Retrieving Answers from Frequently Asked Questions Pages on the Web

Valentin Jijkoun
jijkoun@science.uva.nl

Maarten de Rijke
mdr@science.uva.nl

Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

ABSTRACT

We address the task of answering natural language questions by using the large number of Frequently Asked Questions (FAQ) pages available on the web. The task involves three steps: (1) fetching FAQ pages from the web; (2) automatic extraction of question/answer (Q/A) pairs from the collected pages; and (3) answering users' questions by retrieving appropriate Q/A pairs. We discuss our solutions for each of the three tasks, and give detailed evaluation results on a collected corpus of about 3.6Gb of text data (293K pages, 2.8M Q/A pairs), with real users' questions sampled from a web search engine log. Specifically, we propose simple but effective methods for Q/A extraction and investigate task-specific retrieval models for answering questions. Our best model finds answers for 36% of the test questions in the top 20 results. Our overall conclusion is that FAQ pages on the web provide an excellent resource for addressing real users' information needs in a highly focused manner.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Linguistic processing, Indexing methods; H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms, Measurement, Performance, Experimentation

Keywords

Question answering, FAQ retrieval, Questions beyond factoids

1. INTRODUCTION

Question Answering (QA) is one of several recent attempts to provide highly focused access to textual information. The task has received a great deal of attention in recent years, especially since the launch of the QA track at TREC in 1999. While significant progress has been made in technology for answering general factoids (e.g., *How fast does a cheetah run?* or *What is the German population?*), there is a real need to go beyond such factoids. Our long term goal is to develop QA technology for the web, catering for ad hoc questions posed by web users. Factoids are frequent in the log files of commercial search engines, but other types are at least as frequent, if not more frequent: procedural questions, requests for explanations or reasons. Such questions we refer to as "questions beyond factoids" [29].

What is the appropriate format for answering questions beyond factoids? This is not an easy matter. As we argue in Section 2, even for factoids it may be unclear what the appropriate answer format is in an ad hoc QA setting, i.e., where a system is to respond to general open domain questions without having access to information about the user asking the question. The problem becomes even more difficult when we try to find answers to questions beyond factoids. In this paper we "solve" the answer format issue by leaving it to humans. Our system responds to questions (factoid or beyond) by retrieving answers from frequently asked questions (FAQ) pages. FAQ pages constitute a rich body of knowledge that combine focused questions (mostly beyond factoids) with ready-made human generated answers. Our task, then, is to return, in response to an incoming question Q', a ranked list of pairs (Q, A) extracted from FAQs, where (ideally) the top ranking pair (Q, A) provides an *adequate* answer Q to the user's question Q', and other result pairs (Q'', A'') provide additional *material* information pertaining to Q'. To us, answering questions beyond factoids is a retrieval task, where systems should return a *ranked list* of (*multiple*) focused responses *in context*. This view has clear methodological advantages over the single-shot isolated answer format as required from systems taking part in the TREC and CLEF QA tasks.

In order to retrieve answers from FAQ pages on the web, we face three main tasks: (1) fetching FAQ pages from the web; (2) automatically extracting question/answer (Q/A) pairs from the fetched pages; and (3) addressing users' questions by retrieving appropriate Q/A pairs. In this paper we detail our solutions for each of those three steps, with a focus on the second, and, especially, the third step. We provide evaluation results for all steps.[1]

To the best of our knowledge, this is the first description

CIKM'05, October 31–November 5, 2005, Bremen, Germany.

[1]All the data used for evaluation purposes in this paper is available at http://ilps.science.uva.nl/Resources.

of an open-domain QA system that collects and uses FAQ pages on the web to find answers to user questions. The evaluation of the system on real user questions, randomly sampled from search engine query logs, shows that the system is capable of answering 36% of the questions (in the top 20 results) and provides material information for 56% of the questions. This justifies our claim that FAQ pages on the web constitute an excellent resource for addressing real user's information needs in highly focused manner.

The remainder of the paper is organized as follows. In Section 2 we provide extensive motivation for our proposal to address ad hoc QA as retrieval from FAQs. Sections 3, 4, and 5 are devoted to locating FAQs on the web, extracting Q/A pairs from the found pages, and retrieval against the resulting collection, respectively. In Section 6 we discuss related work, and we conclude in Section 7.

2. BACKGROUND

Our long term aim is to develop question answering technology for web users. By inspecting log files of commercial search engines we can find out what type of questions web users ask, even today. In a sample of 10M queries collected from the MetaCrawler engine (`http://www.metasearch.com`) in the period September–December 2004 we found that at least 2% of the queries were actual questions. While about a quarter of the questions were factoids, another quarter consisted of questions asking for definitions or explanations, and a third of the questions were procedural in nature. At the TREC QA track the importance of questions beyond factoids has been recognized through the introduction of definition questions in 2003 and of so-called "other" questions (that ask for important information about a topic at hand that the user does not know know enough about to task) in 2004 and again in 2005.

