# Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System

**Robin D. Burke, Kristian J. Hammond, & Vladimir Kulyukin**
Intelligent Information Laboratory, University of Chicago
1100 E. 58th St., Chicago, IL 60637
{burke, kris, kulyukin}@cs.uchicago.edu
**Steven L. Lytinen, Noriko Tomuro, & Scott Schoenberg**
School of Computer Science, DePaul University
243 S. Wabash, Chicago, IL 60604
{lytinen, cphdnt, sschoenb}@cs.depaul.edu

January 31, 1997

**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) to improve its ability to match question and answer. We include results from an evaluation of the system's performance, and show that a combination of semantic and statistical techniques works better than any single approach.

## Introduction

In the vast information space of the Internet, individuals and groups have created small pockets of order, organized around their particular interests and hobbies. For the most part those who build 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 wide assortment of frequently-asked question (FAQ) files, many associated with USENET newsgroups, which record the consensus of opinion among a group on some common question. So that newcomers do not ask the same questions again and again, 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 necessarily to those with a more transient interest. What is needed is a centralized means of access to these answers.

We believe that 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 question/answer pairs is feasible and practical. The aim of the FAQ FINDER project is to construct a question-answering system that extends further the aim and intent of the FAQ file phenomenon. The system is an information service, available on the World-Wide Web at `<URL:http://faqfinder.cs.uchicago.edu/>`. A user can pose a question and if it happens to be similar to one of the frequently-asked ones whose answer has been recorded in a FAQ file, the system will have a good chance of answering it.

FAQ FINDER is built on four assumptions about FAQ files. (1) QA format: All of the information in a FAQ file is organized in question/answer format; (2) locality of information: All of the information needed to determine the relevance of a question/answer pair can be found within that question/answer pair; (3) question relevance: The question half of the question/answer pair is the most relevant for determining the match to a user's question; (4) General knowledge: Broad, shallow knowledge of language is sufficient for question matching.

We see assumptions 1-3 as leading to an essentially case-based (Kolodner, 1993) view of the FAQ retrieval problem. A question/answer pair might be loosely considered a kind of case: it is a piece of knowledge that has been considered useful enough to be codified for reuse. The question serves as an index to the knowledge contained in the answer. These assumptions do not hold for all FAQ files, as we discuss below. but, they hold often enough to form a good starting point for research.

### Sample interaction

The following sequence of figures depicts a typical interaction with FAQ FINDER. Figure 1 shows the World-Wide Web interface to the system. Suppose the user enters the following question:

> Is downshifting a good way to slow down my car?

FAQ FINDER compares the question to its set of FAQ files, returning a list of files ranked by their relevance to the question, including in this case the file `auto_consumer_FAQ` as the most relevant file. Some files of questionable relevance are also retrieved, such as the `car_audio_FAQ`, a typical artifact of a keyword-based retrieval system. If the user chooses "Quick Match" when entering a question, the system will skip this first step of manual file selection and automatically choose the top-ranked file to match against.

When a FAQ file is chosen, the system iterates through the QA pairs in that file, comparing each against the user's question and computing a score. The best five matches are returned to the user along with the first line of the answer, so the validity of the answer can be easily determined. Figure 2 shows the results of this matching step comparing our "downshifting" question with the entries in the `auto_consumer_FAQ`. The right answer, a question about downshifting and braking is first, followed by two questions about brakes and two about tires. By selecting the appropriate link, the user can view the entire answer given in the FAQ file.

## How the system works

As the example shows, interaction with FAQ FINDER occurs in a series of stages. The first step is to narrow the search to a small set of FAQ files likely to contain an answer to the user's question. Second, each QA pair is matched against the user's question to find the ones that best match it.

For the first stage of processing, FAQ FINDER uses standard information retrieval technology, the public domain SMART information retrieval system (Buckley, 1985), to perform the initial step

Figure 1: Asking a question of FAQ Finder.

Figure 2: Choosing an answer.

of narrowing the focus to a small subset of the FAQ files. 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 of frequent words. It then forms a term vector from the query, which is 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. We have not found it necessary to tinker with the default configuration of SMART. We treat this part of the system as a black box that returns relevant files.

