Natural Language Processing in the FAQ Finder System: Results and Prospects

Robin Burke, Kristian Hammond & Vladimir Kulyukin
Intelligent Information Laboratory, University of Chicago
1100 E. 58th St., Chicago, IL 60637
Steven Lytinen, Noriko Tomuro & Scott Schoenberg
School of Computer Science, DePaul University
243 S. Wabash, Chicago, IL 60604

Abstract

This paper describes some recent results regarding the employment of natural language processing techniques in the FAQ FINDER system. FAQ FINDER is 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.

FAQ FINDER uses a combination of statistical and natural language techniques to match user questions against known question/answer pairs from FAQ files. We strove in our experiments to identify the contribution of these techniques to the overall success of the system. One unexpected result was that our parsing technique was not contributing as much to the system's performance as we expected.

We discuss some of the reasons why this may have been the case, and describe further natural language processing research designed to address the system's current needs.

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 a user with a casual interest, who might have a specific question about table saws, but not want to read dozens of messages a day about woodworking.

The aim of the FAQ FINDER project is 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 questions. FAQ FINDER answers a user question by searching FAQ files for a similar question. If a successful match is found between a user question and a FAQ question, then the answer supplied in the FAQ file is returned to the user.

Several of the components of FAQ FINDER which match user and FAQ questions have been discussed in some detail in other publications (Burke, Hammond & Cooper, 1996; Hammond, Burke, Martin & Lytinen, 1995). Here we will concentrate on the use of simple NLP techniques in FAQ FINDER, the results we have obtained with the system, and what new directions these results suggest.

The FAQ FINDER system

Input to FAQ FINDER is a question, typed by the user in natural language. The first step for the system is to narrow the search for a matching FAQ question to a single FAQ file. This is done using the SMART information retrieval system (Buckley, 1985). The user question is treated as a set of index terms to be matched against the overall content of individual FAQ files. The system presents the user with a list of FAQ files judged by SMART as most relevant, and the user selects one of these FAQ files for the system to search further.

Once a single FAQ file has been selected, a more complex matching process is used to match and score each individual question in the FAQ file against the user's question. The system uses three metrics in combination to arrive at a score for each question/answer pair: a statistical comparison, a semantic similarity score, and a coverage score, the number of words in the user's question that have some semantic match in the question from the file.

Statistical similarity

The statistical similarity score is computed in a manner quite similar to SMART's document matching. A question-answer (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*, which stands for term frequency times log of inverse document frequency (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 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 vector representing the user's question and the vector representing the QA pair.

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, and typically several paragraphs in length.

The *tfidf* measure has a reasonably long history in information retrieval and has fairly well-understood properties. In particular, it is generally accepted that the metric works best when queries and documents are lengthy. Only long documents have enough words for statistical comparisons to be considered meaningful. Although the question-answer pairs in FAQ FINDER are much shorter than text in typical IR corpora, the term-vector comparison still works well enough for our purposes in FAQ FINDER (see evaluation discussion below), especially when augmented with semantic similarity assessment.

Semantic similarity

We found that statistical matching of questions contributed significantly to FAQ FINDER's retrieval, but statistics alone were not enough to achieve the performance we desired. Term-vector comparison suffers from its inability to 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. The semantic matching algorithm in FAQ FINDER is designed to handle variations in lexical content between input and FAQ questions, variations that might appear naturally in larger corpora. 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 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 for a quick web interaction, and (2) 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 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.

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 is the main association between the two question variants. The fact that an ex-husband is an ex-spouse belongs to the lexical semantics of the two words "ex-husband" and "ex-spouse." This is the level of semantics we wish to capture in the FAQ FINDER system. We call this a shallow lexical semantics, since it is associated directly with the words.

As an example of deeper semantics, we can consider the following pair of questions:

How do I reboot my system?
What do I do when my computer crashes?

Here, there is a causal relation between the question variants: rebooting is a causal consequent 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.

Shallow lexical semantics in WordNet

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.