There are several aspects that make retrieval against Q/A pairs mined from FAQ pages on the web a viable approach to QA. First, factoid questions typically require simple entities or phrases as answers; moving beyond factoids, the TREC QA track required a list of snippets containing important bits of information as the answer to definition and "other" questions; the decisions on answerhood, however, turned out to be problematic [30]. For even more complex questions beyond factoids, we believe that requiring systems to precisely delineate an appropriate answer is beyond current state of the art in language technology. This suggests that returning human generated answers in response to questions beyond factoids is a sensible and realistic solution. Specifically, in our FAQ retrieval scenario, we attempt to locate questions Q that somehow pertain to the user's input question Q' and return the full answer A that comes with Q. Note, by the way, that instead of being isolated text snippets, answers on FAQ pages often come with explanations and/or justifications, while additional Q/A pairs on the page where an answer was found provide users with further context. This is an important feature, especially given that users tend to prefer answers in context [21].

There are further issues with QA that can be addressed by implementing QA as retrieval from FAQs. Recall that the "interaction model" adopted by much current research in QA technology is the following: in response to a question, return a single exact answer. In an ad hoc web setting, with no information about the user and her background or intentions (other than what can be picked up through the user's browsing agent), this single shot format is inappropriate. An example will help explain this. Consider the question *Where is the Rijksmuseum located?*. Without explicit knowledge about the questioner's context or background, the system has to select any of the following as the single best answer:

A1 In a building designed by the architect Cuijpers.

A2 Across Museumplein from the Concertgebouw.

A3 You can take Tram 5 to get there.

A4 Jan Luijkenstraat in Amsterdam.

A5 The Rijksmuseum van Oudheden (RMO) is the national museum of antiquities at Leiden

Each of these is a correct answer, and each contains some information that none of the others contains. This suggests that the best we can do is present *all* of these answers; following standard practice in IR, this is most naturally done in the form of a ranked list.

By returning ranked lists of Q/A pairs in response to an incoming question we are able to implement an important lesson from the "other" questions studied at the TREC QA track. Even if a text snippet does not provide a full answer to a question, it might still supply material information, pertinent to the question at hand, that might help the questioner achieve her information seeking goals [10]. In this manner, our ranked list output format helps facilitate serendipity.[2]

3. FETCHING FAQ PAGES FROM THE WEB

Our first step is to collect web pages containing lists of Q/A pairs. Since only a relatively small portion of the Web pages are actually FAQ pages, it would be quite resource-inefficient to collect such pages using general-purpose crawling. Instead, we use a web search engine (Google) to locate pages that are likely to contain FAQs, specifically, that have the string "faq" in their URL or title. While we obviously miss many potentially useful question/answer pages, the number of found pages is still very large: Google reports 8,900,000 hits for the query "`intitle:faq`" and 16,700,000 hits for "`inurl:faq`."

Ideally, we would simply use these two queries to retrieve a large collection of FAQ pages. Unfortunately, most web search engines, including Google, provide at most 1000 hits for any given query. To obtain more (ideally, all) FAQ pages, various query expansion techniques can be used. We took a simple, but sufficiently effective approach. We used the ontology of the Open Directory Project (`http://dmoz.org/`) to extract a list of reasonably general concepts: we simply extracted all concepts having between 5 and 20 children in the ontology. In the experiments below we used 3,847 out of a total of 9,747 extracted concepts to generate expanded retrieval queries (e.g., "`inurl:faq Party Supplies`"), and for each of the resulting queries the top 900 documents were retrieved using Google's interface. In total, we used 118,742 calls to the Google API, which gave us a list of 404,870 unique URLs (on average 3.4 unique URLs per call to the Google API, or 105.2 unique URLs per expanded query). For the experiments reported below we downloaded pages using 293,031 of these URLs.

[2]In our retrieval experiments in Section 5 we use a three point scale to assess the appropriateness of a Q/A pair in response to a user question: "adequate" (as an answer), "material" (information), and "unsatisfactory."

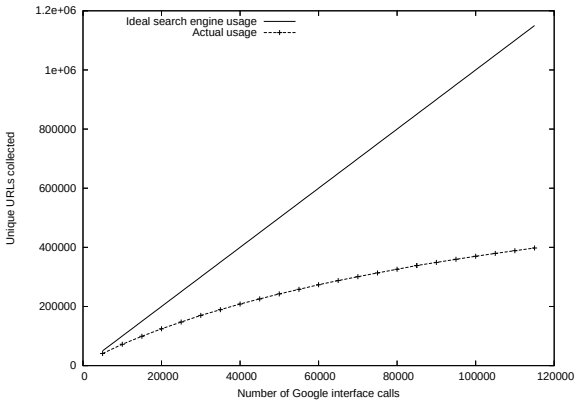

Figure 1: Performance of the FAQ fetching module

Since we do not attempt to partition the search space into non-overlapping segments, result lists for later queries increasingly contain URLs that have already been listed in the results for earlier queries: in 1,201,170 results from Google we found 404,870 (33%) unique URLs. Figure 1 shows how the number of unique URLs grows with the number of calls to the Google API (one call returns at most 10 results). Whereas for the experiments described below our simple fetching method is appropriate, it is not scalable for the creation of a much bigger corpus. In that case, more sophisticated query expansion methods [9] can be used to optimize the utilization of the web search engine.

