Syracuse University
SURFACE

iSchool Faculty Scholarship | School of Information Studies (iSchool)

1-1-2004

Evaluation of Restricted Domain Question-Answering Systems

Anne R. Diekema
Utah State University, diekemar@syr.edu

Ozgur Yilmazel
Syracuse University, oyilmaz@syr.edu

Elizabeth D. Liddy
Syracuse University, liddy@syr.edu

Follow this and additional works at: http://surface.syr.edu/istpub

Part of the Library and Information Science Commons

Recommended Citation
Diekema, Anne R.; Yilmazel, Ozgur; and Liddy, Elizabeth D., "Evaluation of Restricted Domain Question-Answering Systems" (2004). *iSchool Faculty Scholarship.* Paper 52.
http://surface.syr.edu/istpub/52

This Article is brought to you for free and open access by the School of Information Studies (iSchool) at SURFACE. It has been accepted for inclusion in iSchool Faculty Scholarship by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.

Evaluation of Restricted Domain Question-Answering Systems

Anne R. Diekema, Ozgur Yilmazel, and Elizabeth D. Liddy
Center for Natural Language Processing
School of Information Studies
Syracuse University
4-206 Center for Science and Technology
Syracuse, NY 13244
{diekemar,liddy,oyilmaz}@syr.edu

Abstract

Question-Answering (QA) evaluation efforts have largely been tailored to open-domain systems. The TREC QA test collections contain newswire articles and the accompanying queries cover a wide variety of topics. While some apprehension about the limitations of restricted-domain systems is no doubt justified, the strict promotion of unlimited domain QA evaluations may have some unintended consequences. Simply applying the open domain QA evaluation paradigm to a restricted-domain system poses problems in the areas of test question development, answer key creation, and test collection construction. This paper examines the evaluation requirements of restricted domain systems. It incorporates evaluation criteria identified by users of an operational QA system in the aerospace engineering domain. While the paper demonstrates that user-centered task-based evaluations are required for restricted domain systems, these evaluations are found to be equally applicable to open domain systems.

1 Introduction

The Text REtrieval Conference (TREC) organized the first QA evaluation (QA track) in 1999 (Voorhees, 2000) and annual evaluations of this nature are ongoing (Voorhees, to appear). While the tasks and answer requirements have varied slightly from year to year, the purpose behind QA evaluations remains the same: to move from the traditional document retrieval to actual information retrieval by providing an answer to a question rather than a ranked list of relevant documents. The track was originally intended to bring together the fields of Information Extraction (IE) and Information Retrieval (IR). This legacy still continues in the factoid questions that require an IE type answer snippet in response, e.g.: "*What country is the Aswan High Dam located in?"* This style of QA evaluation is spreading with very similar evaluations in Asia (Fukumoto, Kato, Masui, 2003) and Europe (Magnini et al., 2003). Although these evaluations have a multilingual slant, they are strongly modeled after the TREC QA track.

Typical QA systems that participate in these evaluations classify the questions into types which determine what kind of answer is required. After an initial retrieval of documents pertaining to the question, some form of text processing is then applied to identify possible answer sentences in the documents. Sentences that are near or contain keywords from the original question and contain the desired answer pattern are selected for answer extraction. Since it is difficult for systems to determine which part of the sentence is the correct answer, especially if it contains multiple extractions of the desired type, many systems have resorted to redundancy tactics (Banko et al., 2002; Buchholz, 2002). These systems use the Web as an answer verification tool by choosing the answer that appears most often together with the question keywords. While this technique is very successful in open domain evaluations, restricted-domain systems do not have the luxury of using redundancy, making these evaluations inappropriate for systems such as these.

Our QA system participated in the three earlier TREC evaluations, e.g. (Diekema et al., 2002). However, after starting work in the restricted-domain of re-usable launch vehicles, we found that the TREC evaluation no longer suited our system development needs and maintaining two different QA systems was too costly.

