many spurious matches in our experiments. Instead, we decided to use WordNet to create a separate semantic similarity metric for question-matching, which could be combined with the term-vector metric.

The semantic similarity metric is calculated by performing simple marker passing through WordNet's "hypernym" (essentially, isa) links. The problem of multiple meanings occurred here as well, a general problem in marker passing systems (Collins & Quillian, 1972). We dealt with this problem by making the marker passing dependent on an initial parsing step. The question is parsed and only those word senses compatible with the parse are allowed to be used in semantic matching. For example, a parse of the question "How do I get my ex-wife's name off of my credit history?" would mark "name" as being a noun, and verbal senses of the word would not need to be considered. Parsing greatly improved the efficiency and utility of marker passing.

The final metric used in question matching is the comparison of question type. FAQ FINDER has a taxonomy of question types as defined by their syntactic expression. For example, it would assign the type `Q-HOW` to the credit history question above. Questions with different question types would be penalized such as "Who keeps track of my credit history?" which has

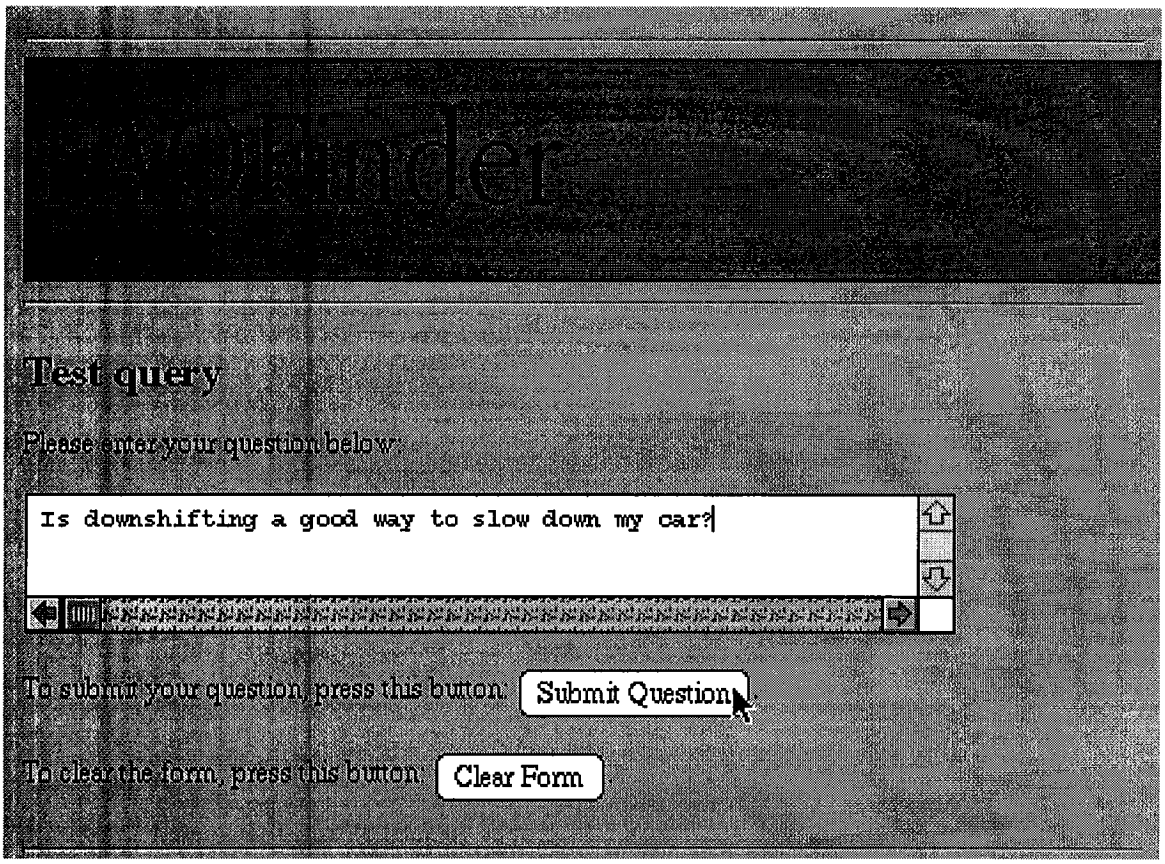

Figure 1: Submitting a question to FAQ FINDER.