The WordNet database provides the underlying framework for the FAQ FINDER semantic matcher. By using classical marker-passing algorithms, the FAQ FINDER system uses the WordNet database to accept variations such as "ex-husband" for "ex-spouse." In particular, we rely on the network's *hypernym* links, which function as *is-a* relationships for nouns and verbs, and the synonym links for adjectives and adverbs. Thus, "ex-husband" and "ex-wife" can be judged to be semantically related, through the common hypernym "ex-spouse."

The matching algorithm we use is based on the classical marker-passing algorithms of Quillian (1968). 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 are interested mainly in semantic relatedness and not on preserving constraints. The marker-passing phase relies solely on the shallow lexical semantics of WordNet; the relationships between words are not verified. Because there is no checking to make sure that complex semantic or syntactic relations are satisfied and we have a fixed length of marker propagation, this marker-passing phase is polynomial in the branching factor of the WordNet network.

Score computation

For the purposes of comparing questions, we are interested in arriving at a similarity score using the marker passing technique. Given two questions, we want to know how closely related they are. We have designed FAQ FINDER as a modular system so that many scoring functions can be tested. At the moment, we are using a scoring technique that involves building a matrix of scores and then reducing that matrix to a single number.

The first step in this metric is the word-by-word comparison of questions. 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 ith word of the user's question. Let f_j be the jth 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. H and L are constants that define the range of s. The score is therefore in the range of H and L, inclusive, and linearly inverse to the number of links traversed between the two words.

For example, "ex-wife" and "ex-husband" have a total of two links between them, up from "ex-wife" to "ex-spouse" and then down to "ex-husband." Under the system default match scoring scheme, their match score is 0.24. "Ex-husband" and "ex-spouse" which are separated by a single link have a score of 0.32. Words that are identical or a morphological variants 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:

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

This matrix 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

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

We compute a third measure, c, the degree of coverage of 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 is the percent of words in u that have a non-zero s for some f_j.

Finally, we combine all of the scores: the term vector similarity value t, the semantic similarity value w, and the coverage value c in a weighted average to arrive at a overall match score m.

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

where T, W, and C are constant weights associated with term vector, WordNet and coverage scores, respectively. Adjusting these weights adjusts the reliance of the system on each component of the match score.

Initial Experiments

Before the creation of the current version of FAQ FINDER, the system went through many different incarnations as we tested matching techniques. The most important areas of experimentation were our experiments with managing the marker passing process and our attempts to make use of abstract question-type as an input to the matching process.

Restricting marker passing

WordNet is not a single semantic network; separate networks exist for nouns, verbs, adjectives, and adverbs. Syntactically ambiguous lexical items, such as "name," 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. Our first attempt was to use an existing part-of-speech tagger, the Xerox Tagger (Cutting, et al., 1992). This system uses a hidden Markov model learned from a large corpus of English text to statistically determine the most likely sense for any given word in the context of a sentence. The system worked well in many cases, but it had a significant drawback in that it is not robust in the face of unknown words. By default, it assumes that unknown words are nouns. Since unknown words are unlikely to appear in WordNet anyway, this is not an immediate problem, however the tagger uses its determination that the unknown word is a noun to influence the rest of its tagging. For example, the word "reboot" is unknown to the tagger. When tagging the sentence "How do I reboot faster?" the system marks "reboot" as a noun and cannot then mark "faster" as an adverb because there is no verb nearby. "Faster" is marked as a noun, presumably meaning a person who fasts, a wildly inappropriate word sense in this context. Because FAQ files contain technical vocabulary from many different fields, and therefore many terms that are unknown to the tagger, we found we could not use this method for disambiguation.

Our next attempt at identifying unambiguous part-of-speech information was to use natural language parsing. We built a simple context-free grammar for questions, and implemented it in a bottom-up chart parser. The parser's lexicon was compiled from a combination of the on-line Oxford English dictionary and the Moby part-of-speech dictionary (Grady Ward, 1994). A successful parse would resolve syntactic category ambiguities of individual lexical items in a question (e.g., whether "name" in the above example is a noun or a verb). If the parser finds more than one parse for a question, the parses are ranked according to word sense frequency information from the on-line dictionaries, and the top parse selected.