2 Restricted-domain system characteristics

The restricted-domain systems of today are different from the toy systems from the early years of QA (Voorhees and Tice, 2000), which might be what first comes to mind when reading the term 'restricted-domain'. Early systems like LUNAR (with a domain somewhat tangentially related to ours, namely lunar archeology) were developed by researchers in the field of natural language understanding. These early systems encoded large amounts of domain knowledge in databases. The restricted-domain systems of today are far less dependent on large knowledge bases and do not aim for language understanding per se. Rather, they use specialized extraction rules on a domain specific collection. The one thing that both types of restricted-domain systems have in common is that they are often developed with a certain goal or task in mind. As we will see later, this task orientation becomes equally important in the evaluation of these QA systems.

An example of a modern-day restricted-domain system is our Knowledge Acquisition and Access System (KAAS) QA system. The KAAS was developed for use in a collaborative learning environment (Advanced Interactive Discovery Environment for Engineering Education or AIDE) for undergraduate students from two universities majoring in aeronautical engineering. While students are working within the AIDE they can ask questions and quickly get answers. The collection against which the questions are searched consists of textbooks, technical papers, and websites that have been pre-selected for relevance and pedagogical value. The KAAS system uses a two-stage retrieval model to find answers in relevant passages. Relevant passages are processed by the Center for Natural Language Processing's eQuery information extraction system using additional rules in the domain of reusable launch vehicles. Users are aided in their question formulations through domain specific query expansions.

3 Initiating a restricted domain evaluation

When it came time to evaluate the KAAS system, we initially defaulted to the TREC style QA evaluation with short, fact-based questions, adjudicated answers to these questions, and a test collection in which to find those answers. This choice of evaluation was not surprising since early versions of our system grew out of that environment. However, it quickly became apparent that this evaluation style posed problems for our restricted-domain, specific purpose system.

Developing a set of test questions was easier said than done. Unlike the open domain evaluations, where test questions can be mined from question logs (Encarta, Excite, AskJeeves), no question sets are at the disposal of restricted-domain evaluators. To build a set of test questions, we hired two sophomore aerospace engineering students. Based on class project papers of the previous semester and examples of TREC questions, the students were asked to create as many short factoid questions as they could, i.e "*What is APAS?*" However, the real user questions that we collected later did not look anything like the short test questions in this initial evaluation set. The user questions were much more complex, e.g. "*How difficult is it to mold and shape graphite-epoxies compared with alloys or ceramics that may be used for thermal protective applications?*" A more in depth analysis of KAAS question types can be found in Diekema et al. (to appear).

Establishing answers for the initial test questions proved difficult as well. The students did fine at collecting the questions that they had while reading the papers, but lacked sufficient domain expertise to establish answer correctness. Another issue was determining recall because it wasn't always clear whether the (small) corpus simply did not contain the answer or whether the system was not able to find it. A third student, a doctoral student in aerospace engineering, was hired to help with these issues. To facilitate automatic evaluation we wanted to represent the answers in simple patterns but found that complex answers are not necessarily suitable for such a representation, even though patterns have proven feasible for TREC systems.

While a newswire document collection for general domain evaluation is easy to find, a collection in our specialized domain needed to be created from scratch. Not only did the collection of documents

take time, the conversion of most of these documents to text proved to be quite an unexpected hurdle as well.

As is evident, the TREC style QA evaluation did not suit our restricted domain system. It also leaves out the user entirely. While information-based evaluations are necessary to establish the ability of the system to answer questions correctly, we felt that they were not sufficient for evaluating a system with real users.

4 User-based evaluation dimensions

Restricted domain systems tend to be situated not only within a specific domain, but also within a certain user community and within a specific task domain. A generic evaluation is neither sufficient nor suitable for a restricted domain system. The environment in which KAAS is situated should drive the evaluation. Unlike many of the systems that participate in a TREC QA evaluation, the KAAS system has to function in real time with real users, not in batch mode with surrogate relevance assessors. This brings with it additional evaluation criteria such as utility and system speed (Nyberg and Mitamura, 2003).