the type Q-WHO. While this system helps keep FAQ FINDER from answering some questions inappropriately, we found that syntactic analysis alone was insufficient for identifying what was being asked in a question. For example, questions of the syntactic type Q-WHAT can be used to ask almost anything: "What is the right way to do X?" (should be Q-HOW), "What should I expect to pay for X?" (should be Q-COST), etc. We plan to integrate semantic information along the lines of (Lehnert, 1978) into the analysis of question type to improve its accuracy.

Figure 2 shows the result of matching the downshifting question against the Automobile Consumer's FAQ, a correct answer presented to the user.

The above description is an accurate high-level picture of the matching processes that underlie FAQ FINDER, but efficiency considerations led us to implement the system somewhat differently. We have attempted to do as much processing as possible off-line, so that only the analysis of the user's question and comparison with it need be performed while the user is waiting. As a first step, we put the parsing and analysis of the FAQ files in an off-line step. Auxiliary files are constructed for each FAQ file, containing a term-vector representation of each question/answer pair and a parsed version of the question.

Another significant efficiency gain was achieved by rebuilding the WordNet semantic network. WordNet, especially in its new 119,000 word incarnation, is too large to keep in core memory all at once. However, much of WordNet is unnecessary for our purposes. All FAQ FINDER needs is what words are linked to others via the "hypernym" link. We used WordNet to build a "tree" dictionary: associated with each word is a tree of hypernyms, for example the entry for "wife" in this hypernym dictionary is

```
(wife
  ((woman
     (female
       (person
         ((life_form (entity ()))
          (causal_agent (entity ()))))))
   (spouse
     (relative
       (person
         ((life_form (entity ()))
```

```
(causal_agent (entity ()))))))))
```

With these trees, the matter of marker passing is reduced to identifying the topmost level at which two such trees have a common element. As part of the pre-processing of FAQ files, we also record the index into the tree dictionary for each word in each question. Then, at run time, the only lookup that is required is for the words in the user's question. For the on-line version of FAQ FINDER, we have reduced the lookup overhead even further by caching hypernym trees for all of the words that occur in any FAQ file.

Evaluating FAQ Finder

We began evaluating FAQ FINDER by soliciting a corpus of questions on a range of topics from undergraduate students. Since the system has been brought online for local use, we have gathered more questions from actual use. The FAQ FINDER test suite now consists of 18 FAQ files drawn from the RTFM archive at MIT, 99 user questions for which there are answers in the FAQ files, and 72 questions that do not have answers.

In our earliest tests with 50 FAQ files, we discovered that SMART was extremely effective at identifying appropriate files. 77% of the time the correct file was listed first in relevance and 86% of the time the correct file could be found in the top five displayed to the user. While SMART's effectiveness remains to be tested on the full set of RTFM FAQ files (about 2500 files), we believe that it will be continue to be satisfactory.

Evaluation in FAQ FINDER is complicated by the fact that the task of the system is different than the information retrieval problem as it is typically posed. In Informational retrieval, the assumption is that there is a document collection in which there are some documents relevant to the users' query and it is the system's job to return as many of these relevant documents as possible. In FAQ FINDER, we are not interested in relevant answers – probably all answers in a FAQ file are somewhat relevant to the user's query – we want the system to return the right answer, where the right answer is defined as the information that best answers the user's question as it was posed.

This alternative stance has several important consequences for evaluating FAQ FINDER. Most significantly, we must try to identify questions that the system cannot answer. If there is no relevant document in a collection, it is considered acceptable for an IR system to return what is essentially garbage – the closest things it can find. This is not acceptable for a question-answering system, which should be able to say "Your question is not answered in this FAQ." One benefit of being able to make such a determination is that the system can collect unanswered questions and potentially have them answered and added to the FAQ by experts. To measure this property of the system, FAQ FINDER compute what we call the "rejection rate": the percentage of times that the system correctly asserts that it does not have the answer.