Analysis of the effects of parsing on FAQ FINDER performance indicated that the desired goal of improving semantic matching using WordNet was not achieved. We compared a version of the system using parsing with one in which lexical items which appear in the on-line dictionaries are always tagged according to their default (most frequent) word sense. Thus, parsing could, in theory, improve system performance by correctly tagging words whenever it correctly tags a word with its non-default syntactic category, and additionally by correctly identifying which WordNet tree to search for lexical items which are not in the on-line dictionaries. Although tagging does not cause the system to match a FAQ question that otherwise would not be, recall could be improved if rejecting a FAQ question resulted in the correct question/answer pair being promoted to the "top five" list returned to the user. Actual analysis, however, indicates that parsing currently did not have a significant effect.

There are several possible reasons for why this is the case. First, although the parser found a parse for 80% of user questions, it is possible that the parser sometimes finds the wrong parse, and on those occasions mistags words. If this were happening, however, we would expect parsing to actually decrease performance, rather than having no significant effect. A more likely explanation is that default word sense tagging of user questions turns out to be highly accurate; thus, parsing is not significantly improving the tagging of lexical items. Because parsing is computationally expensive and did not seem to be a significant contributor to the system's performance, our current version of the program simply uses the default part-of-speech information in the on-line dictionary for disambiguation.

Parsing for question type

Another area where we devoted significant research attention was the issue of question type. Our intuition was that abstract categories of questions could be a useful part of the matching equation. For example, a "temporal" question from the user should probably match against a "temporal" question in a file as opposed to a "how to" or "where" question on the same topic.

We used the same parser that was employed for lexical disambiguation to determine question type. The grammar included question-types as nonterminal symbols in the grammar; thus, categorization of a question occurred automatically as a by-product of parsing. The grammar included nonterminals for `Q-WHAT`, `Q-HOW`, `Q-ADVICE`, `Q-COST`, `Q-LOC`, `Q-TIME`, `Q-WHY`, `Q-YES-OR-NO` and other subtypes of these question types.

What became immediately obvious from our implementation, however, was that syntactic information was insufficient to determine abstract question type with any reliability. The question "How often should I change the oil on a shovelhead Harley?" can be easily identified as having the `Q-TIME` type by the phrase "How often," but the same question can be easily rephrased into a different syntactic type without changing its meaning "What is the proper oil change interval for a shovelhead Harley?" Our parser would categorize this as a different syntactic question type, `Q-WHAT`. The conclusion of our evaluation was that assessment of question type by purely syntactic means did not contribute to successful matching. It is an open question what other techniques might be employed to more accurately assess semantic question type, possibly using categories such as Lehnert's (1978). We are continuing to hold this option open as an area for future research.

Current Status

As of this writing, the FAQ FINDER is implemented in approximately 9000 lines of Allegro Common Lisp. Another 1000 lines of Lisp code is devoted to the generation of HTML for the system's web interface. The program runs under the CL-HTTP server (Mallery, 1994). Currently the system incorporates 187 FAQ files, but we are in the process of building the system's FAQ file library, aiming to cover all of the files in the USENET FAQ file archive[1] that are in question-answer format, approximately 1000 files. The system will be available publicly as a web utility in the Spring of 1997.

We have evaluated the performance of FAQ FINDER on a corpus of questions drawn from the log files of the system's local use at the University of Chicago 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. There are hundreds more questions in the system logs awaiting incorporation into our testing regime.

Evaluation metrics

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 users' query. It is the system's job to return as many of these relevant documents as possible. In contrast, FAQ FINDER works under the assumption that there is such a thing as a "right answer": an answer 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, all the answers are probably somewhat relevant to the user's query.