We did not have the resources to perform a large scale evaluation of the accuracy of the fetching method. However, in a random sample of 109 pages from the 293,031 pages collected using the method described above, 83 pages (76%) turned out to be true FAQ pages. (See Section 4.2 for more details on this evaluation.)

4. EXTRACTING Q/A PAIRS

Once web pages potentially containing lists of Q/A pairs have been collected, the next task is to automatically extract Q/A pairs. While FAQ files abound on the web,[3] the main challenge is that the details of the Q/A mark-up vary wildly between FAQ files. In HTML, questions and answers can be simply marked as separate paragraphs, or questions can be made to "stand out" using italics, boldface, headers, fonts of different size and color etc. Various types of indentation (tables, lists, text alignment) and specific line prefixes (e.g., `Question:`, `Subject:`, `Q:`, `A:` etc.) are used. Most FAQ pages share two important aspects: the annotation on a single page is usually consistent (all questions look "similar"), and pages typically contain several Q/A pairs.

4.1 Extraction Methods

We propose two methods for extracting Q/A pairs: one is heuristics-based, and the second bootstraps the heuristic method using memory-based learning. The second method is supervised in that it builds on the output of our handcrafted heuristic method; it is unsupervised in that it uses no further manual annotation. Both methods detect questions on potential FAQ pages. Once the questions have been identified, we use rules to locate corresponding answers.

Heuristic Extraction. We first use the HTML mark-up of a page to split it into a sequence of text blocks that would be visually separated if the page were viewed in a web browser. We parse the HTML code and consider tags `table`, `br`, `p`, `li`, `div`, `h1-h6`, etc. as always starting or ending a text block. Since many "real world" web pages contain non-well-formed HTML (non-nested tags, missing closing tags, etc.), we use simple heuristics, emulating the behavior of a browser.

Each text block is described by a set of features like the first and second words of the block, templates of these words (e.g., "digits followed by a semicolon"), the first and last characters of the block, length, all HTML tags and tag attributes affecting the block, etc. Most features are obvious, and some have been used for detecting questions [24]. The actual number of features for a page is typically between 25 and 45. Once the content of a page is split into a sequence of blocks, we use simple heuristics to select questions: a block should contain at most 200 characters, not be a hyperlink, and contain a question mark or a capitalized question word. This extraction method is called *Heuristics* below.

Machine Learning Extraction. Naturally, the *Heuristics* method is not perfect, neither for *precision* nor for *recall*. We extend it by exploiting another aspect common to many FAQ pages: there are typically *several* true questions on a page, many of which are *consistently annotated*. For each FAQ file we assume a consistent mark-up (e.g., all questions are section headers). We make no assumptions on the common annotation details, but do assume that there are mark-up features that are consistent among questions on the page and distinguish questions from other text blocks (e.g., answers). We use TiMBL, a memory-based learner [8], to bootstrap from the questions selected with the *Heuristic* method. The learner detects which block features are *consistent* and *characteristic* for the questions on a given web page. We use the learner as a re-classifier, where it uses as its training base all text blocks from a page, classified into {*question*, *not-question*} using our heuristics. The learner calculates feature weights and then re-classifies the same sequence of blocks of the page in the leave-one-out mode (i.e., ignoring the original decision of our heuristic method). We use k-nearest-neighbors classification with inverse distance voting and the number of neighbors equal to the number of questions found with the heuristics. This is the *ML* method.

Identifying Answers. Once the blocks corresponding to questions have been determined, we use a simple but reliable rule to extract corresponding answers: up to three consecutive non-question text blocks immediately following a question in the HTML file constitute the answer.

4.2 Evaluation of the Extraction Methods

To assess our extraction methods, we created our own evaluation corpus, which consists of 109 pages taken randomly from the results of the fetching step (Section 3). The descriptive statistics of the corpus can therefore be extrapolated to large collections retrieved from the web in this way.

From the 109 pages, annotators manually extracted all Q/A pairs. Annotators were asked to extract as questions bits of texts that are easily interpreted as questions *in the context of the page* and that have explicit and clearly identifi-

[3]There is even a markup language, QAML, to represent FAQs: `http://xml.ascc.net/en/utf-8/qaml-index.html`.

able answers. The annotators were instructed not to include question numbers or identifiers and various prefixes (e.g., *Q:*, *A:*, *Question:*, *Subject:* etc.). Moreover, to make the annotation task easier and faster, they were not requested to annotate *complete* answers, but only initial portions (one or two first sentences or lines) that were sufficient to locate any given answer on a page. Questions (without prefixes or identifiers) had to be extracted in full.

The table below gives some statistics for our evaluation corpus.

Q/A extraction evaluation corpus		
Total number of pages	109	
True FAQ pages	83	(76%)
Total number of Q/A pairs	1418	
Avg. number of Q/A pairs per page	13	

We evaluated our Q/A extraction methods against the evaluation corpus using conventional precision and recall measures. The table below gives precision and recall scores for the two methods.

Extraction method	Precision	Recall
Heuristic	0.94	0.93
ML	0.92	0.94

The performance of the heuristic Q/A pair extraction method is striking, given its simplicity. With precision and recall around 0.93 the system can be used to create a large clean collection of Q/A pairs. Bootstrapping shows the expected behaviour: the *ML* method finds other questions, not identified with the simple heuristics, but having similar mark-up features as the questions already found. Although the differences are fairly small, using machine learning to bootstrap the heuristic method allows us to somewhat improve the recall, at the cost of a minor drop in precision.

