FAQ finder: a case-based approach to knowledge navigation

Conference Paper · March 1995

DOI: 10.1109/CAIA.1995.378787 · Source: IEEE Xplore

Kristian Hammond
Northwestern University
173 PUBLICATIONS **4,237** CITATIONS
SEE PROFILE

Robin Burke
DePaul University
194 PUBLICATIONS **9,256** CITATIONS
SEE PROFILE

Steven L. Lytinen
DePaul University
49 PUBLICATIONS **799** CITATIONS
SEE PROFILE

FAQ Finder: A Case-Based Approach to Knowledge Navigation

Kristian Hammond, Robin Burke, Charles Martin
The Artificial Intelligence Laboratory
The University of Chicago
1100 East 58th Street, Chicago, IL 60637

Steven Lytinen
Department of Computer Science
Depaul University
Chicago, IL 60606

Information Infrastructure

One of the important prospects for the national information infrastructure is the wide-spread availability of on-line data and information services on a large scale. However, existing means of information access will not scale up to a network of this size. We believe that access to appropriate information, the ability to contact the right source at the right time, is the most significant obstacle to making a large-scale information infrastructure work.

We see two significant consequences of having a large pool of information distributed over wide-area networks.

- First, information is placed in publicly-accessible locations by individuals or organizations who have a stake in seeing that the information is correct, useful and accessible.
- Second, the sheer size and distribution of these resources means that users will have difficulty knowing where to look for answers to specific questions.

These consequences are already observable in today's largest, most widely-available electronic information service, USENET newsgroups. Because certain questions come up over and over in such groups, a mechanism has evolved to handle so-called "Frequently Asked Questions" or FAQs. News group contributors put together frequently asked questions and their answers into files called FAQ files, which reside at publicly-accessible sites. New readers of the newsgroup can get a copy of the FAQ file and look up answers to their questions without tying up network resources. Unfortunately, given a question, few users know which FAQ files to look for or where they might be found. This means that the information, while available in theory, is hidden from the more infrequent user.

This is not an isolated phenomenon. Commercial networks such as CompuServe and America On-Line are also producing structured archives similar to those on the Internet. They face the same problem, however, in that only fairly sophisticated users are able to navigate through the information space that results. Likewise, telecommunications companies such as Ameritech have developed systems such as TOUCH FOUR which organize information in question/answer formats that are maintained by outside interests such as advertisers and trade organizations.

We are developing a class of systems, called FAQ FINDER systems, that use a natural language question-based interface to access distributed text information sources, specifically text files organized as question/answer pairs such as FAQ files. In using these systems, a user will enter a question in natural language and the system will attempt to find an information source that answers the question, and then find the closest matching question/answer pair. These systems combine three technologies: statistically based IR engines, syntactic natural language analysis and semantic networks. In particular, they combine the SMART information retrieval system, a natural language parser based on the XEROX tagger, and a semantic net derived from Princeton's WORDNET.

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. 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. While the general problem of understanding questions stated in natural language remains open, we believe that the simpler task of matching questions to corresponding answers is feasible and practical.

As it stands, the FAQ FINDER project is, an automated question-answering system that uses the files of "Frequently-Asked Questions" (FAQs) associated with many USENET newsgroups [1]. These files are compendiums of the accumulated wisdom of the newsgroup on topics that are of frequent interest. FAQ FINDER takes a user's query on any topic, attempts to find the FAQ file most likely to yield an answer, searches within that file for similar questions, and returns the given answers.

In general, the FAQ FINDER idea is to use existing on-line resources of "questions asked" and "answers given" to provide a simple and natural interface between users and information networks. In particular, the goal of this project is to develop a FAQ FINDER system that will provide immediate natural language access to a large corpus of medical information.

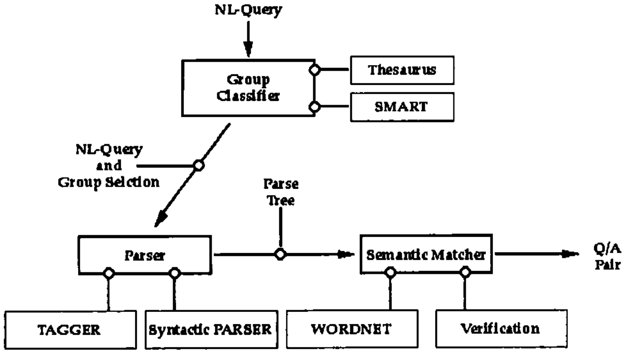


Figure 1: Flow of control through FAQ FINDER

The FAQ Finder system

The technology used to develop FAQ FINDER is fairly simple (See figure 1). We have combined three central technologies in constructing the system:

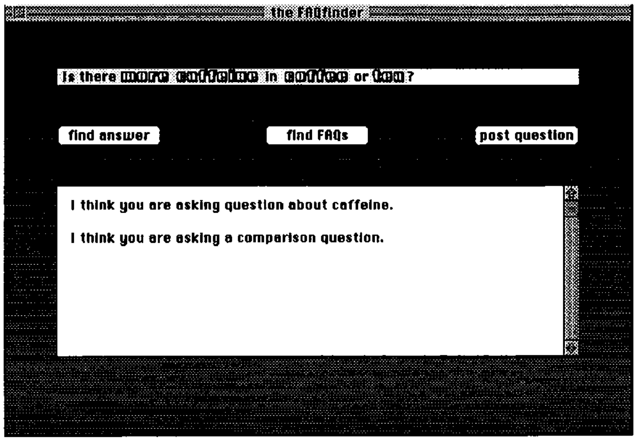
- Statistical information retrieval, embodied in SMART [2], is used to select FAQ files, given a particular question.
- Syntactic parsing, embodied in the Xerox tagger and a bottom up chart parser, is used to construct a simple parse tree and identify the primary verb and noun phrases in a question.
- Semantic concept matching, through the use of the WORDNET network of lexical semantics, is used to select possible matches between the query and target questions in the FAQ files.

At each stage of the process, the user is provided with some, though not all of the system's decision making. This allows the user to make adjustments to questions, select between answers, and, in case the system is unable to retrieve and answer, choose which News Group the query should be posted to.

For example, given a question such as:

Is there more caffeine in coffee or tea?

FAQ FINDER uses SMART to identify the **caffeine_FAQ** as a possible source of an answer. It provides the user with feedback in the form of highlighting the words used in the identification. Given alternative FAQ files, a user can either choose one directly or simply change the words in the question so as to redirect the selection of the FAQ (Figure 2).



Is there more caffeine in coffee or tea?

I think you are asking question about caffeine.

I think you are asking a comparison question.

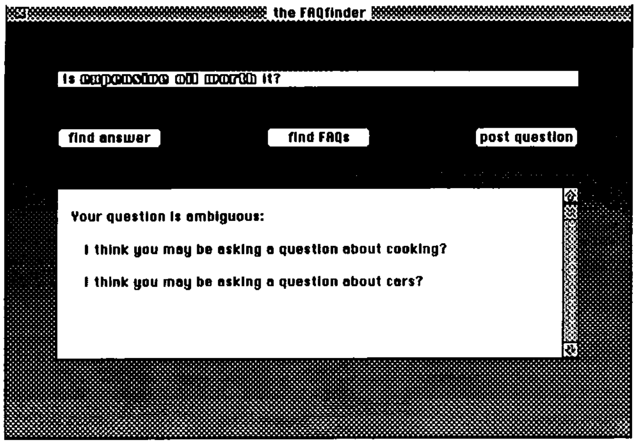
Figure 2: FAQ FINDER identifying a FAQ

On the other hand, given the question

Is expensive oil worth it?

FAQ FINDER finds plausible matches with both the **automotive_FAQ** and the **cooking_FAQ**, and highlights the words "expensive", "oil", and "worth" as being relevant (Figure 3). Given alternative FAQ files, a user can either choose one directly or simply change the words in the question so as to redirect the selection of the FAQ.

Once an appropriate FAQ file has been identified, the system parses the query into a syntax tree. The goal here is not to parse into an unambiguous structure, but instead to parse into a rough representation that can be used to support matching. Part of the "syntax" of the queries is the category of question type into which it falls. For example, questions beginning



Is expensive oil worth it?

Your question is ambiguous:

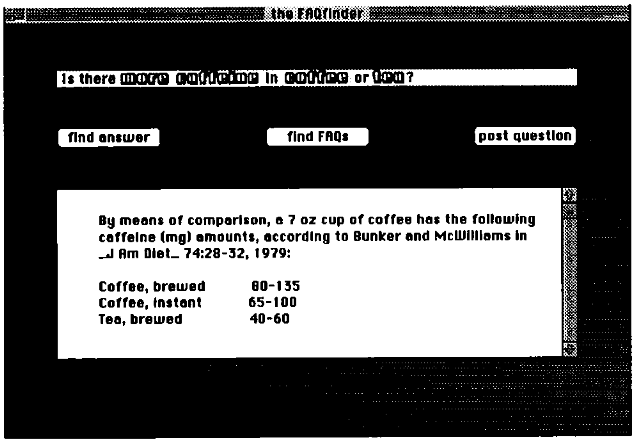
I think you may be asking a question about cooking?

