We thank The MITRE Corporation for their administrative support of the workshop.
We thank the Advanced Research and Development Activity (ARDA) Northeast Regional Research Center (NRRC) for their technical support in the organization of the workshop.

nrrc.mitre.org

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730

MITRE
www.mitre.org

Question Answering: Strategy and Resources

Workshop Program

Tuesday May 28, 2002

Palacio de Congreso de Canarias

8:00 a.m. Welcome and Introduction
Mark Maybury, The MITRE Corporation, USA

8:15 a.m. Invited Keynote – What's the Next Big Thing in Question Answering?
John Lowe, UC Berkeley and formerly Ask Jeeves, Inc.

QA Evaluation

9:00 a.m. The Evaluation of Question Answering Systems:
Lessons Learned from the TREC QA Track
Ellen M. Voorhees, National Institute of Standards and Technology, USA

9:25 am Why are People Asking these Questions?
A Call for Bringing Situation into Question-Answering System Evaluation
Elizabeth D. Liddy, Syracuse University, USA

9:50 am A Curriculum-based Approach to a QA Roadmap
John Prager, IBM, USA

10:15 am Evaluating QA Systems on Multiple Dimensions
Eric Nyberg and Teruko Mitamura, Carneige Mellon University, USA

10:40 am Evaluation Roadmap Discussion
All

11:00 – 11:20 a.m. Morning Break

11:20 am QA Roadmap
All

13:00 p.m. Lunch and Demos

Inference

14:30 p.m. Inference in Question Answering
Bonnie Webber, University of Edinburgh, Scotland
Claire Gardent, CNRS-LORIA, France, and
Johan Bos, University of Edinburgh, Scotland

Applications

14:55 p.m. The Challenge of Technical Text
Fabio Rinaldi, James Dowdall, and Michael Hess,
University of Zurich, Switzerland

15:20 p.m. Question Answering in the Infosphere: Semantic Interoperability and Lexicon Development
Paul Thompson and Steven Lulich, Dartmouth College, USA

Multilingual and Multiperspective QA

15:45 p.m. Summarization Based Japanese Question and Answering System for Newspaper Articles
Yohei Seki and Ken'ichi Harada, Keio University, Japan

16:10 p.m. Multiple Perspective and Temporal Question Answering
James Pusteyovsky, Brandeis University, USA,
Janice Wiebe, University of Pittsburgh, and
Mark Maybury, The MITRE Corporation, USA

16:35 - 17:00 p.m. Afternoon Break

17:00 p.m. Final Group Roadmap Session on Question Answering
All

19:00 p.m. Close

Lunch Time Demos

Question Answering system for POLISH (POLINT)
Zygmunt Vetulani, Adam Mickiewicz University, Poland

QA from Technical Manuals
Fabio Rinaldi, James Dowdall, and Michael Hess, University of Zurich, Switzerland

Statistical Web-based Question Answering
Drago Radev, University of Michigan, USA

Table of Contents

Page

Evaluation

Inference

Applications

Multiperspective and Temporal

Table of Contents (Concluded)

Page

Multilingual

Preface

Effective question answering is crucial for proper human-system interaction, and systems that can answer questions help to realise the artificial intelligence dream of a machine as a collaborative agent. Question answering draws on many capabilities including information retrieval, language processing, and human computer interaction. Effective question interpretation and answer generation require technologies that index, retrieve, transcribe, extract, translate, and summarize. Question answering can occur in multilingual, multimedia, and multiparty environments. The applicability of question answering ranges across all domains and tasks including learning, playing and conducting business.

Topics in the call for papers, listed in its entirety at www.lrec-conf.org/lrec2002/lrec/wksh/QuestionAnswering.html, included but were not limited to:

- Roadmaps for question answering language resources (LR) and scientific algorithm developments
- Existing question answering language resources
- Guidelines, standards, specifications, models and best practices for question answering LR
- Methods, tools, and procedures for the acquisition, creation, management, access, distribution, and use of question answering LR
- LR and evaluation and benchmarking of question answering systems and algorithms for tasks including:
  - Advanced question analysis
  - Answer discovery and integration
  - Answer explanation and presentation generation
  - Interactive question answering
- LR and evaluation methods for advanced question answering challenges, including but not limited to:
  - Question answering from heterogeneous (structure, unstructured, semi-structured) sources.
  - Multimedia (e.g., text, graphics, audio, video) and Multimodal (i.e., auditory, visual) question answering
  - Multilingual question answering
  - Answering questions from multiple perspectives
    (e.g, political/economic/legal, local/national/international)
- Question answering components, architectures or instrumentation that facilities evaluation

This one day workshop aims to refine a roadmap (www-nlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc) for question answering applications and the methods for the creation and evaluation of resources for the next decade in support of these systems. The workshop will draw upon research in the TREC Q&A track, the AQUAINT program, and efforts planned for the ARDA Northeast Regional Research Center (NRRC). Participants will help formulate grand challenge problems, discuss possible data sets and/or evaluation metrics/methods, articulate the role of and necessary advances in resources and evaluation to solve these challenges, as well as strategize jointly about the most effective and efficient path forward. Possible joint products arising from the workshop include:

- A list of existing resources and ones under development (with planned release dates)
- Joint formulation of a Q&A roadmap, motivated by ARDA's roadmap (www-nlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc)
- List of evaluation methods and benchmarks of question answering systems
- List of unresolved research problems and/or areas in question answering
- Shared knowledge of research groups and efforts

Table 1 below lists the papers included in the workshop, the primary focus of the article, question answering issues addressed in the papers, and the kinds of sources focussed on.

TABLE 1. Overview of Contributions

Primary Focus	Title	Technical Issues Addressed	Sources	Author(s)
Evaluation	The Evaluation of Question Answering Systems: Lessons Learned from the TREC QA Track	Evaluation, benchmarking, TREC, existing resources	Newspapers	Ellen Voorhees
Evaluation	Why are People Asking these Questions? A Call for Bringing Situation into Question-Answering System Evaluation	Evaluation, Application	Statistical tables; Mechanical Engineering Papers; Web Sites	Elizabeth Liddy
Evaluation	A Curriculum-based Approach to a QA Roadmap	Question answering, roadmap, evaluation, resources, computational linguistics	documents	John Prager
Evaluation	Evaluating QA Systems on Multiple Dimensions	Ambiguity resolution, evaluation methodology	TREC QA track corpora (Future: Chinese and Japanese newswire)	Eric Nyberg, Teruko Mitamura
Inference	Inference in Question Answering	Inference, question-answering, test suites	documents	Bonnie Webber, Claire Gardent, Johan Bos
Applications	The Challenge of Technical Text	Question Answering, Technical Domains, Technical Terminology, XML	Technical Manuals	Michael Hess, James Dowdall, Fabio Rinaldi
Applications	Question Answering in the Infosphere: Semantic Interoperability and Lexicon Development	question answering systems, query optimization, semantic interoperability, lexicons, connectivistic databases	Sensors (Plans to address documents)	Steven Lulich, Paul Thompson
Multiperspective and Temporal	Multiple Perspective and Temporal Question Answering	question answering systems, multiperspectives, temporal expressions, events	Newspapers	James Pustejovsky, Jan Wiebe, Mark Maybury
Multilingual	Summarization Based Japanese Question and Answering System for Newspaper Articles	Japanese Q A System, summarization technique, information fusion from multiple newspaper articles	Newspapers	Yohe Seki, Ken'ichi Harada
Multilingual	Question Answering system for POLISH (POLINT) and its language resources	question answering, language resources, grammars, dialogue corpora, Polish language	Question-answer corpus	Zygmunt Vetulani

Any international workshop demands the selfless contributions of many individuals. We first thank the authors and participants for their important contributions. We next thank the Organizing Committee for their time and effort in providing detailed and high quality reviews and counsel. And we thank Paula MacDonald at MITRE for her tireless and excellent administrative workshop support.

Workshop Organiser

Mark Maybury
The MITRE Corporation
maybury@mitre.org

Workshop Program Committee

Sanda Harabagiu
University of Texas at Austin
sanda@cs.utexas.edu

Liz Liddy
University of Syracuse
liddy@syr.edu

John Prange
Advanced Research and Development Activity (ARDA)
jprange@nsa.gov

Karen Sparck Jones
University of Cambridge
sparckjones@cl.cam.ac.uk

Ellen Voorhees
National Institute of Standards and Technology (NIST)
ellen.voorhees@nist.gov

Author Index

Page

Invited Keynote

What's the Next Big Thing in Question Answering?

John B. Lowe*
UC Berkeley / LACITO Paris / Formerly of Ask Jeeves, Inc. and W3C AC

Question answering as a computational craft has been around just long enough to have a colorful history and a track record of successes and failures. This checkered past provides object lessons and touchstones in the quest for an effective roadmap for further research.

Early attempts to answer questions by computer -- valiant, creative, and ambitious -- enjoyed limited success due to a number of constraints both foreseen and unforeseen. The importance of certain now well-understood principles governing conversation (e.g. Austin 1962, Grice 1957, 1969, Searle 1969, Dreyfus 1972, 1979) and indeed linguistics generally (Harris 1995, Lakoff 1989) were only dimly appreciated three or four decades ago. Computational resources, both hard and soft, were scarce -- NLP and IR accessories (tokenizers, POS taggers, and parsers, for example) which today are taken for granted often did not exist or had to be re-invented in each instance. While the early research program did not always realize its ambitious goals, a large number of approaches were tried and to some extent evaluated. Much was learned.

The advent of the web and other technological developments of the mid- to late-nineties injected new vigor into the question-answering field. For the first time in a long time commercial and intellectual opportunity was seen in open-domain question answering and a number of companies, both startups and established firms, rushed into the fray.

Yet another wave of twenty-first century technology promises to both enable and challenge future QA systems. The first of these is the so-called Semantic Web. A gleam in the eye of the web inventor Tim Berners-Lee and others for some time now (Dertouzos 2001), the Semantic Web is to be partially enabled by Web Services, another initiative which is now the subject of a turf war between major players in information services.

If the mark of a mature research programme is a group of focused researchers working together within an accepted paradigm judged on the basis of impartial evaluation criteria then the question answering field is mature. Nevertheless, even the best systems today handle only a few classes of the known range. Furthermore, the prospects for general solutions are anxiously dependent on developments in other fields as disparate as linguistic semantics, sociolinguistics, and knowledge representation (KR).

The roadmap presented as part of this workshop demonstrates the maturity of the field. It also indicates that question answering is at a crossroads and how important it is to pick the right path. As part of my talk, I will critique some of the major points and suggestions made therein, with an eye to clarifying their achievability and the consequences of success.

* Department of Linguistics
1203 Dwinelle Hall
University of California at Berkeley
Berkeley, CA 94720-2650
voice: (510) 643-9910
fax: (208) 567-2107
email: jblowe@socrates.berkeley.edu

The Evaluation of Question Answering Systems: Lessons Learned from the TREC QA Track

Ellen M. Voorhees

National Institute of Standards and Technology
100 Bureau Dr. STOP 8940
Gaithersburg, MD 20899-8940
ellen.voorhees@nist.gov

Abstract

The TREC question answering (QA) track was the first large-scale evaluation of open-domain question answering systems. In addition to successfully fostering research on the QA task, the track has also been used to investigate appropriate evaluation methodologies for question answering systems. This paper gives a brief history of the TREC QA track, motivating the decisions made in its implementation and summarizing the results. The lessons learned from the track will be used to evolve new QA evaluations for both the track and the ARDA AQUAINT program.

1. The TREC QA Task

TREC is a workshop series designed to provide the infrastructure required for large-scale evaluation of text retrieval and related technologies (National Institute of Standards and Technology, 2002). A "track" for the investigation of question answering systems was introduced into TREC-8 in 1999, and has been run each year since then for a total of three times to date.

The original motivation for the track was to foster research that would move retrieval systems closer to *information* retrieval systems rather than *document* retrieval systems. Document retrieval systems' ability to work in any domain was considered an important feature to maintain. At the same time, the technology that had been developed by the information extraction community appeared ready to exploit. Thus the task for the TREC-8 QA track was defined such that both the information retrieval and the information extraction communities could work on a common problem. The task was very similar to that used in the MURAX system (Kupiec, 1993), which used an on-line encyclopedia as a source of answers for closed-class questions, except that the answers were to be found in a large corpus of documents rather than an encyclopedia. Since the documents consisted mostly of newswire and newspaper articles, the domain was essentially unconstrained. However, only closed-class questions were used, so answers were generally entities familiar to information extraction systems.

Participants were given a document collection and a test set of questions. The questions were fact-based, short-answer questions such as *How many calories are there in a Big Mac?* and *Where is the Taj Mahal?*. Each question was guaranteed to have at least one document in the collection that answered it. For each question, participants returned a ranked list of five [*document-id*, *answer-string*] pairs such that each answer string was believed to contain an answer to the question. Answer strings were limited to either 50 or 250 bytes depending on the run type. Human assessors read each string and made a decision as to whether or not the string contained an answer to the question in the context provided by the document. Individual questions received a score equal to the reciprocal of the rank at which the first correct response was returned (or 0 if none of the five responses contained a correct answer). The score for a run was the mean of the individual questions' reciprocal ranks.

2. Evaluation

The TREC QA evaluations have been based on the assumption that different people will have different ideas of what constitutes a correct answer. This assumption was demonstrated to be true during the TREC-8 evaluation. For TREC-8, each question was independently judged by three different assessors. The separate judgments were combined into a single judgment set through adjudication for the official track evaluation, but the individual judgments were used to measure the effect of differences in judgments on systems' scores. Assessors had legitimate differences of opinion as to what constituted an acceptable answer even for the deliberately constrained questions used in the track. Two prime examples of where such differences arise are the completeness of names and the granularity of dates and locations.

Fortunately, as with document retrieval evaluation, the relative scores between QA systems remain stable despite differences in the judgments used to evaluate them (Voorhees and Tice, 2000). The lack of a definitive answer key does mean that evaluation scores are only meaningful in relation to other scores on the same data set. Absolute scores *do* change if you use a different set of judges, or a different set of questions. However, this is an unavoidable characteristic of QA evaluation. Since assessors' opinions of correctness differ, the eventual end users of the QA systems will have similar differences of opinion, and an evaluation of the technology must accommodate these differences.

A [*document-id*, *answer-string*] pair was judged correct if, in the opinion of the NIST assessor, the answer-string contained an answer to the question, the answer-string was responsive to the question, and the document supported the answer. If the answer-string was responsive and contained a correct answer, but the document did not support that answer, the pair was judged "Not supported" (except in TREC-8 where it was marked correct). Otherwise, the pair was judged incorrect. Requiring that the answer string be responsive to the question addressed a variety of issues. Answer strings that contained multiple entities of the same semantic category as the correct answer but did not indicate which of those entities was the actual answer (e.g., a list of names in response to a who question) were judged as incorrect. Certain punctuation and units were also required. Thus "5 5 billion" was not an acceptable substitute for "5.5 billion", nor was "500" acceptable when the correct answer was "$500". Finally, unless the question specifically stated otherwise, correct responses for questions about a famous entity had to refer to *the* famous entity and not to imitations, copies, etc. For example, two TREC-8 questions asked for the height of the Matterhorn (i.e., the Alp) and the replica of the Matterhorn at Disneyland. Correct responses for one of these questions were incorrect for the other.

One of the problems of judging entire strings for correctness is that the resulting judgments do not create a reusable test collection. The primary way TREC has been successful in improving document retrieval performance is by creating appropriate test collections for researchers to use when developing their systems. While creating a large collection can be time-consuming and expensive, once it is created researchers can automatically evaluate the effectiveness of a retrieval run. Unfortunately, different QA runs very seldom return exactly the same answer strings, and it is quite difficult to determine automatically whether the difference between a new string and a judged string is significant with respect to the correctness of the answer. Word recall (Breck et al., 2000) and answer patterns (Voorhees and Tice, 2000) have been suggested as ways of approximating a reusable test collection. These approximations have been well-correlated with human judgments in tests to date, but they mis-judge broad classes of responses. Since the mis-judged classes are frequently the cases that are difficult for the original systems being evaluated, the approximations are likely to be less useful as QA systems continue to improve. Nonetheless, they are currently helpful for providing quick feedback as to the relative quality of alternate question answering techniques.

3. Retrieval Results

The most accurate of the TREC-8 systems were able to answer more than 2/3 of the questions. When an answer was found at all, it was likely to be highly ranked. Not surprisingly, allowing 250 bytes in a response is an easier task than limiting responses to 50 bytes. Indeed, traditional passage retrieval techniques are effective when a response as long as 250 bytes is acceptable (Singhal et al., 2000).

Most participants used a version of the following general approach to the question answering problem. The system first attempted to classify a question according to the type of its answer as suggested by its question word. For example, a question that begins with "who" implies a person or an organization is being sought, and a question beginning with "when" implies a time designation is needed. Next, the system retrieved a small portion of the document collection using standard text retrieval technology and the question as the query. The system performed a shallow parse of the returned documents to detect entities of the same type as the answer. If an entity of the required type was found sufficiently close to the question's words, the system returned that entity as the response. If no appropriate answer type was found, the system fell back to best-matching-passage techniques.

The absolute value of the scores for TREC-9 systems was lower than for TREC-8, but in fact the systems were significantly improved (the TREC-9 task was much more difficult as described below). The improvement in QA systems came from refinements to the individual steps of the general strategy described above rather than an entirely new approach. TREC-9 systems were better at classifying questions as to the expected answer type, and used a wider variety of methods for finding the entailed answer types in retrieved passages. Many systems used WordNet (Fellbaum, 1998) as a source of related words for the initial query and as a means of determining whether an entity extracted from a passage matched the required answer type.

Many systems continued to refine this approach in the TREC 2001 track. However, the TREC 2001 track also saw a resurgence of approaches that relied on simpler pattern matching methods using very large corpora (generally the web) rather than sophisticated language processing. The idea exploited in the massive data approach is the fact that in a large enough data source a correct answer will usually be repeated often enough to distinguish it from the noise that happens to occasionally match simple patterns.

4. Creating a Question Set

The manner in which the test set of questions was assembled has had a big effect on the results of the QA evaluations. In TREC-8, the majority of the questions were created expressly for the track, and thus tended to be back-formulations of a statement in a document. In TREC-9, the questions were selected from an Encarta log that contained actual questions, and a raw Excite log. Since the raw Excite log did not contain many grammatically well-formed questions, NIST staff used the Excite log as a source of ideas for actual questions. All the questions were created without looking at any documents. The resulting test set of questions was much more difficult than the TREC-8 set, mainly because the TREC-9 set contained many more high-level questions such as *Who is Colin Powell?*. For the TREC 2001 track, the source of the questions was again web logs, this time from Microsoft and AskJeeves who automatically filtered their raw logs to select queries containing question words. NIST did additional human filtering of the logs, selecting a final set of 500 questions. Except for some tweaking of the spelling and punctuation, the questions were as they appeared in the log.

NIST has made no attempt to control the relative number of different types of questions in the test set from year to year. Instead, the distribution of question types in the final test set has reflected the distribution in the source of questions. The TREC 2001 test set contained a dramatically greater proportion of definition questions than the previous years. While a large fraction of definition questions is "real" in that the filtered MSNSearch and AskJeeves logs contain many definition questions, there are easier ways to find the definitions of terms than searching for a concise definition in a corpus of news articles. As a result, NIST intends to exert somewhat more control over the distribution of question types in future tracks.

5. Other Tasks

Each of the TREC QA tracks have differed slightly from one another in ways other than the manner in which the test set of questions was assembled. To investigate whether QA systems are robust to the variety of different ways a question can be phrased, the TREC-9 question set contained 500 questions drawn from the logs, plus an additional 193 questions that were syntactic variants of an original question. For example, the test set contained four variants for the question *What is the tallest mountain?*: *What is the world's highest peak?*, *What is the highest mountain in the world?*, *Name the highest mountain.*, and *What is the name of the tallest mountain in the world?*. Systems that parsed questions into a common representation generally had fewer differences in their responses to question variants than did systems that relied on templates to classify questions by answer types. Overall, however, most variant sets showed little variability in the average score obtained by the different participants, indicating that the difficulty of obtaining the underlying information being sought dominated the results. For the few variant sets that did have a wide range of average scores, the difference was usually caused by different word choices in the variants. For example, the original question *Where was Poe born?* had a much higher average score than any of the variants that all asked for Poe's birthplace.

The TREC 2001 track contained three tasks, the main task, the list task, and the context task. The main task was similar to the previous tracks except questions were not guaranteed to have an answer in the document collection. Recognizing that there is no answer is a challenging task, but it is an important ability for operational systems to possess since returning an incorrect answer is usually worse than not returning an answer at all. The majority of the systems did not attempt to do no-answer processing.

The list task was designed to require systems to assemble an answer from information located in multiple documents. Such questions are harder to answer than the questions used in the main task since information duplicated in the documents must be detected and reported only once. The test set of questions consisted of 25 questions constructed by NIST assessors, each of which specified a target number of instances of a particular kind of information to be retrieved. For example, *What are 9 novels written by John Updike?* was one of the question used in the task. Systems returned an unordered list of [*document-id*, *answer-string*] pairs where each pair represented a single instance. The list could contain no more than the target number of instances. Each individual instance was judged as in the main task. The evaluation metric used was average accuracy, where the accuracy for a single question was the number of distinct correct instances retrieved divided by the target number of instances. The best performing system had an average accuracy of 76%, suggesting that the list task as defined is feasible with current technology.

The context task was intended to test systems' ability to track discourse objects (context) through a short series of questions. However, system performance was so dominated by whether the system could answer the particular type of question posed that differences in ability to track context were not detectable. More research is needed to create an evaluation that actually measures a system's ability to track context.

6. Future Evaluations

The TREC QA track has stimulated research on open-domain question answering and has created a foundation on which future evaluations can build. The data used in the TREC tracks, including questions, answer patterns, sentences containing answers, and evaluation scripts are available on the TREC web site (National Institute of Standards and Technology, 2002).

To date, the TREC QA track has used only factoid questions. This allows the evaluation of the answers to be judged using a binary decision of correct/incorrect. While assessors' opinions as to correctness differ even for this basic question type, evaluation is at least stable in that the relative quality of different QA systems is not materially affected by such differences in opinion. Answers to other types of questions require a more fined-grained scoring procedure: answers that are explanations or summaries or biographies or comparative evaluations cannot be meaningfully rated as simply right or wrong. The appropriate dimensions along which such answers should be judged, scoring mechanisms that reflect quality in those dimensions, and the stability of evaluations using those scoring mechanisms all need to be investigated.