For our retrieval experiments (Section 5), we took the output of the fetching module (Section 3), and extracted Q/A pairs using the *Heuristic* method as it shows slightly better precision. In total, 2,824,179 Q/A pairs were extracted.

5. RETRIEVING Q/A PAIRS

We now describe our approach to retrieving Q/A pairs from the collection built in Section 4. We discuss various retrieval models, and then describe experiments aimed at assessing their effectiveness.

5.1 Retrieval Models

Our task is to return a ranked list of Q/A pairs in response to a user's question. The answer parts of one or more of the Q/A pairs returned should provide an adequate answer to the user's question; in addition, we want to return Q/A pairs that provide additional important information about the topic at hand. We treat the task as a fielded search task, where the user's question is treated as a query, and the items to be returned (Q/A pairs) have multiple fields: a question part, an answer part, and the text of the document from which it was extracted. There are several aspects of any FAQ collection that make the retrieval task different from ad hoc retrieval:

- Q/A pairs are often "incomplete," in that they implicitely assume the topic of the FAQ or anaphorically refer to entities mentioned elsewhere on the FAQ page (e.g., "*Who produced the program?*").
- Following our user model (discussed in Section 2), we focus on *initial precision*: the user prefers to find an adequate answer in top 10–20 ranked Q/A pairs.
- Question words and phrases (such as "*Who*", "*How do I*" etc.), typically consisting of stopwords, might be valuable indications of adequacy of an Q/A pair.

The retrieval models we investigate exploit these aspects.

Indexing. We use the implementation of the vector space model in Lucene [1] as the core of our retrieval system. For each Q/A pair we index the following fields:

- question text
- question text without stopwords
- answer text without stopwords
- title of its FAQ page without stopwords
- full text of its FAQ page without stopwords

We build both stemmed and non-stemmed indices. Question texts are indexed both with and without stopwords, whereas stopwords were removed from all other fields.

Given a user question, the retrieval status value of a Q/A pair is calculated as a linear combination of similarity scores between the user question and the fields of the Q/A pair. The models we detail below correspond to different weights of the linear combination.

Models. As a baseline (B), we consider a simple model that calculates a linear combination of vector space similarities of the user question and fields of the Q/A pair: question text (with weight 0.5), answer text (weight 0.2) and FAQ page title (weight 0.3) (no stemming, but stopwords are removed).

The model B+D adds to the Q/A pair score the vector space similarity between the user question and the text of the entire FAQ page. In essence, we *smooth* the Q/A pair model with a document model.

The model B+D/2 is similar to B+D, but the document similarity is added with weight 0.5 rather than 1, so as to give the background model less weight.

The model B+D/2+S is similar to B+D/2, but all similarities are calculated after stemming (we use the English Porter stemmer [26]).

The model B+D/2+S+Q adds a score for the similarity between the user question and the question from the Q/A pair, calculated *without removing stopwords* and without stemming. The idea is to boost exact matches of questions, taking question words (such as "what", "how", etc.) into account that are typically treated as stopwords.

The model B+D/2+S+Q+PH is similar to B+D/2+S+Q, but the similarities are calculated using 1, 2 and 3 word n-grams (without stopwords) from the user question, rather than just single words. The intuition here is to take possible multi-word expressions (e.g., "sign language" or "super model") in the user's question into account. Recent work on topic distillation suggests that this may have a positive impact on early precision [25].

5.2 Experimental Setting

We experimented with a collection of 2,824,179 Q/A pairs, created as described in Sections 3 and 4; the collection adds up to 3.6Gb of raw text of the FAQ pages. One of our aims was to see how well questions asked by real users on the

web can be answered using FAQ files. To this end, we obtained samples from query logs of the MetaCrawler search engine (http://www.metacrawler.com) during September–December 2004, and extracted 44,783 queries likely to be questions: we simply selected queries that contained at least one question word ("what", "how", "where", etc.). For our retrieval experiments, we randomly selected 100 user questions from this sample. To facilitate our error analysis of the retrieval system, we manually classified questions into 8 categories; Table 1 gives statistics and examples. The categories are the same as the question types used by the FAQ Finder system [22], except that but we merged its *time*, *location*, *entity*, etc. categories into a single *factoid* category. (Note that the classification of user questions was *not* used during the retrieval.)

Type	#Q	Example
procedural	38	how to cook a ham
factoid	17	what did the caribs eat
description	13	who is victoria gott
explanation	10	info on why people do good deeds
non-question	10	what you did was rude e-cards
definition	8	what is cpi trotting
direction	2	where can i find physician employment
other	2	what present to get for my boyfriend

Table 1: Manual classification of 100 user questions.

For each retrieval model and each user question we retrieved a ranked list of top-20 Q/A pairs from the collection. In line with our user modeling concerns in Section 2, each combination of user question and retrieved Q/A pair was manually assessed on a three point scale:

- *adequate* (2): the Q/A pair contains an answer to the user question and provides sufficient context to establish the "answerhood";
- *material* (1): the Q/A pair does not exactly answer the user question, but provides important information pertaining to the user's information need; and
- *unsatisfactory* (0): the Q/A pair does not address the user's information need.

