Evaluating Question Answering Systems Using FAQ Answer Injection

Jochen L. Leidner and **Chris Callison-Burch**
Institute for Communicating and Collaborative Systems (ICCS),
School of Informatics, University of Edinburgh,
2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, UK.
⟨{jochen.leidner,callison-burch}@ed.ac.uk⟩

Abstract

Question answering (NLQA) systems which retrieve a textual fragment from a document collection that represents the answer to a question are an active field of research.

But evaluations currently involve a large amount of manual effort.

We propose a new evaluation scheme that uses the insertion of answers from Frequently Asked Questions collections (FAQs) to measure the ability of a system to retrieve it from the corresponding question. We describe how the usefulness of the approach can be assessed and discuss advantages and problems.

1 Introduction

Automatic open-domain question answering systems are an active field of research (e.g. (Harabagiu et al., 2000) or (Leidner, 2002)). These systems retrieve one or a set of textual fragments from a document collection that represents the answer to the question. Since the Eighth Text Retrieval Conference (TREC-8), annual competitions have been carried out to assess system performance. Such evaluations currently involve a large amount of manual effort and are therefore costly and relatively small-scale. We propose a new, fully automatic evaluation scheme that uses *Frequently-Asked Questions* (FAQs).

Such documents are available in large number on the Internet, both on commercial Websites and especially in archives of USENET newsgroups in large concentration on the dedicated server ⟨ftp://rtfm.mit.edu/⟩. Each FAQ file is about a specific domain, but due to their vast number ranging from cooking recipes over Frank Zappa or Monty Python fan issues, politics to compiler construction and artificial intelligence. Due to their wide availability, we hope FAQs will be further recognized as a valuable resource for the question answering community: they have already been used as a knowledge base for question answering (Burke et al., 1995), learning document segmentation rules (McCallum and Pereira, 2000), automating call centers (IBM, 2002) and summarization (Berger and Mittal, 2000).

We propose a new application area for FAQs, namely to automate the natural language question answering (NLQA) *evaluation* task: we propose that the intrinsic knowledge in a question-answer pair be used to measure the performance of question answering systems without having to resort to human-created answer keys.

Section 2 describes the current TREC evaluation briefly and mentions other attempts towards automating evaluation. Section 3 points out some drawbacks of existing practice. Section 4 describes our method, called FAQ Answer Injection. Section 5 points out future work that will be necessary to establish the quality of the method. Section 6 describes the benefit of FAQ Answer Injection for component evaluation. Section 7 discusses some pros and cons of our method and Section 8 concludes this paper.

2 Previous Work

The amount of human effort required in the TREC question answering evaluations is enormous (Voorhees and Tice, 2000): for instance, for the TREC-8 Question Answering track there were 37,927 system responses that were judged by the NIST judges. 35,648 of these responses were unique answer strings. There were 198 questions, and each run could produce up to 5 ranked answers to each question. 25 sites participated, each submitting one or both of a 50-byte answer run and a 250-byte answer run, for a total of 41 submitted runs. The 1999 TREC-QA evaluation additionally used an adjudicated system which required three judges examine each answer.

Automating this process while ensuring that it would remain as accurate as human evaluation is a difficult undertaking (Voorhees and Tice, 2000). The method of automating evaluation in subsequent TREC evaluations was to use a set of hand-designed "answer patterns." For every question, five answer candidates were examined, and a set of regular expressions that describe answer patterns was defined by the U.S. NIST organizers to automate the evaluation process. To do this, either manual screening of the document collection was necessary, or an information retrieval system could be used by a human expert to find answers. He or she would then refine or extend the regular expression pattern so as to subsume all answers contained in the collection. For example, for

the question

When did Shakespeare die? (1)

a regular expression[1] matching all possible or expected ways of expressing the (an essential part of the) answer (with respect to the document collection) must be formulated, e.g.

```
(1616|sixteen(hundred\s+and)?\s+sixteen)
```
(2)

Usually, these patterns overgenerate, so an automatic match might need another human inspection that can decrease the score. A script is subsequently used to compute a score based on matching the expressions (describing human-prepared gold-standard answers) against the systems' retrieval results. The process of human answer key generation is time-consuming, costly and requires human experts.

(Breck et al., 2000) present Qavia, an automated evaluation system that compared human evaluation and automatic computation of recall of stemmed content words from a human [*sic*] answer key. They showed that their automatic metric agrees with the human 93%–95% of the time. 41 TREC NLQA systems yielded an correlation coefficient of $\tau = 0.92$ (Kendall, 1970), compared to $\tau = 0.96$ between human assessors.