The impact the way in which the test set of questions was assembled has had on system effectiveness in TREC illustrates the balancing of tensions required to create an effective test. One the one hand, careful selection of questions allows specific features of QA systems to be tested, enabling crisper conclusions to be drawn. On the other hand, such selection generally reduces the realism of the test. Designed tests usually lack the diversity of subject matter, vocabulary, and sentence constructions that are represented in large samples of naturally occurring questions. Such diversity can be particularly important to include in initial evaluations when the features that affect performance on the task are not well understood.

The TREC track will continue, with the goal of increasing the kinds and difficulty of the questions that systems can answer. The main task in the TREC 2002 will focus on having systems retrieve the *exact* answer. In past tracks, responses could contain extraneous information and still be judged correct provided the extraneous information was not distracting. Such fuzziness in the definition of correct was used in the first track when it was unclear what the systems' abilities were, and it has remained. However, the fuzziness is masking true differences in systems in the final scores. Forcing system to be precise will not only allow scores to better distinguish among technologies, but also improve QA technology.

An evaluation effort related to the TREC QA track is the new AQUAINT (Advanced QUestion and Answering for INTelligence) program sponsored by ARDA (Advanced Research and Development Activity), a research center within the U.S. Department of Defense (see `http://www.ic-arda.org/`). The main focus of AQUAINT is to move beyond factoid questions, including the investigation of scoring mechanisms for complex answer types. Within the first year of AQUAINT (2002), AQUAINT contractors and NIST will run pilot studies to experiment with different measures.

7. References

Eric Breck, John 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 the Second International Conference on Language Resources and Evaluation (LREC-2000)*, volume 3, pages 1495–1500.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Julian Kupiec. 1993. MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In Robert Korfage, Edie Rasmussen, and Peter Willett, editors, *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 181–190. Special issue of the SIGIR FORUM.

National Institute of Standards and Technology. 2002. The Text REtrieval Conference web site. `http://trec.nist.gov`.

Amit Singhal, Steve Abney, Michiel Bacciani, Michael Collins, Donald Hindle, and Fernando Pereira. 2000. AT&T at TREC-8. In E.M. Voorhees and D.K. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*. NIST Special Publication 500-246. Electronic version available at `http://trec.nist.gov/pubs.html`.

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*, pages 200–207, July.

Why are People Asking these Questions? :
A Call for Bringing *Situation* into Question-Answering System Evaluation

Elizabeth D. Liddy
Center for Natural Language Processing
School of Information Studies
Syracuse University
Syracuse, New York 13210
315-443-5484 (v) 315-443-5806 (f)
liddy@syr.edu; www.cnlp.org

Introduction

I believe that in order for the field of Question-Answering (QA) to evolve to the stage where it will provide maximum utility, the environment in which a QA system is to be used should become a parameter in the evaluation of QA systems. That is, the current evaluation paradigm is becoming restrictive and may well push development in a single direction that will not produce systems that will prove useful in multiple environments. Even a quick review of the potential scenarios in which QA can be utilized suggests two key facts: 1) what is considered *'a useful answer'* in one context might not be useful in another, and; 2) currently permissible methods that systems can utilize to determine correct answers are not feasible in many real world QA environments. This paper will advance this position and suggest a range of situational dimensions that should be considered for inclusion in the QA evaluation roadmap.

QA Evaluation

While there was significant early research in Question Answering in the fields of logic and linguistics (Belnap, 1963; Belnap & Steel, 1976), automatic QA was first focused on in a large-scale evaluation framework in the TREC Conferences, beginning with TREC-8 in 1999 (Voorhees & Tice, 1999). The paradigm established in TREC-8 and continued in the next two TREC Conference QA tracks is simple fact-based, short-answer questions. Initially, answer strings were limited to either 50 or 250 bytes depending on the run type. In TREC-10, the 250 byte condition was eliminated and the list task was added. The list task consisted of 25 questions which specified the number of unique responses to be retrieved, e.g., *"What four countries are the top producers of wheat in the world?"* All other parameters of the main QA task remained the same (Voorhees, 2001).

Discussion at the TREC 2001 Workshop on QA intimated that the QA track in TREC 2002 will accept as correct only fragments which contain the minimal answer to the question. Any explanatory text, even if within the 50 byte limit, will cause the answer to be marked as incorrect. Additionally, the practice introduced in TREC 2001 of a system first determining the most frequent potential answer by searching the web, and then finding a document in the TREC collection which contained that answer fragment will continue to be allowed.

Potential Problems

The need for a more refined evaluation of answer strings was evident from some sample answers shown at the Workshop as they contained text that was non-contributory to the answer and just happened to contain the correct answer that had been provided to the relevance assessors. However, this was not always true. In some instances, the additional text can be argued to have provided useful supportive or confirmatory information. The potential problem I see in the requirement of a minimal answer is that this evaluation paradigm, which does not permit the inclusion of supporting information that might be useful in some QA scenarios, will foster the development of systems which will be useful in only a subset of the contexts in which QA systems are truly needed.

Furthermore, the decision to allow systems to utilize redundancy on the web to select answers (Brill et al, 2001) will also foster methods that may not be useable in many QA environments. It is highly unlikely that the redundancy approach will transfer to QA systems that are developed for specialized resource environments. While the simple factoid questions for which multiple instances of responses can be found on the web have been the norm in the QA track, this is not typical in other environments for which QA systems provide great utility.

While the existing QA evaluation scenario has utilized very simple questions, has focused on a narrow definition of length of useful answer to the exclusion of other issues, and has permitted the use of a method of determining an answer which will not work in other than the simple query environment, some QA system builders have begun to call for an evaluation paradigm that considers dimensions above and beyond correctness (Breck et al, 2000). We strongly agree with this view and encourage the discussion of a broader evaluation paradigm for the QA Roadmap that will take into account the wide range of environments in which QA is already providing an essential service.

Range of Possible QA Environments

Consider the three following real-life environments for which we have developed QA systems. In each of these environments, the collection, the type of queries, how the system determines answers, and what constitutes an acceptable answer formulation for the user vary dramatically.

1. Scientific Questions from Undergraduate Students

We have developed a QA system (Liddy, 2001) with funding from NASA and AT&T for use within a collaborative learning environment for undergraduate students from two universities majoring in aeronautical engineering who are taking courses that are taught within the AIDE (Advanced Interactive Discovery Environment for Engineering Education). The students are able to ask questions and quickly get answers in the midst of their hands-on collaborations within the AIDE. The collection against which the questions are searched consists of textbooks, technical papers, and websites that have been pre-selected for their relevance and pedagogical value. We are currently working towards the addition of transcripts of class lectures and accompanying power point slides. The students questions are not typically simple factoid questions, but tend more towards '*Why*' and '*How*' questions and require more than bare answers, such as:

? *How do ablating materials minimize energy conducted into a RLV?*
? *What are the changes made to the design of the Shuttle SRM since the Challenger Accident?*
? *How are malfunctions detected for the pitch and yaw gimbal actuators of the space shuttle OMS engines?*

Answers are provided in increasing window sizes, allowing the student to gradually expand the amount of text by mouse-clicking from 'answer-providing passage', to paragraph (s) containing the 'answer-providing passage' to full document(s) containing the 'answer-providing passage'. The system is currently undergoing user testing. The U S Army has funded us to create a similar capability for the students in the Army's intel training programs. They share NASA's vision that work in the future will consist largely of virtual collaborative situations in which questions that arise will need to be answered electronically from selected sources.

2. Citizens’ Search for Statistical Information

Naïve users need to access statistical information, but frequently do not have the sophisticated understanding required in order to translate their information needs into structured database queries using the controlled vocabulary which are currently required. However, these users can articulate quite straightforwardly in their own terms what they are looking for. One approach to satisfying the masses of citizens with needs for statistical information is to automatically map their natural language expressions of their information needs into the metadata structure and terminology that defines and describes the content of statistical tables. To accomplish this goal, under funding from NSF’s Digital Government Initiative (http://istweb.syr.edu/~tables/), we undertook an analysis of 1,000 user email queries seeking statistical information from federal agencies which provide internet access to their statistical tables. Our goal was to understand the dimensions of interest in naïve users’ typical statistical queries, as well as the linguistic regularities that could be captured in a statistical-query sublanguage grammar. We developed an ontology of query dimensions using this data-up analysis of the queries and extended the ontology where necessary with values from actual tables. We proceeded to develop an NLP statistical-query sublanguage grammar that enabled the system to semantically parse users’ queries and produce a template-based internal query representation which was then mapped to the tables’ metadata, in order to retrieve relevant tables which were displayed to users with the relevant cell’s value highlighted (Liddy & Liddy, 2001). Typical queries were:

? *I am trying to find the percentage of women in the workforce from the years 1900 to 1998*.

? *I want to know how many people worked for small businesses last year.*

? *What was the average amount of time women spent on housework per week in 1900; 1950; 1995?*

This project made it eminently clear that the situation predicts the nature of the questions, the resources searched, and the acceptable answer formulation.

3. Speech-based Inquiries in Travel and Tourism

In an exciting project in the commercial world, we worked with a speech understanding technology company to provide answers to travelers who were planning Caribbean vacations via interaction with a voice-activated system. While the business idea was well-researched, the current status of speech-understanding technology was not, and the corporation failed to pull off the application. However, I mention it here because it introduces a third and very different set of users, answer-providing resources, and answer formulation in which appropriate supporting detail is essential.

? *We're looking for a family resort in the Caribbean with baby sitting, other activities for a family with a one and three year old. Any suggestions?*

? *My fiancee and I were wondering if there was anywhere we could go in October that would not be extremely crowded, yet more secluded?*

? *When is the best time to go on a Caribbean Cruise - and do you recommend bring our 16 year-old so? He is very bright.*

Again this situation points out that evaluation needs to reflect an environment – we do not foresee that all questions will be ones that can be satisfied with short answers which are found redundantly present on the web. Requirements in this particular situation contradict the TREC QA evaluation requirement that evidence supporting the answer should not be provided.

Conclusion

We have found that the collection of documents that will be available for querying, the nature of queries generated by real users, as well as the breadth vs. narrowness of what constitutes a useful answer in each of these instances is not the same. Therefore, it would only seem appropriate that an evaluation should fully specify the user, the purpose for which they are asking their question, and the nature of an acceptable answer. These should be parameters that can be varied in QA evaluations. It is essential that the situational aspects be known so that the criteria provided to the human relevance assessors truly reflect what users in that particular context would require. Evaluations should be designed that simulate as closely as possible the dimensions of the context in which users will be posing their questions. Clearly the use of multiple scenarios would enhance the possibility that evaluation would lead to a range of QA systems, each defined by the parameters of the situation in which they are to be used.

References

Belnap, N. D. (1963). An analysis of questions: Preliminary report. Scientific Report TM-1287. Santa Monica, CA.

Belnap, N. D. & Steel, T. B. (1976). The logic of questions and answers. New Haven, CT., Yale University Press.

Brill, E., Lin, J., Banko, M., Dumais, S. & A. Ng. (2001). Data-Intensive question answering. Notebook Proceedings of the Text Retrieval Conference. Gaithersburg, MD: NIST Special Publications.

Breck, E.J., Burger, J.D., Ferro, L, Hirschman, L., House, D., Light, M. and Mani, I. (2000). How to evaluate your question answering system every day…and still get real work done. Proceedings of Language Resources and Evaluation (LREC).

Liddy, E.D. (2001). Breaking the Metadata Generation Bottleneck. Joint Conference on Digital Libraries. Roanoke, VA., June 25, 2001.

Liddy, E.D. & Liddy, J.H. (2001). An NLP approach for improving access to statistical information for the masses. Proceedings of the Federal Committee on Statistical Methodology Research Conference. Arlington, VA.

Voorhees, E. and Tice, D. (1999). The TREC-8 question answering track evaluation. In Voorhees, E. and Harman, D. Proceedings of the Eighth Text Retrieval Conference. Gaithersburg, MD: NIST Special Publications.

Voorhees, E. (2001). Overview of the TREC 2001 question-answering track. In Voorhees, E. and Harman, D. Notebook Proceedings of the Text Retrieval Conference. Gaithersburg, MD: NIST Special Publications.

A Curriculum-Based Approach to a QA Roadmap

John Prager
IBM T.J. Watson Research Center
Yorktown Heights, N.Y. 10598
Tel (914) 784-6809; Fax (914) 784-6078
jprager@us.ibm.com

Abstract

The QA community is beginning to understand the core problems in the field, and they largely coincide with those of Natural Language Understanding. The difficulty of answering a question by a current QA system is a function of the match or lack of it between the question or its expression and the resources used to answer it, not how difficult it is for a human to answer it. A prominent factor in making a question hard now is not so much in finding an answer but in validating whether a candidate answer is correct. The problem in many ways parallels that of reading comprehension for children, which suggests a graduated approach to developing and evaluating the field. The difficulties faced by QA systems include long-standing issues in computational linguistics, such as anaphora resolution, metonymy etc.; logic-oriented issues such as scope and quantification as introduced by adverbs and articles; structural problems where the answer must be assembled from many sources, as well as reasoning about space, time and numbers. These problem areas are largely orthogonal, and can be introduced progressively with at each step accepted criteria for success.

Introduction

The approach TREC has been taking to Question Answering has been rather like asking fourth-graders to read and understand *Hamlet*, and when they show even some rudimentary success, moving them on to *War and Peace* and then *Finnegan's Wake*. While it is very understandable that members of the community - or indeed several communities: academic, government, military and web-users - wish to push the state of the art as far and as fast as possible, it is inescapable that complete success at QA requires mastering all of the core problems of NLP. This has not been done over the last fifty years and is not going to be achieved anytime soon.

Approximately two years ago, a first QA Roadmap was drafted on behalf of ARDA (ARDA, 2000), based on input from many key researchers in the field (including the present author). That document developed the question taxonomy previously proposed by researchers at SMU (Moldovan et al., 2000). That taxonomy lists a series of increasingly difficult questions, characterizing them by the kind of questioner who would ask them. The taxonomy is very well intentioned but, in hindsight, unfortunately wrong in some of its details or its emphasis and difficult to work with because of two inherent assumptions that appear to have been made, or at least not rejected.

The problematic assumptions are (1) that it is possible to grade the difficulty of questions by semantics independent of the corpus and/or other resources that will be used to answer them, and (2) that what is difficult for a human will also be difficult for a computer. As for the first point, we observe that understanding the question is indeed part of the QA process, but it is only a part. Understanding the corpus (plus ontologies and other kinds of data) is equally important, as is being able to match these resources to the question. Sometimes such a match is trivial, sometimes it requires considerable linguistic processing and/or reasoning: which is the case cannot be determined from the question alone.

For example, consider the question:

> When was Queen Victoria born?.

It is very easy to answer if there is a text passage of the form:

> ... Queen Victoria was born in 1819...,

and only a little trickier if the text reads

> ... Queen Victoria (1819-1901) ....

However, if the text contains no such statements, but instead just the indirect reference

> ... King George III's only granddaughter to survive infancy was born in 1819 ...,

along with text (possibly elsewhere) that states

> ... Victoria was the only daughter of Edward, Duke of Kent,

along with more text (possibly yet elsewhere) that states

> ... George III's fourth son Edward became Duke of Kent ...

the question becomes considerably harder to answer.

By contrast, the seemingly difficult question

> Should the Fed raise interest rates?

becomes much simpler to answer in the presence of a news article quoting Alan Greenspan as saying

> All of the current leading economic indicators point in the direction of the Federal Reserve Bank raising interest rates at next week's meeting.

On a lighter level, even the perennial

> What is the meaning of life?

is a cinch to answer if one consults *The Hitchhiker's Guide to the Galaxy* (Adams, 1982)[1].

If one accepts that questions by themselves cannot be arranged in order of difficulty, the very notion of a Roadmap might seem to be called into question. However, it is the thesis of this paper that a systematic approach mirroring somewhat an academic curriculum can achieve the desired goals. A basic method of the classical Western educational system is the incremental dissemination of new information and skills, building on previous knowledge (as opposed to, say, the immersion approach to language learning). Evaluation is performed continually, with testing materials crafted either to examine as closely as possible just the new material, or a combination of new and old, as the teacher sees fit.

Going beyond TREC

The problem with the current TREC-style evaluation using real user's questions and real news articles is that every question can potentially test a different variety and combination of system skills and knowledge, so a system's performance can vary widely from question set to question set. A given system can fare remarkably differently on seemingly isomorphic questions because of idiosyncrasies of the data resources. Granted, using large enough question sets it becomes possible to rank order QA systems, as TREC does (Voorhees & Tice, 2000), but the current setup does not enable one to easily assert exactly what is being tested in a system (except QA in a holistic way), or what, if anything, a system is good at. Amongst other things, this makes it difficult to predict performance when a QA system is to be deployed in a new domain, or how it will behave with different user groups.

Three trends in TREC QA, from the first instance in TREC8 to the proposed TREC2002, have had and are having the benefit of forcing systems to "know" what they are doing. These are: (1) the trend from 250-byte answers to 50-bytes to "exact answer", (2) from 5 submitted answers to a single answer, and (3) the (as yet largely unexploited) possibility of "no answer". These refinements of the track are fine and do a great service in that they greatly reduce the chances that systems get the right answer "by accident", but they represent the end of the line in this particular kind of evaluation development. It should be mentioned, though, that while these improvements are necessary for the evolution of QA systems whose output will in turn be used by other automatic systems, they are not so necessary when the consumers of the output are real users, who can tolerate a set of candidate answers and who will generally be pleased to see the answers in the context of text passages. Having said that, it is true that if a system can do well in the more constrained context it can only benefit its performance in the less constrained one.

The essential difficulty with question answering stems from the fact that textual material is in natural language, and that to consistently answer questions posed against text corpora requires understanding the text. Since these texts were written with human readers in mind, they make copious use of all of the linguistic and stylistic devices that make reading pleasurable and computer understanding difficult: anaphora, definite noun phrases, synonyms, subsumption, metonyms, paraphrases, nonce words, not to mention idioms, figures of speech and poetic or other stylistic variations. For example, in answer to "How did Socrates die", we find from the TREC corpus:

> His chapter on wifely nagging traces nagging back to the late Cretaceous period and notes that one of the all-time nags was Socrates' spouse, Xanthippe. Hemlock was a pleasure by comparison.

and

> We also meet snake root, which is toxic, and poison hemlock, which for over two thousand years has been famous for curing Socrates of life.

In fact, all of the other mentions of Socrates and hemlock together in this corpus happen to be indirect, thus making this simple sounding question particularly difficult. Usually, though, in a large corpus such as TREC uses there are multiple mentions of facts interesting enough to be the subject of questions, and for every obscure reference there are often several plain ones.

Following the train of the argument in this paper, it would seem that by far the easiest way to provide a Roadmap for QA would be to mimic the progression of reading comprehension tests in school, by using texts written for progressively higher grade-levels. These would start with texts employing only short sentences using simple syntax and little imagery, and progress to adult-level texts such as news articles and beyond. The difficulty here, though, is that these elementary texts do not exist in sufficient quantity, especially online, to provide a meaningful-sized corpus (the current TREC QA corpus is 3GB). If we cannot fix the corpora, then at least we can fix the questions. [We should mention here a recent posting by Karen Spark-Jones to the TREC web site (Spark-Jones, 2001). The posting lists a set of questions, and for each one a

[1] The answer is 42.

large number of candidate answer sentences that address some aspect of the questioner's concern, but may or may not answer the question itself. This is in the same spirit as the theme of this paper, as it finesses the issue of finding such sentences, but allows one to concentrate on the problems of question-answer match.]

Impedance match

Using as background the earlier argument that multiple mentions of interesting facts should generally reduce problems of text complexity, we can again advance the suggestion that sets of increasingly difficult questions be developed. The measure of difficulty, though, will be quite different from that espoused in the first QA Roadmap. The notion is to identify components of the QA task that are difficult for a machine to perform, rather than difficult for a human. In some cases, the difficulty will ensue from the absence of a direct answer in the resources used, as discussed above. In other cases, the difficulty will derive from the linguistic and/or logical structure of the question, rather than its semantics (that is, the semantics of the individual content words). Take for example the question "What is the population of France's capital?". Assuming that there is no text that directly restates the question, the task is to first find the capital of France (Paris), and then to find the population of Paris; these two steps may well be performed using different documents or different knowledge bases or databases. The level of difficulty of the question does not stem from the fact that two resources must be searched. Given the problem breakdown, it is straightforward to construct the two necessary queries. The difficulty comes from the question's structure: the system must know that the phrase "France's capital" is a reference to an entity that must itself be found before the outer question can be answered.

The structure of the problem in general ensues from not only the structure of the question but also the availability of *knowledge sources*: both the information resources and the kinds of processing needed to make use of them. The question "What is the largest city in France?" can be answered in a variety of ways: from a direct statement in text; from a table listing French cities and their sizes; from discovering that Lyon is the second largest French city, and that Paris is larger than Lyon; from an enumeration of separately discovered pairs of {city, size} (making assumptions of completeness), and others. The difficulty of the task can be varied by making available or unavailable any of the pertinent knowledge sources. To summarize, the measure of difficulty of the questions mentioned so far in this paper stems from what might be called the *impedance match* (or *mismatch*) between question and knowledge sources. Moving on, we can orthogonally mine the linguistic dimension for incremental difficulty.

The Linguistic Dimension

In what follows we present an unordered and non-exhaustive list of the kinds of linguistic capabilities that a full-fledged QA system should have. These capabilities can be expressed and evaluated by question sets that require that particular competence for successful performance. We have already seen some examples of questions that derive their difficulty from the absence of a direct and straightforward representation of the answer in the available resources. The remaining examples are for the most part easy for humans to address, but illustrate difficulties that computers have with NLP.

Consider the following two questions:

> Name a US state where automobiles are manufactured

and

> Name a US state where automobiles are not manufactured

The vast majority of present-day QA systems will pay no attention to the *not*, although it is critical for correct behaviour. Likewise, other adverbial modifiers such as *just* and *only* can play havoc with the system's performance. Sometimes the presence of a single such modifier can require large amounts of real-world knowledge. Consider

> Name an astronaut who made it to the moon

versus

> Name an astronaut who nearly made it to the moon

One can easily come up with half-a-dozen reasonable interpretations of *nearly* here, each giving different sets of correct answers.

In a similar vein, articles play an important role in question interpretation. The TREC community has been arguing for years whether Atlantic City is a correct answer to "Where is the Taj Mahal?". Making the article indefinite would generate much less of a dispute whether casinos, hotels and restaurants were allowable answers; having a computer understand the difference, though, would be a challenge. Interesting questions arise when articles are absent and the end-user is unknown. Is the question:

> What is mold?

really a hurried form of

> What is a mold?