The second stage of processing in FAQ FINDER is a 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 statistical term-vector similarity score $t$, a semantic similarity score $s$, and a coverage score $c$. These metrics will be described in detail below. Overall match similarity $m$ is a weighted average

$$m = \frac{tT + sS + cC}{T + W + C}$$

where $T$, $S$, and $C$ are constant weights that adjust the reliance of the system on each metric.

The statistical similarity score at the QA pair level is computed in a manner similar to SMART's document matching. A QA pair is represented by a term vector, a sparse vector that associates a significance value with each term in the QA pair. The significance value that we use is commonly-known as *tfidf* (Salton & McGill, 1983). If $n$ is the term frequency (the number of times that a term appears in a QA pair), $m$ is the number of QA pairs that the term appears in the file, and $M$ is the number of QA pairs in the file, then *tfidf* is equal to $n \times \log(M/m)$. The idea behind this measure is to evaluate the relative rarity of a term within a space of documents, and use that as a factor to weight the frequency of that term in a particular document. A term that appears in every QA pair in a file is probably of little value, and its *idf*, or $\log(M/m)$ value, would correspondingly be zero. A term that appears in only a single QA pair would have the highest possible *idf* value. Term vectors for user questions are computed similarly by using the *idf* values associated with terms in a given FAQ. Term vectors are then compared using another standard information retrieval metric, the cosine of the angle between the vector representing the user's question and the vector representing the QA pair.

The term-vector metric allows 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 several paragraphs or more in length.

The *tfidf* measure has a reasonably long history in information retrieval, and has generally been thought to work best only on relatively lengthy documents. This is because only long documents have enough words for statistical comparisons to be considered meaningful. However, we have found that this metric contributes significantly to the overall performance of our matching algorithm (see evaluation discussion below), despite the extremely short length of the "documents" (QA pairs) involed in matching.

The semantic similarity metric enhances term-vector comparison by taking into account a very shallow level of semantic analysis of lexical items that appear in user and FAQ questions. The semantic matching algorithm in FAQ FINDER is designed to handle variations in lexical content between input and FAQ questions. For example, consider the following questions:

How can I get my ex-spouse's debts off my credit report?

Can I get credit if my ex-husband had a lot of debts?

Here, the difficulty is that there are many ways of expressing the same question, all using different (but semantically related) words and phrases. In large documents, these lexical variations might not affect term-vector comparison greatly, because over the course of the document a variety of synonymous terms might be used. However, in FAQ FINDER since matching is being performed on a very small number of terms, the system needs a means of matching such synonymous.

This suggests the need for a level of semantic analysis of user and FAQ questions. However, in the FAQ FINDER system, it is important to balance the depth of analysis with the breadth of coverage. Deep causal reasoning about questions would not be feasable because it would require too much knowledge engineering to cover all of the necessary areas of knowledge. For FAQ FINDER, we believe that a *shallow lexical semantics* provides an ideal level of analysis for the system. Such a semantics has three important advantages: (1) It provides critical semantic relations between words; (2) It does not require expensive computation to compute relations; and (3) It is readily available.

For example, since the consumer credit FAQ file is full of questions about credit reports and debts, it is important that the system identify the relation between "ex-spouse" and "ex-husband." This can be done at the level of the words themselves, hence our term "shallow lexical semantics." As an example of deeper semantics, we can consider the following pair of questions: (a) How do I reboot my system? (b) What do I do when my computer crashes?

Here, there is a causal relation between the question variants: rebooting is a causal consequence of having one's computer crash. In order to match these questions, the system would have to understand the causality of the computer domain. Since FAQ FINDER is intended to encompass the whole gamut of USENET topics, not just computers, it is impractical to expect even this simple level of domain-specific knowledge representation.

FAQ FINDER obtains its knowledge of shallow lexical semantics from WordNet, a semantic network of English words (Miller, 1995). 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 relations. By using a marker-passing algorithm (Quillian, 1968), the FAQ FINDER system uses the WordNet database to accept variations such as "ex-husband" for "ex-spouse." In particular, marker-passing is performed over the network's *synonym* and *hypernym* (i.e., *is-a*) links.[1]

Marker passing is performed to compare each word in the user's question with each word in the FAQ file question. Let $u_i$ be the $i$th word of the user's question. Let $f_j$ be the $j$th word of the FAQ file question. The similarity score $s$ of these two words is given by $s(u_i, f_j) = H - (p\frac{H-L}{D})$ where $p$ is the length of the path between $u_i$ and $f_j$ and $D$ is the maximum path length permitted by the system and $H$ and $L$ are constants. The score is in the range of $H$ and $L$, inclusive, and linearly inverse to the number of links traversed between the two words. Marking passing is not used for words that are identical or related by morphology: they are given fixed scores.