Because of the differences in the FAQ FINDER task, 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

---

[1] ftp://rtfm.mit.edu

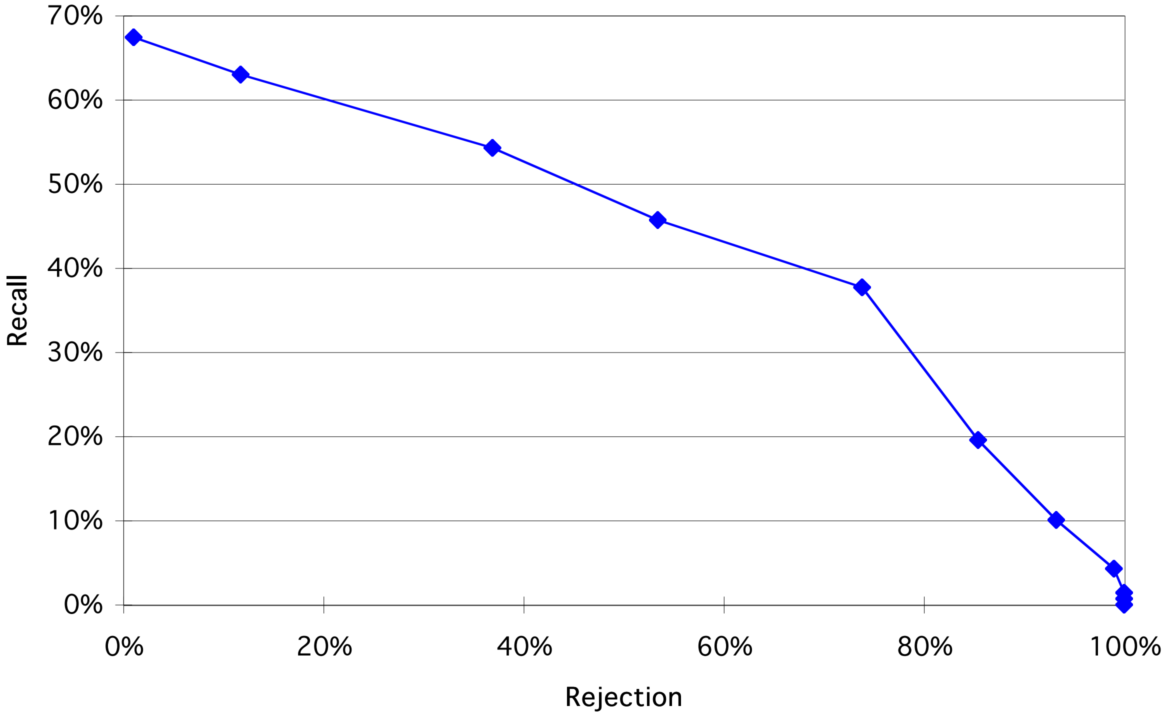

Figure 1: Recall vs. rejection for FAQ Finder

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, 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 a variant 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 would.

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 rate. 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 garbage 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. Even when tested against the entire FAQ file corpus, 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. Using standard IR metrics, it would stand at 88% recall and 23% precision.[2] Note that this step is the only part of the system directly affected by scale, since the question-matching step is always performed within the scope of a single FAQ file regardless of the size of the corpus. The remaining tests described below were performed with an 187-file subset of the FAQ file archive.

Figure 1 shows the recall vs. rejection results that we obtained for FAQ Finder's question matching component. As the graph shows, rejection is disappointingly 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

---

[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.