What if the end user is the native speaker of a language that doesn't use articles? One can imagine an exercise where the system is given a set of questions to be answered in the context of each of a set of user-profiles. These profiles may be no more than simple age/profession/nationality descriptors, but sufficient to elicit different maximum-likelihood interpretations for each question in the set.

An important area where difficulty can be introduced in an orthogonal manner is in that of ungrammatical questions. Although NIST has tried to make the TREC QA questions immune from this problem, by the author's count about two percent contain one or more misspellings, incorrect capitalizations, incorrect compoundings, or syntax errors. Observing the first such errors in TREC8 has had the unintended beneficial consequence of causing some groups to develop and deploy spell-checkers and other fault-tolerant mechanisms. Raw questions from real users undoubtedly contain a much higher percentage of such errors than in TREC; keyword-based queries, so common on the Web, can be considered to be degenerate cases of ungrammatical sentences.

A common cause of problems, not only in QA but also in basic Information Retrieval, is the lack of lexical match between two equivalent or ontologically-related concepts. Question sets that specifically test subsumption, synonymy, meronymy and other relationships can easily be generated, in the obvious way.

QA systems today don't do well with numbers. "How many"-type questions are easy to answer if the sought figure is discussed in text, but not so if the system has to enumerate instances. Ability to convert between units is largely absent. Ability to evaluate reasonable magnitudes is also missing.

QA systems are currently monolingual. It is clearly desirable to be able to query in one language texts in another, but there is scope for awareness of other languages that falls far short of full CLIR, or maybe that should be CLQA. Even simple questions like "What does ciao mean?", bearing no explicit indication of foreign language presence, can benefit greatly from systems having some notion of what is English.

Summary

Developing a Roadmap for QA entails developing a series of tasks which, when mastered, would result in an extremely capable system. The current TREC approach of requiring QA systems to do everything in the first year, and just be better at it in subsequent years, does not provide the right kind of incremental basis. Instead, rather like in a modular school curriculum, technical areas to be addressed should be identified and codified in question-sets that require the requisite capability to answer. The question-sets may be accompanied by restrictions on resources that may be used. Such "learning modules" can be either orthogonal or incremental, or even some combination. Developing them will not be as easy as generating the TREC question-sets, since, in many cases, knowledge by the question-set compiler of the resources available (text corpora, ontologies, databases) will be necessary to judge how and where a given question is appropriate, just as a textbook author must know the subject matter in order to set appropriate questions for each chapter.

Acknowledgments

The author would like to thank Jennifer Chu-Carroll and David Ferrucci for their helpful comments and suggestions. This work was supported by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program under contract number MDA904-01-C-0988.

References

Adams, D. (1982). The Restaurant at the End of the Universe, book 2 of the Hitchhiker's Guide to the Galaxy trilogy, Pocket Books, NY.

ARDA (2000). Issues, Tasks and Program Structuresto Roadmap Research in Question & Answering (Q&A) (http://www-nlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc)

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R. Goodrum, R. Girju, R and Rus, V. (2000). "LASSO: A Tool for Surfing the Answer Net", Proceedings Eighth Text Retrieval Conference, E.M. Voorhees and D.K. Harman Eds., NIST, Gaithersburg, MD.

Spark-Jones, K. (2001) "Question-Answering Data" http://trec.nist.gov/data/qa_no_pword/qa_task.txt.

TREC8 (2000) "The 8th Text Retrieval Conference", E.M. Voorhees and D.K. Harman Eds., NIST, Gaithersburg, MD.

TREC9 (2001) "The 8th Text Retrieval Conference", E.M. Voorhees and D.K. Harman Eds., NIST, Gaithersburg, MD.

TREC2001 (2002) "The 8th Text Retrieval Conference", E.M. Voorhees and D.K. Harman Eds., NIST, Gaithersburg, MD (to appear).

Voorhees, E.M. and Tice, D.M. (2000). "Building a Question Answering Test Collection", *Proceedings of SIGIR 2000*, pp. 184-191, Athens, Greece.

Evaluating QA Systems on Multiple Dimensions

Eric Nyberg & Teruko Mitamura

Language Technologies Institute
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA 15213 USA
{ehn,teruko}@cs.cmu.edu

Abstract

Question-answering systems are expanding beyond information retrieval and information extraction, to become full-fledged, complex NLP applications. In this paper we discuss the evaluation of question-answering systems as complex NLP systems, and suggest three different dimensions for evaluation: objective or information-based evaluation; subjective evaluation; and architectural evaluation. We also discuss the role of ambiguity resolution in QA systems, and how ambiguity resolution might be evaluated.

1. Introduction

The recent QA Roadmap (Burger et al., 2001) expanded the scope of question answering along several dimensions, including: multiple question types, multiple answer types, multiple media, multiple languages; interactive dialog with the user to refine/guide the QA process; multiple answer perspectives; and ultimately, answers which provide an evaluation or judgment based on retrieved data. QA systems are expanding beyond information retrieval and information extraction, to become full-fledged, complex NLP applications.

We present three different types of evaluation: a) *information-based evaluation*, which (like the TREC QA track) focuses on the completeness and correctness of the answers given; b) *utility-based evaluation*, which focuses on the usability of the QA system for the end-user; and c) *architectural evaluation*, which focuses on the characteristics of the software architecture used to implement the QA system. For each type of evaluation, we discuss possible ways to define test data and carry out an evaluation.

These three types of evaluation are relevant for next-generation QA systems such as JAVELIN (Nyberg et al., 2001). The ideas presented here draw upon our experience with the evaluation of other complex NLP systems (e.g. Machine Translation (Mitamura et al., 1999), Integrated Information Management (Nyberg & Daume, 2001)) that are directly relevant to advanced QA.

2. Extending Information-based Evaluation: Ambiguity Resolution

At the core of current QA evaluation methods is the objective evaluation used in the TREC QA track. Objective evaluation requires the creation of questions and correct answers for each question, given a corpus and some pre-defined criteria for judging “correctness”. The QA Roadmap describes the evolving capabilities of QA systems, which will require new objective measures (i.e. new TREC QA tasks). Although objective evaluation is extremely useful and easy to carry out once the data sets have been created, it is probably not feasible to create a single suite of questions that adequately tests all dimensions of a QA system in an objective manner. For example, different suites might evaluate system’s performance on various question types, answer types, document sets, etc. More global capabilities, such as ambiguity resolution, cut across all of the question and answer types and should be evaluated separately. In the remainder of this section we discuss the specific challenges of creating an objective evaluation for ambiguity resolution.

Starting with TREC 2002, the QA evaluation track will include question ambiguities. In general terms, an ambiguous question is one that has more than one meaning or interpretation. In a QA system, question ambiguity is significant when the different meanings imply different answers. If there is a high degree of ambiguity (many different meanings), or the ambiguity implies a much greater degree of information processing (many more texts to be searched), the system should attempt to resolve the ambiguity.

Ambiguity in natural language has been studied in detail in the fields of computational linguistics and machine translation, and all of the classic forms of ambiguity can affect a QA system (lexical ambiguity, syntactic ambiguity, pronominal anaphora, scope ambiguity, etc.). When designing a QA system, it is important to consider a) whether (and how) to detect a particular type of ambiguity; b) whether (and how) to resolve the ambiguity before searching for an answer; c) whether (and how) to resolve ambiguity as part of composing the answer. The diagram in Figure 1 illustrates the difference between approaches b) and c). In either case, the system can resolve the ambiguity automatically, or interact with the user to resolve the ambiguity.

In the following subsections, we discuss three specific types of automatic disambiguation that can be evaluated in an advanced QA system: context disambiguation, structural attachment disambiguation, and word sense disambiguation.

2.1. Context Disambiguation

Context can be disambiguated automatically by using the analyst profile or past session memory. The context category (e.g. economy, politics, geography, etc.) can be used for disambiguation. Questions might include words that can belong to different domains; for example, the words "line, defense, conference", may indicate an academic context or a sports context.

Questions that refer to attributes of objects may also be ambiguous in different contexts. As noted in the QA Roadmap (Burger et al., 2001), the same attribute name might imply different answer types. For example, the general notion of "dimension" as queried in questions like *How big is New York*? or *How big is the Pacific Ocean?* implies different possible answer types (e.g. a population count, a geographical area in square miles, etc.). In each case, the QA system must select more specific query terms that are appropriate to the particular meaning intended. For example, *How old is Koizumi?* can be answered by searching for a birth date, where a question like *How old is Siemens?* requires searching for events like *incorporated*, *founded*, etc. The strategy depends on knowing whether the question refers to a person or an organization, in this case.

It is a large task to address this type of ambiguity for unrestricted English text, since this presupposes a well-defined semantic model with broad coverage ("world knowledge"). A more feasible method for developing test data and evaluations might be to construct an model of the most relevant contextual ambiguities for intelligence gathering tasks. For the most relevant query object and answer types associated with a particular corpus (e.g. person, organization, location, country), it should be possible to determine the set of salient attributes of each type (e.g. age, location, size), along with the potentially ambiguous question terms that are typically used to refer to those attributes. This type of empirical data gathering presupposes that a set of sample questions are available for analysis. Once all of the attributes and their source language query terms have been identified, a set of questions could be constructed to evaluate a system's ability to search for the correct attribute given an ambiguous query.

Although this discussion has focused on single attributes, realistic questions will also include nested attributes, such as *How big is support for Koizumi?* For this question, it is important to know that a) Koizumi is a person in the political arena, b) *support* in this context implies public opinion concerning job performance, and c) big is a relative measure of public opinion, perhaps based on the results of a public opinion poll. Handling this type of nested ambiguity will require not only the disambiguation of nouns such as *Koizumi* and *support*, but also an understanding of syntactic structure and the relationships represented by prepositions like *for.*

2.2. Structural Disambiguation

One of the challenges for phrase level analysis is the resolution of structural attachment ambiguity (e.g. prepositional phrase attachment). In building the JAVELIN system, we plan to extend the automatic structural attachment heuristics developed for the KANT system (Mitamura et al., 1999) to handle structural disambiguation in question analysis. If the system cannot automatically resolve structural ambiguity, then it will ask the analyst for clarification.

In our work on machine translation, we have developed two fundamental ways to evaluate ambiguity resolution: a) by testing analysis results (meaning interpretations) against a pre-defined "gold standard", and b) by checking the correctness of the translation results (Mitamura et al., 2002). Interestingly, an incorrect ambiguity resolution sometimes has no impact on the quality of the translation result, because the input sentence can be translated correctly in spite of the mistake. The analogy for QA systems is that there will be ambiguous questions that can be answered correctly using simple methods without ambiguity resolution, e.g. simple query term search without reformulation. An adequate test suite for ambiguity is one where the probability of getting the correct answer is significantly increased if some form of ambiguity resolution takes place.

Another type of structural ambiguity is seen in the phrase *domination of China*, which could be interpreted as *someone is dominated by China*, or as *China is dominated by X* If we think of *dominate* as a binary predicate accepting two organizations or countries as arguments, then the nominal form *domination of X* will be a common way to ask questions about *dominate* events when one party is unknown. The ambiguity arises when the pattern $V_{nominal}$ *of N* can be interpreted such that N is either the subject or the object of *V*.

For both types of ambiguity, designing test suites depends on analyzing a set of representative questions to determine what kinds of structural ambiguity arise in realistic scenarios. Since solving the general problem of ambiguity resolution in English is a large, difficult problem, QA evaluations should narrow their focus initially to the types of structural ambiguity that are relevant for QA systems. Once a set of ambiguous constructions is identified (e.g. the *of* case illustrated above), a variety of test cases should be constructed with respect to the evaluation corpus. Effective test cases will be those where more than one potential answer exists, depending on the interpretation of the question, and getting the right answer involves some form of disambiguation.

We also note that there are structural ambiguities that should always be resolved automatically, because only one structural interpretation is semantically valid.

2.3. Word Sense Disambiguation:

During question analysis, word sense disambiguation may follow from identification of the question context (as mentioned above). When there is more than one word sense for a particular term that is not resolved automatically, the system will ask the analyst to choose a term definition from a given list. Evaluating word sense disambiguation can be broken down into two parts: a) does the system represent all of the possible meanings for ambiguous terms in the corpus, and b) can the system correctly select the appropriate meaning in a given sentence (in the absence of contextual or structural cues). For nouns, this involves assigning all possible object types (person, organization, location); for verbs, it involves assigning all possible event meanings.

Once a set of common ambiguous words are identified, based on an analysis of realistic scenarios, a variety of test cases should be constructed with respect to the evaluation corpus. Effective test cases will be those where more than one potential answer exists, depending on word sense disambiguation, and getting the right answer involves correct choice of word meaning. There may also be cases where only a single answer exists, and all but one sense of a particular word are invalid in the domain context.

2.4. Discussion

For objective evaluation, the question is "*How well does System X resolve ambiguity type Y*?"[1]. Ambiguity resolution is important if resolving the ambiguity significantly enhances the system's probability of getting the right answer. Conversely, when constructing a test suite, it is useful to select questions where the probability of getting the right answer is significantly lower if the system does not resolve the ambiguity. For each of the ambiguity phenomena, an effective test suite will contain questions that have multiple answers. The TREC answer format (regular expressions) can be utilized. The real challenge is in crafting questions that differentiate between systems that disambiguate and those that do not, since the probability of getting the right answer is also influenced by the specific documents in the corpora and the degree of evidence for alternative answers.

Contextual ambiguity has important considerations for question answering systems. When a single, isolated question is asked, the context is unconstrained and the question can be assigned any meaning that is valid in the scope of the entire corpus. When a question is asked in the context of a question answering dialog, the context may be constrained to the particular topic of that session. Note that a continuation question may include ambiguous references (e.g. pronominal anaphora) that refer to concepts originally introduced in either a prior question or an answer. The QA system should automatically resolve ambiguities by referring to the existing context whenever possible.

For information-based evaluation, it is essential to construct test questions and answers that address the purpose of the evaluation. This is true not only for ambiguity resolution, but also the other QA phenomena that can be evaluated objectively (e.g., answer justification, answer completeness, multilingual QA, etc.).

[1] Note that objective evaluation does not consider the processing time used by the system. A system that resolves ambiguity during question analysis might in general be faster than a system that resolves ambiguity during answer generation, since it prunes the search space earlier.

3. Utility-Based Evaluation – How Good is the Tool?

As QA systems move beyond the laboratory to real-world applications, objective information-based evaluations must be supplemented by utility-based evaluations that evaluate the effectiveness of the software for real tasks. End-to-end system evaluations must focus on realistic analyst scenarios, and characterize the overall system's performance under different operating conditions. We envision at least three ways to evaluate end-to-end performance, described in the following subsections.

3.1. Percentage of Task Completion

The most important functional metric is whether or not the system can retrieve the desired information. Of course, a comprehensive test suite for task completion should exercise all of the question types and answer types to be covered by the system. But it is also necessary to consider other dimensions, such as the specificity or “vagueness” of the user’s question.

If a question is precise and unambiguous (e.g., When was Enron incorporated?), then the system should retrieve the desired information quickly, with no further interaction with the user. On the other hand, if the question is vague (e.g., Where is Enron?), the evaluation could focus on at least two different outcomes: a) the system finds all possible answers (place of business, global markets, etc.), or b) the system refines the question interactively to focus on the “correct” answer (e.g., *Where is Enron’s headquarters located?*).

Once a set of reference questions and answers should is created to exercise all of the possible question types and answer types, the test set should be expanded to include various “vague” reformulations of each question, to test task completion under varying levels of initial specificity.

3.2. Efficiency of Task Completion

This efficiency metric will measure how easy it is to get the desired information using the system. This dimension is crucial for a realistic evaluation; since JAVELIN will support interactive planning with the user, it will be necessary to strike a balance between accuracy (task completion) and automaticity (how much burden is placed on the analyst during the resolution of ambiguity, clarification, etc.). We can measure the overall time elapsed (how long the analyst has to wait for the answer), the amount of time spent by the analyst in responding to clarifications, and the total number of clarifications per question.

When evaluating the efficiency of machine translation systems, we often compare the time required for a complete manual translation to the time required for a machine translation plus human post-editing. To make an analogous comparison in QA evaluation, we should compare the time required by an unaided human (using only a search engine) to retrieve an answer with the time required by a human plus QA system. If a given task takes less time when using the QA system (despite the need for user interaction, refinement, etc.), then the QA system is more efficient than a human using a search engine.

3.3. N-Point Subjective Measure

Researchers in human factors have noted that the fastest system is not always the "best" - users may prefer a system that is up to slower than another, if it provides better feedback regarding its progress. In open-domain QA, it will be important to measure the user's perception of various subjective measures, e.g., *How well do you understand what the system is doing?*; *Does the system provide you with adequate feedback?*; *Is the system easy to use?*; *Does the system ask you too many questions?*, etc. Such measures are important in that they help to determine what the user considers a "usable" system - note that a system which performs no clarifications may not inspire confidence in an analyst who expects to spend a certain amount of time guiding the search.

In our work with machine translation systems, we have observed two important phenomena with respect to subjective evaluation: a) there is a definite threshold regarding interactivity – if the system asks too many questions on a particular task, the user will lose patience and select the default response, especially when under time pressure; and b) if the content of or motivation for a clarification question is not apparent to the user, they will lose confidence in the system. The subjective evaluation of QA systems should attempt to determine whether these two phenomena are also relevant for information-seeking tasks.

4. Architectural Evaluation

An objective “black-box” evaluation focuses on only those characteristics that are important to the end user, who cannot “see inside” the actual system as it is working. But it is also important to consider glass-box evaluation, which has two important benefits: a) the ability to evaluate the performance of individual system modules can help developers to rapidly locate and address problems in functionality, performance, etc.; b) an understanding of how easy it is to tune, extend and maintain the system. Therefore architectural evaluation is primarily for the system developer and the system client, who are concerned with the global characteristics of the QA system as a product of software engineering.

Architectural evaluation can be performed in the context of a design review (Pressman, 2000), which focuses on the architectural design and system documentation rather than an information-based evaluation. Although QA systems are designed and implemented using a variety of paradigms and techniques, a global set of design criteria that can be evaluated in a more or less subjective manner for each QA system. The requirements for an ideal QA architecture are similar to those summarized by the TIPSTER II architecture working group (Grishman, 1996):

? **Standardization**. Does the system specify a standard set of functions and interfaces for information services? Is it possible to mix and match different modules in a straightforward manner? In the IIM system (Nyberg & Daume, 2001) we specified a set of standard interfaces for system components that allow the end-user to perform unlimited customization without recompilation of the main system.

? **Rapid Deployment**. How easy is it to create new applications from existing components? A system with an inherently modular design is easier to reconfigure for new applications.

? **Maintainability**. Is it possible to update one module in the system without affecting the others? One key for rapid progress in QA research is the ability to work on the different aspects of the problem (question analysis, retrieval, answer formulation, etc.) in parallel, with frequent system-level testing.

? **Flexibility**. How easy is it to alter the performance of the system by allowing novel combinations of existing components?

? **Evaluation**. Is it possible to isolate and test specific modules (or versions of modules) side-by-side in the same application? If a system incorporates multiple strategies or “loops” (Harabagiu, et al., 2000), how can we evaluate the contributions made by each strategy or algorithm to the overall utility of the system?

Complex QA systems incorporate several different algorithms, modules, processing loops, etc. Effective glass-box evaluation requires a certain degree of instrumentation inside the software, so that various measurements, logging, etc. may be done before, during and after key processing steps (Nyberg & Daume, 2001). This allows the developers to identify component-specific effects and perform ablation studies that clearly evaluate the contribution of a particular component to the system’s overall performance.

If a QA research effort is focused purely on initial discovery of new algorithms, then perhaps architectural evaluation is of secondary importance. However, for longer-term efforts aimed at building a reusable technology base for ongoing development, we argue that architectural evaluation and attention to software engineering are of paramount importance. The JAVELIN project is intended to produce a general, extensible architecture, and we intend to evaluate the JAVELIN system design along dimensions such as reusability (of components, operators, etc.) and external extensibility (e.g., by ARDA's chosen third-party integrator).

5. Conclusion

Ongoing research is expanding the scope of question-answering systems beyond information retrieval and information extraction to include complex NLP techniques. In this paper, we advanced the idea that the evaluation of advanced QA systems can and should be carried out on three different levels: information-based (objective) evaluation, utility-based (subjective) evaluation, and architectural evaluation. As the field moves beyond its focus on information-based (TREC-style) evaluation, we must develop new test suites and test methods to improve the quality of QA systems along all three dimensions.

6. References

Burger, J., C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C. Y. Lin, S. Maiorano, G. Miller, D. Moldovan, B. Ogden, J. Prager, E. Riloff, A. Singhal, R. Shrihari, T. Strzalkowski, E. Voorhees, R. Weishedel (2001). Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A), http://www-nlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc

Harabagiu, S. D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus, and P. Morarescu (2000). FALCON: Boosting knowledge for answer engines. In

9th Text REtrieval Conference, Gaithersburg, MD.

Mitamura, T., E. Nyberg, E. Torrejon, and R. Igo (1999). Multiple strategies for automatic disambiguation in technical translation. In *Proceedings of 8th International Conference on Theoretical and Methodological Issues in Machine Translation*.

Mitamura, T., E. Nyberg, E. Torrejon, D. Svoboda, A. Brunner and K. Baker (2002). Pronominal anaphora resolution in the KANTOO multilingual MT system, *Proceedings of 9th International Conference on Theoretical and Methodological Issues in Machine Translation*.

Nyberg, E., J. Callan, J. Carbonell, R. Frederking, J. Lafferty, A. Lavie, T. Mitamura (2002). JAVELIN: Justification-based Answer Valuation through Language Interpretation, proposal submitted to ARDA BAA 01-01. See http://www.lti.cs.cmu.edu/Research/JAVELIN

Nyberg, E. and H. Daume (2001). Integrated information management: An interactive, extensible architecture for information retrieval, *Proceedings of HLT 2001*.

Pressman, R. (2000). *Software Engineering: A Practitioner's Approach*, 5th Edition, New York: McGraw-Hill.