Precision and recall, the traditional evaluation metrics in IR, use a retrieved set of data and measure how much of what is retrieved is relevant and how much of what was relevant was retrieved. In our case, there is only one right answer in a FAQ, so if precision is non-zero, recall will always be 100% – in other words, if FAQ FINDER retrieves anything worthwhile, it will retrieve everything. Recall is therefore not a useful measure, and we do not compute it. For user interface reasons, we return a small fixed-size set of results to the user. Currently we return five items and if the correct answer is found, this would be considered, in IR terms, to be 20% precision. However, since there is only one correct answer, it is impossible for the system to do better than this. So, we consider each such retrieval to be a success, and compute the percentage of times that success is achieved.

Our two evaluation metrics therefore are success rate, the percent of questions for which FAQ FINDER returns the correct answer (when one exists), and rejection rate, which is the percent of questions that FAQ FINDER correctly reports as being unanswered in the file. We feel that the use of these metrics better reflects FAQ FINDER's real-world performance under its expected conditions of use than recall and precision would. As in the case of recall and precision, there is a trade-off between success rate and rejection rate. If the rejection threshold is set too high, some correct answers will be eliminated.

In our evaluation, The different matching techniques discussed above were tested independently, and then tested in combination using a linear weighted average. We found that WordNet matching and term-vector matching were approximately equal in power with WordNet alone having a success rate of 71% and term-vector matching a success rate of 78%. Better still, an equal parts combination of the two techniques yielded a phenomenal success rate of 86%. However, the rejection rates at this level success was unacceptably low, 3%, meaning the system could not tell the difference between good answers and bad ones.

Contrary to our earlier experiments with a smaller data set, question type turned out to contribute little to the matching equation, probably because it could not be reliably assigned for the reasons discussed above. We will continue to experiment with this part of the matching criteria. Another factor that we intro-

duced penalizing questions that failed to match against words in the user's question, did turn out to be beneficial, creating a reduction in success rate but substantial gains in rejection rate. We are still in the process of managing the trade-off between the different factors to achieve optimal behavior, but our best results to date are success rate 60%, rejection rate 51%, which was achieved with the following weights: 42% term-vector, 42% WordNet, and 16% unmatched words. These preliminary results are summarized in Figure .

We expect to have FAQ FINDER operational as a public web service in the Summer of 1996. One of the chief hurdles remaining is the large-scale tagging of FAQ files. FAQ FINDER needs to have the question/answer pairs labeled within each file in order to do its comparisons. The small corpus of files we have been using as a testbed were all tagged manually, a laborious task. To scale up the system, we are actively developing automated and semi-automated tagging tools that will recognize different types of FAQ files and parse them into quesion/answer pairs. See (Kulyukin, Hammond, & Burke, this workshop.)

Future Work

FAQ FINDER is built on four assumptions about FAQ files:

- All of the information in a FAQ file is organized in question/answer format.
- All of the information needed to determine the relevance of a question/answer pair can be found within that question/answer pair.
- The question half of the question/answer pair is the most relevant for determining the match to a user's question.
- Broad general knowledge is sufficient for question matching.

Unsurprisingly, we have found many instances in which these assumptions are violated. For example, FAQ writers frequently use headings to mark sections of their documents and rely on the reader's interpretation of those headings in their question writing. In the "Investment FAQ" file, the following text can be found:

```
Subject: Analysis - Technical:
...
    Q: Does it have any chance of working?
...
```

The "it" is of course intended to refer to technical analysis. However, FAQ FINDER is currently not capable of making use of this referent because it lies outside the question/answer pair. Part of our intent as we automate the tagging process is to make heading information available to the matcher.

There are other more difficult cases of ellipsis found in FAQ files. In the "Wide-Area Information Server FAQ," the following passage can be found:

```
Q: What is Z39.50?
A: ...
Q: Do they interoperate?
A: ...
```

The reference "they" refers to both Z39.50, an information retrieval standard, and WAIS, the subject of the FAQ. We do not expect FAQ FINDER to be able to dissect references that are this oblique. It would, however, be useful to refer back to earlier questions if there is no heading information with which to resolve a referent.