I think you may be asking a question about cars?

Figure 3: FAQ Finder suggesting alternative FAQs and its reason for picking them.

with the phrase "What is the difference between..." tend to fall into the category **Q-COMPARISON**. This can be recognized during the syntactic parse.

As we will discuss in a later section, the resulting tree structure is then used to validate any matches that are found in the semantic processing. When a match is found, the matching question/answer pair is presented to the user (Figure 4). At every stage of this process, the user is given some control over disambiguation and selection of the final answer.



Is there more caffeine in coffee or tea?

By means of comparison, a 7 oz cup of coffee has the following caffeine (mg) amounts, according to Bunker and McWilliams in _J Am Diet_ 74:28-32, 1979:

Coffee, brewed	80-135
Coffee, instant	65-100
Tea, brewed	40-60

Figure 4: An answer found in the Caffeine_FAQ

If no answer is found, the user is given the option of posting the question to the News Group that the FAQ FINDER suggests. If the question is to be posted, it is handed to a FAQ manager that posts the question to the newsgroup and incorporates the resulting answer into the FAQ file. The FAQ FINDER project has shown that when there is an existing collection of questions and answers, as found in the FAQ files, question answering can be reduced to matching new questions against question/answer pairs. This is a considerably more tractable task than question understanding.

In general, the FAQ FINDER project is interesting in that it uses not just the existing archives on the InterNet, but also the existing sociology. One of the more powerful aspects of the new-groups is the collective desire on the part of the users to "get it right." This drive has already resulted in the FAQ files themselves, and the FAQ FINDER that extend beyond the range of the "most" frequently asked questions.

Technical issues

There are several research issues that remain for the creation of a useful question-based retriever. The process used in FAQ FINDER has three stages: first, a small set of relevant files is chosen from the library of possible files; second, syntactics parse of the query is constructed; and third, questions found in the files are matched against the user's question. Each of these phases has it's own issues.

Statistical IR

The statistical retrieval techniques that FAQ FINDER uses have been shown to be fairly accurate at locating sets of relevant files. The most significant problem is that of matching questions entered by the user against the questions in the file. The route taken by FAQ FINDER is to use question templates to assign a question class to each user question. The system has a set of answering strategies that are applicable to each question type. For example, the question "Is purified water better than tap water for my houseplants?" is a comparison question. FAQ FINDER tries to find questions in the file that belong to the same class and contain the same terms: a comparison question that contains "tap water" and "purified water." If there is no direct comparison question in the file, the system can look for descriptive questions for each of the terms separately.

Analysis of questions

Both user questions and the questions found in FAQ files are analyzed, using natural language processing techniques, in order to improve the matching done during retrieval. The analysis of questions are divided into several stages. First, the question is processed by a part-of-speech tagger. We are currently using the Xerox part-of-speech tagger, developed by Brill. The tagged text is then passed on to a context-free parser, which syntactically analyzes the question. The resulting parse tree is then used in two different ways: (a) to categorize the question according to a set of "question type" categories; and (b) to assign generic (i.e., nonspecific) semantic cases to noun phrases. Question-type and semantic case information are used in retrieval, to rank matches and/or to filter out irrelevant matches. Matches are ranked higher if question types are the same, and if matching keywords appear in the same semantic case in both the user question and the FAQ file question.

The parser which we are using is a simple bottom-up chart parser. We are currently designing a grammar for parsing a wide variety of questions, which uses the tags produced by the part-of-speech tagger. We are designing our own grammar so that we can include our question-types as nonterminal symbols in the grammar; thus, categorization of a question occurs automatically as a by-product of parsing. For example, the final grammar may include rules such as the following:

```
S            → Q-HOW-TO | Q-DEFINITION |...
Q-HOW-TO     → How do NP VP
Q-VERIFY     → Is NP NP | Does NP VP | ...
...
```

A parse of a question using this grammar would result in a category such as Q-HOW-TO, Q-VERIFICATION, etc., appearing in the parse tree.

Generic semantic cases are computed from the parse tree using a set of rules which map syntactic role to semantic case. For example, in a HOW-TO question, the NP directly after "How do" should be assigned the ACTOR/AGENT case, the direct object of the verb the OBJECT/PATIENT case, and so on. Objects of prepositions are assigned very general cases, so as to allow for variations in use of prepositions across questions. Different question types may require different mappings between syntactic and semantic roles.

The FAQ Finder Matcher

The matching algorithm is designed to handle variations in lexical content between input and FAQ questions. For example, consider the question:

> How do I reboot my computer after a power failure?