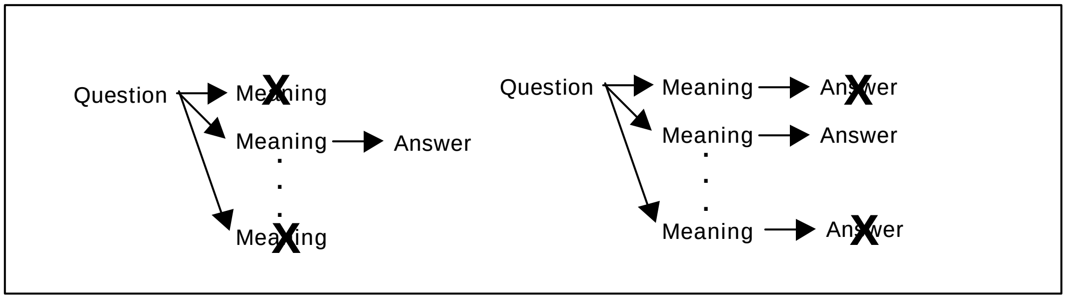


Figure 1: Ambiguity Resolution

Position statement: Inference in Question Answering

Bonnie Webber[*], **Claire Gardent**[†], **Johan Bos**[*]

[*]Division of Informatics
University of Edinburgh
Edinburgh EH8 9LW, UK
{bonnie,jbos}@cogsci.ed.ac.uk

[†]CNRS – LORIA
BP 239 – Campus Scientifique
54506 Vandoeuvre-les-Nancy, FRANCE
claire.gardent@loria.fr

Abstract

One can only exploit inference in Question-Answering (QA) and assess its contribution systematically, if one knows what inference is contributing to. Thus we identify a set of tasks specific to QA and discuss what inference could contribute to their achievement. We conclude with a proposal for *graduated test suites* as a tool for assessing the performance and impact of inference.

1. Introduction

Our point in this position statement is that, to use inference in Question-Answering (QA) in a way that will support what Barr and Klavans (2001) call *component performance evaluation* – assessing the performance of system components and determining their impact on overall system performance – one must identify specific *question-answering tasks* that can potentially gain by exploiting inference. In the first generation of QA systems (i.e., those designed to answer questions in terms of information in structured databases), only a few QA tasks were seen to need inference. In all cases, inference complemented the extensional process of relational (SQL) database querying, through reasoning on the concepts involved:

- Stallard (1986) used terminological reasoning (in a description logic) for the task of mapping from the logical form (LF) representation of a user's query and the concepts it was couched in, into the concepts and relations that formed the *data model* for the database.

- In the context of QA from multiple databases, inference was used in (Hendrix et al., 1978) in the task of developing plans for what databases to access for concept extensions, which would then be combined to produce an answer.

- Kaplan (1982) used inference on the query and its presuppositions for the task of generating a response to a question whose direct answer was not deemed useful.

- Pollack (1986) used inference on the query and an enhanced *data model* for the task of identifying and correcting user misconceptions that underlay otherwise unanswerable (or not usefully answerable) questions.

- In (Mays, 1984; Mays et al., 1982), when a question couldn't be usefully answered at the time it was asked, inference in the form of a temporal tableaux reasoner was used to generate a response to a question whose direct answer was not deemed useful. Specifically, it was used to identify whether the situation described in the question could occur in the future. If so, the QA system could offer to monitor for its occurrance, at which time the question could be answered.

Not all of these QA tasks are relevant to today's (or even tomorrow's) Open-Domain QA systems, which are designed to answer questions on the basis of *unstructured data* (i.e., free text). Nevertheless, it is still the case that there are places where inference can enhance the capabilities of Open-Doman QA systems (Burger et al., 2000; Hirschmann and Gaizauskas, 2001) and/or improve the quality and/or accuracy of their answers. As already noted, our point in this position statement is that, to use inference to these ends, one must identify specific *question-answering tasks* that will drive inference. This will then allow development of the kinds of *graduated test suites* with respect to which *evaluation* can be carried out on both the QA system and the inference engines themselves.

Note that the position we are taking here is very similar to that in (Hobbs et al., 1993), where the authors identify a set of *discourse tasks* that need to be solved in order to explain why the sentences of a text, in combination, would be true. These *discourse tasks* include (but are not limited to): interpreting compound nominals; resolving definite referring expressions; further specifying vague predicates; identifying how predicates apply to their arguments; disambiguating the arguments to predicates; determining coherence relations between adjacent segments of text; and detecting relation of an utterance to the speaker's overall plan. These, in turn, may depend on solving lower-level tasks such as resolving attachment and/or word sense ambiguities, resolving anaphora, and filling in missing (semantic) arguments. But by first specifying the discourse tasks, the authors can show exactly how inference (in their case, *weighted abduction*) can potentially – with efficient search and sufficient background knowledge – be used to solve them. (Note that weighted abduction is not a technique for *forward reasoning.* So any discourse task that requires determining the additional conclusions that can be drawn from a text may require another form of reasoning.)

In the first part of this statement, we identify a set of *question-answering tasks* in which inference could allow enhanced or extended QA services. Our goal is not to comment on what has or has not already been done in using inference in Open-Domain QA systems, but rather to lay out general areas where inference can contribute. We conclude by saying a bit more about *graduated test suites.*

2. QA Tasks

For this short position paper, we restrict the label *QA tasks* to ones that follow from a *functional role* of question or answer, rather than as text *per se*. That is, it is well known that inference can support discourse processing: texts can be parsed using *deduction* – it is what DCGs are all about – and (theoretically) they can be assigned a consistent explanatory interpretation using a combination of *weighted abduction* (Hobbs et al., 1993) and *consistency checking* (Blackburn and Bos, forthcoming). While this kind of interpretation can knit together elements of a text and supply missing (implicit) elements of its fabric, and thereby be critical for deriving answers to particular questions or even particular classes of questions, discussing the role that inference can play in discourse understanding requires its own paper, which we or other people should write.

Similarly, QA interactions are *dialogues*, and work done by Perrault, Cohen, Allen, Litman, Pollack, Walker and others has clearly shown that inference is needed to support dialogue processing – e.g., to decide what a question is really asking for. But this too is a large enough area to require its own paper.

Our focus in this paper then is on the significant set of tasks that remain after both discourse and dialogue understanding are, for the moment, put aside. Among these, we can identify several where inference could provide enhanced or extended QA services.

2.1. Expanding the search criteria for *potential* answers

It is standard procedure in QA to establish search criteria based on the question that has been posed. These search criteria make up the formal *query*, which is used to find *potential* answers in the form of candidate documents that may provide evidence for or contain a *proper* answer.

To increase the yield of potential answers, alternative terms can be added to the query. While this does not intrinsically require inference, what inference can do is expand queries with truth-functionally or defeasibly equivalent *global* reformulations of the original question. These can be used to augment the query with terms that could not have been identified using essentially *local* translation of individual words that ignores their context and functor-arguments dependencies, including implicit (semantic) arguments. For example, abductive reasoning on the question

(1) What do penguins eat?

(solving the implicit argument of *when* the eating event takes place – the same generic "in general" as the generic subject penguins) might produce a defeasibly equivalent version in terms of their *staple diet.* This term would not be added for a question like

(2) What did the characters eat in the seduction scence from the film "Tom Jones"?

which has its (optional) event argument instantiated.

Inference can also expand a query with one-way *entailments* of the original question. For example, being *awarded a degree in Computer Science* (CS) entails being *enrolled for a CS degree.* Given the question

(3) How many students were enrolled in Computer Science at Cambridge last year?

computing its one-way entailments would allow the query to be expanded with *award*$\wedge$*degree*.

Finally, inference can expand queries through subconcepts that form a *partition* (i.e., disjoint cover) of a concept in the original query; a distinct sub-query can be formed for each one. In this way for instance, the query

(4) How many people work for IBM?

could be decomposed into a set of sub-queries such as e.g., *How many men work for IBM? How many women work for IBM* or *How many white collar workers does IBM have? How many blue collar workers does IBM have?*.

Although we have discussed these expansion techniques in terms of constructing a query (either initial or follow-up, in case the initial query does not produce sufficient results), the same techniques could benefit the *ranking* of potential answers with respect to the question, if *recall* on the original query is felt to be sufficient.

2.2. Determining *proper* answers from *potential* answers

A proper answer to a wh-question may be found within a single clause, or it may be distributed through the potential answer (*answer locality*). Moreover, a proper answer may be explicit in the text (i.e., derivable simply by pattern matching), or it may require inference or other method of information fusion (*answer derivability*).

Even where an answer appears to be *explicit* in a text, inference can help determine whether it is a *proper* answer (Bos and Gabsdil, 2000), as with the following potential answers to:

(5) Q: Who invented the electric guitar?
A1: Mr. Fender did not invent the electric guitar.
A2: The electric banjo, cousin of the electric guitar, was invented by Bela Fleck.

A proper answer to this question must entail either (1) that there is someone who invented the electric guitar, or (2) that there is no such person, or (3) that it is true of everyone. All of these are logical relations between a potential answer and a representation of the question in terms of its question domain D (here, persons) and its body B (here, inventing the electric guitar). As such, inference can be used to determine whether any of these relations hold.

Inference can also help when *proper answers* are only implicit in *potential answers*. In (Hobbs et al., 1993), Hobbs et al. show that *weighted abduction* can be used to solve a variety of *discourse tasks*, thereby making explicit information that is implicit in a text. This can be applied to potential answers. For example, a potential answer to the question

(6) Where do condors live?

might contain the compound nominal *the California condor*. As in resolving "the Boston office" (Hobbs et al., 1993), this can be (abductively) resolved to condors whose location is California. That this is a matter of abductive inference rather than simple pattern matching, can be seen by not wanting to draw similar conclusions in determining proper answers to the similar question

(7) Where do terriers live?

Here, compound nominals such as "Yorkshire terrier", "Boston terrier", "West Highland terrier", etc. in potential answers would yield such incorrect proper answers as Yorkshire, Boston, etc.

There is much more to be explored here. Nevertheless, it is clear that inference can be used to support more than one aspect of this task.

2.3. Comparing *proper* answers to wh-questions

The way in which answers are sought in open-domain QA means that one cannot avoid the problem of determining whether proper answers derived from different potential answers (candidate documents) are the same (i.e., mutually entail one another) or different. In the latter case, one may also not be able to avoid the problem of determining whether (i) one answer is more specific than another (i.e., the more specific answer entailing the more general one, but not vice versa); (ii) two answers are mutually consistent but not entailing in either direction; or (iii) two answers are inconsistent. Determining such relations among proper answers becomes a QA task for Open Domain QA, where it was not one for database QA because the underlying relational DB query system was able to recognize and remove all duplicates.

The outcome of such determination depends on whether the original question is taken to have a single answer (a unique individual or property or set) or alternative answers, the set of which is of unknown cardinality. Whatever the reason, these are problems that inference can help solve.

- Answers determined to be equivalent (mutually entailing) can be replaced by a single member of the equivalence class;
- Answers that differ in specificity (one-way entailing) can be replaced by either the most specific one (as with the answer to *When was the Bastille taken?*, where *14 July 1789* is preferred over the less specific *14 July* and *1789*) or by a conjunction of the most specific answers (as with answers to *Who is Noam Chomsky?*, where *MIT linguist*$\wedge$*left-wing activist* is the preferred way to combine the answers in the set *MIT linguist, linguist, MIT academic, political activist* and *left-wing activist*);
- Answers that are mutually consistent but not entailing can be replaced by their conjunction (as with *MIT linguist* and *left-wing activist* above);

- Answers that are inconsistent are the only true alternatives. In the case of questions with unique answers, only one of them can be correct. In the case of questions with alternative answers such as *Where do penguins live?*, all the alternatives may be distinct proper answers.

2.4. Comparing questions

Where efficiency is a goal of QA, it can be supported by determining whether a new question is one that has previously been answered (Harabagiu et al., 2001) or is related in a systematic way to one that has previously been answered. (This is the reason that FAQ-lists exist.) Inference is a valid way of computing both *equivalence* relations between questions and *subsumption* – i.e., whether one question is more specific than another one. The latter allows two different forms of answer re-use. Consider the questions

(8) Where can I go skiing in the Northern Hemisphere in June?

(9) Where can I go for winter sports in the Northern Hemisphere in June?

If one has cached the answer to (8), then one has a partial answer to question (9), which subsumes it. Conversely, if one has already cached the answer to the subsuming question (9), that answer may contain or provide a basis for an answer to question (8). That is, if (9) has been answered by answering the set of questions that follow from each possible way of instantiating the general term "winter sports", then one already has an answer to (8). On the other hand, if question (9) has been answered in general, then (much as with the "linked" questions in TREC-10) sources for that answer might prove a good place to start looking for an answer (8), rather than posing it against a completely open domain.

2.5. Determining *proper* answers to yes/no questions

One may take the set of proper answers to a yes/no question to comprise simply *yes* and *no*, or one may take it more broadly to include temporal and/or modal qualifiers as well – eg. *possibly*, *sometimes*, *it depends*, etc. In the first case, determining a proper answer requires identifying what support exists for a positive answer (*yes*); what support exists for a negative answer (*no*); and on which side the support is stronger. Practically, this could involve separate queries – one seeking evidence for the positive assertion, the other, for the negative assertion. These queries could differ because lexical items can have distinct negative-polarity counterparts. For example, given the question

(10) Does Anacin contain any stimulants?

a query seeking evidence for the positive statement might contain the terms ANACIN, CONTAIN and STIMULANT, while the query seeking evidence for the negative statement might contain the terms ANACIN, LACK and STIMULANT. But because *potential answers* retrieved in response to such questions may themselves contain explicit negation (i.e., *no* or *not*), deciding what they support requires determining the scope of negation. Here, inference can determine which of the readings are consistent. Inference can also be used as discussed in Section 2.2. to determine whether two pieces of *evidence* are the same or different, so that instances of the same evidence or instances of stronger and weaker evidence aren't multiply counted.

In general, it is easier to find positive evidence than negative evidence, as what does not hold is most often conveyed implicitly, by the lack of evidence for it (i.e., the *closed-world assumption*). But for certain yes/no questions, evidence for a negative answer may be easier to come by than for a positive one. For example, in a question with a universal quantifier such as

(11) Did Larsson score in every game he played for Celtic?

a single piece of negative evidence (e.g., "Larsson failed to score in Tuesday's game") is needed to justify a negative answer, while a positive answer requires either a potential answer that itself contains a universal quantifier or a set of potential answers that cover the entire set of games. The latter is essentially (extensional) database question-answering, with the *closed-world assumption* that the database covers all positive instances.

2.6. Generating responses in lieu of or support of a direct answer

Unlike in TREC-9, TREC-10 systems were asked to identify when they couldn't answer a question. In database QA, finding no answer to a question was not an uncommon occurence. One reason for this occurring was failure of a presupposition in the question. For example, the question

(12) Have any women been awarded a Pulizer prize for sports journalism?

may have the direct answer *None* because the existential presupposition that there is a Pulizer prize for sports journalism is false. Hence, techniques were developed (Kaplan, 1982) for recognising presupposition failure and for generating responses such as *There is no Pulizer prize for sports journalism.* But as shown in (Blackburn and Bos, forthcoming), verifying presuppositions involves inference in order to check their consistency and informativity in context.

Another reason for not being able to answer a question is that *positive* information is lacking. Here, a partial response can be formulated if *negative* information can be found that *excludes* something from the set of proper answers. For example, given the question

(13) Which French cities did Reagan like?

information to the effect *Reagan disliked Paris* provides a useful partial response. Inference can be used to recognize that an individual is excluded from the set of proper answers.

A third situation motivating a response is the case of negative answers to extensional yes/no questions, which are rarely very informative – e.g.

(14) Q: Did Hearts played a home game against Celtic in January?
A: No.

In such cases, the answer to a "weaker" question – one that can be computed from the original one by subsumption reasoning, may provide the basis for a useful response – e.g. *Did Hearts play a game against Celtic in January?* or *Did Hearts play a home game against Celtic?* or *Did Hearts play a home game in January?*. More complex questions, such as ones containing quantification and/or negation, may require more complex subsumption reasoning to establish weaker questions that are worth posing.

Note that weakening the question only makes sense for questions answered extensionally, not ones answered through inference or pattern matching such as

(15) Do penguins migrate?[1]

Other situations in which responses are useful in lieu or support of a direct answer, many of which require forms of inference, are described in (Webber, 1986).

3. Graduated Test Suites

While TREC evaluation of QA systems has focussed on the full end-to-end task, some systems have also carried out what Barr and Klavans (Barr and Klavans, 2001) call *component performance evaluation* – assessing the performance of system components and determining their impact on overall system performance. The components of interest here are those that use inference. We see *graduated test suites* as a tool for assessing their performance and impact, allowing: (1) comparison against similar components that do not use inference; (2) comparison of components that differ in what inference tools they use; and (3) assessment of the impact of improvements in inferential ability. We also see graduated test suites as a way of evaluating automated reasoning tools on the inference problems raised by QA.[2]

We now discuss two of the above QA tasks, making explicit what one would expect to see in a distinct test suite for each. As in TREC, developing the test suites would involve carefully crafting a set of examples to the correct level of difficulty, fixing evaluation criteria and delimiting in a more precise way the linguistic task involved.

Expanding the query. Section 2.1. identifies four ways of expanding the query: through equivalence, through entailment, through multiple sub-queries and through abduction. For each of these tasks, inference can be involved as follows.

When expanding the query with semantically equivalent reformulations, inference can be used in at least one of two ways: First, given a subsumption based hierarchy KB encoding relations between word meanings, inference can be used to *find* the set of (structured) concepts which are logically equivalent to the structured concept representing the initial query. Alternatively, for reformulations produced by some other mechanism (e.g, parsing the query and then generating paraphrases from the resulting semantic representation(s)), inference can be used to *check* that they are indeed semantically equivalent.

Similarly, when expanding the query with more specific variants, inference can be used either to *find* within a hierarchy, the set of most specific concepts subsumed by the concept representing the query, or, for potential variants found by other means, simply to *check* that each indeed stands in some kind of entailment relation to the initial query.

Thirdly, when expanding concepts (and/or sets of concepts) in the query into *partitions* (i.e., disjoint covers) of more specific sub-concepts, the task for automated reasoners would be to check that the conjunction of queries $Q_1, \ldots, Q_n$ obtained by replacing a concept in the original query Q by a partition of its immediate sub-concepts is equivalent to the original query.

Finally, queries can be expanded by making implicit information explicit. This requires some kind

[1] Many types of penguin migrate, swimming north each autumn in the Southern Hemisphere and south each spring.

[2] Automated reasoners have been optimised for their performance on problems from mathematics and logic. As this is not necessarily optimal for NL problems, we need to drive their optimisation in this direction. That is the reason for having test suites for both QA components and automated reasoners.

of abduction – e.g, weighted abduction (Hobbs et al., 1993) or model building (Gardent and Konrad, 2000a; Gardent and Konrad, 2000b). With the first, the reasoner is given a semantic representation of the query, along with relevant world, domain and/or lexical knowledge and returns the cheapest explanation (proof) of the query, making explicit the hypotheses (either abduced or assumed) that support it. Similarly, model building will produce a (minimal) model satisfying the formula which encodes the explicit and implicit information expressed by the query.

In all cases, the information (facts in model or logical formulae) resulting from query expansion can be converted to a form appropriate to the query. If queries are Boolean combinations of key words and/or phrases, NL Generation techniques can be applied to each semantic component to produce a parse tree whose leaves constitute a string of lexical *lemmas*, from which key words and phrases can be identified and added to the query.

Determining proper answers. For **wh-questions with a single answer**, the problem of determining a proper answer from a potential answer depends on (i) the *expected answer type* (positive, negative, unknown); (ii) the *answer locality* (whether the answer is contained in a single clause or distributed over the text), and (iii) the *derivability* of the answer (whether it is explicit in the text and derivable simply by pattern matching, or it requires inference or other method of information fusion).

Test-suite examples could therefore be divided into 12 classes, of different complexity, depending on the values of these factors. For example, consider *expected answer type*. Formulated in first-order logic, with ϕ_A representing the meaning of the potential answer A, D the domain of the question and B its body, (1) if the expected answer type is positive, there is at least one object having the properties set by the question. So the inference task is simply: **Prove** $\models \phi_A \rightarrow \exists x(D(x) \wedge B(x))$**.** (2) Alternatively, if the expected answer type is negative, there is no object having the properties set by the question. So the inference task is: **Prove** $\models \phi_A \rightarrow \neg\exists x(D(x) \wedge B(x))$**.** (3) Finally, if the expected answer type is unknown, then *both* the above inference tasks are required.

For **questions with multiple answers**, we can only comment now on the use of inference for questions that can be expanded into a set of more specific sub-queries with known cardinality, such as

(16) What is the longest river on each continent?

which can be expanded into *What is the longest river in Europe? What is the longest river in Asia?* .... Once expanded in this way, each sub-query is a simple wh-question with a single answer. This is then the case discussed earlier.

4. Summary

There is no question that QA would not also be enhanced through the use of inference in *discourse tasks* involved in finer-grained examination of the texts retrieved in response to user-queries. It would likewise be enhanced by the use of inference in *dialogue tasks* involved in understanding the user's current utterance with respect to the current QA dialogue. Here we have focussed solely on the use of inference in *QA tasks* – tasks that follow from the *functional role* of a question or an answer – and how it could contribute to achieving these tasks, over and beyond methods that don't use inference.

When considering the development of *graduated test suites* to assess system performance on QA tasks and its impact on overall system performace (and also the performance of automated reasoning tools), it makes sense to consider the use of previous TREC questions and the set of passages (potential answers) that the retrieval components of TREC QA systems have returned in response. The usefulness of doing so is most obvious in the case of two of the tasks discussed here: determining proper answers from potential answers and comparing proper answers to wh-questions. What now requires discussion is what to do next.

5. References

Valerie Barr and Judith Klavans. 2001. Verification and validation of language processing systems: Is it evaluation? In *Proceedings of ACL Workshop on Evaluation Methodologies for Language and Dialogue Systems*, Toulouse, France.

Patrick Blackburn and Johan Bos. forthcoming. *Computational Semantics*. Current draft available from http://www.comsem.org.

Johan Bos and Malte Gabsdil. 2000. First-order inference and the interpretation of questions and answers. In *Proceedings of Gotelog 2000*, pages 43–50, Goteborg, Sweden.

John Burger, Claire Cardie, Vinay Chaudhri, Robert Gaizauskas, and *et al.* 2000. Issues, tasks and program structures to roadmap research in question & answering. Technical report, National Institute of Standards and Technology. Available on-line at http://www-nlpir.nist.gov/projects/duc/papers/QA.roadmap-paper_v2.pdf.

Claire Gardent and Karsten Konrad. 2000a. Interpreting definites using model generation. *Journal of Logic, Language and Information*, 1(2):193–209.