The matrix $S$ for a user question of length $n$ and a FAQ file question of length $m$ is an $n \times m$ matrix representing all possible comparisons of words in the two questions:

[1]Hypernym links are only provided in Wordnet for nouns and verbs; thus only synonym links are used for adjectives and adverbs.

$$S_{u,f} = \begin{bmatrix} s(u_1, f_1) \cdots s(u_1, f_m) \\ \vdots \\ s(u_n, f_1) \cdots s(u_n, f_m) \end{bmatrix}$$

$S$ is reduced to a single value, $w$, by choosing the maximum match score for each user question and then averaging these maxima for all words. The value $w(u, f)$ for semantic relatedness of the two questions $u$ and $f$ is given by

$$w(u, f) = \frac{\sum_{i=1}^{n} \max(s(u_i, f_0), \ldots s(u_i, f_m))}{n}$$

WordNet is not a single semantic network; separate networks exist for nouns, verbs, adjectives, and adverbs. Syntactically ambiguous lexical items, such as "run," which could be either a noun or a verb, appear in more than one network. We found that unrestricted marker passing, using all networks in which a term appears, led to too many spurious matches, a common problem in marker passing systems in general (Collins & Quillian, 1972). We tried several approaches to disambiguate terms to a single WordNet network, including using an existing part-of-speech tagger (Cutting, et al., 1992) and context-free parsing of questions, but in the end, we found that simply relying on the default (most common) word sense for each word worked as well as any of the more sophisticated techniques.

The third metric measures the degree of coverage of user terms by the FAQ question. The intuition behind this measure is to penalize questions that are lacking corresponding words for each word in the user's question. In other words, we do not care if the FAQ file question answers many questions at once, but we want to make sure that the important concepts in the user's question are covered. The coverage value $c$ is the percent of words in the user question that have a non-zero $s$ (computed in the semantic similary metric) for some term in the FAQ question.

## Evaluating FAQ Finder

We evaluated the performance of FAQ FINDER on a corpus of questions drawn from the log files of the system's use during the period May to December of 1996. A total of 241 test questions were used to perform the evaluation. We manually scanned each FAQ file for answers to each question, and determined that there were 138 questions that had answers in the FAQ file corpus, and 103 questions that were unanswered.

The most obvious precedents to FAQ FINDER are information retrieval systems, and standard information retrieval evaluation techniques are a starting point for the evaluation of the system. However, 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. Normally, the assumption is that there is a document collection in which there may be a number of documents relevant to the user's query. In contrast, FAQ FINDER works under the assumption that there is such a thing as a "right answer": a single FAQ QA pair that best addresses the user's question as it was posed. The system's job is to return that answer within the small fixed-size set of results that can be displayed on a single web page. Relevance is not that useful a measure to us because, within a given FAQ, most of the answers are probably somewhat relevant to the user's query.

Because FAQ FINDER's task is different, the traditional IR evaluation metrics of recall and precision must be modified somewhat. Recall normally is a measure of the percentage of relevant documents in the document set that are retrieved in response to a query, whereas precision is a measure of the percentage of retrieved documents that are relevant. In our case, however, since there is typically one right answer to be retrieved from a FAQ, these are not independent measures of performance. Assuming that an answer to a user question exists in a FAQ file and that the system returns 5 QA pairs, FAQ FINDER will perform at either 100% recall and 20% precision (if the answer is retrieved), or 0% recall and precision (if it is not). If no answer exists, then precision will be 0%, and recall is undefined.

To measure the quality of retrieval, then, we calculate our version of recall, which amounts to the percent of questions for which FAQ FINDER returns a correct answer when one exists. Our calculation is slightly different from traditional recall measures because it does not penalize the system if there is more than one right answer in the file. If there are several answers within a file that answer a user's question, it does not make sense to regard retrieval of only one of these answers as only partial success. If the user's question is answered, it is irrelevant that there was another QA pair that also answered it. Instead of precision, we calculate a value called *rejection*, the percentage of questions that FAQ FINDER correctly reports as being unanswered in the file. We feel that these metrics better reflect FAQ FINDER's real-world performance than traditional recall and precision.