(Magnini et al., 2002) go a step further and use a combination of strict and lenient *validation patterns* against the Web as an oracle. A lenient pattern is meant to retrieve answer candidates, quite like in the NLQA system itself, whereas the strict pattern is meant to measure the degree of justification via the number of hits. For example, in the question

Who shot Ronald Reagan? (3)

the query

```
('shot' NEAR 'Reagan')
```

could be used to find candidate answers, with

X='John W. Hinkley, Jr.'

as one of the retrieved candidates. Subsequently,

⟨'X shot Ronald Reagan'⟩

could be used as a justification template to show that

```
'John W. Hink-
ley, Jr. shot Ronald Reagan'
```

yields more hits than instantiations with any other candidate. Magnini et al. (2000) use two measures to estimate the relevance of the pattern searches, Mutual Information (MI) and Corrected Conditional Probability (CCP), and find the best agreement between decisions made by human judges occur in 84.82% of the cases with their method using the CCP metric over the TREC-10 question set.

[1] We use Perl notation.

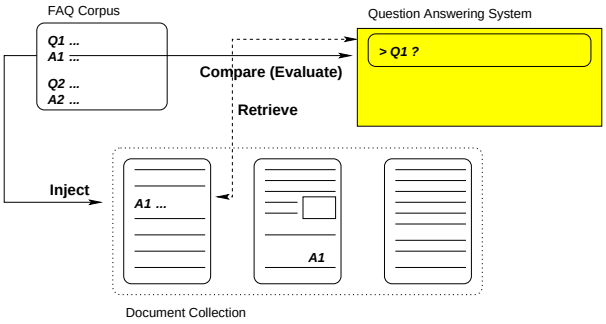

Figure 1: FAQ Answer Injection.

3 Motivation

The current evaluation scenario has a some drawbacks:

1. A human expert is needed to prepare the answer keys by sifting through the document collection in order to find the answers. The process is *costly* and *time intensive*.
   The absence of fully-automatic procedures has been repeatedly pointed out as a bottleneck for progress in the field (ARDA, 2002).

2. The expert is required to find *all* valid answer instances that could potentially be extracted by a system.

3. Naturally, answer keys cannot be used for different questions; by they also can't be re-used if the same questions are run against the same system, but using a different, or even dynamically changing, document collection. This scenario is important when we think about applying a system to the Web.

We propose a new evaluation method for question answering system evaluation in the following section that does not require human intervention and addresses issues (1.) and (3.), and sketch how to assess its usefulness.

4 Method

FAQs are plain text files that follow some format conventions which are not enforced, but tools have been developed in the *FAQ Finder* project to convert them into a more rigid form (Burke et al., 1995).

Formally, an FAQ is a set of pairs $\langle Q_i;A_i \rangle$ comprising individual question answer pairs. Extracting these, we can insert an answer A_i into a random document of the collection used by the question answering system, at a random position which we remember (**Injection** phase, Figure 1). This can be repeated with as many questions as we want. Then the question answering system is run with the questions (Q_i), to see whether the corresponding answers A_i are indeed retrieved. For example, consider the pair

Q: *Where should I report bugs and other problems with Emacs?*
A: *The correct way to report Emacs bugs is by e-mail to bug-gnu-emacs@prep.ai.mit.edu. Anything sent here also appears in the newsgroup gnu.emacs.bug, but please use e-mail instead of news to submit the bug report. This ensures a reliable return address so you can be contacted for further details.*

We inject the paragraph "*The correct way to report Emacs bugs...*" into the document collection, either as an individual document (if paragraph indexing is used) or into an existing document, keeping track of its paragraph number or document number and position. We then send the question "*Where should I report bugs and other problems with Emacs?*" to the question answering system to be evaluated.

For the evaluation proper, we can generate answer keys from the FAQ answer, which is known to us, e.g. `(correct way.+report.+Emacs bugs.+e-mail.+bug-gnu-emacs@prep.ai.mit.edu)` and use the standard TREC evaluation script. Alternatively, we can also directly use the retrieved *position* to check whether the retrieved answer list contains the fragment from the position of injection (which we had kept track of) to avoid recall errors by retrieving similar-looking, but different answers.