KAAS users were asked in two separate surveys about their use and experiences with the system. The surveys were part of larger scale, cross-university course evaluations which looked at the students' perceptions of distance learning, collaboration at a distance, the collaborative software package, the KAAS, and each participating faculty member. While there was some structure and guidance in the user survey of the QA system, it was minimal and the survey is mainly characterized by the open nature of the responses. There were 25 to 30 students participating in each full course survey, but since we do not have the actual surveys that were turned in, we are not certain as to exactly how many students completed the survey section on the KAAS. However, it appears that most, if not all of the students provided feedback.

Given the free text nature of the responses, it was decided that the three researchers would do a content analysis of the responses and independently derive a set of evaluation dimensions that they detected in the students' responses. Through content analysis of the user responses and follow-up discussion, we identified 5 main areas of importance to KAAS users when using the system: system performance, answers, database content, display, and expectations (see Figure 1). Each of the categories will be described in more detail below.

4.1 System Performance

System Performance is the category that deals with system speed and system availability. Users indicated that the speed with which answers were returned to them mattered. While they did not necessarily expect an immediate answer, they also did not want to wait, e.g. "*took so long, so I gave up*". Whenever users have a question, they want to find an answer immediately. If the system is down or not available to them at that moment, they will not come back later and try again.

Possible system performance metrics are the "answer return rate", and "up time". The answer return rate measures how long it takes (on average) to return an answer after the user has submitted a question. "Up-time" measures for a certain time period how often the system is available (system available time divided by the length of up-time time period).

4.2 Answers

What users find important in an answer is captured in the Answers category. The users not only wanted answers to be accurate, they also wanted them to be complete and, something that is not tested at all in a regular evaluation, applicable to their task. e.g. "*in general what I received was helpful and accurate*", "*it* [the system] *was useful for the Columbia incident exercise…*".

Possible metrics concerning answers are "accuracy or correctness", "completeness", "relevance", or "task suitability". While the first three metrics are used in some shape or form in the TREC evaluations, "task suitability" is not. Perhaps this measure requires a certain task description with a question to test whether the answer provided by the system allowed the user to complete the task.

1 System Performance
- 1.1 Speed
- 1.2 Availability / reliability / upness

2 Answers
- 2.1 Completeness
- 2.2 Accuracy
- 2.3 Relevance
- 2.4 Applicability to task / utility / usefulness

3 Database Content
- 3.1 Authority / provenance / Source quality
- 3.2 Scope /extensiveness / coverage
- 3.3 Size
- 3.4 Updatedness

4 Display (UI)
- 4.1 Input
  - 4.1.1 Question understanding / info need understanding
  - 4.1.2 Querying style
    - 4.1.2.1 Question
      - 4.1.2.1.1 NL query
    - 4.1.2.2 Keywords
    - 4.1.2.3 Browsing
  - 4.1.3 Question formulation assistance
    - 4.1.3.1 Spell Checker
    - 4.1.3.2 Abbreviation recognition
- 4.2 Output
  - 4.2.1 Organization
  - 4.2.2 Feedback Solicitation

5 Expectations
- 5.1 Googleness

Figure 1: User-based evaluation dimensions.

4.3 Database Content

Users also shared thoughts about the Database Content or source documents that are searched for answers. They find it important that these documents are reputable. They also shared concerns about the size of the database, fearing that a limit in size would restrict the number of answerable questions, e.g. "*it needs more documents*". The same is true for the scope of the collection. Users desired extended coverage to ensure that a wide range of questions could be fielded by the collection, e.g. "*I found the data too limited in scope*".