Assessors were asked to "back-generate" an actual information need given a user question before starting to make assessments. Adequacy assessments had to be based only on the text of the Q/A pair and the title and URL of the FAQ page. Table 2 gives examples of the assessors' decisions. In total, 5645 assessment decisions were made, 131 of which were *adequate* and 173 *material*.

Our main evaluation measure is *success rate*: the number of questions with at least one appropriate response in the top n results (where n equals 10 or 20). More precisely, we calculated $S_{1,2}@n$ (success rate with *adequate* or *material* Q/A pairs in the top n), and $S_2@n$ (*adequate* Q/A pair in the top n). These evaluation measures closely correspond to the user model we are interested in.

5.3 Results

We report on three groups of experiments, all aimed at understanding the effectiveness of our fielded search approach to FAQ retrieval. Table 3 presents the scores. In the first group in the table (rows 2–4) we compare the baseline model against smoothing with the document model at different

Model	$S_{1,2}@20$	$S_2@20$	$S_{1,2}@10$	$S_2@10$
B	48	28	44	25
B+D	49	31	45	26
B+D/2	49	28	45	27
B+D/2+S	47	30	41	26
B+D/2+Q	**56**	**36**	50	**29**
B+D/2+S+Q	54	34	45	28
B+D/2+S+Q+PH	50	34	44	27
B+Q	**56**	35	50	**29**
B+D+Q	55	35	**51**	**29**

Table 3: Evaluation of FAQ retrieval approaches on 100 user questions. Three groups of models (explained in Section 5.3), with the highest scores in boldface.

weights for the document model. In the second group (rows 5–8) we take one of the smoothed models from the first group, and investigate the impact of further smoothing, normalization and phrasal techniques. Finally, in the third group (rows 9–10) we compare the contribution of smoothing with the question model (B+Q) vs. smoothing with a document model (B+D and B+D+Q).

The best performing models provide adequate answers in the top 10 for 29% of the user questions, and material information for 50% of the questions (35% and 56% at top 20, respectively). This is a surprisingly high performance, given the limited size of our FAQ collection, the real-world and open-domain nature of the evaluation questions, and the simplicity of the retrieval models.

Clearly, the success rates at 10 and at 20 are highly correlated, especially where the differences between the scores are substantial. In particular, three of the models (smoothing with the question model and, optionally, the document model, i.e., B+Q, B+D+Q, and B+D/2+Q) behave very similarly and outperform others for all four measures.

5.4 Discussion and Error Analysis

Interestingly, the best performing models use matches on questions, without removing stopwords. As mentioned above, question stopwords ("how", "why", "what", etc.) are often good indicators of the question type, which has been shown to be useful in FAQ retrieval [22]. Even without explicit identification of types of user questions we are able to exploit this information by simply keeping question stopwords.

The evaluation results indicate that stemming does not help in retrieving Q/A pairs. Stemming is generally used to improve recall, but it does not seem to be helpful for the success rate, the main measure we are interested in. Interestingly, the FAQ Finder system [5] used stemming for retrieving Q/A pairs, which did make sense given the small size of its FAQ collection. The size and redundancy of our corpus may play an important role: user questions often have several adequate and material Q/A pairs in different FAQ pages, which increases the chances of finding these Q/A pairs even without stemming. In Figure 2 we show the effect of stemming on the reciprocal rank (the inverse of the rank of the first adequate or material Q/A pair) for the evaluation questions. We compare one of the best runs (B+D/2+Q) and its stemmed version (B+D/2+S+Q). Although stemming does somewhat improve the ranking of Q/A pairs for some questions (right-hand side of Figure 2), for many questions it moves appropriate Q/A pairs down the ranking. Due

question	how to beat traffic radar	*type*: procedural
adequate	Q What is Radar Detection?	
	A Modern traffic radar guns can clock vehicles in front of or behind the patrol car. For this reason, it's important that your detector is able to capture radar behind your vehicle...	
material	Q Are radar detectors legal?	
	A They are legal to use in New Zealand have been for over 26 years...	
unsatisf.	Q How does the radar work?	
	A NEXRAD (Nex t Generation Rad ar) obtains weather information (precipitation and wind) based upon returned energy...	
question	what are internet browsers	*type*: definition
adequate	Q What browsers are there available?	
	A There are numerous browsers available to surf the WWW...	
adequate	Q What is a browser?	
	A To take a look at web pages your will need what is called a browser...	
material	Q What Internet browsers are supported?	
	A This site is best viewed with any w3c.org compliant browser like Mozilla , Firefox , Camino (Mac) and Safari (Mac) . However, we have yet to test our pages with Opera , Konqueror , OmniWeb...	
unsatisf.	Q Why would someone use a browser other than Netscape or Internet Explorer?	
	A See the reasons listed above under " Why can't we just ask people to upgrade their browsers or switch browsers?	

Table 2: Examples of adequacy assessments.

to the restricted size of our question test set, we have not yet been able to determine whether the usefulness of stemming correlates with types of user questions.