One FAQ-specific phenomenon we have encountered is the use of *metasyntactic variables*, meaningless pieces of text that stand in for a filler, which can vary. For example, the "Pool and Billiards FAQ" contains the question

```
Q: What are the rules for XXX?}
A: STRAIGHT POOL...
   EQUAL OFFENSE...
   NINE BALL...
```

Metasyntactic variables often have a distinct form and can be easily recognized. We anticipate that a mechanism similar to a heading recognizer could be used to recognize the sub-answers within a multi-part answer such as this. Not every variable can be so treated, however. The "Woodworking FAQ" contains the question

```
Q: Should I buy a Sears blurfl?
```

The answer does not enumerate the entire catalogue of Sears power tools: the same advice is intended to apply to all. The reader is supposed be capable of matching the nonsense word against the name of any power tool. This is exactly the type of domain-specific knowledge that we have sought to avoid including in FAQ FINDER. FAQ FINDER can successfully match this question against questions like "Are Sears power drills a good buy?" because the word "Sears" is sufficiently distinctive, but it would fail to match against a question like "What kind of power drill should I buy?"

The problem of domain-specific knowledge will probably surface in a more intractable form as we incorporate more FAQ files into the system. Many existing FAQs are technical; a large proportion address the quirks of hardware and operating systems. In these files, WordNet's general-purpose word knowledge will be less applicable. For example, WordNet does not

know about proper names that are common knowledge: A question like "What is the best GM car?" would not match against a question that mentioned Chrysler.

We are investigating several possibilities for capturing domain-specific semantic knowledge from FAQ files. One possibility is to use machine learning techniques, using as data reformulated questions posed by users whose initial question is not answered. If the user who originally asked about GM cars rephrases his or her question to ask specifically about Chrysler, the system might be able to postulate a relationship between the two terms. Such relationships could be treated as FAQ-specific extensions to WordNet. We are also considering various semi-automated approaches such as using word co-occurrence information to suggest candidate terms, and requiring a user to establish appropriate links.

Conclusion

We have described FAQ FINDER, a functioning knowledge-based information retrieval system, that relies on the knowledge engineering inherent in FAQ files distributed on the Internet. The system combines statistical measures, shallow lexical semantics, and natural language processing in matching users' questions against question/answer pair recorded in FAQ files. Our evaluations, conducted with a small subset of FAQs and small corpus of questions, have demonstrated the feasibility of the system, which we now intend to scale up into an information utility.

Ultimately, FAQ files are a social phenomenon, created by people to record and make public their understanding of a field. Our aim in FAQ FINDER is to further this goal by making the answers recorded in FAQs more widely available. Along with FAQ FINDER itself, we developing an interactive FAQ file maintenance and indexing tool that will allow a user to create FAQ files and to build the annotations required by FAQ FINDER. Our long-term aim is the distribution of this tool to individual FAQ maintainers.

References

Buckley, C. 1985. Implementation of the SMART Information Retrieval Retrieval [sic] System. Technical Report 85-686, Cornell University.

Collins, A. M. and Quillian, M. R. 1972. How to Make a Language User. In E. Tulving and W. Donaldson, *Organization of Memory.* New York: Academic Press.

Lang, K. L.; Graesser, A. C.; Dumais, S. T. and Kilman, D. 1992. Question Asking in Human-Computer Interfaces. In T. Lauer, E. Peacock and A. C. Graesser *Questions and Information Systems* (pp. 131-165). Hillsdale, NJ: Lawrence Erlbaum Assoc.

Lehnert, W. G. 1978. *The Process of Question Answering.* Hillsdale, NJ: Lawrence Erlbaum Assoc.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11).

Ogden, W. C. (1988). Using natural language interfaces. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 205-235). New York: North-Holland.

Salton, G., & McGill, M. 1983. *Introduction to modern information retrieval.* New York: McGraw-Hill.

Souther, A.; Acker, L.; Lester, J. and Porter, B. 1989. Using view types to generate explanations in intelligent tutoring systems. In *Proceedings of the Eleventh Annual conference of the Cognitive Science Society* (pp. 123-130). Hillsdale, NJ: Lawrence Erlbaum Assoc.