Rejection is adjusted in FAQ FINDER by setting a cut-off point for the minimum allowable match score. As with precision, there is a trade-off between recall and rejection. If the rejection threshold is set too high, some correct answers will be eliminated; on the other hand, if the threshold is too low, then incorrect responses will often be given to the user when no answer exists in the FAQ file.

## Results

SMART is highly effective at the file retrieval task. The correct file appears 88% of the time within the top five files returned to the user, and 48% of the time in the first position. This translates to 88% recall and 23% precision.[2]

Figure 3 shows the recall vs. rejection results that we obtained for the second stages of the FAQ FINDER system. As the graph shows, rejection is somewhat low for reasonable values of recall, meaning that the system confidently returns garbage in most cases when there is no right answer in the file. If the rejection threshold is lowered to make it easier to identify questions without good answers, recall drops dramatically. However the top value for recall is encouraging: there is a better than a two-thirds probability that the system will find the right answer.

### Ablation study

Our next step was to evaluate the contribution of different components of the matching scheme through an ablation study. We selectively disabled different parts of the system and ran the same corpus of questions. There were four conditions: a *random* condition, in which QA pairs were selected randomly from each FAQ file; a *coverage only* condition, in which the coverage score for each question was used by itself; a *semantic score only* condition, in which only the semantic scores

[2]We do not use rejection in evaluating SMART's performance since it is a standard IR system, designed to find relevant documents, not answer questions.

Figure 3: Recall vs. rejection for FAQ Finder

Figure 4: Recall for ablation study

derived from WordNet were used in evaluating answers; and, a *statistical score only* case, in which the term vector comparison was used in isolation.

Figure 4 shows average recall results for these conditions. Interestingly, both WordNet and our statistical technique are contributing strongly to system performance. These two methods had very similar average recall, but are clearly not equivalent measures, since their combination yields results that are better than either individually. These results confirmed our earlier results with a small corpus of questions, which showed an even more dramatic benefit from the combination of methods.

Figure 5, which shows the recall vs. rejection analysis for these conditions, provides even more evidence that the two measures differ. The curve for the semantic scoring condition is particularly striking. Although recall in this condition is weaker than the system as a whole, this metric shows good rejection performance. This suggests that the application of semantic information might be used specifically to improve rejection.

Figure 5: Recall vs. rejection for ablation study

## Discussion

A preliminary analysis of the failure cases, questions for which the system failed to find answers, suggests that the biggest culprit is usually undue weight given to semantically useless words. For example, a question such as “where can I find woodworking plans for a futon?” retrieves questions that incorporate the word “woodworking” as strongly as those that contain “futon”, even though “futon” should be a much more informative term inside the `woodworking_FAQ` than “woodworking”, which applies to everything. The problem is that the term “woodworking” does not appear that often in the FAQ despite its close semantic relation to words that do appear.

Another type of problem commonly encountered with FAQ FINDER is related to violations of the assumptions about FAQ files discussed at the beginning of this paper: the assumptions of question/answer format, locality of information, question relevance, and sufficiency of general knowledge. 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, making it more difficult to match against a question like “Does technical analysis work?”[3] 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?
```

[3]The statistical component of the matcher may enable the question to be retrieved anyway, but we would prefer that the semantic component also be able to contribute in this situation.

```
A: ...
Q: Do they interoperate?
```

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...
```

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?"

## Future Work

The previous discussion suggests many areas that deserve future research attention. One of the most obvious open questions is the problem of improving the system's rejection of unanswerable questions. We have concentrated our tuning of the system to maximize recall, so that answerable questions will be answered. However, it is also useful to be informed that an answer does not exist within a FAQ file. This may suggest to the user that the question should be submitted to the FAQ's related newsgroup.[4]

One way of approaching this problem is to focus on the small retrieved set of QA pairs before they are returned to the user. We know from our evaluation that if an answer is present in the FAQ file, the system is likely to find it. Therefore if none of the QA pairs returned by the system are in fact a good answer, the chances are that the system should report that the answer does not exist. We also know that semantic information seems to have better rejection characteristics than statistical information. We may be able to perform a more in-depth analysis, involving deeper natural language processing to accept or reject each returned set of questions. Because this set is by definition small, such intensive processing would not be as computationally prohibitive as performing deeper natural language processing throughout the entire matching process.