Claire Gardent and Karsten Konrad. 2000b. Understanding each other. In *Proceedings, 1st Annual Meeting of the North American Chapter of the ACL*, Seattle WA.

Sanda Harabagiu, Dan Moldovan, and et al. 2001. Falcon: Boosting knowledge for answer engines. In *Proceedings of the 9th Text Retrieval Conference (TREC 9)*, pages 479–488, National Institute of Standards and Technology. Available on-line at http://trec.nist.gov/pubs/trec9/papers/smu.pdf.

Gary Hendrix, Earl Sacerdoti, Daniel Sagalowicz, and Jonathan Slocum. 1978. Developing a natural language interface to complex data. *ACM Transactions on Database Systems*, 3(2):105–147.

Lynette Hirschmann and Rob Gaizauskas. 2001. Natural language question answering: The view from here. *Natural Language Engineering*, 4.

Jerry Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards. 1993. Interpretation as abduction. *Artificial Intelligence*, 63(1-2):69–142.

Jerrold Kaplan. 1982. Cooperative responses from a portable natural language database query system. In Michael Brady and Robert Berwick, editors, *Computational Models of Discourse*, pages 167–208. MIT Press, Cambridge MA.

Eric Mays, Aravind Joshi, and Bonnie Webber. 1982. Taking the initiative in natural language data base interactions: Monitoring as response. In *Proceedings of the European Conference on Artificial Intelligence*, pages 255–256, Orsay, France.

Eric Mays. 1984. *A Modal Temporal Logic for for Reasoning about Changing Data Bases with Applications to Natural Language Question Answering*. Ph.D. thesis, Dept of Computer and Information Science, University of Pennsylvania, Philadelphia PA.

Martha Pollack. 1986. *Inferring Domain Plans in Question-Answering*. Ph.D. thesis, Department of Computer & Information Science, University of Pennsylvania.

David Stallard. 1986. A terminological simplification transformation for natural language question answering systems. In *Proceedings of the 24th Annual Meeting, Association for Computational Linguistics*, pages 241–246, Columbia University.

Bonnie Webber. 1986. Questions, answers and responses. In Michael Brodie and John Mylopoulos, editors, *On Knowledge Base Systems*, pages 365–401. Springer-Verlag, New York.

The Challenge of Technical Text

Michael Hess, James Dowdall, Fabio Rinaldi

University of Zürich, Institute of Computational Linguistics
Winterthurerstrasse 190, CH-8057 Zürich, Switzerland
{hess,dowdall,rinaldi}@ifi.unizh.ch

Abstract

When evaluating and comparing Answer Extraction and Question Answering systems one can distinguish between scenarios for different information needs such as the “Fact Finding”, the “Problem Solving”, and the “Generic Information” scenarios. For each scenario, specific types of questions and specific types of texts have to be taken into account, each one causing specific problems. We argue that comparative evaluations of such systems should not be limited to a single type of information need and one specific text type. We use the example of technical manuals and a working Answer Extraction system, “ExtrAns”, to show that other, and important, problems will be encountered in the other cases. We also argue that the quality of the individual answers could be determined automatically through the parameters of correctness and succinctness, i.e. measures for recall and precision on the level of unifying predicates, against a (hand-crafted) gold standard of “ideal answers”.

1. Introduction

The classical type of *information need* satisfied by existing IR systems can be described with the scenario of “Essay Writing”: If you have to write an essay on a given topic you need to locate as much backup material dealing with this topic as possible, i.e. preferably whole documents[1].

Increasingly, more specific types of information needs become important. *First*, one need not catered for by the “Essay Writing” scenario is a determination to locate factual knowledge about individually identifiable entities, concerning their location in time or space, their properties, or their identity with other entities. This could be called the “Fact Finding” scenario, and it is the situation assumed by the QA Track of TREC. The questions are factual questions (“where is/who is XYZ”). One source of such information is, of course, news items but also includes encyclopedias, text books, and fact sheets.

A *second*, equally important, information need beyond the “Essay Writing” scenario arises in situations where concrete problems require explicit solution(s) from a collection of documents. This could be called a “Problem Solving” scenario, and the questions asked are procedural (“how do I do XYZ”). A typical, real world, example is that of an airplane maintenance technician who needs to repair a defective component. He must locate in the massive maintenance manual of the aircraft the exact description of the specific repair procedure. Other text types that contain procedural information are “case data bases” used for trouble shooting purposes, operational handbooks, and some types of scientific articles (e.g. diagnostic and therapeutic reports in medicine).

Third is the situation where you need to find information about principles and regulations, i.e. what one might call the “Generic Information” scenario. The typical questions are definitional (“what is”), and the typical texts consulted in this situation are on-line encyclopedias, but also technical standards publications. Many technical manuals also contain numerous definitions of concepts or devices. It can also be argued that deontic texts (laws etc.) also fall under this heading, and they are extremely important in society.

What users need in the “Fact Finding”, “Problem Solving”, and “Generic Information” scenarios are systems capable of finding those exact (parts of) sentences in document collections that constitute the answer to their question. Depending on the type of question (“where is/who is”, “how do I”, “what is”) different problems will be prominent to different degrees. Thus, named entities are important for answering factual questions but less so for problem solving and definitional questions. There is also evidence that for the latter two types of questions a deeper (syntactic and semantic) analysis of questions is needed than for the factual ones. In order to define standards for comparative evaluations that are not biased towards one particular type of information need, examples of queries and texts of different types should be used from the very beginning.

In the present position statement we will briefly describe ongoing research in the related fields of Question Answering (QA) and Answer Extraction (AE), primarily in the dual context of the TREC QA track (section 2.) and of our own work on the first text type mentioned above, i.e. technical manuals (section 3.). Later we will present some of the problems that are specific to different text types (section 4.), briefly consider the difficulties of evaluating AE systems (section 5.), and finally mention the resources used in our work (section 6.). As relative ’outsiders’ we explicitly aim at providing a critical and, in some respects, dissenting voice, giving the view of somebody approaching Question Answering from a perspective different from that defined (and circumscribed) by the TREC QA track.

2. Results from TREC

Results from the two first TREC Question Answering Tracks (Voorhees, 2000; Voorhees and Harman, 2001) seemed to show that standard, keyword based, IR techniques are not sufficient for satisfactory Answer Extraction. When the answer is restricted to a very small window of text (50 bytes), systems that relied only on those techniques fared significantly worse for the kind of questions used in

[1] It has been often observed that Information Retrieval should rather be called “Document Retrieval”.

the QA track than systems that employed some kind of language processing.

More successful approaches employ special treatment for some terms (Ferrett et al., 2001) and named entity recognition (Humphreys et al., 2001), or a taxonomy of questions (Hovy et al., 2001). Interestingly, some sort of convergence appears to be emerging towards a common base architecture which is centered around four core components (Abney et al., 2000; Pasca and Harabagiu, 2001). Passage Retrieval (Clarke et al., 2001) is used to identify paragraphs (or text windows) that show some general similarity to the question (according to some system specific metric), a Question Classification module is used to detect possible answer types (Hermjakob, 2001), an Entity Extraction module analyzes the passages and extracts all the entities that are potential answers, and finally a Scoring module (Breck et al., 2001) ranks these entities against the question type, thus leading to the selection of the answer(s).

The results of this general design are promising for the kind of factual questions that make sense in the context of news messages. Since such questions ask mostly about properties of individually identifiable entities, good named entity recognition can go a long way towards finding informative text passages. However, for other types of questions (procedural and definitional) we need to be able to analyze other types of constructions, and pinpoint answers more precisely. This means that the choice of a single type of text for the purpose of comparative evaluation creates the risk of "over-fitting" in that all competitors converge on the techniques used by the most successful system for this particular type of text. This effect tends to stifle innovation rather than foster it, and we think that a wider range of texts should be used in comparative evaluation from the beginning to counteract this danger.

It appears that, partly, the problem has already begun to emerge in the latest TREC QA track (TREC10). On one hand, many systems are converging towards the 'generic AE system design' described above, on the other hand, the system that did best (Soubbotin and Soubbotin, 2001) made massive use of heuristics and patterns, that might have limited portability to other domains and other types of applications.

3. ExtrAns

Over the past few years our research group has developed an Answer Extraction system (ExtrAns) (Rinaldi et al., 2002; Mollá et al., 2000) that is mainly geared towards procedural and definitional questions over technical texts.

Two real world applications have so far been implemented with the same underlying technology. The original ExtrAns system is used to extract answers to arbitrary user queries over the Unix documentation files ("man pages"). A set of 500+ unedited man pages has been used for this application. An on-line demo of ExtrAns can be found at the project web page.[2]

More recently we tackled a different domain, the Airplane Maintenance Manuals (AMM) of the Airbus A320. The combined challenges of an SGML-based format and the more technical nature of the text and a larger size (120MB)[3] have been met using the original basic architecture (Fig.1), plus a specialized XML based tokenizer and a new CSS-based display utility.

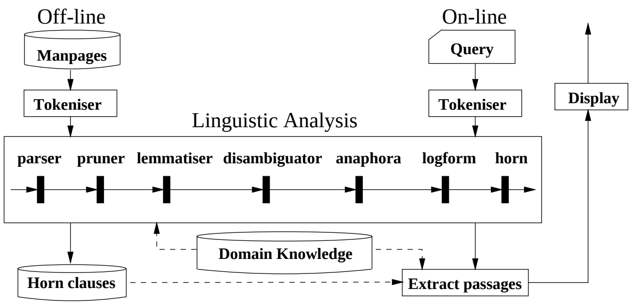


Figure 1: Architecture of the ExtrAns system

Essentially, ExtrAns extracts answers from documents by semantically comparing queries against document sentences. This is achieved by deriving, from documents and queries, the basic semantic relationships of each sentence and representing them as Minimal Logical Forms (MLF). These are representations that use selected reification and underspecification to keep them open to dynamic, incremental and non-destructive extension, depending on requirements. Answers are derived from these logical forms by deductive proof. This representation is both expressive enough to allow non-trivial comparison and computationally "light" enough for real world applications. True, this approach requires expensive deep linguistic analysis of questions and documents, involving syntax, semantics and consideration of lexical alternations (synonyms and hyponyms) but it returns, in exchange, the exact answer sentences (ideally) and often manages to even determine the individual parts of sentences constituting the exact answer(s) to user questions.

The general design of the system is fairly standard. A (very powerful) tokenizer identifies word and sentence boundaries as well as domain specific multi-word terms. Once tokenized, sentences are parsed using Link Grammar (LG) (Sleator and Temperley, 1993). Link Grammar's ability to predict the syntactic requirements of unknown words ensures that an analysis of all sentences is returned. So ExtrAns always produces MLFs, possibly extended with special predicates that mark any unprocessed tokens as "keywords". Multi-word terms (to be extracted independently and beforehand) are parsed as single syntactic units. Relieving LG of the need to compute the internal structure of such terms reduces the time and space involved for parsing technical text by almost 50%.

A corpus-based approach (Brill and Resnik, 1994) then disambiguates prepositional phrase attachments as well as gerund and infinitive constructions. An anaphora resolution algorithm (Lappin and Leass, 1994) resolves sentence-internal pronouns. The same algorithm can also be applied

[2] http://www.ifi.unizh.ch/cl/ExtrAns/

[3] Still considerably smaller than the size of the document collections used for TREC.

to sentence-external pronouns but this is not (yet) done in ExtrAns.

From the resulting disambiguated linkage, semantic relations between verbs and arguments as well as modifiers and adjuncts are expressed as a MLF. Strict underspecification ensures this only involves objects, eventualities and properties. These predicates are conjoined, and all variables are existentially bound with maximal scope. By way of an example, (1) represents the sentence, *"A coax cable connects the external antenna to the ANT connection"*:

```
(1) holds(o1),
    object(coax_cable,o2,[v3]),
    object(external_antenna,o3,[v4]),
    object(ANT_connection,o4,[v5]),
    evt(connect,o1,[v3,v4]),
    prop(to,p1,[o1,v5]).
```

ExtrAns identifies three multi-word terms, translated into (1) as the objects: v3, a coax_cable, v4 an external_antenna and v5 an ANT_connection. The entity o1 represents the fact of a 'connect' eventuality involving two objects, the coax_cable and the external_antenna. This reified argument, o1, is used again in the final clause to assert the eventuality happens '*to*' v5 (the ANT_connection).

The utility of reification, yielding the additional arguments o1, o2, o3 and o4 as hooks to the abstract entities they denote is that the expression (1) can now be modified by monotonically adding constraints over these entities without destructively rewriting the original expression (Schneider et al., 1999). So the sentence *"A coax cable* ***securely*** *connects the external antenna to the ANT connection"* changes nothing in the original MLF, but additionally asserts (2) that o1 (i.e. the fact that the coax cable and the external antenna are connected) is *secure*:

```
(2) prop(secure,p8,o1).
```

This MLF only needs to refer to the reification of an **eventuality** for further modification but other, more complex, sentences will need to refer to the reifications of **objects** (e.g. for non-intersective adjectives) or of **properties** (e.g. for adjective modifying adverbs).

ExtrAns extracts the answers to questions by forming the MLF of the question and running Prolog's theorem prover to find the MLFs from which the question can be derived. So,

"How is the external antenna connected ?'

becomes:

```
(3) holds(V1),
    object(external_antenna,O2,[V5]),
    evt(connect,V1,[V4,V5]),
    object(anonymous_object,V3,[V4]).
```

If a sentence in the text used as a knowledge base asserts that the *external antenna* is connected to or by *something*, the query will succeed. This *something* is the anonymous object of the query. If there are no answers (or too few) ExtrAns relaxes the proof criteria by introducing hyponymy related tokens as part of the MLF. Additionally, a sentence identifier indicates from which tokens the predicate is derived (not shown in the example above). This information is used to highlight the (relevant parts of the) answer in the context of the document (see Fig. 2).

This kind of very parsimonious representation could appear too "semantically weak" for general QA. This may be true but it is optimized for the task at hand (AE) and can be extended, at will, for more demanding tasks (such as full QA). The MLFs can also be used to ensure that sentences are retrieved that are, in strictly logical terms, not correct answers, but they are useful nevertheless. Thus (4i-ii) are useful (albeit not logically correct) answers, in addition to the correct answers (4iii-iv).

(4) i. The external antenna must not be directly connected to the control panel.

ii. Do not connect the external antenna before it is grounded.

iii. The external antenna is connected, with a coax cable, to the ANT connection on the ELT transmitter.

iv. To connect the external antenna use a coax cable.

4. Text Types, Question Types, and Problem Types

At present, discussions in the TREC community around the further development of Answer Extraction and Question Answering (e.g. in the "Roadmap" document (Burger et al., 2001)) address a very large number of problem and question types, many of them very thorny. However, they do so almost exclusively against the background of *one specific document type*, viz. newspaper texts.

We feel, on the basis of six years' of development and experimentation with Answer Extraction systems, that this exclusive focus on a single, very specific, type of document is not ideal, and that other document types should be considered from the beginning. There are three reasons for this:

1. Processing Strategies developed for newspaper texts become less relevant to users accessing increasing volumes of technical data.

2. Some important problems of AE/QA hardly occur in newspaper texts.

3. Some of the problems that are quite fundamental to any kind of AE/QA can be found in a more isolated, "pure", form in other types of text.

Concerning the *first* point, it is our experience that better access to archived newspaper texts and similar documents is low on the list of priorities for most potential users of QA/AE-Systems in industry, administration, and academia. One exception may be intelligence agencies with interests in monitoring news streams. However, systems that allow high-precision access to the information stored in texts covering narrower, more technical, domains would be welcomed by many organisations in business, administration, and research. Cases in point are (among others):

- Technical manuals of complex systems (any large technical system comes with massive manuals, most often in machine-readable form)
- On-line help systems (for software or other complicated products, such as some financial products)
- Customer queries (systems that process and answer e-mails and/or Web inquiries)
- Access to abstracts and full texts of scientific articles (such as Medline).

Concerning the *second* point, there are some important problems *not* given sufficient weight in the Roadmap document, due to the fairly specific characteristics of newspaper texts:

- **Domain specific terminology:** It is generally recognized that the compilation and use of terminologies is a top priority for the automatic processing of texts in technical applications. The *use* of a (reliable) terminology for a given domain makes the processing of texts vastly simpler, faster, and more useful than without (the quality of Machine Translation systems, for instance, remains dismal without terminology). However, the automatic *compilation* of terminologies ("term extraction") is basically an unsolved problem (none of the available methods produce really useful results). More work is needed in this field but the problem is very peripheral in the Roadmap document.
- **Procedural Questions:** In many of the applications mentioned above (apart from natural language interfaces to technical manuals also on-line help systems and customer e-mail processing systems) the procedural questions of the type *"How do I do X?"* ("How do I convert Apple files to UNIX text format?", "How can I move funds from checking to savings?") are of paramount importance. However, this type of question makes little sense in the framework of newspaper texts, and is therefore given too little attention in the Roadmap document.
- **Generic Questions:** In the documents used for the above-mentioned types of applications (but also in on-line encyclopedias etc.) many sentences are *generic* (timeless rules). Typical questions directed at such texts are *"How do you stop a Diesel engine?"* or *"What is a typhoon?"*. These, too, are relatively rare in newspaper texts (which normally describe individual, time-bound facts), and they are consequently not mentioned in the Roadmap document [4]. Although generic sentences are admittedly a thorny problem they must not be ignored, due to their general importance.

Concerning the *third* point, there is a number of problems that are fundamental to any kind of AE/QA system, and that do occur in newspapers texts, but which are "drowned" by the numerous other difficulties resulting from the characteristics of newspaper texts. Among them are:

- **Intensional constructions**: Contrary to (almost) common belief, intensional constructions are fairly common in perfectly normal language, and not treating them properly results in wrong answers. Cases in point are "higher order verbs" (as in "pack **attempts** to store the specified files in a packed form" - it may not succeed) and intensional uses of adjectives (as in "Only the super-user can allocate **new** files" - they don't exist yet).
- **Anaphoric references:** Although it has been argued that anaphoric reference (by means of pronouns or definite noun phrases) is irrelevant for document retrieval purposes (or even damaging) the situation is definitely different for AE/QA. Crucial information is often contained in sentences that refer to entities *only* by anaphoric references. Moreover, information is often given in technical manuals just once, so even one missed pronominal reference may seriously impair retrieval performance. Even for the relatively simple task of named entity recognition we must often have recourse to some of the techniques needed for reference resolution ("Bill Gates of Microsoft" &... "Gates" ... "the Gates company" etc.).
- **Pluralities**: Reference to groups of objects (be it through plurals ["dogs"] or through conjunctions ["Fido and Rover"]) is a well-known headache, in particular due to the different possible readings of plural noun phrases (collective/distributive/cumulative: "Fido and Rover fought/barked/ate up the food"). While in many cases it is possible to leave underspecified the exact number of objects introduced by pluralities this is no option when we want to get exact numbers from textual documents (e.g. via "how many"-questions).

The specific characteristics of newspaper texts that somehow overshadow these problems are:

1. **Range of topics:** Due to the vast range of topics covered by newspapers the topic of *sense ambiguity* becomes a top priority problem (cf. "Where is the Taj Mahal?"). In more restricted domains we can usually get away with little or no sense disambiguation (and if we have to perform it, it is much simpler than in open domains). Since sense disambiguation is a very thorny problem, domains where it is not of primary importance would be most useful.

   The wide range of topics also creates the rather ill-understood problem of the type "original vs. copy" ("What is the height of the Statue of Liberty?" - only the original, no models thereof).

2. **Time-dependence of information:** The things described by newspapers are mostly time-dependent

[4] A small number of definitional questions were included in TREC9. In TREC10 their number was significantly higher, due to the different source of the questions. It has however been observed that a corpus of newspaper articles is not the best place to search for answers to that type of questions (Voorhees, 2001).

("When was Yemen reunified?" or *"Who is the president of Ghana?")*. Keeping track of stages (i.e. the changes that the world is undergoing) is difficult (not least as we can, of course, refer to past states of affairs, and would therefore be able to process the various ways in which natural language encodes such information [the whole tense system!]).

3. **Volume of information:** The sheer volume of information in newspapers archives puts such a heavy burden on processing systems that a strong bias towards shallow analysis is created. One case in point is SRI's TACITUS which was replaced by FASTUS for the MUC competitions, for reasons of speed alone, although TACITUS is a much more powerful system.

Naturally, all these problems will have to be solved sooner or later but, in our opinion, the far more fundamental problems mentioned above could be approached best when kept somewhat sheltered from these minefields.

We certainly do not argue against the use of very large, TREC-like, collections of newspaper texts in the development and evaluation of AE/QA systems but argue for the early inclusion of more moderate volumes of technical texts representative of other, very important, types of documents.

5. Evaluation of AE/QA Systems

As experience gained in the past QA tracks has shown the question of how AE and QA systems shold be evaluated consists of at least two components:

1. What should the answer sets look like?

2. How should the quality of an answer be determined?

The *first* question concerns, among other things, the question of the size of the answer string and, connected with it, that of answer justifications. There is agreement that a fixed-length string that happens to contain the correct answer but in a wrong document context should not be counted as correct (e.g. the answer string "Bush" taken from a document written when George Bush was president but dealing exclusively with shrubs). However, this requirement forces assessors to consult the original document and determine whether the answer string is justified. Clearly a considerable element of uncertainty is entered into the evaluation that way (Is the justification allowed to be implicit in, and/or distributed over, the document? When is an answer justified?)[5].

For a pure AE system, i.e. one *retrieving* explicit answers rather than *computing* answers from possibly distributed, possibly implicit, information (as done by true QA systems) this problem can be contained somewhat by requiring systems to retrieve not fixed-length strings but (not necessarily contiguous) fragments of sentences of potentially unlimited length that, when concatenated, constitute the complete answer, ideally as a well-formed sentence, as seen in Fig. 2. That this is a sensible requirement becomes particularly obvious in technical domains. Consider, for instance, the question:

> **Do I need write permissions to remove a symbolic link?**

A 50-byte answer window may retrieve from the Unix manual, among others, the string:

```
" need write permission to remove a
symbolic link, "
```

Checking the document sentence will reveal that this string is a completely wrong answer as the sentence from which it was taken is:

> **Users do not need write permission to remove a symbolic link, provided they have write permissions in the directory.**

The arbitrary limit of 50 bytes just happened to cut off the crucial negation. However, requiring the AE system to return a complete, ideally well-formed, sentence will result in the justification to be part of the answer itself (in this case, the entire document sentence should be returned).