Surprisingly, phrasal search has a clear negative effect on the success rates, although it proved useful for improving early precision in other settings such as topic distillation in web retrieval [25]. We see two reasons for these findings. One is that the questions are much longer (6.9 words on average) than the queries on which phrasal search was found to be most effective in the topic distillation setting (2.5 words on average at TREC 2003). Another reason is the presence of duplicate or near duplicate FAQ pages and Q/A pairs in our collection; phrasal search seems to help in giving highly similar Q/A pairs very similar retrieval scores, which often forced other potentially useful Q/A pairs to drop out of the top 10 or even top 20 results. Offline (at indexing time) and/or online (at query time) clustering and (near) duplicate detection are essential to eliminate this problem.

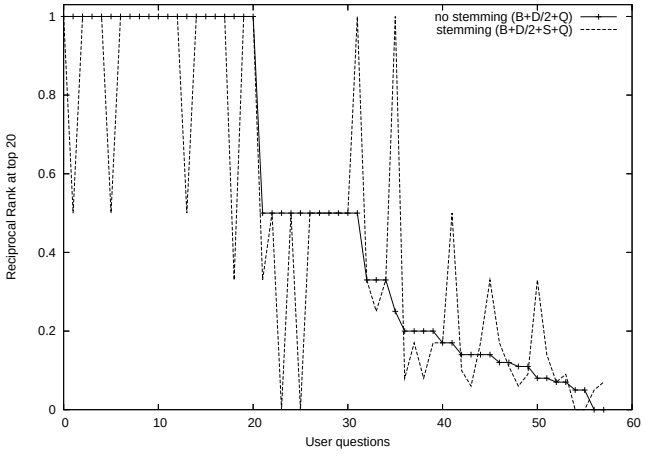

Figure 2: The impact of stemming on reciprocal rank, B+D/2+Q vs. B+D/2+S+Q; only test questions with a non-zero reciprocal rank for at least one of the two runs are plotted.

A further source of errors are frequent typos in user questions: "*who made the got milk?*" with *goat* misspelled, "*how do yousay youre wlecome in sign language*", or "*how glidders take off*" with *gliders* is misspelled. Since we do not try to correct these mistakes, the system does not find appropriate responses for the questions. For a real web application detection and correction of typos is essential.

A different type of problem we faced was the difficulty of assessing adequacy for some of the questions. First, none of the non-questions (10% in Table 1) sampled from the MetaCrawler query log received adequate or material responses from the system. However, even for some well-formed questions, (e.g., "*how to move in combat*" or "*how to teach the sac*") assessors were unable to reconstruct the original information need and, hence, identify any appropriate Q/A pairs in the results of the system. Nevertheless, we decided not to exclude these non-questions and unclear questions from the evaluation, so as not change our sample.

6. RELATED WORK

6.1 Question Answering

FAQ files are a natural source for knowledge. AutoFAQ [32] was an early system that provided access to a small number of FAQs. The FAQ Finder system [4, 5] retrieves existing answers from a small and homogeneous set of FAQ files (mostly associated with USENET newsgroups, with a standardized Q/A pair markup). FAQ Finder was evaluated using questions drawn from the system's log files. It uses a weighted sum of the term vector similarity (between user question and Q/A pair), word overlap between user question and indexed question, and a WordNet-based lexical similarity between user question and indexed question. We use fielded search, based on a mixture model to which indexed

questions, answers, and full FAQ pages contribute, optionally together with phrasal search and stemming. Later work on FAQ Finder [22, 23] studied the impact of sense tagging and question typing on computing the similarity between user questions and indexed questions. As discussed previously, in the model in which we represent questions with a separate model we implictly do question typing.

Operating in a helpdesk environment, eResponder stores Q/A pairs that have previously been asked and answered; these pairs can be used to respond to user questions, or to assist customer service representatives in drafting new responses [6]. See [16] for a similar system and [15] for a variation that uses cluster-based retrieval.

QA beyond factoids is largely unexplored, with a few notable exceptions [2, 3, 12]. Much of today's QA research is centered around the QA tracks at TREC and CLEF, where systems respond to factoids by returning a single exact answer extracted from newspaper corpora. While TREC is slowly moving beyond factoids (by also considering definitions questions in 2003, "other" questions in 2004 and 2005, and "relationship finding" in 2005), it focuses on newspaper corpora, instead of the wealth of information on the web.[4]

As to systems that perform QA against the web, AnswerBus [34] is an open-domain QA system based on sentence level web IR. Focused on factoids, AnswerBus returns isolated sentences that are determined to contain answers. In its later incarnations, the Start QA system [13] uses an abstraction layer over diverse, semi-structured online content, centered around "object-property-value" queries [14]. Through this layer, Start has access to online sources such as `biography.com` and the Internet Movie Database to answer factoids. MULDER [18] was a TREC-style QA system used to investigate whether the QA techniques studied in IR can be scaled to the web; its focus was on factoids. AskJeeves (`http://www.ask.com`) is a commercial service that provides a natural language question interface to the web, relying on human editors to map between question templates and authoritative sites.