However, not all FAQ question-answer pairs are equally suitable for question answering evaluation. Some filtering will be helpful in order to ascertain a certain realistic scenario: for instance, the USENET FAQ for the 386BSD operating system contains the following question:

Q: *How can this be happening?*
A: *...*

The demonstrative pronoun requires a context that is simply not available in the current TREC scenario. The following measures reduce (but not eliminate) this issue (**Pre-processing** phase):

1. Filter out headings (non-questions) Exclude question-answer pairs the question of which contains *no question*, only section headers:

   Q: *Subject: 3.3 Archives*
   A: *...(a very long paragraph)...*

2. Cut out section information:

   Q: *3.2.4 How do I get ddb, the kernel debugger, compiled into the kernel and running?*
   A: *...*

   Initial section information—often followed by punctuation—should be deleted.

3. Filter out by length: Questions and/or answers that are either too short (e.g. 2 words) or too long (e.g. more than 20 words) should be deleted.
4. Filter out pairs with exophoric references: Do not consider pairs where the question contains pronouns

   Q: *As more and more investors look for these factors to identify stocks, will they still be as effective?*
   A: *...*

   or

   Q: *7 Distilled Wisdom on Equipment*
   A: *This is a new section, designed to contain small articles people have put together on various topics pertaining to cooking equipment*

   Heuristics must be used here, such as the number of anaphora for which there is no antecedent in the same segment.

Some more example FAQ pairs are given Appendix A.

5 Assessing the Method

Before applying the proposed method, it should be determined how well it is correlated to human judgment, or TREC-style human answer keys. Magnini et al. (2000) measure agreement between their evaluation method and TREC-10 answer keys prepared by the human experts, whereas (Breck et al., 2000) rank the TREC-8 using their Qaviar system based on stemmed human answer key recall and correlate the ranking with the original TREC system ranking using τ, Kendall's τ correlation coefficient (Kendall, 1970). τ provides a way to measure the strength of the tendency of one variable to follow the trends of a second variable without making assumptions with respect to underlying distributions.[2] It is easy to compute, has a simple interpretation and penalizes for ties. Furthermore it has the advantage that it was also used by NIST to show the strong correlation between three-judge rankings and single-judge rankings (Voorhees and Tice, 2000). To compute $\tau(X,Y)$ for a number of observation pairs $\langle x_i;y_i\rangle$, $\langle x_j;y_j\rangle$, ... we simply compare whether the signs are the same (*concordant*) or not (*discordant*). If we do this for all $n(n-1)$ pairs and normalize so that $\tau \in [-1;+1]$ we obtain:[3]

$$\tau = \frac{2 \cdot (\#concordant - \#discordant)}{n(n-1)} \qquad (4)$$

τ can be interpreted as a probability estimate that to pairs a concordant; other, more frequently used measures

[2] Actually, τ is a family of three tests τ_a, τ_b and τ_c. However, they behave identical in the absence of ties. SPSS implements two commands, `NONPAR CORR` (τ_a) and `CROSSTABS` (τ_b, τ_c) to compute these coefficients.

[3] Note that (Breck et al., 2000) give a slightly different definition.

(such as the Spearman Rank Correlation Coefficient), do not offer such an interpretation.
Therefore, we intent to use τ to in our future work for assessing the usefulness of FAQ Answer Injection.

6 Application to Component Evaluation

As pointed out in (Breck et al., 2000), automating the evaluation of question answering systems does not simply save on the cost associated with hand evaluation; it also allows a system to be repeatedly evaluated during its development cycle. Whereas the Qaviar system relies on there already being an "answer key" which it can use to score responses and only allows a system as a whole to be evaluated, our method of FAQ answer injection is fully automatic and further allows a fine-grained component-level evaluation.

Question answering systems such as Webclopedia (Hovy et al., 2001) are modular in design. Common components in QA systems include:

- Question Analysis – analyzes a question and performs an action such as assigning an expected answer type
- Document Retrieval – fetches documents which are thought to be relevant from the document collection
- Passage Selection – identifies paragraphs or short segments from the retrieved documents which likely answer the question
- Answer Ranking – selects the single best or *n* best passages as the answer to be returned

Since each component may operate independently from each other and since they might be developed simultaneously it would be useful to have a method for assessing how well each component was performing. FAQ Answer Injection allows us to evaluate components. Figure 2 gives an illustration of one way that a pool of candidate documents which includes an answer-injected document could be used to diagnose faulty components. Assuming that each component in a pipelined architecture limits the possible candidate set, then knowing which document and passage was the correct answer would allow an inspection to be performed to determine which component incorrectly excludes correct answers. This problematic component could then be improved.