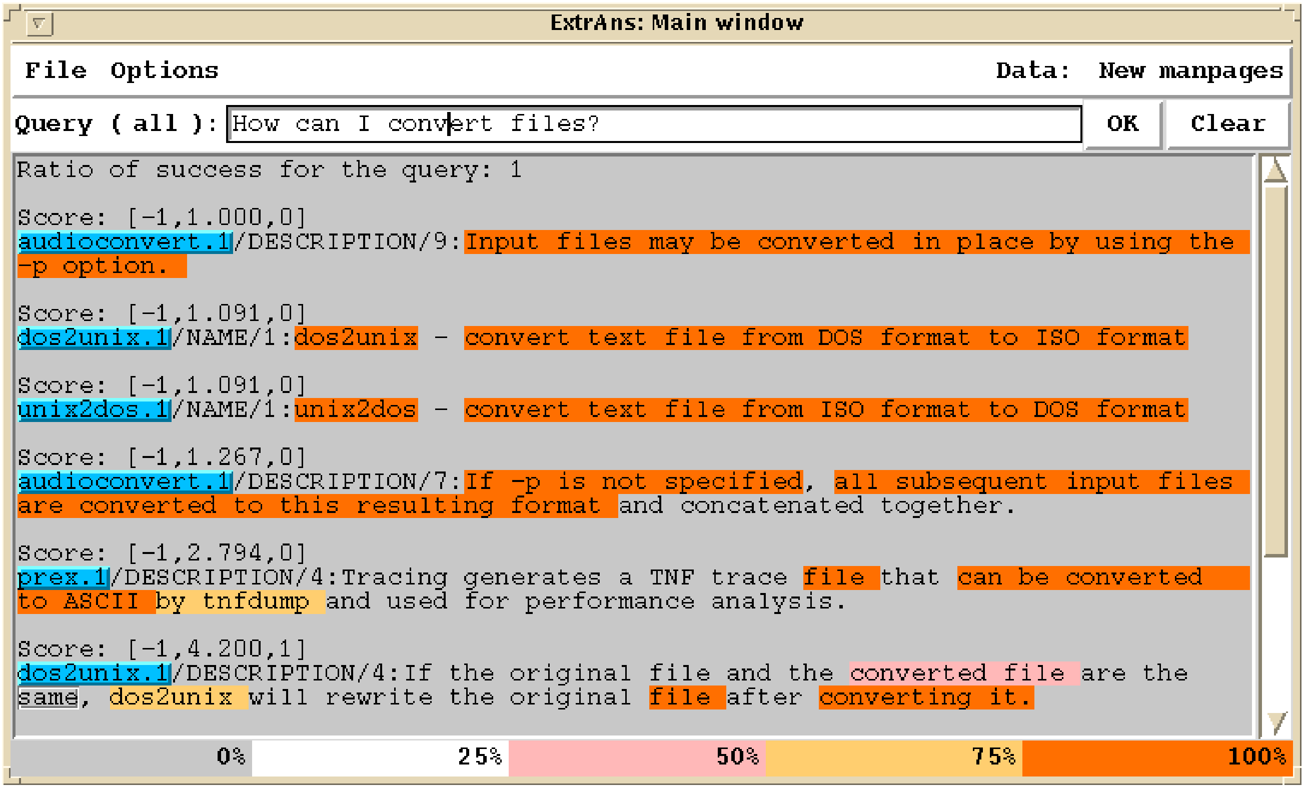
Another aspect of the first question concerns the *test queries*. Clearly, it is always better to use real world queries than queries that were artificially constructed to match a portion of text. By using, as we suggest, manuals of real world systems, it is possible to tap the interaction of real users with this system as a source of real questions (we do this by logging the questions submitted to our system over the Web). Another way of finding queries is to consult the FAQ lists concerning a given system available on the Web. By combining those two sources we compiled a list of 524 questions about the Unix domain. However, a large proportion of them is problematic as they have no answers in the document collection or are clearly beyond the scope of an automatic system (for example, if the inferences needed to answer a query are too complex even for a human judge). Nevertheless they are a useful starting point for a set of test queries in this domain.

Concerning the *second* issue, that of answer quality, the standard measures of Precision and Recall are not ideal for an Answer Extraction system, when applied to individual answer sentences. It can, in particular, be argued that Recall is significantly less important than Precision, as the aim of such a system is to provide (at least) one correct answer, rather than all the possible answers in a given collection. The user needs to find one good answer to a question and they are not interested in repeatedly finding the same answer.

In the Question Answering track of TREC a measure of precision is therefore used that takes this into account, viz. the Mean Reciprocal Rank (MRR). The Rank of a given result is the position in which the first correct answer is found in the output list of the system. Over a given set of answers the MRR is computed as the mean of the reciprocals of the ranks for all the answers.

The problem with this approach is that the underlying assumption, that an answer returned by an AE system is either completely correct or completely wrong, is not entirely realistic. Quite often we get a series of answers

[5] for the latter see:
http://www.isi.edu/natural-language/
projects/webclopedia/controv-trec10-eval.html



How can I convert files?

```
Ratio of success for the query: 1

Score: [-1,1.000,0]
audioconvert.1/DESCRIPTION/9:Input files may be converted in place by using the
-p option.

Score: [-1,1.091,0]
dos2unix.1/NAME/1:dos2unix - convert text file from DOS format to ISO format

Score: [-1,1.091,0]
unix2dos.1/NAME/1:unix2dos - convert text file from ISO format to DOS format

Score: [-1,1.267,0]
audioconvert.1/DESCRIPTION/7:If -p is not specified, all subsequent input files
are converted to this resulting format and concatenated together.

Score: [-1,2.794,0]
prex.1/DESCRIPTION/4:Tracing generates a TNF trace file that can be converted
to ASCII by tnfdump and used for performance analysis.

Score: [-1,4.200,1]
dos2unix.1/DESCRIPTION/4:If the original file and the converted file are the
same, dos2unix will rewrite the original file after converting it.
```

Figure 2: Identifying Relevant Parts of Sentences.

which are all correct to some degree but not entirely correct. We need some kind of weighting, exactly as in document retrieval, but again on the sentence level. The way this weighting should be performed is, however, less clear. One approach might be to find a representative set of correct answers by making a person write the ideal answers to a number of questions (labour-intensive but feasible), and then to find the sentences in the documents that are "semantically close" to these ideal answers automatically.

Semantic closeness between a sentence and the ideal answer, i.e. the weight of an answer sentence, could be computed by combining the two measures that one might call *"succinctness"* and *"correctness"*. Both measures compare a potential answer sentence with the ideal answer. Succinctness and correctness are the counterparts of precision and recall, respectively, but now on the sub-sentential level. These measures can be computed by checking the overlap of words between the sentence and the ideal answer (Hirschman et al., 1999), but we suggest a more content-based approach. Our proposal is to compare not words in a sentence, but their logical forms. Of course, this comparison can be done only if it is possible to agree on how logical forms should look like, to compute them, and to perform comparisons between them. The second and third conditions can be fulfilled if the logical forms are simple conjunctions of predicates that contain some minimal semantic information. In this paper we will use a simplification of the minimal logical forms used by ExtrAns (Schwitter et al., 1999). Below are two sentences with their logical forms:

(5) *rm removes one or more files.*
`remove(x,y), rm(x), file(y)`

(6) *csplit prints the character counts for each file created, and removes any files it creates if an error occurs.*
`print(x,y), csplit(x),`
`character-count(y),` **`remove(x,z),`**
`file(z)``, create(x,z), occur(e),`
`error(e)`

As an example of how to compute succinctness and correctness, take the following question:

Which command removes files?

The ideal answer is a full sentence that contains the information given by the question and the information requested. Since *rm* is the command used to remove files, the ideal answer is:

(7) *rm removes files.*
`remove(x,y), rm(x), file(y)`

Instead of computing the overlap of *words*, succinctness and correctness of a sentence could now be determined by computing the overlap of *unifying predicates*. The overlap of the unifying predicates ("overlap" henceforth) of two sentences is the maximum set of predicates that can be used as part of the logical form in both sentences. The predicates in boldface in the two examples above indicate the overlap with the ideal answer: 3 for (5), and 2 for (6).

Correctness of a sentence with respect to an ideal answer (recall on the predicate level) is the ratio between the overlap and the number of predicates in the ideal answer. In the examples above, correctness is 3/3=1 for (5) and 2/3=0.66 for (6). This means that (5) is completely correct in that it returns all the relevant predicates while (6) is only partially correct in that it describes the removal of files by a command but that this command is not the "ideal command" (the removal is, in fact, merely a side-effect of a command whose primary purpose has nothing to do with file removal).

Succinctness of a sentence with respect to an ideal answer (precision on the predicate level) is the ratio between the overlap and the total number of predicates in the sentence. Succinctness is, therefore, 3/3=1 for (5), and 2/8=0.25 for (6). This means that (5) returns only relevant predicates while (6) contains some extraneous material.

Finally, a combined measure of succinctness and correctness could be used to determine the semantic closeness of the sentences to the ideal answer. By establishing a threshold to the semantic closeness, one can find the sentences in the documents that are listed as answers to the user's query.

The advantage of using overlap of unifying predicates against overlap of words is that the (semantically highly relevant) *relations between the words* also affect the measure for succinctness and correctness. We can see this in the following artificial example. Let us suppose that the ideal answer to a query is:

(8) *Madrid defeated Barcelona.*
`defeat(x,y), madrid(x),`
`barcelona(y)`

The following candidate sentence produces the same predicates:

(9) *Barcelona defeated Madrid.*
`defeat(x,y),` **`madrid(y)`**`,`
`barcelona(x)`

However, at most two predicates can be chosen at the same time (in boldface), because of the restrictions of the arguments. In the ideal answer, the first argument of "defeat" is Madrid and the second argument is Barcelona. In the candidate sentence, however, the arguments are reversed. The overlap is, therefore, 2. Succinctness and correctness are 2/3=0.66 and 2/3=0.66, respectively.

While these ideas have not been implemented yet they may be useful as a contribution to the question of how answers in AE systems should be weighted according to their quality. While the "gold standard" (the ideal answers) would have to be compiled by hand, comparisons against this standard could be done in a wholly automatic fashion.

6. Resources

Some of the resources that we used in our work are:

a The Aircraft Maintenance Manual (AMM) for the Airbus A320. The original SGML markup has been converted into XML for simpler processing (in English, 120 MB total, 45 MB excluding markup).

b The Aircraft Troubleshooting Manual (ATM) for the Airbus A320. Original SGML converted into XML (in English, 62 MB total).

c The on-line manual of Unix (Solaris) in English.

d A list of 524 real user questions about Unix.

e A terminology database (semi-automatically extracted) for the aircraft manuals (approx. 3000 terms).

f Terminology Visualization Tools.
Additional XML markup that denotes the extracted terms is automatically inserted into the manual. The new markup tags can be tied to presentational information (given e.g. by CSS stylesheets), so that when the manual is browsed the terms are highlighted and differentiated from the rest of the text. Most modern web browsers are capable of handling such specification of the information.

Of these resources all the manuals are copyrighted but the lists (questions, terms) are not.

7. References

Steven Abney, Michael Collins, and Amit Singhal. 2000. Answer Extraction. In Sergei Nirenburg, editor, *6th Applied Natural Language Processing Conference*, pages 296–301, Seattle, WA.

Eric Breck, John Burger, Lisa Ferro, Warren Greiff, Marc Light, Inderjeet Mani, and Jason Rennie. 2001. Another system called qanda. In *(Voorhees and Harman, 2001)*.

Eric Brill and Philip Resnik. 1994. A rule-based approach to prepositional phrase attachment disambiguation. In *Proceedings of COLING*, volume 2, pages 998–1004, Kyoto, Japan.

John Burger, Claire Cardie, Vinay Chaudhri, Robert Gaizauskas, Sanda Harabagiu, David Israel, Christian Jacquemin, Chin-Yew Lin, Steve Maiorano, George Miller, Dan Moldovan, Bill Ogden, John Prager, Ellen Rilo, Amit Singhal, Rohini Shrihari, Tomek Strzalkowski, Ellen Voorhees, and Ralph Weischedel. 2001. Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A). http://www-nlpir.nist.gov/projects/pub/roadmapping.html.

C.L.A. Clarke, G.V. Cormack, D.I.E. Kisman, and T.R. Lynam. 2001. Question Answering by Passage Selection (MultiText experiments for TREC-9). In *(Voorhees and Harman, 2001)*.

Olivier Ferrett, Brigitte Grau, Martine Hurault-Plantet, and Gabriel Illouz. 2001. Qualc - the question-answering system of limsi-cnrs. In *(Voorhees and Harman, 2001)*.

Ulf Hermjakob. 2001. Parsing and Question Classification for Question Answering. In *ACL'01 workshop "Open-Domain Question Answering"*, pages 17–22.

Lynette Hirschman, Marc Light, Eric Breck, and John D. Burger. 1999. Deep Red: A reading comprehension system. In *Proceedings of ACL'99*, University of Maryland.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. 2001. Question answering in webclopedia. In *(Voorhees and Harman, 2001)*.

Kevin Humphreys, Robert Gaizauskas, Mark Hepple, and Mark Sanderson. 2001. University of Sheffield TREC-8 Q&A System. In *(Voorhees and Harman, 2000)*.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Diego Mollá, Gerold Schneider, Rolf Schwitter, and Michael Hess. 2000. Answer extraction using a dependency grammar. *Traitement Automatique de Langues (T.A.L.), Special Issue on Dependency Grammar*, 41(1):127–156.

Marius Pasca and Sanda Harabagiu. 2001. Answer mining from on-line documents. In *ACL'01 workshop "Open-Domain Question Answering"*, pages 38–45.

Fabio Rinaldi, Michael Hess, Diego Mollá, Rolf Schwitter, James Dowdall, Gerold Schneider, and Rachel Fournier. 2002. Answer extraction in technical domains. In *Proceedings of the CICLING02 Conference*, pages 360–369, February.

Gerold Schneider, Diego Mollá Aliod, and Michael Hess. 1999. Inkrementelle minimale logische formen für die antwortextraktion. In *Proceedings of 4th Linguistic Colloquium*, University of Mainz, September 7-10. FASK.

Rolf Schwitter, Diego Mollá, and Michael Hess. 1999. Extrans - Answer Extraction from Technical Documents by Minimal Logical Forms and Selective Highlighting. In *Proceedings of the Third International Tbilisi Symposium on Language, Logic and Computation*, Batumi, Georgia.

Daniel D. Sleator and Davy Temperley. 1993. Parsing english with a link grammar. In *Proceedings of the Third International Workshop on Parsing Technologies*, pages 227–292.

M. M. Soubbotin and S. M. Soubbotin. 2001. Patterns of Potential Answer Expressions as Clues to the Right Answers. To appear in Proceedings of TREC-10.

Ellen M. Voorhees and Donna Harman, editors. 2000. *Eighth Text REtrieval Conference (TREC-8)*. NIST.

Ellen M. Voorhees and Donna Harman, editors. 2001. *Ninth Text REtrieval Conference (TREC-9)*, Gaithersburg, Maryland, November 13-16.

Ellen M. Voorhees. 2000. The TREC-8 Question Answering Track Report. In *(Voorhees and Harman, 2000)*.

Ellen M. Voorhees. 2001. Overview of the TREC 2001 Question Answering Track. To appear in Proceedings of TREC-10.

Question Answering in the Infosphere: Semantic Interoperability and Lexicon Development

Steven Lulich*, Paul Thompson†

* Program in Linguistics
& Cognitive Science
Dartmouth College
Hanover, NH 03755
steven.m.lulich@dartmouth.edu

† Institute for Security Technology Studies
Dartmouth College
45 Lyme Road, Suite 200
Hanover, NH 03755, U.S.A
Paul.Thompson@dartmouth.edu

Abstract

Much recent question answering research has focussed on supporting the textual retrieval needs of intelligence analysts. Question answering may also play a role in other less textual domains, such as sensor networks, or the Joint Battlespace Infosphere (JBI). We propose a connectivistic database to serve as the core of a lexicon which may be used to improve current methods of question answering, as well as other natural language and ontology processing application

1. Introduction

The question answering vision (Carbonell et al., 2000) and roadmap (Burger et al., 2000) documents describe a five year program for research and development for question answering systems with a focus on how such systems could support the needs of an intelligence analyst. DARPA's Office of Information Exploitation (IXO) program has the mission to ". . . develop sensor and information systems with application to battle space awareness, targeting, command and control, and the supporting infrastructure required to address land-based threats in a dynamic, closed-loop process." IXO is developing 1-, 5-, and 20-year vision statements to meet the challenges of these systems. These dynamic information environments require intelligent middleware to broker services to connect information users and sources. For example, users pose natural language questions, which must be translated into the query languages and ontologies of the heterogeneous systems making up the JBI (United States Air Force Scientific Advisory Board, 1999, 2000; Infospherics, 2001). While technologies in this area will build on current DARPA programs providing tools for efficient human creation of ontologies (DARPA Agent Markup Language, 2002; DARPA Rapid Knowledge Formation, 2002), because of the dynamic, rapidly changing environment represented by the JBI, it is necessary that more automated approaches to semantic interoperability be developed, as well.

We suggest the desirability of a connectivistic database to serve as the core of a lexicon which may be used to improve current methods of question answering, as well as other natural language and ontology processing applications. Specifically, we illustrate the use of such a lexicon in the Joint Battlespace Infosphere (JBI). Related work has been done on statistical tools that automate the process of mapping from one ontology or grammar to another (Thompson, 2001). We are interested in building on this work, as well as using mixed-initiative approaches (Haller et al., 1999) to provide human input, where needed.

2. Lexicon Development: Application of Linguistic Knowledge to Natural Language Processing

2.1. Properties of natural language which may be mimicked computationally

Three aspects of natural language are submitted for consideration:

? Grammars consist of categories which may be cognitively manipulated synchronically or altered diachronically (Heine, 1997), such as phones, morphs, words, and grammatical classes. The categories within grammars are defined with respect to each other, much as the words of a dictionary are defined with respect to other words in the dictionary, and do not therefore line up evenly across languages (Whaley, 1997). For instance, the study of languages as diverse as English, Tagalog, Manchu, and !Xhosa has resulted in the understanding that lexical classes in different languages do not all conform to the same mould. Some languages employ lexical classes which are not employed in English, and vice versa. Furthermore, the same class in different languages may not be easily reconciled with each other, and the distinctions between classes, even noun and verb, can sometimes become blurred. Morphologists and psycholinguists such as Joan Bybee (1988) and Ardi Roelofs (1992), to name only two, have explored the idea of a connectivistic lexicon with some success, both conceptually and experimentally.

? Grammars do not consist only of minimal units and rules for combining them. It has been found that the human brain stores a far more redundant amount of linguistic information than had previously been thought. Work with aphasic patients shows that the use of rules in combining morphemes may be thought of as a back-up method for producing morphologically complex words when access to the lexicon fails (Badecker & Caramazza, 1998). Psycholinguistic experiments have shown that the timing of lexical access for morphologically simple words is not significantly different from the timing of lexical access for morphologically complex words, and phonetic and psycholinguistic studies indicate that some prosodic structures are stored as whole units alongside of the individual segments of which they are comprised (Levelt, 1999; Grzegorz Dogil, personal communication).

? Grammars are learned best by immature brains – brains with degraded short term memory – which may learn only general principles of grammar before narrowing down to specific principles (Deacon, 1997). Deacon outlines work done by others in cognitive and computer science which involved training of neural networks to learn a grammar to a relatively large degree of accuracy when the "short-term memory" of the network was disturbed. Studies by MacWhinney (1978) and Peters (1983) indicate that generalizations (rules) gradually emerge from stored rote forms, which are initially processed and stored as unanalyzed wholes, cf. (Bybee, 1988). These studies corroborate both the work done by Deacon, and the evidence that linguistic data stored in the lexicon is often redundant.

2.2. Proposal for the design of a lexicon which mimics these properties

A lexicon with five main components may serve to mimic these properties of natural language: a Pattern Finding Engine (PFE), Short Term Memory (STM), Long-Term Memory (LTM), Connectivistic Database (CD), and an Anchor Set (AS),

2.2.1. Pattern Finding Engine and Memory

The Pattern Finding Engine (PFE) searches a text for patterns, and, during the training phase of the lexicon, stores those patterns in the Short-Term Memory (STM), while the strings predictable from those patterns are stored in the Connectivistic Database. For instance, starting from scratch, the PFE recognizes a sentence such as "Johnny ate the apple" as a single unit. This imitates the theory derived from the work of Deacon, MacWhinney, and Peters above. This single unit is stored as a whole in the CD as an object of class "lexical unit." Exposure to more sentences, such as "Johnny ran away" and "The apple is red", enables the PFE to recognize "Johnny" and "the apple" as units, and to store them in the CD, along with "ran away" and "is red". Further exposure to sentences such as "Apples taste good" and "Jack and Jill ran up the hill" allows the PFE to recognize "ran" as separate from "away again", and "s" as a morpheme attached to "apple".

Initially, PFE is not better than chance at finding correct patterns. Therefore, potential patterns are stored in STM. As more and more occurrences of patterns in STM are found by PFE, the patterns in STM are stored in Long-Term Memory (LTM). Because some units larger than the segment or the word may occur with great frequency, the work of PFE together with STM and LTM allows an imitation of the theory that the lexicon is not redundancy free. This also allows us to capture idioms as whole chunks (Nunberg et al, 1994).

2.2.2. Connectivistic Database

An object of class "lexical unit" represents all of the information concerning a single unit. Within the object of class "lexical unit" is a set of objects of class "link". Each object of class "link" contains two variables: a pointer, pointing to one other object of class "lexical unit"; and a value corresponding to the strength of that connection. Each "lexical unit" also contains an activation value, which records and keeps track of the activation of that unit at all times. Activation is a measure of the probability that a certain unit will be the next one chosen out of the lexicon, and is determined by the amount of activation flowing to it through its connections with other activated units. Each "lexical unit" also has an abstract position variable, represented by an n-dimensional vector, which identifies a location for the "lexical unit" in an abstract n-dimensional Minkowsky space.

Throughout the training phase, with the help of PFE, STM, and LTM, the CD automatically organizes itself into an n-dimensional Minkowsky space. Categories are automatically approximated by defining opposing categories with respect to each other along a similar dimension. Sets of categories which are not defined with respect to each other are defined along different dimensions. Such definitions may be approximated without prior human or machine coding (Klein, 1998; Levine et al., 2001).