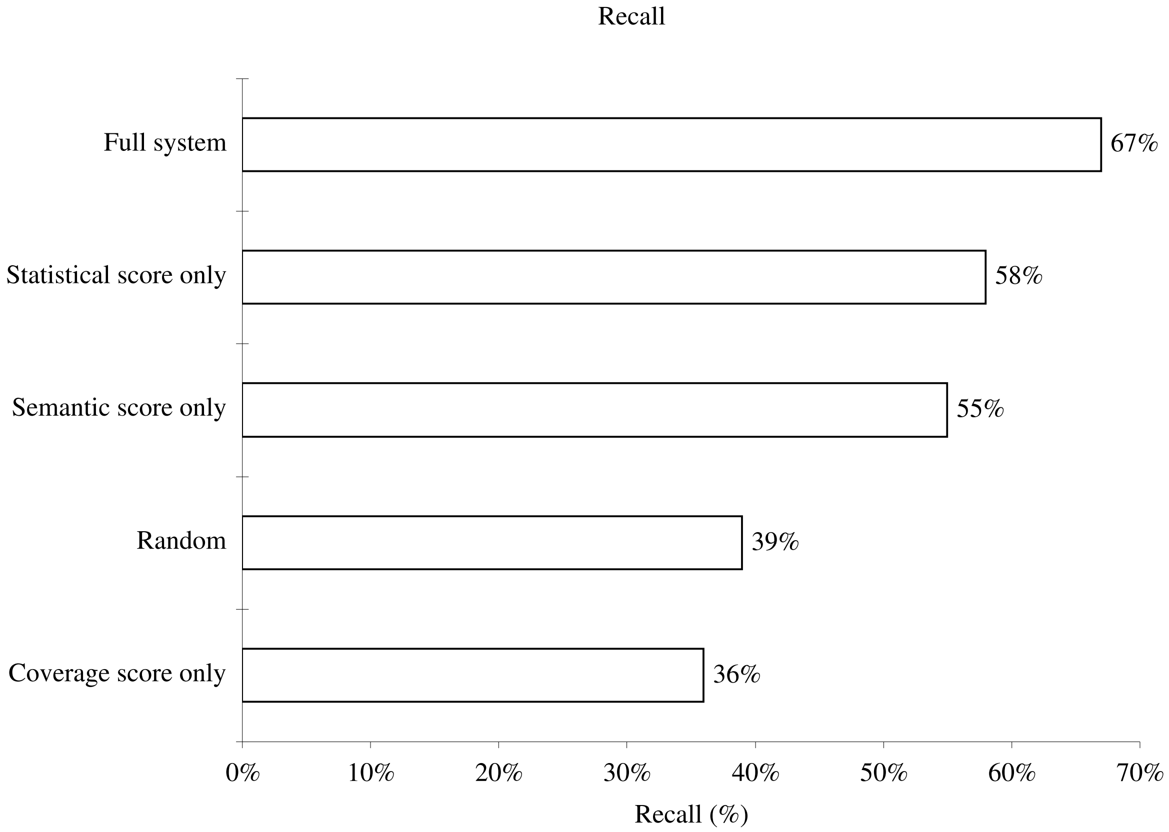

Figure 2: Recall for ablation study

threshold is lowered to make it easier to identify questions without good answers, recall drops. However the top value for recall is encouraging: better than a two-thirds probability that the system will find the right answer when one exists.

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 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 2 shows average recall results for these conditions. Interestingly, both WordNet and our statistical technique are contributing strongly to system performance. Coverage score, which is an extremely weak measure in isolation, turned out even worse than selecting questions randomly. Semantic scoring and statistical scoring 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 to the combination of methods.

Figure 3, which shows the recall vs. rejection analysis for these conditions, has even more evidence for the difference between the two measures. 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.

Future Work

Given the success of the system in its current form, our primary research goal is to improve the system's rejection results. Poor rejection occurs when the system retrieves answers with high ratings even though they do not actually answer the user's question. There are two ways to improve rejection:

1. Improve the accuracy of the scoring method so that good answers are always scored more highly, or

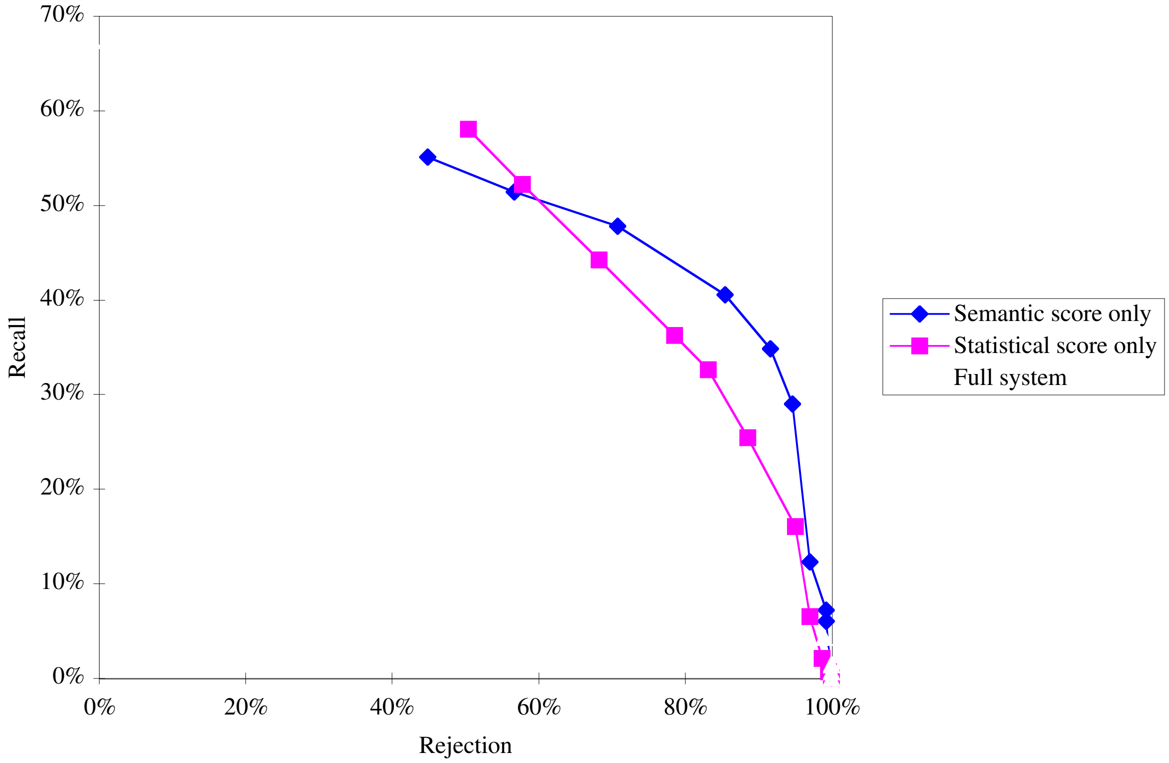

Figure 3: Recall vs. rejection for ablation study

2. Perform some post-retrieval analysis to verify the system's retrieval.

We are continuing to experiment with FAQ FINDER's scoring mechanism, hoping to improve both rejection and recall. However, in any system with a statistical component, a certain amount of noise is unavoidable.

We believe that deeper semantic analysis for verification (option 2) is a promising research direction. We know from our results so far that the semantic analysis we do has better rejection characteristics than the system as a whole. Also, it is possible to perform deeper and more complex natural language processing for verification than it is for retrieval, since a verification step would be applied only to a small set of candidates, rather than the whole FAQ file.

The parsing technique described above, which did not contribute significantly to the retrieval step, might be useful in filtering retrieval results. In particular, the position of matched terms in a parse tree could be used to further evaluate the scoring of the term-vector and semantic matching components. It might make sense to increase the score of a match between two terms that appear in similar positions in a parse tree, and to discount matches of terms that are in dissimilar positions. For example, if two matched nouns both appear in their respective parse trees as direct objects of semantically similar verbs, this suggests a good match between the terms; if, on the other hand, the verbs are semantically dissimilar, or if syntactic roles of the matched nouns are different, the system may consider removing the FAQ question from the final list. If all of the returned questions fare poorly when scrutinized more deeply, FAQ FINDER can conclude that the entire set is spurious and that a null result should be returned instead.