This question might be expressed in any of the following forms:

1. How do I reboot my computer after a power failure?
2. How do you restart the system after a crash?
3. What do I do after the system goes down?

Here, the difficulty is that there are many ways of expressing the same question, all using different words and phrases. The FAQ FINDER system needs a means of matching such synonymous but varied inputs against its FAQs. Since the similarity lies in the meaning of these questions, recognizing similarity and matching must make use of *knowledge representation.*

Knowledge representation is a classic AI endeavor. In the FAQ FINDER system, it is important to balance the depth of representation with the breadth of coverage. The goal of FAQ FINDER is to provide fast answers to an amazingly varied set of questions; deep causal reasoning about questions can be excluded because: (1) it would take too long, and (2) it would require too much knowledge engineering.

For FAQ FINDER, we believe that a *shallow lexical semantics* provides an ideal level of knowledge representation for the system. Such a semantics has three important advantages: it provides critical semantic relations between words; it does not require expensive computation to compute relations; and it is readily available.

The Wordnet system provides a level of shallow lexical semantics appropriate for FAQ FINDER. The Wordnet system provides a system of relations between words and "synonym sets," and between synonym sets themselves. The level of knowledge representation does not go much deeper than the words themselves, but there is an impressive coverage of basic lexical semantics.

The Wordnet database will provide the underlying semantic framework for the FAQ FINDER matcher. By using classical marker-passing algorithms, the FAQ FINDER system will use the Wordnet database to accept variations such as "husband" for "spouse."

Each FAQ is represented as a node in the Wordnet semantic space, with links to the lexical semantics of its components. Each link is annotated with markers as to the syntactic role of that component in the FAQ. For example, the FAQ "How do I reboot my computer after a power failure?" is represented by the following links:

	Lexicon	*Annotations*
	how	question type
	i	subject
Links:	reboot	verb; main action
	computer	direct object; genitive "my"
	failure	prep object; prep "after"; adj "power"; article "a"

Indices: how, reboot, computer, power, failure Note that some of the annotations are indices into the lexicon as well.

The matching algorithm

The matching algorithm we use is based on the classical marker-passing algorithms of Quillian. In Quillian's system, marker-passing in semantic space was used to identify candidate structures which were then compared to *form tests* to judge their linguistic accuracy. For example, the input phrase "lawyer's client" would cause *marker* data structures to be passed through a network from the **lawyer** and **client** concepts. One concept discovered by this search would be the **employment** concept, with the form test: "*first's* **second**". The form test verifies that the input actually was of the proper form to identify the **employment** concept.

From the point of view of FAQ FINDER, Quillian's basic algorithm had the particularly useful feature that it was *fast.* In FAQ FINDER, we use the indices of the FAQ representation given above to do initial identification of candidate FAQs. The marker-passing phase relies solely on the shallow lexical semantics of Wordnet; the annotations of each FAQ link are not taken into account. Because there is no checking to make sure that complex semantic or syntactic relations are satisfied, this marker-passing phase is very fast.

After identifying a candidate set of FAQs, each element of the set is rated heuristically on the basis of its link annotations. This phase of analysis is more time-consuming, but only operates on the relatively small set of candidate FAQs. After heuristic evaluation, the top candidates are passed on to the user interface for consideration.

We anticipate that adjustments to the heuristic evaluation function will be the primary means of improvement in the FAQ FINDER matcher. We are also considering various case-based learning schemes to improve the efficacy of the heuristic evaluation over time.

Depth and Breadth

Our current work in FAQ FINDER is aimed at providing a tool that can handle the full breadth of the InterNet. This would result in a tool that could handle the literally thousands of FAQ files that exist. Our conviction that this is possible is based on 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. 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.). We see this as a huge advantage in that the FAQ FINDER will be able to use existing on-line resources of "questions asked" and "answers given" to provide a simple and natural interface between users and information networks.

Bibilography

[1] Hammond, K.J.; Burke, R., and Schmitt, K. (1994). A Case-Based Approach to Knowledge Navigation. In AAAI Workshop on Knowledge Discovery in Databases. AAAI. August 1994. Seattle WA.

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

[3] 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.

[5] 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.

[6] Graesser, A. C.; Byrne, P. J. and Behrens, M. L. (1992). Answering Questions about Information in Databases. In T. Lauer, E. Peacock and A. C. Graesser *Questions and Information Systems* (pp. 131-165). Hillsdale, NJ: Lawrence Erlbaum Assoc.

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

[8] Martin, C. (1991). Direct Memory Access Parsing. PhD Thesis, Yale University.