Recently, several researchers used large numbers of Q/A pairs mined from the web for QA system development. E.g., Ramakrishnan et al. [28] use them to train a strongly data-driven TREC-style factoid QA system. Soricut and Brill [29] describe a statistical QA system that is trained using roughly 1 million Q/A pairs to deliver answers to questions beyond factoids, where the answers are extracted from documents returned by commercial web search engines.

6.2 Retrieval

Mixing global and local evidence in retrieval has been the focus of much of the research in semistructured retrieval, dating back to the work of Wilkinson [33]. Unlike in more recent applications (e.g., XML retrieval [11]), the retrieval unit (Q/A pair) is fixed in our task, but smoothing the Q/A pair model with the model of the whole FAQ page does indicate some improvements.

6.3 Extraction

Web content mining, i.e., mining, extraction and integration of useful information from web page contents, is a large area. Wrapper induction and automatic data extraction have been actively pursued since the mid-1990s [17]. While there has not been much work on extracting Q/A pairs from FAQ pages on the web, there is work on detecting lists and/or repeated items on web pages. E.g., building on [31], Limanto et al. [20] build an extractor for web discussion forms which tries to find repeated patterns using a suffix tree. Closer to our work is the work by Lai et al. [19], who propose a heuristic method for classifying FAQ pages on the web and a list detection method for extracting Q/A pairs; their rule-based extraction method is not evaluated, while their classifier achieves a precision of 80%. McCallum et al. [24] evaluate a supervised method (maximum entropy Markov model) on a segmentation task that is similar to but different from our Q/A pair extraction task: tagging lines of text files as questions and answers. They report scores of 0.86 (Precision) and 0.68 (Recall) for this task.

[4]Many TREC and CLEF participants use the web; they use it not as their primary answer source but, for instance, to mine extraction patterns from or for smoothing purposes.

6.4 Crawling

As stated previously, general-purpose crawling is not a suitable way to obtain FAQ pages. Focused crawling [7] is not appropriate either, since we are not restricting ourselves to any particular domain. The only robust criterion for "wanted" pages for us is that they contain question/answer pairs. Our ontology-based method of expanding queries that are aimed at retrieving FAQ pages from a web search (such as Google) differs from techniques in the literature; e.g. Etzioni et al. [9] used recursive query expansion for a similar task, partitioning the set of a search engine's results until the full list becomes accessible. The method works by recursively adding query terms `+word` or `-word` to the original query for reasonably frequent words. Our experiments with this method for locating FAQ pages, however, showed that the quality of the retrieved pages (i.e., the proportion of true FAQ files) deteriorates quickly with the length of the expanded query. This may be due to subtle interactions between terms in Google's fielded search ("faq" was requested in the title or URL, while expanded terms should occur in the body of a page). Unlike the technique used in [9], our query expansion method does not try to partition the search space into non-overlapping segments. As a consequence, results for later queries increasingly contain URLs already encountered.

7. CONCLUSIONS

We have presented an open-domain Question Answering system that retrieves answers to user's questions from a database of Frequently Asked Questions automatically collected on the Web. The system performs offline datamining (locating FAQ pages on the web and extracting Q/A pairs) to create the database, and then appropriate Q/A pairs are retrieved from the database as responses to users' questions. By making the semantic structure of FAQ pages (i.e., relations between questions and answers) explicit, we are able to avoid an ad hoc definition of expected answer format, ultimately leaving it to authors of the FAQ pages to define what a good answer to a question should look like. This is especially important for non-factoid questions, which are the most frequently asked questions on the web. The performance of the system is measured on a 3 point scale, corresponding to the user modeling we detailed in Section 2. Our system shows an impressive performance, even with fairly simple fetching, extraction and retrieval methods. This suggests that FAQ pages on the web are indeed an excellent

resource for focused information access providers.

Looking forward, there are some obvious techniques, such as WordNet relations [23], and question classes [22] are waiting to be integrated into our system. To make the system usable in real world scenarios, many technical and scientific questions need to be addressed. First, different, more robust and scalable fetching methods have to be designed to obtain substantially larger FAQ corpora. The system has to operate in the dynamic environment of the web, where new data emerges or becomes outdated every hour. More effective Q/A pairs extraction techniques should deal with different types of FAQ pages [19]. Duplicate and near duplicate detection (both offline and online) should be used to eliminate redundancy in presenting results to the user. We are currently experimenting with result clustering and topic detection, both of which we regard as essential, given the often ambiguous nature of user queries.

Another direction for future research concerns our evaluation. We are extending our retrieval evaluation. Additionally, to which extent do our evaluation measures reflect the reality of real users? What happens if we restrict the questions and FAQ pages to be in the same subject area?

In sum, we believe that "QA as FAQ retrieval" is an important step towards providing highly focused information access to web users.

Acknowledgements. We thank our referees for their feedback. This research was supported by the Netherlands Organization for Scientific Research (NWO) under project numbers 017.001.190, 220-80-001, 264-70-050, 354-20-005, 612-13-001, 612.000.106, 612.000.207, 612.066.302, 612.069.006.

8. REFERENCES

[1] Apache Lucene: A high-performance, full-featured text search engine library. http://lucene.apache.org.

[2] E. Agichtein, S. Lawrence, and L. Gravano. Learning to find answers to questions on the web. *ACM Trans. Inter. Tech.*, 4(2):129–162, 2004.