2.2.3. Anchor Set

Initial training of the lexicon is supervised by a human assigning certain "lexical units" to corresponding absolute concepts. Such "anchor points" provide the basis for translation from one grammar or ontology to another via the lexicon. English "chair" and German "Stuhl", for instance, refer to roughly the same concept. Therefore, the word "chair" in an English trained lexicon, and the word "Stuhl" in a German trained lexicon will both be anchored to the concept of "CHAIR". The Anchor Set (AS) can be used then to manipulate and align the abstract n-dimensional vector spaces of the two lexicons such that, by extrapolation, lexical units with nearly identical position vectors should theoretically be nearly identical in meaning or use, depending on the dimension. The more anchor points that are explicitly taught to the AS, the more accurate this alignment will be.

2.3. Discussion

To the best of our knowledge, though the ideas and evidence outlined in this paper in favor of a connectivistic view of the lexicon have been explored by linguists already, there has been no attempt to apply such a model to challenges in natural language processing. Certainly this may partially be attributed to the fact that the computing power necessary to undertake such a task has not long been available.

We believe that development of such a lexicon is relevant to Question Answering technology in several ways. First, the lexicon, whatever shape it may take, is an important and central part of any natural language processing application. Without it, language is simply noise. We believe therefore that the form of the lexicon has a direct effect on the overall performance of the application. Second, in answering a single question, it is often necessary to extract information from multiple sources of varying media and ontologies. The information coming from these disparate sources must somehow be fused together and outputted into yet another ontology or medium. Because this conception of a lexicon is easily trained, it is easily transportable across multiple domains and ontologies or grammars. As discussed in section 2.2.3, the Anchor Set allows translation from one ontology to another via the lexicon, thus enabling this kind of fusion of information. Finally, though certainly not exhaustively, the automatic categorization of words along different dimensions, and the connections between words may be helpful as a tool for word sense disambiguation.

3. Questions in the Infosphere

3.1. Background

Question answering in heterogeneous sensor networks involves some of the same issues as question answering in more textual domains, but also introduces other aspects. The answer to the question may not exist in the network at the time the question is asked. Sensors may need to be tasked to provide the answer. A mapping must be made between the language of the user and the descriptions of the functionalities of various sensors. There is high transaction volume in the Joint Battlespace Infosphere (JBI) and questions may overlap in various ways. Efficient question answering calls for query planning and optimization along the lines of work done in relational databases (Jarke & Koch, 1984) and knowledge bases, but with additional factors introduced by the distributed, mobile, highly dynamic nature of sensor networks. Also, because much of the data in these networks will be structured, question answering in this environment can also build on research on natural language interfaces to relational databases (Adroutsopoulos et al., 1995; Urro & Winiwarter, 2001).

The JBI consists of client users, databases, sensors, and filtering or fusion operations. These filtering or fusion operations are carried out by fuselets, lightweight data fusion elements. Fuselets use simple logical rules to take inputs from other elements of the JBI, such as sensors, or other fuselets, to derive fused information. The functionality of each fuselet is described using a Fuselet Markup Language (FML). The JBI is implemented as a publish and subscribe architecture, where each fuselet publishes its services and subscribes to the outputs of other elements of the JBI. Questions in the JBI are answered by breaking the question into components and efficiently routing the components through the JBI network of fuselets, databases, and sensors.

Although ontologies may be provided for various sub-domains, it may be necessary to rapidly create and map among ontologies on the fly. For example, a fuel truck may be represented in separate ontologies for target tracking and for logistics. It must be possible to: a) determine that the two representations are of the same type of entity, b) reason within the joint probability space represented by the two ontologies, and c) answer questions by fusing information from the two domains. We will investigate a variety of tools to achieve semantic interoperability. In addition to the linguistic approaches to lexicon development discussed in section 2, we plan to explore statistical, text-based mapping and subsumption tools (Woods, 1997; Buckland et al., 1999; Gey et al., 2001; Schatz, 2002).

3.2. A JBI Fuselet Example

As a simplified example of question answering in the Infosphere, consider the following. In a battlefield situation when an enemy target is to be fired upon, it is first necessary to ascertain that no friendly assets are in the vicinity that might be adversely affected. A subset of the JBI involving a network of sensors, radio transmitters operated by groups of soldiers, advanced Land Warrior personal GPS systems, current roster information, other sources of information, and fuselets would be needed to make this determination. The current location, velocity, and vector of all friendly assets would need to be determined. If processing this information takes too much time, the target opportunity might be missed. If the enemy target is fired upon without the information being processed accurately, friendly assets may become casualties. Personnel in the tactical operation center would submit a natural language query, "Are any friendly assets in danger of being hit, if the target at UTM grid coordinate XY123456 is fired upon?" This query would then be interpreted by the question answering system. Fuselet 1 would aggregate the outputs from the soldiers' radio transmitters. Fuselet 2 would aggregate the output of the GPS systems. Fuselet 3, with situational tracking software, would fuse the outputs of Fuselets 1 and 2. Fuselet 4 in the personnel services center would fuse outputs from databases with current roster information, as well as with outputs from other databases making adjustments to the current roster, e.g., lists of soldiers on medical leave. Fuselet 5 would fuse the outputs of Fuselets 3 and 4 and produce as output a report for the tactical operations center, answering the query.

4. Conclusions

We intend to address question answering issues in the JBI, in particular those concerning closed-loop sensor networks. Our domain has some overlap with that of the intelligence analyst described in the question answering vision and roadmap documents, but has significant differences, as well. We intend to build a sensor network integrated with textual messages. We will make use of ontologies, such as a sensor markup language, but we will also explore connectivistic lexicon, corpora linguistic, and other techniques to learn about our domains in a more dynamic manner, as necessary.

5. References

Androutsopoulos, I.; Ritchie, G.D.; & Thanisch, P. (1995). Natural language interfaces to databases – An introduction. Journal of Natural Language Engineering, 1(1), p.29-81

Badecker, W. & Caramazza, A. (1998). Morphology and Aphasia. In A. Spencer, & A.M. Zwicky (Eds.), The Handbook of Morphology (pp. 390-405). Oxford: Blackwell Publishers Ltd.

Buckland, M., Chen, A., Chen, H., Gey, F., Kim, Y., Lam, B., Larson, R., Norgard, B., & Purat, Y. (1999). Mapping Entry Vocabulary to Unfamiliar Metadata Vocabularies. D-Lib Magazine, 5(1).

Burger, J.; Cardie, C.; Chaudhri, V.; Gaizauskas, R.; Harabagiu, S.; Israel, D.; Jacquemin, C.; Lin, C..; Maiorano, S.; Miller, G.; Moldovan, D.; Ogden, B.; Prager, J.; Riloff, E.; Singhal, A.; Shrihari, R.; Strzalkowski, T.; Voorhees, E.; Weischedel, R. (2000). Issues, tasks and program structures to roadmap research in question & answering (q&a). Gaithersburg: National Institute of Standards and Technology.

Bybee, J. (1988). Morphology and Lexical Organization. In M. Hammond & M. Noonan (Eds.), Theoretical Morphology: Approaches in Modern Linguistics (pp. 119-142). San Diego, CA: Academic Press.

Carbonell, J.; Harman, D.; Hovy, E.; Maiorano, S.; Prange, J.; & Sparck Jones, K. (2000). Vision statement to guide research in question & answering (Q&A) and text summarization. Final version 1. Gaithersburg : National Institute of Standards and Technology.

DARPA Agent Markup Language (DAML) (2002). http://dtsn.darpa.mil/ixo/daml%2Easp.

DARPA Rapid Knowledge Formation (RKF). (2002). http://dtsn.darpa.mil/ixo/rkf%2Easp.

Deacon, T. (1997). The Symbolic Species: The Co-evolution of Language and the Brain. New York: W.W. Norton.

Gey, F.; Buckland, M.; Chen, A.; & Larson, R. (2001). Entry vocabulary – a technology to enhance digital search. In Proceedings of HLT 2001: First International Conference on Human Language Technology Research (pp. 91-95). San Francisco: Morgan Kaufmann.

Haller, S.; McRoy, S.; and Kobsa, A. (Eds.). (1999). Computational Models of Mixed-Initiative Interaction Boston: Kluwer.

Heine, B. (1997). Cognitive Foundations of Grammar. New York: Oxford University Press.

Infospherics: Science for Building Large-scale Global Information Systems. (2001). http://actcomm.dartmouth.edu/infospherics/

Jarke, M. & Koch, J. (1984). Query optimization in database systems. Computing Surveys, 16(2), 111--152.

Klein, A. (1998). Textual Analysis Without Coding: It Can be Done. Dissertation, Mathematical Social Sciences, Dartmouth College.

Levelt, W.J.M. (1999). Producing spoken language: a blueprint of the speaker. In P. Hagoort & C.M. Brown (Eds.) The neurocognition of language (pp. 94-122), Oxford: Oxford University Press.

Levine, J.H.; Klein, A.; & Mathews, J. (2001). Data Without Variables. Journal of Mathematical Sociology, 23(3), 225--273.

MacWhinney, B. (1978). The Acquisition of Morphophonology. Child Development Publication, Chicago: University of Chicago Press.

Nunberg, G; Sag, I.; & Wasow, T. (1994). Idioms. Language, 70, 491--538.

Peters, A.M. (1983). The Units of Language Acquisition. Cambridge, U.K.: Cambridge University Press.

Roelofs, A. (1992). A spreading-activation theory of lemma retrieval in speaking. Cognition, 42, 107--142.

Schatz, B.R. (2002). The Interspace: Concept navigation across distributed communities. IEEE Computer. 35(1), 54--62.

Thompson, P. (2001). Classification Crosswalks: From Interchange to Interoperability. Classification Crosswalks: Bringing Communities Together The 4th NKOS Workshop at ACM-IEEE Joint Conference on Digital Libraries (JCDL).

United States Air Force Scientific Advisory Board. (2000). Report on Building the Joint Battlespace Infosphere, vol. 1 Summary SAB-TR-99-02.

United States Air Force Scientific Advisory Board. (1999). Report on Building the Joint Battlespace Infosphere, vol. 2 Interactive Information Technologies SAB-TR-99-02.

Urro, R. & Winiwarter, W. (2001). Specifying Ontologies – Linguistic Aspects in Problem-Driven Knowledge Engineering. In Proceedings of the 2nd International Conference on Web Information Systems Engineering, Los Alamitos, IEEE Computer Society Press.

Whaley, L.J. (1997). Introduction to Typology: The Unity and Diversity of Language. Thousand Oaks, CA: Sage Publications.

Woods, W.A. (1997). Conceptual indexing: A better way to organize knowledge. Sun Microsystems Research Technical Report TR-97-61.

Multiple-perspective and Temporal Question Answering

James Pustejovsky
Dept. of Computer Science
415 South Street
Brandeis University
Waltham, MA. 02254
jamesp@cs.brandeis.edu
www.cs.brandeis.edu/~jamesp

Janyce Wiebe
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
wiebe@cs.pitt.edu
www.cs.pitt.edu/~wiebe

Mark Maybury
Information Technology Division
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
maybury@mitre.org
nrrc.mitre.org

Keywords: question answering systems, multiperspectives, temporal expressions, events

1.0 Introduction

The question answering vision (Carbonell et al. 2000) and roadmap (Burger et al.,2002) articulate a research and development direction for the next five years. Although a range of question and answer types are described, the ability to interpret a question and provide an answer with respect to different perspectives and the ability to answer questions involving temporal dimensions are largely unaddressed. This position paper argues for the importance of multiple perspective and temporal question answering and attempts to outline some aspects of the problem that would be important to capture on the Q&A roadmap. We address these problems in the context of two ARDA Northeast Regional Research Center (NRRC) Workshops, held in the summer of 2002, focused on time and multiple perspectives (nrrc.mitre.org).

2.0 Multiple-Perspective Question Answering

2.1 Multiple-Perspective Questions

A question may explicitly request multiple perspectives, for example ``What are the positions of German political parties on UN resolution 53?" or "What opinions are being expressed in the world press about US plans to invade Afghanistan?" In addition, questions asking for speculations or opinions might most appropriately be interpreted as asking for answers from multiple perspectives. Examples are ``How should the US response be to the terrorist incident?'' and "Will the US economy improve in the next six months?" Even for other questions, a multiple-perspective treatment may be very useful to an analyst or consumer. The user could be given the option to ask for multiple perspectives, whatever the specific form of the question. Finally, questions themselves can signal the perspective of the source or speaker (who could hold distinct views) while at the same time eliciting a multiperspective response as in "What do the Europeans think about the short-sighted US policy in the Middle East?".

2.2 Multiple-Perspective Answers

Perhaps the most obvious situation in which a question may be answered differently from multiple perspectives is when people or groups hold different beliefs about what is factually true. However, answers from different perspectives also include ideological beliefs, religious beliefs, evaluations, judgments, and speculations. They might reflect personally held beliefs, or official positions in legal, political, religious, or ideological platforms. In addition, the source of the belief might be a specific person, a group, a political or economic sector, or even the general culture at large. Recognizing the type of perspective reflected in an answer is essential for knowing how to interpret the information and what we can learn about the source.

We can envision a system that does not provide a single answer but rather presents the various positions on a topic currently being expressed in the world press, to help the user answer the question for himself or herself.

For the results to be useful, they should be characterized and clustered for presentation to the user. Storing the results in a knowledge base would support reasoning about multiple perspectives on a topic, and detecting changes in perspective and trends over time.

Thus, five main aspects of the problem are the following:

? Retrieval of text segments containing candidate answers from multiple perspectives (Wiebe 1994, Wiebe et al. 1999).

? Characterization of the type of perspective of each answer. The answer may be presented as factual in the original source, or as a belief or opinion. It might reflect personally held beliefs, or official positions in legal, political, religious, or ideological platforms. It might be positive or negative evaluative, or speculative.

? Characterization of the source of the perspective. The source of the perspective may be an individual, a group, a political or economic sector, etc. Because beliefs about beliefs about beliefs, etc., may be presented, a structured representation of sources is needed.

? Comparison and clustering of the answers into similar perspectives, for presentation to the user.

? Representation of the answers in a knowledge base. As questions are answered from multiple perspectives over time, storing the results in a knowledge base would support queries such as which sources have expressed negative evaluations toward various topics, or which perspectives have changed over time.

Following are examples of multiple perspectives expressed in text. First, here are different views expressed about the same topic in editorials.

“General Musharraf has wisely chosen to throw in his lot with the US.” (from *The India Times*).

“Looking at the event from the beginning most people including myself were convinced that President Musharraf's decision to support the USA was ill-thought, ill advised and was only taken for financial reward in a hurry.” (from *The Frontier Post, Pakistan*)

In the following passage, which describes a factual dispute, the sources of the perspectives are people mentioned in the text:

“Agha [Tayab Agha, spokesman for Taliban leader Mohammad Omar] claimed the Taliban continued to rule in Kandahar, Oruzgan, Zabol, Ghazni and Helmand provinces. Afghan and Western sources, along with travelers who arrived today in Spin Boldak, disputed his claim, saying the Taliban only control parts of most of these provinces and had no influence over Ghazni at all (from *The Washington Post Foreign Service*).

A rich representation is needed to capture the characteristics of perspectives, their sources, and their objects, which may themselves be perspectives.

In addition to involving answers from multiple perspectives, questions often refer explicitly to time sensitive information, the area of question answering which we consider next.

3.0 Temporal Question-Answering: When time makes a difference

Humans live in a dynamic world, where actions bring about consequences, and the facts and properties associated with entities change over time. For this reason, temporally grounded events are the very foundation from which we reason about how the world changes. To be sure, named entity recognition is crucial to analyst reporting, information extraction, and question-answering systems; but without a robust ability to identify and extract events and time-stamps from a text, the real "aboutness" of the article can be missed. Moreover, entities and their entities change over time as well; hence a database of assertions about entities will be incomplete or incorrect if it doesn’t reflect such time-stamps (e.g., the status of the World Trade Center Buildings before and after Sept. 11, 2001). To this end, event recognition drives basic inferences from text.

The focus of the Time and Event Recognition for Question Answering (TERQAS) workshop (time2002.org) is to address the problem of how to answer temporally-based questions about the events and entities in news articles. Currently, questions such as those shown below are not generally supported by Q&A systems:

1. Is Gates currently CEO of Microsoft? (*time-stamp* question)
2. When does the seminar take place? (*punctual event* question)
3. How long did the hostage situation in Berlin last? (*Duration of event* question)
4. On what days were there bombings in the Middle East? (*Quantified event* question)
5. What airplane crashes occurred shortly after assassinations? (*Quantified event* question with *relative event ordering*)
6. What terrorist actions occurred within a week of political speeches by extremist governments? (*Quantified event* question with *relative event ordering*)
7. What bombings have occurred during the occupation of the West Bank? (*Quantified event* question with *durative event overlapping*)

What characterizes these questions as beyond the scope of current systems is the following: they refer, respectively to the temporal properties of the entities being questioned, the relative ordering of events in the world, and events that are mentioned in news articles, but which have not occurred at all.

3.1 Temporal Question-Answering Challenges

There has recently been a renewed interest in temporal and event-based reasoning in language and text, particularly as applied to information extraction and reasoning tasks (cf. Pustejovsky and Busa 1995; Mani and Wilson 2000; 2001 ACL Workshop on Spatial and Temporal Reasoning). Several papers from the workshop point to promising directions for time representation and identification (cf. Setzer and Gaisauskas, 2001, Filatova and Hovy, 2001, Schilder and Habel, 2001). Many issues relating to temporal and event identification remain unresolved. In our efforts we aim to (a) to examine how to formally distinguish events and their temporal anchoring in text (news articles); and (b) to evaluate and develop algorithms for identifying and extracting events and temporal expressions from texts.

Relative to the first goal above, we are addressing four basic research problems:

1. Time stamping events (identifying an event and anchoring it in time)
2. Ordering events with respect to each other (relating more than one event in terms of precedence, overlap, and inclusion)
3. Reasoning about the ramifications of an event (what is changed by virtue of an event)
4. Reasoning about the persistence of an event (how long does an event or the outcome of an event persist)

3.2 TimeML and TIMEBANK

To answer these problems, we are presently working to define a specification language and an annotated Gold Standard corpus. A specification language, TimeML, will be defined and developed. This XML-compliant language should formally model most of the following properties of time and events:

1. How to represent the interval values of events (time-stamping);
2. How to represent aspectual properties of an event (what phase of an event is being time-stamped);
3. How to represent all possible temporal ordering relations between two events;
4. How to model shallow (entailed) ramifications of an event (what related events are triggered by an event's occurrence);
5. How to model when a state persists and when it does not (what states follow from an event)

Once the initial definition and specification of TimeML is complete, it will be necessary to begin annotation on a large number of news articles, in order to create a Temporal Gold Standard (TIMEBANK). This entails the annotation of at least 400 articles, taken from four separate sources: 100 DUC articles; 100 ACE articles; 100 AP News articles; and around 100 PropBank annotated articles. We are presently in the process of the construction of TIMEBANK, the annotated corpus that we will provide as a community resource when completed, subject to appropriate copyright restrictions.

The specification language TimeML will suggest but not determine the nature of how answers to temporal questions are best presented to the user. This remains largely an issue of habitability and usability of the application. Nevertheless, answers to temporal questions may take one of several forms:

1. Selections from database entries, populated from the appropriate information extraction algorithms;
2. Textual fragments from news articles, indicating total or partial answers to the question;
3. Answers may be abstracted and represented visually in terms of a timeline or a hyperbolic visualization algorithm.

The second goal mentioned above involves the evaluation of existing, and development of new temporal extraction algorithms. The four research problems given above correspond roughly to extraction algorithms of increasing degrees of sophistication and complexity. Time stamping events is not too dissimilar from named entity recognition; event ordering identification is somewhat similar to relational parsing; and capturing persistence and ramification properties of events is similar to identifying dependencies in a dependency grammar.

The algorithms will be applied and tested against the development corpus of the gold standard, TIMEBANK. Evaluation against a blind test set will measure for accuracy of answers for a range of questions, as defined by the participants, paying particular attention to target the specific temporal properties of the text with different questions.

Significantly, the results of our workshop will enable the community to begin addressing an entirely new type of question-answering capability, and one that is necessary for answering questions pertaining to the deeper content of news articles.

4.0 Implications for Q&A Road Map

The above observations point to the importance of research into multi-perspective and temporal Q&A. Some of the key milestones on the roadmap include:

- Characterize the types and nature of multiple perspectives and temporal aspects
- Establish and iteratively refine an ontology of multiple perspectives both for question analysis and answer generation. Do the same for temporal questions.
- Create corpora that include both multiple perspective and temporal phenomena
- Create annotation standards for multiple perspective and temporal markup

The two NRRC workshops described in this article will contribute in the next three months to advancing the state of the art by creating:

- An ontology of perspective
- An annotated corpus of multiple perspective questions and answers
- A repository of linguistic clues indicative of perspective
- A baseline of experimental results (segmentation, property annotation, clustering)
- A standard markup language for temporal and event expressions, TimeML
- A gold standard corpus for temporal expressions, TIMEBANK

5.0 Conclusion

This paper describes two important aspects of question answering that have gone largely unaddressed: time and multiple perspectives. These are important elements that should be reflected in the Q&A roadmap.

6.0 References

Allen, J. "Maintaining Knowledge about Temporal Intervals", Communications of the ACM, 26(1):"832-843.

Burger, John; Cardie, Claire; Chaudhri, Vinay; Gaizauskas, Robert; Harabagiu, Sanda; Israel, David; Jacquemin, Christian; Lin, Chin-Yew; Maiorano, Steve; Miller, George; Moldovan, Dan; Ogden, Bill; Prager, John; Riloff, Ellen; Singhal, Amit; Shrihari, Rohini; Strzalkowski, Tomek; Voorhees, Ellen; Weischedel, Ralph. (2002) “Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A). www-nlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc

Carbonell, Jaime; Harman, Donna; Hovy, Eduard; Maiorano, Steve; Prange, John; and Sparck Jones, Karen. 2000. “Vision Statement to Guide Research in Question & Answering (Q&A) and Text Summarization”. Final version 1.www-nlpir.nist.gov/projects/duc/papers/Final-Vision-Paper-v1a.pdf

Filatova, E. and E. Hovy (2001) "Assigning Time-Stamps To Event-Clauses", in Proceedings of ACL Workshop on Temporal and Spatial Information Processing, Toulouse, France, July, 2001.

Mani, I., Wilson, G., Ferro, L., and Sundheim, B. 2001. Guidelines for annotating temporal information. Proceedings of Human Language Technology Conference. hlt2001.org/papers/hlt2001-31.pdf

Mani, I. and G. Wilson (2000) Robust Temporal Processing of News", in Proceedings of the 38th Annual Meeting of the ACL, Hong Kong.