Possible database content metrics are "authority", "coverage", "size", and "up-to-dateness". To measure "authority" one would first have to identify the core authors for a domain through citation analysis. Once that is established, one could measure the percentage of database content created by these core researchers. "Coverage" could be measured in a similar way after the main research areas within a domain are identified. "Size" could simply be measured in megabytes or gigabytes. "Up-to-dateness" could be measured by calculating the number of articles per year or simply noting the date of the most recent article.

4.4 User Interface

The User Interface of a system was also found of importance. Users were critical about the way they were asked to input their questions. They did not always want to phrase their question as a question but sometimes preferred to use keywords, e.g. "*a keyword search would be more useful*". They also expected the system to prompt them with assistance in case they misspelled terms, or when the system did not understand the question, e.g "*sometimes very good at correcting you to what you need, other times not very good*". Users also care about the way in which the results are presented to them and whether the system desires any additional responses from them. They did not like being prompted for feedback on a document's relevance for example, e.g. "*...the 'was this useful' window was disruptive*".

Measuring UI related aspects can be done through observation, questionnaires and interviews and does not typically result in actual metrics but rather a set of recommendations that can be implemented in the next version of the system.

4.5 Expectations

Another interesting aspect of user criteria is Expectations , e.g. "*the documents in the e-Query database were useful, but Google is much faster*". All users are familiar with Google and tend to have little patience with systems that have a different look and feel.

Expectations can be captured by survey so that it can be established whether these expectations are reasonable and whether they can be met.

5 Restricted domain QA Evaluation

If we consider a restricted domain QA system as a system developed for a certain application, it is clear that these systems require a situated evaluation. The evaluation has to be situated in the task, domain, and user community for which the system is developed.

How then can a restricted domain system best be evaluated? We believe that the evaluation should be driven by the dimensions identified by the users as important: system performance, answers, database content, display, and expectations.

The system should be evaluated on its performance. How many seconds does it take to answer a question? Once the speed is known, one can determine how long users are willing to wait for an answer. It may very well be that the answer-finding capability of a system will need to be simplified in order to speed up the system and satisfy its users. Similarly, tests to determine robustness need to be part of the system performance evaluation. Users tend to shy away from systems that are periodically unavailable or slow to a crawl during peak usage hours.

Systems should also be evaluated on their answer providing ability. This evaluation should include measures for answer completeness, accuracy, and relevancy. Test questions should be within the domain of the QA system in order to test the answer quality for that domain. Answers to certain questions require a more fine-grained scoring procedure: answers that are explanations or summaries or biographies or comparative evaluations cannot be meaningfully rated as simply right or wrong. The answer providing capability should be evaluated in light of the task or purpose of the system. For example, users of the KAAS are learners in the field and are not well served with exact answer snippets. For their task, they need answer context information to be able to learn from the answer text.

The evaluation should also include measures of the Database Content. Rather than assuming relevancy of a collection, it should be evaluated whether the content is regularly updated, whether the contents are of acceptable quality to the users, and whether the coverage of the restricted domain is extensive enough.

Another system component that should be evaluated is the User Interface. Is the system easy to use? Does the interface provide clear guidance and/or assistance to the user? Does it allow users to search in multiple ways?

Finally, it may be pertinent to evaluate how far the system goes in living up to user expectations. Although it is impossible to satisfy everybody, the system developers need to know whether there is a

large discrepancy between user expectations and the actual system, since this may influence the use of the system.

6 Cross-fertilization between evaluations

How different are restricted-domain evaluations from open-domain evaluations? Are they so diametrically opposed that restricted-domain systems require separate evaluations from open-domain systems and vice versa? As pointed out in Section 1, we stopped participating in the TREC QA evaluations because that evaluation was not well suited to our restricted-domain system. However, we regretted this as we believe we could, nevertheless, have gained valuable insights.

Clearly, open-domain systems would benefit from the evaluation dimensions discussed in Section 4. The difference would be that the test questions used for evaluation would be general rather than tailored to a specific domain. Additionally, it may be harder to evaluate the database content (i.e. the collection) for a general domain system than would be the case for restricted-domain systems.