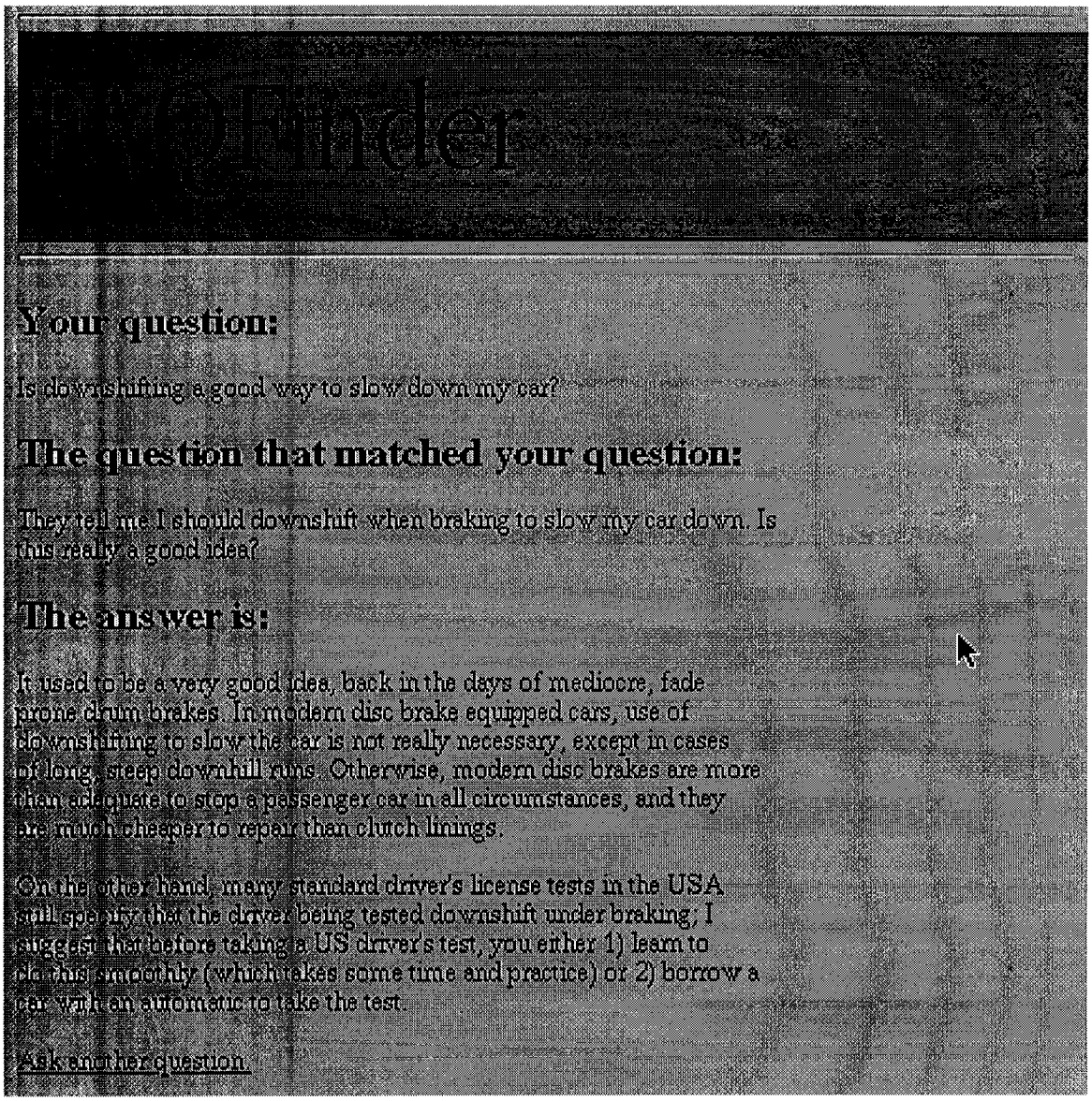

Figure 2: Answers returned by FAQ FINDER.

Matching technique	Success Rate	Rejection Rate
WordNet alone	71%	NA
Term vector alone	78%	NA
WordNet & Term vector	86%	3%
WordNet, Term vector & Unmatched	60%	51%

Figure 3: Preliminary FAQ FINDER evaluation results