Even if a system did not conform to the architecture of Figure 2, individual tests could be designed to test the components. For example, if the question analysis component assigned an expected answer type which corresponded to a named entity type that was assigned by the passage selection component, then FAQs would provide a way of analyzing how often the expected answer types were in line with the named entities. Similarly, an answer ranking component could be tested by giving it a number of candidate answers from a FAQ collection, and testing how often it was able to select the appropriate answer for a particular question.

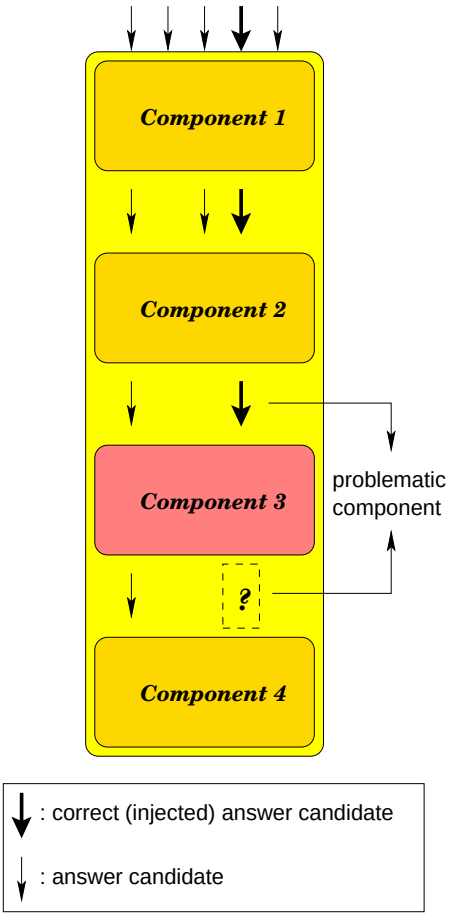

Figure 2: Detection of Components Responsible for Recall Drop.

Having a large corpus of FAQs allows tests to be tailored in order to verify assumptions being made in the design stage. For instance we might conjecture that a certain percentage of key words in question will also appear in answers, or that using WordNet synonyms will help expand the pool of correct answers, or so forth. Since QA systems are engineering endeavors it would be useful to see what components are weak and what assumptions are unfounded.

7 Discussion

Though we have not yet verified that FAQ answer injection correlates with human judgments (as proposed in Section 5), we expect that it will at very least be a useful method for evaluating upward or downward trends in system performance. The approach is not completely without problems, however. For instance,

1. A system might choose to return a *different* answer from A_i found in a different document that still answers Q_i correctly. Since we are tracking only those

answers which we injected, we would not count this as a correct answer.

2. The artificially injected answers are unrelated to the topic of the document that they have been inserted into, so systems which rely heavily on surrounding document will be penalized unfairly under this evaluation scheme.

3. Because FAQs are collections of questions, they may contain extrasentential anaphora or other heavily context-dependent material. Such questions might not be representative of the types of questions that would be normally be posed to QA systems.

These problems are not necessarily fatal. Problem (1) might be lessened by the fact that the evaluation can use the top-*n* answers, and the FAQ answer is likely to be expected among what are considered the *n* candidate answers. Furthermore, the standard TREC procedure suffers from the same drawback, since the human patterns might not cover all the answer snippets in all documents. Problem (2) might be countered by a variation of the scheme which controls for same domains of the injected answer and the document that hosts it by using a statistical document classifier. Problem (3) might be controlled for by adapting the preprocessing stage so as to exclude such questions.

The strong advantage of our evaluation method is its *fully* automatic nature. The procedure therefore allows for more questions to be used for evaluation than in other, *semi*-automated techniques. The relative impact of problems (1) and (3) would be minimized simply by running the evaluation using a very large corpus of FAQs.

The approach is not completely unproblematic, as the system might choose to return a *different* answer from A_i found in a different document that still answers A_i correctly. However, since the evaluation can use the top-*n*, the FAQ answer is likely to be expected among what are considered the *n* candidate answers. Furthermore, the standard TREC procedure suffers from the same drawback, since the human patterns might not cover all the answer snippets in all documents.

The largest benefit is that due to its fully automatic nature, the procedure allows for *more* questions to be used for evaluation.

The artificially injected answers are unnatural, but this can be countered by a variation of the scheme which controls for same domains of the injected answer and the document that hosts it by using a statistical document classifier.

Not all questions from FAQs can be utilized for evaluation. If the question contains extrasentential anaphora that cannot be safely resolved or other heavily context-dependent material, we might want to skip them.

Interestingly, the absence of overlap between question and answer, which marks a very hard class of questions (for automatic systems), can be tested just as easily, since we possess the link between question and answer. Furthermore, those questions that fall in this difficult class can be determined automatically, because the detection of lacking surface overlap requires only string processing.