[4] If rejection were sufficiently good, we could incorporate such an option into the system itself.

An important part of maintaining the performance of FAQ FINDER on a large set of FAQ files will be the incorporation of new vocabulary items into WordNet. Since WordNet was formed from a corpus of everyday English, its vocabulary does not include many technical terms or proper nouns. Unfortunately, due to the technical nature of many FAQ files, technical terms and proper nouns constitute a significant portion of the domain vocabulary of these files. In addition, these terms can be the most useful in retrieving relevant FAQ question/answer pairs, since they are often the most specific and discriminating terms. Thus, the fact that they are missing from WordNet can significantly impair the performance of FAQ FINDER.

We are investigating ways in which information obtained from the parses of questions can be used to automatically acquire additional terms and build the appropriate synonym and hypernym links for these terms in one of the WordNet hierarchies. We will rely on feedback from the user to tell the system when a good match has been found between a user question and a FAQ question/answer pair.[5] If the user indicates the system retrieved a relevant answer, then any words in either the user or the FAQ question that are not contained in WordNet have the potential to be acquired. The system will then attempt to match the unknown word with a synonym in the other question. Both questions will be parsed, and position in the parse tree used to determine which words are candidate synonyms of the unknown word.

Since the matching process between question pairs is likely to incorrectly propose some synonyms of unknown words, our approach is to collect synonyms for unknown words over time, and propose new WordNet entries by analyzing collections of possible synonyms for each unknown term. Clustering algorithms are likely to be of use here in determining the likely best entry(ies).

## Conclusion

FAQ FINDER is a Web-accessible knowledge-based information access system that relies on the knowledge engineering inherent in FAQ files distributed on the Internet. The FAQ FINDER project has shown that when there is an existing collection of questions and answers, question answering can be reduced to matching new questions against question/answer pairs, a considerably more tractable task than question understanding. The system combines statistical measures and shallow lexical semantics to match users' questions against question/answer pairs from FAQ files. Our evaluation, although conducted with a small corpus of questions, have demonstrated the effectiveness of the system.

The power of our approach rises from the fact that we are using highly organized knowledge sources that are designed to "answer" the commonly asked questions. 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.

Ultimately, FAQ files are a social phenomenon, created by people to record and make public their understanding of a field. 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 newsgroups is the collective desire on the part of the users to "get it right." This drive has already resulted in the existence FAQ files themselves. Our aim in FAQ FINDER is to further this goal by making the answers recorded in FAQ files more widely available.

[5]Using our previous local version of FAQ FINDER, about 20% of users gave this type of feedback. We expect that the improved interface of FAQ FINDER 2.0 will increase the frequency of user feedback on performance of the system.

**Acknowledgments**

In addition to the authors, the development team for FAQ FINDER 2.0 consisted of Jay Budzik and Benjamin Young of the University of Chicago; and David Mortman of Commonwealth Edison. Previous versions of the system benefited from contributions by Edwin Cooper, Xiaobin Fu, Julia Kozlovsky, Charles Martin, and Tom McDougal. Research on FAQ FINDER has been supported with funds from the Office of Naval Research and with hardware donations from Apple Computer.

## References

Cutting, D., Kupiec, J., Pederson, J., & Sibun, P. 1992. A Practical Part-of-Speech Tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing.* ACL.

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

Burke, R., Hammond, K., & Cooper, E. 1996. Knowledge-based information retrieval from semi-structured text. In *AAAI Workshop on Internet-based Information Systems*, pp. 9-15. AAAI.

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.

Grady Ward, 1993. *Moby Part-of-Speech II.* Computer file. Arcata, CA: Grady Ward.

Kolodner, J. 1993. *Case-Based Reasoning.* San Mateo, CA: Morgan Kaufmann.

Kulyukin, V., Hammond, K. & Burke, R. 1996. Automated analysis of structured on-line documents. In *AAAI Workshop on Internet-based Information Systems*, pp. 78-86. AAAI, 1996.

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.

Lenhert, W. 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).

Quillian, M. R. 1968. Semantic Memeory. In *Semantic Information Processing*, Marvin Minsky, ed., pp. 216-270. Cambridge, MA: MIT Press.

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.