Another type of analysis that could contribute to a verification step is a more robust categorization of the type of question the user has asked, and the types of questions the system has returned. Our earlier experiments showed that simple syntactic recognition of question type was insufficient, but it may be worth applying a more comprehensive semantic approach to parsing for question type at verification time.

Natural Language Learning

One indirect result of scaling FAQ FINDER up to include a much larger corpus of FAQ files is that we will incorporate more technical FAQ files. Our initial sample of hand-processed FAQ files was skewed towards non-technical topics. As a result, we believe that we will begin to see a degradation in the contribution of WordNet to our matching algorithm. The reason is that WordNet was formed from a corpus of everyday English, and does not incorporate technical terms or proper nouns. For many domains, technical terms and proper nouns constitute a significant portion of the domain vocabulary. In addition, these terms can be the most useful in retrieving relevant FAQ question/answer pairs. Thus, the fact that these terms 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 by the system to automatically acquire additional terms and build the appropriate synonym and hypernym links for these terms in one of the WordNet hierarchies. We 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.[3] If the user indicates the system retrieved the right 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 attempts to match the unknown word with a synonym in the other question. The parse tree is used to determine likely synonyms. As with the use of syntactic information to enhance matching discussed above, position in the parse tree is used to determine which words are candidate synonyms of the unknown word.

Consider the following example. Say the user asks the question, "How can I recover an unsaved Word file if my Macintosh crashes?" Let us assume that "Macintosh" is not encoded in WordNet. If FAQ FINDER retrieves question such as "Is it possible to recover unsaved files when my computer crashes," and the user indicates that this is a good match, then the parse trees of the user and FAQ questions can indicate that "Macintosh" and "computer" are likely to be synonyms, since both appear as the subject of the verb "crashes". This suggests that "Macintosh" should be entered in WordNet as a synonym or hyponym of "computer".[4]

Since the matching process between question pairs is likely to incorrectly propose some synonyms of unknown words, our plan 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) for an unknown word. Proposed synonyms which are "outliers" from a cluster of other proposed synonyms could be discarded as probably incorrect, based on a clustering analysis. In addition, ambiguous unknown words could be detected by finding more than one cluster of synonyms. For example, for the word "Macintosh", our system might collect a list of synonyms which include terms such as "computer," "pc," "workstation," "apple," and "fruit." This would suggest two clusters, corresponding to the two possible meanings of "Macintosh."

Acknowledgments

The FAQ FINDER projects has benefited from the contributions of many individuals. Other contributors at the University of Chicago include Charles Martin, Edwin Cooper, Julia Kozlovksy, Kass Schmitt, and Benjamin Young. The natural language learning component of the system is being developed by Tom Carlton of DePaul University.

References

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.

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.

Grady Ward. 1994. *Moby Part-of-Speech II* (data file). Grady Ward, 3449 Martha Ct., Arcata CA.

Hammond, K., Burke, R., Martin, C. & Lytinen, S, 1995. FAQ Finder: A Case-Based Approach to Knowledge Navigation. In *Proceedings of the Eleventh Conference on Artificial Intelligence for Applications.* Los Angeles: IEEE Computer Society 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 An-*

---

[3] About 20% of the users give this type of feedback.

[4] Another approach to the vocabulary learning problem involves more user feedback — users can be asked directly to rephrase questions that contain unknown words. Such feedback would be more reliable than looking for synonyms in matched questions. We are testing user response to this feature in the current version of FAQ FINDER.

swering. Hillsdale, NJ: Lawrence Erlbaum Assoc.

Mallery, J. C. 1994. A Common LISP Hypermedia Server. In *Proceedings of The First International Conference on The World-Wide Web*, Geneva: CERN.

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.

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

Salton, Gerard, ed. 1971. *The SMART Retrieval System: Experiments in Automatic Document Processing.* Englewood Cliffs, NJ: Prentice-Hall.

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.