[3] A. Berger, R. Caruana, D. Cohn, D. Freitag, and V. Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings SIGIR 2000*, pages 192–199, 2000.

[4] R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. Natural language processing in the FAQFinder system: Results and prospects. In *Proceedings 1997 AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, pages 17–26, 1997.

[5] R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked question files: Experiences with the FAQFinder system. *AI Magazine*, 18(2):57–66, 1997.

[6] D. Carmel, M. Shtalhaim, and A. Soffer. eResponder: Electronic question responder. In *Proceedings CoopIS 2002*, pages 150–161, 2000.

[7] S. Chakrabarti, M. Van Den Berg, and B. Dom. Focused crawling: A new approach to topic-specific Web resource discovery. *Computer Networks*, 31:1623–1640, 1999.

[8] W. Daelemans, J. Zavrel, K. Van Der Sloot, and A. Van Den Bosch. *TiMBL: Tilburg Memory Based Learner, version 5.0*. Tech. Report 03-10, 2003. URL: http://ilk.kub.nl/downloads/pub/papers/ilk0310.ps.gz.

[9] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Web-scale information extraction in KnowItAll: (preliminary results). In *Proceedings WWW 2004*, pages 100–110, 2004.

[10] A. Foster and N. Ford. Serendipity and information seeking: an empirical study. *J. Documentation*, 59(3):321–340, 2003.

[11] N. Fuhr, M. Lalmas, S. Malik, and Z. Szlavik, editors. *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval (INEX 2004)*, LNCS 3493, Springer, 2005

[12] R. Girju. Automatic detection of causal relations for question answering. In *Proceedings ACL 2003 Workshop on Multilingual Summarization and Question Answering*, 2003.

[13] B. Katz. Annotating the World Wide Web using natural language. In *Proceedings RIAO'97*, 1997.

[14] B. Katz, S. Felshin, D. Yuret, A. Ibrahim, J. Lin, G. Marton, A. McFarland, and B. Temelkuran. Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings NLDB 2002*, 2002.

[15] H. Kim and J. Seo. High-performance FAQ retrieval using an automatic clustering method of query logs. *Information Processing & Management*, in press.

[16] L. Kossseim, S. Beauregard, and G. Lapalme. Using information extraction and natural language generation to answer e-mail. *Data & Knowledge Engineering*, 38(1):85–100, 2001.

[17] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1–2):15–68, 2000.

[18] C. Kwok, O. Etzioni, and D. Weld. Scaling question answering to the web. In *Proceedings WWW 2001*, pages 150–161, 2001.

[19] Y.-S. Lai, K.-A. Fung, and C.-H. Wu. FAQ mining via list detection. In *Proceedings Coling Workshop on Multilingual Summarization and Question Answering*, 2002.

[20] H. Limanto, N. Giang, V. Trung, N. Huy, J. Zhang, and Q. He. An information extraction engine for web discussion forums. In *Proceedings WWW 2005*, pages 978–979, 2005.

[21] C.-Y. Lin, D. Quan, V. Sinha, K. Bakshi, D. Huynh, B. Katz, and D. Karger. What makes a good answer? The role of context in question answering systems. In *Proceedings INTERACT 2003*, 2003.

[22] S. Lytinen and N. Tomuro. The use of question types to match questions in FAQFinder. In *Proceedings AAAI-2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases*, pages 46–53, 2002.

[23] S. Lytinen, N. Tomuro, and T. Repede. The use of WordNet sense tagging in FAQFinder. In *Proceedings AAAI-2000 Workshop on AI and Web Search*, Austin, TX, 2000.

[24] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings ICML 2000*, pages 591–598, 2000.

[25] G. Mishne and M. de Rijke. Boosting Web Retrieval through Query Operations. In *Proceedings ECIR 2005*, pages 502–516, 2005.

[26] M. Porter. An algorithm for suffix stripping. *Program*, 14 (3):130–137, 1980.

[27] D. Radev, W. Fan, H. Qi, H. Wu, and A. Grewal. Probabilistic question answering on the web. In *Proceedings WWW 2002*, pages 408–419, 2002.

[28] G. Ramakrishnan, S. Chakrabarti, D. Paranjpe, and P. Bhattacharya. Is question answering an acquired skill? In *Proceedings WWW 2004*, pages 111–120, 2004.

[29] R. Soricut and E. Brill. Automatic question answering: Beyond the factoid. In *Proceedings HLT/NAACL*, 2004.

[30] E. Voorhees. Evaluating answers to definition questions. In *Proceedings HLT 2003*, 2003.

[31] J. Wang and F. Lochovsky. Data extraction and label assignment for web databases. In *Proceedings WWW 2003*, pages 197–196, 2003.

[32] S. Whitehead. Auto-FAQ: An experiment in cyberspace leveraging. *Computer Networks and ISDN Systems*, 28(1–2): 137–146, 1995.

[33] R. Wilkinson. Effective retrieval of structured documents. In *Proceedings SIGIR 1994*, pages 311–317, 1994.

[34] Z. Zheng. AnswerBus question answering system. In *Proceedings HLT 2002*, 2002.