To make open-domain evaluations more applicable to restricted-domain systems, they could be extended to include metrics about answer speed, and the ability of answering within a certain task. For example, the evaluation could include system performance to get an indication as to how much processing time, given certain hardware, is required in getting the answers. As for answer correctness itself, it may be interesting to require extensive use of task scenarios that would determine aspects such as answer length and level of detail. It may also be desirable to evaluate runs without redundancy techniques separately. Ideally, users would be incorporated into the evaluation to assess the user interface and the ability of the system to assist them in completion of a certain task.

7 Summary

Restricted-domain systems require a more situated evaluation than is generally provided in open-domain evaluations. A restricted-domain evaluation extends beyond domain specific test questions and collections to include the user and their task. Users of the restricted-domain KAAS system identified five areas that should be included in an evaluation: System Performance, Answers, Database Content, Display, and Expectations. Most of these evaluation dimensions could be applied to open-domain evaluations as well. Adding system performance metrics (such as answer speed) and specific task requirements may allow a convergence between open domain and restricted domain QA evaluations.

Acknowledgements

Funding for this research has been jointly provided by NASA, NY State, and AT&T.

References

Banko, M., Brill, E., Dumais, S. and Lin, J. 2002. AskMSR: Question answering using the worldwide Web. In *Proceedings of the 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, March 2002, Palo Alto, California.

Buchholz, S. 2002. Using Grammatical Relations, Answer Frequencies and the World Wide Web for TREC Question Answering. In: E. M. Voorhees and D. K. Harman (Eds.), *The Tenth Text REtrieval Conference (TREC 2001)*, volume 500-250 of NIST Special Publication, Gaithersburg, MD. National Institute of Standards and Technology, 2002, pp. 502-509.

Diekema, A.R., Chen, J., McCracken, N, Ozgencil, N.E., Taffet, M.D., Yilmazel, O. and Liddy, E.D. 2002. Question Answering: CNLP at the TREC-2002 Question Answering Track. In: *Proceedings of the Eleventh Text Retrieval Conference (TREC-2002)*. E.M. Voorhees and D.K. Harman (Eds.). Gaithersburg, MD: Department of Commerce, National Institute of Standards and Technology, 2002.

Diekema, A.R., Yilmazel, O., Chen, J., Harwell, S., He, L., and Liddy, E.D. Finding Answers to Complex Questions. To appaer. In Maybury, M. (Ed.) *New Directions in Question Answering*. AAAI-MIT Press.

Fukumoto, J., Kato, T., and Masui, F. 2003. Question Answering Challenge (QAC-1): An Evaluation of Question Answering Tasks at the NTCIR Workshop 3. In *Proceedings of the AAAI Spring Symposium*: New Directions in Question Answering, p.122-133, 2003.

Magnini, B., Romagnoli, S., Vallin, A., Herrera, J. Peñas, A., Peinado, V., Verdejo, F., M. de Rijke, The Multiple Language Question Answering Track at CLEF 2003. In Carol Peters (Ed.), *Working Notes for the CLEF 2003 Workshop*, 21-22 August, Trondheim, Norway, 2003.

Nyberg E. and T. Mitamura. 2002. Evaluating QA Systems on Multiple Dimensions. In *Proceedings of LREC 2002 Workshop on QA Strategy and Resources*, May 28th, Las Palmas, Gran Canaria.

Voorhees, E.M. 2003. DRAFT Overview of the TREC 2003 Question Answering Track. To appear in *Proceedings of TREC 2003*. Gaithersburg, MD, NIST, to appear.

Voorhees, E.M. Overview of the TREC-8 Question Answering Track Report. In *Proceedings of TREC-8*, 77-82. Gaithersburg, MD, NIST, 2000.

Voorhees, E.M. & Tice, D.M. Implementing a Question Answering Evaluation. In Proceedings of LREC'2000 Workshop on Using Evaluation within HLT Programs: Results and Trends. 2000.