8 Conclusions

In this paper we propose a method for evaluating question answering systems using an extremely common Internet resource: Frequently Asked Questions collections. FAQs are a useful resource for evaluating QA systems in two respects. First, the vast range of topics covered by FAQs is a good simulation for truly open-domain systems. Second, and more importantly, FAQs provide a way of fully-automating the evaluation process.

By exploiting the fact that FAQs contain both questions and their answers, we nullify the need for human intervention that is required by other semi-automated techniques for QA evaluation such as those which use handbuilt regular expression answer keys. Correct answers are already associated with questions.

We simulate the task of retrieving the answers from a document set by injecting the answers into a particular document within the collection. By keeping track of which document the answer has been inserted into, and the position of the answer within the document, we are able to evaluate whether a system is able to correctly retrieve the answer. By scoring the system over a large number of questions, a rough idea of the system performance may be gathered. We further describe how FAQs can be used to produce a more fine-grained evaluation of the individual components within a QA system and describe how the value of our method can be assessed.

9 Acknowledgments

The first author is indebted to Karen Spärck-Jones and John Prange for their support. We are further grateful to Robert Burke for making available the USENET FAQ pre-processor from the FAQ Finder project.

References

ARDA. 2002. *AQUAINT Phase I Mid-Year Workshop*. Monterey, CA.

Adam Berger and Vibhu O. Mittal. 2000. Query-relevant summarization using faqs. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*.

Eric J. Breck, John D. Burger, Lisa Ferro, Lynette Hirschman, David House, Marc Light, and Inderjeet Mani. 2000. How to evaluate your question answering system every day and still get real work done. In *Proceedings of LREC-2000*, Athens, Greece.

Robin Burke, Kristian Hammond, and J. Kozlovsky. 1995. Knowledge-based information retrieval for semi-structured text. In *Working Notes from the AAAI-95 Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, Menlo Park, CA. AAAI Press.

Sanda Harabagiu, Marius Pasca, and Steven Maiorano. 2000. Experiments with open-domain textual question answering. In *Proceedings of COLING-2000*.

Edward H. Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. 2001. Question answering in webclopedia. In *Proceedings of The Ninth Text REtrieval Conference (TREC-9)*. NIST Special Publication.

IBM. 2002. Method for FAQ generation. Online (Quoted 2002-11-04) ⟨http://www.trl.ibm.com/projects/s7710/tm/FAQ/datagen_e.htm⟩.

M. G. Kendall. 1970. *Rank Correlation Methods*. Griffin, London, fourth edition.

Jochen L. Leidner. 2002. Robust question-answering without domain restrictions. (Practical Report), Computer Laboratory, University of Cambridge, Cambridge, UK.

Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. 2002. Towards automatic evaluation of question/answering systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC-2002)*, Las Palmas, Spain.

Andrew McCallum and Fernando Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the Twenty-Third Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

A Some Examples

1. Easy example (a lot of surface overlap):

   Q: *Can you feed a cat a vegetarian diet? A dog?*
   A: *Both animals can be fed a vegetarian diet, although neither is a vegan by nature – dogs are omnivores, and cats are carnivores. While both dogs and cats belong to the class carnivora, this doesn't mean a lot, so does the panda bear and their diet is basically vegan. By nature cats and dogs wouldn't eat anything like what is commonly found in a can of pet food either.*

2. Problematic example with no surface overlap:

   Q: *When a stock's Relative Price Strength Rating is 99, hasn't most of the big money already been made?*
   A: *Not necessarily. A 99 indicates tremendous leadership.*

3. Very indirect link between question and answer (wellness/exercise–pollution, bike–cyclist):

   Q: *Commuting - Do cyclists breathe more pollution than motorists?*
   A: *The sources for this information vary in credibility, but most of it comes directly from published studies or other reputable sources like the Berkeley Wellness letter.*

   1. Exercise will extend your life by about the amount of time you spend doing it. So if you spend an hour on your bike, you've added an hour to your life.

   ...

4. Another easy example:

   Q: *Do airbags really work?*
   A: *Preliminary statistics suggest the following: Airbags work much better than no belts; good 3 point belts alone work much better than Airbags alone, and AirBags + 3 point belts work slightly better than 3 point belts alone. The con to airbags is that some designs tend to burn the driver's hands when venting the byproducts of the explosion that occurs inside the bag, and that some designs (but not all) may knock the driver's hands from the wheel, making retention of control of the vehicle after the bag deflates more difficult.*