Northeast Regional Research Center (nrrc.mitre.org)

2001 ACL Workshop on Spatial and Temporal Reasoning.

Pustejovsky, J. and F. Busa (1995) A Revised Template Description for Time in MUC-6 (v3), http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html.

Schilder, F. and C. Habel (2001) "From Temporal Expressions to Temporal Information: Semantic Tagging of News Messages", in Proceedings of ACL Workshop on Temporal and Spatial Information Processing, Toulouse, France, July, 2001.

Setzer, A. and R. Gaizauskas (2001) “A Pilot Study On Annotating Temporal Relations In Text ", in Proceedings of ACL Workshop on Temporal and Spatial Information Processing, Toulouse, France, July, 2001.

Wiebe, J., R. Bruce, and T. O'Hara (1999) “Development and Use of a Gold Standard Data Set for Subjectivity Classifications”, in *Proc. 37th Annual Meeting of the Assoc. for Computational Linguistics (ACL-99),*

Wiebe, J. (1994) “Tracking Point of View in Narrative”, *Computational Linguistics*, 20.2: 233-287.

Summarization-Based Japanese Question and Answering System from Newspaper Articles

Yohei Seki[*] and Ken'ichi Harada[*]

[*]Department of Computer Science, Keio University
Kanagawa, Japan 223-8522
yohei.seki@dream.com

Abstract

Recently, many researchers are focusing on the application of Natural Language Processing (NLP) techniques such as summarization, information extraction, and text mining. One of the challenges with these technologies is developing an accurate Question and Answering System (Burger et al., 2001). In this paper, we will discuss Japanese Q&A problematic issues that have appeared in our experimental system. Our system is implemented with multi-document summarization (MDS) techniques.

Keywords: Japanese Q&A System, multi-document summarization technique, information fusion from multiple newspaper articles, and QAC (Question and Answering Challenge)

1. Introduction

There is a year long workshop being held by the National Institute of Informatics in Japan called NTCIR-3. We participated in the 'Question and Answering Challenge' (QAC) dryrun (Fukumoto and Kato, 2001) in the winter of 2001: Japanese Q&A tasks. We created an experimental system for the Japanese Q&A to detect problems specific to the Japanese language. Our input data was Mainichi Newspaper articles from 1998 and 1999 Year. This included about 230,000 articles. In this paper, we propose a multi document summarization based approach for Q&A. We also discuss some Japanese related problematic issues.

This paper consists of seven sections. We explain the tasks of QAC in Section 2, and discuss details of our system design and approach in Section 3. Section 4 provides an overview of our system user interface. Section 5 contains a brief evaluation of our system with QAC problems. In Section 6, some problematic issues are discussed. Finally, we present our conclusions in Section 7.

2. Question and Answering Tasks in QAC

The Question and Answering Challenge workshop (QAC) (Fukumoto and Kato, 2001) consisted of three tasks. The first and second task contained the same 50 questions. A list of five accurate answers was the goal in the first task; The goal of the second task was to extract the correct answer set. The third task had 10 problems and each problem had one follow-up question. The dryrun with these three tasks was held on five consecutive days in December, 2001.

The Answers were to be noun phrases which indicated a person's name, organization names, money, size, date and so on. The source documents were a two-year-period of Japanese newspaper articles.

3. Our Multi-Document Summarization Based Approach for the Q&A System

Our approach for the Q&A System consisted of three procedures: question analysis, summarization of questions from various articles, and answer formation.

3.1. Question Analysis

The Question analysis process is basically divided in two parts. One is the detection of question type, and the other is the extraction of keywords with a numeric score that summarizes documents. We use the Japanese part-of-speech tagger, 'Chasen'[1] in order to break the question sentences into morphemes. Question types are categorized with keywords as follows:

Interrogative pronoun		modifying suffix	
Nan(-i)	(What)	Nen	(Year)
		Gatsu	(Month)
		Nichi	(Day)
		Nin	(How many people)
		Kai	(How much times)
		Ken	(How many units)
Dare	(Who)		
Doko	(Where)	Kuni	(Which country)
		Kaisha	(Which company)
Itsu	(When)		
Ikura	(How much)		
Dono, Dore	(Which)	Kikan	(How long)
		Ryou	(The amount)

Figure 1: Japanese Question Taxonomy

[1] http://chasen.aist-nara.ac.jp/

The question taxonomy above shows that Japanese question types are determined by a combination of an interrogative pronoun and a modifying suffix.

Another process is keyword detecting and scoring. We score keywords in each question as follows:

1. Each matching noun morpheme receives 1 point.

2. The proper noun or phrase containing the proper noun receives 3 points.

3. A time related adverb/noun receives 0.5 points.

4. Each verb or adjective morpheme (except some basic elements) receives 1 point.

3.2. Sentence Extraction with Multi-Document Summarization Technique

Next, we extracted sentences related to each question keywords from a two year supply of newspaper articles. The question keyword scores determine these individual sentence scores.

If a sentence contains a keyword, the keyword score is added to the sentence score, then the sentence score is divided by the sum of all the keyword scores in that question. Therefore, a maximum score of a sentence is 1. If a score of any sentence is more than 0.4, the sentence is extracted and stored into the answer file for that question. This is a kind of cut and paste summarization technique (Jing and McKeown, 2000) from a wide source of newspaper articles (McKeown and Radev, 1995). In order to accelerate our system's performance, some multi-document summarization techniques (Mani, 2001) with text segmenting and clustering (Stein et al., 1999) were also needed. When this MDS approach is adopted, the Q&A accuracy performance must be kept in mind. MDS has some information fusion or aggregation processes to avoid overlapping information. If this process was applied wrongly, the correct answer would be removed from summary. We did not implement this process at this stage but implemented a similar process at the answer formation stage.

3.3. Answer Formation from Summary Sentences

Answer Formation is the process of extracting answers from summary sentences using question types. We implemented this step as pattern matching according to question type information with Perl. We use question type information like Nan-Nen Nan-Gatsu (In what year and month did the event happen?), and encode that information in regular expressions like $/(0-9)\{1,2\}gatsu(0-9)\{1,2\}nichi/$ in order to detect answer candidates.

Some question types were needed to extract distance patterns or make answers with a parsing technique. We implemented noun formation functions according to question types with a recursive function about part-of-speech information (concerned with noun morpheme type). The noun phrase formation process was different according to question types and was localized with Perl functions. Some examples are as follows:

1. Who (Dare) Questions
   'Chasen' tagged personal names as 'noun-proper noun-personal name'. When 'Chasen' tagged a personal name correctly, the personal name is extracted based on the noun formation. In addition, an abbreviated name like 'J.F.K.' or some hard to place place noun needs to be extracted with an answer formation process. This type answer was not tagged correctly with the morpheme tagger. Therefore, we need some parsing technique to look before and after the part-of-speech information.

2. When (Itsu) Questions
   'When' questions' difficulties mainly stemmed from unknown details: What year, month, day, or time? We extracted answers from 'when' questions with time-related number extraction and formation. When some time-related suffixes were matched, this pattern was formed following Japanese conventional time-expressing order; year, month, and day. When time information was expressed with 'of' or other modifying terms, there might be gaps between some time expressions. For example, 'In Keicho 5 (1600), the war of Sekigahara started on the 15th September.' The year and the date are separated in the sentence but both are necessary in an answer. If that information together was expressed in one sentence, our system would have no problem extracting the correct answer to form one time expression.

3. Where (Doko) Questions
   'Where' questions also varied in their answers according to the details. To find a specific location of an event such as a war in East Timor in Indonesia, the initial input question might not be able to place 'Daerah Istimewa Aceh' province without wider geographic information. The morpheme tagger tagged a place noun as 'noun-...-place' and a country name noun as 'noun-...-place-country'. In our system, this distinction is judged mainly based on question keyword information. When the question was judged to be concerned with country name, the corresponding function was called.

4. Amount Questions
   In the Japanese language, amount information is characterized with a modifying suffix like 'liter' or 'cubic meter'. Therefore, this suffix information is key in extracting an answer. Number information was tagged correctly as 'noun-number' or 'prefix-auxiliary-number'. Our system formed these elements to make quantity noun phrases.

Extracted answers were scored with their source sentence score and their occurring frequencies. Some answer candidates with same meanings were merged to a single answer with information fusion or aggregation techniques to avoid overlapping answers.

3.4. Detecting Answers for Follow-up Questions

In Task 3, we employed a different approach because follow-up questions often contain pronouns instead of nouns and don't contain specific keywords. To extract an answer in a follow-up question, we use a summary from the first question and the question type pattern in the follow-up question. Some follow-up question examples are shown as follows:

1. (a) What are the titles of Mr. Natsume Soseki's most famous work?
   (b) What was his eldest son's occupation?
   (his = Mr. Natsume Soseki)
2. (a) When did the 'Aerosmith' make their debut?
   (b) What was their first hit at that time?
   (at that time = their debut time)
3. (a) What are the three biggest festivals in Japan?
   (b) Where are those festivals held?
   (those = the three biggest)

4. System User Interface

The Q&A system produced summaries including sentence weights and source article ID numbers. They were tagged in XML-style formats. When the answer formation process was executed, answers were provided with their occurring articles by using summary information. This system is shown in Figure 2.

5. Evaluation

QAC results were evaluated with MMR (Maximal Marginal Relevance) scoring (Mani, 2001) and F-score (or F-measure) (Stein et al., 2000) metrics. Some bugs in our system were removed after the dryrun was finished. The results of our present system are shown as follows:

1. Task 1 (Top five Q&A)
   Task 1 had 50 questions. We scored the top 5 answers as follows: if the best answer was in fact correct, 1 point was added to the score; if second best answer was correct, 0.5 points was added to the score; ...; if the fifth best answer was correct, 0.2 points was added to the score. The total score ranges are shown in Table 1.

Score	Rates
$1 \leq \; -$	$\frac{13}{50}$
$0.5 \leq \; - \; < 1$	$\frac{7}{50}$
$0 < \; - \; < 0.5$	$\frac{8}{50}$
0	$\frac{22}{50}$

Table 1: Scoring in Task 1

   Answer scores with over 1 point contained four time-related questions, two questions about organization and personal names, one question about great literary and artistic works, money, people, units, and countries.

2. Task 2 (Answer Set)
   Task 2 had the same questions as Task 1. The goal of Task 2 was to extract the correct answer set. Our system answered this task as the best 10 answers. F-score ($\frac{2 \times Precision \times Recall}{Precision + Recall}$) ranges are shown in Table 2.

F-score	Rates
$0.6 < \; - \; \leq 1$	$\frac{2}{50}$
$0.4 < \; - \; \leq 0.6$	$\frac{3}{50}$
$0.2 < \; - \; \leq 0.4$	$\frac{9}{50}$
$0 < \; - \; \leq 0.2$	$\frac{19}{50}$
$-$	$\frac{17}{50}$

Table 2: F-score in Task 2

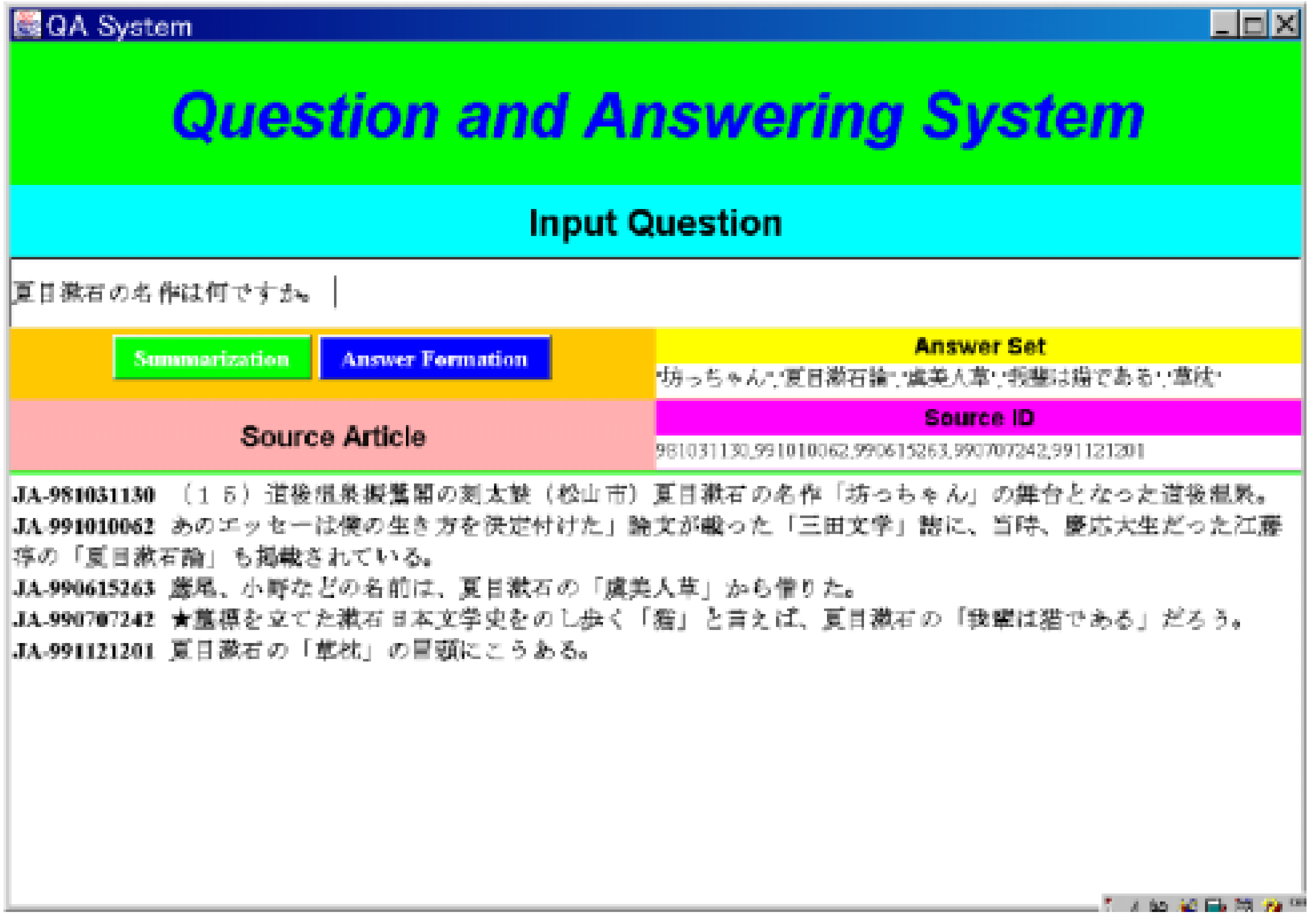
   Questions with the best two scores were concerned with literary and artistic works and countries. Both questions contained multiple answers.

3. Task 3 (Follow-up Q&A)
   Task 3 had 10 follow-up questions to each of the original questions. Out of the 10 questions, two questions contained correct answers in the top rank: they were a time-related question and a question about debut work. Another three questions contained correct answers in the top five ranks. Another two questions contained correct answers. The remaining three questions did not come up with a correct answer: questions concerning occupations, ranks, and personal names.

6. Some Problematic Issues

In this research, we only used surface information and didn't use deeper semantic information like a thesaurus would provide. Our result set contained erroneous elements, but in Task 2, $\frac{2}{3}$ of the correct answers were found. There are two reasons why correct answers were not found: there was too much erroneous information extracted and the correct answers were not extracted and put in the initial summary.

The source input data of QAC contained a very large (about 230,000) amount of articles. Our system caused some time-consuming problems because our system extracted summaries with common weighing values for every question type. Some questions extracted too many summary and others didn't extract enough summaries. In fact, the assigned threshold 0.4 was very sensitive according to question types. When this threshold was set as '> 0.4' (not equal), some questions contained more accurate answers in the best 10 answer candidates, but other questions' answers were missed. Although our threshold , of course, can be changed easily according to question type, some explicit criteria between threshold values and question types were hard to establish. In addition, when commonly used and polysemous question keywords were detected, many sentences with erroneous elements were extracted.



Input Question

夏目漱石の名作は何ですか。

Answer Set

'坊っちゃん','夏目漱石論','虞美人草','我輩は猫である','草枕'

Source ID

981031130,991010062,990615263,990707242,991121201

Source Article

JA-981031130 （１５）道後温泉振鷺閣の刻太鼓（松山市）夏目漱石の名作「坊っちゃん」の舞台となった道後温泉。
JA-991010062 あのエッセーは僕の生き方を決定付けた」論文が載った「三田文学」誌に、当時、慶応大生だった江藤淳の「夏目漱石論」も掲載されている。
JA-990615263 藤尾、小野などの名前は、夏目漱石の「虞美人草」から借りた。
JA-990707242 ★道標を立てた漱石日本文学史をのし歩く「猫」と言えば、夏目漱石の「我輩は猫である」だろう。
JA-991121201 夏目漱石の「草枕」の冒頭にこうある。

Figure 2: Q&A System

On the other hand, answer quality problems mainly stemmed from the question analysis quality. Questions which extracted too much erroneous information were mainly concerned with unique personal names or too specific place names. Other questions which did not contain correct answers were relatively unique-patterned questions. In order to increase the accuracy, we need to use a more semantic sensitive program.

We explained our improvement strategy for the Japanese Q&A problematic issues. In Japanese, there are two ways to say 'in the second place': "Dai-ni-i" and "ni-i". In the latter, the prefix "Dai" is omitted. We implemented a noun phrase formation to detect an answer with a parsing technique, but the two Japanese examples above came up with two different answers. A technique in detecting same meanings to make a single answer is also needed. This technique is a kind of multi-document summarization technique (Mani, 2001), especially for information fusion from multiple sources.

7. Conclusions and Future Direction

We tested our experimental Q&A System mainly using morpheme type information and the multi-document summarization based technique. Our results contained $\frac{2}{3}$ of the correct answers and each answer was provided with its occurring article ID number. Therefore, our system is useful for checking results with people.

In Japanese, question analysis process is a little more complex than English because question type is determined with the combination of interrogative pronoun and modifying suffix. A parsing and information fusion techniques regarding Japanese morphemes are needed in implementing the answer formation process.

In order to improve our results, some semantic information for the question category or taxonomy of inquiries (Burger et al., 2001) may be needed to reduce the amount of incorrect answers from a large summary source. In addition, if the assigned threshold for summarization is changed according to question type information, better results will follow. In order to determine precise thresholds according to question types, we will try more Q& A tasks and adjust our system.

Acknowledgements

We thank the National Institute of Informatics and members held for NTCIR-3 QAC, and also thank Robert B. Whitehead for re-reading the English.

8. References

J. Burger, C. Cardie, V. Chaudhri, R. Gaizauskas, and S. Harabagiu et al. 2001. Issues, tasks and program structures to roadmap research in question & answering (q & a). http://www-nlpir.nist.gov/projects/duc/roadmapping.html.

J. Fukumoto and T. Kato. 2001. An overview of question and answering challenge (qac) of the next ntcir workshop. http://www.nlp.cs.ritsumei.ac.jp/qac/qac-ntcirWS2.pdf.

H. Jing and K. McKeown. 2000. Cut and past based text summarization. In ANLP-NAACL 2000, Seattle, WA USA, May.

I. Mani. 2001. Automatic Summarization, volume 3 of Natural Language Processing. John Benjamins, Amsterdam, Philadelphia, first edition.

K. McKeown and D. R. Radev. 1995. Generating summaries of multiple news articles. In the 18th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pages 74–82, Seattle, WA USA, July.

G. C. Stein, T. Strzalkowski, and G. B. Wise. 1999. Summarizing multiple documents using text extraction and interactive clustering. In Pacific Association for Computational Linguistics (PACLING-1999).

G. C. Stein, T. Strzalkowski, G. B. Wise, and A. Bagga. 2000. Evaluating summaries for multiple documents in an interactive environment. In 2nd Int. Conf. on Language Resources & Evaluation (LREC2000).

Question Answering system for POLISH (POLINT) and its language resources

Zygmunt Vetulani

Adam Mickiewicz University
Dept. of Computer Linguistics and Artificial Intelligence
ul. Umultowska 87, PL-61614 Poznan, Poland
http://main.amu.edu.pl/~vetulani
vetulani@amu.edu.pl

Abstract

In this paper we would like to present several issues related to our long-term research on question answering in Polish. Experiment-generated corpus of question-answer pairs, as well grammatical resources for developing Q&A systems for Polish language are presented.

1. Introduction

The research reported in this paper is a part of a long-term project aiming at the software platform with emulated linguistic competence to study man-machine interaction (Vetulani, 2000b; Vetulani & Marciniak, 2000). A question answering system constitutes an essential part of this project. The name POLINT stands for successive versions of systems derived from the Polish module to the ORBIS system (Colmerauer, Kittredge). What makes an essential difference with respect to its predecessors is that POLINT may be used as an interface in real-time systems because of substantial efficiency improvement. The recent version[1] of the system enables the user to ask questions concerning an episode of a football match (cf. Appendix 1). Two further systems are now being developed: the ACALA virtual robot controlled by the natural language interface (Vetulani& Marciniak, 2000) and a virtual "interactive glass case" for Archeological Muzeum in Poznan (MUZARP, by Vetulani and Gribko).

Our research, the substantial part of which is presented in this paper, focussed first of all on the following issues among those mentioned in the Q&A Roadmap Paper (Burger et al., 2002):

1) question taxonomies (and formal models),
2) question processing (syntax, semantics, parsing, understanding),
3) real time question answering (efficient processing),
4) interactive Q&A (dialogue structure),
5) user profiling for Q&A.

2. Empirical background: reference corpus for system design and evaluation

The development of POLINT was preceded by empirical studies on question answering in Polish. This preparatory work consisted in collection of a small but highly annotated corpus of information-acquisition-oriented question-answering dialogues. This corpus contains of 582 question-answer pairs collected during 30 sessions with human subjects. The questions were collected at sessions involving two participants: the information seeker and the information provider. The information seeker was supposed to formulate written questions to the information provider about the content of a picture (with regard to an intentionally banal subject: a scene with St. Claus, children, gifts, etc.). The information seekers were given a partial knowledge of the scene: the same picture with several blank areas. This very special setting and a particular mode of communication amounted with a number of observations, which, despite obvious limitations, are of interest especially at the early stage of QA system design. Examples of the observed syntactic phenomena of general interest are:

- short questions (average between 6 and 7 words in a question and between 2 and 3 words in a nominal group),
- rare ellipsis of whole constituents,
- low complexity of questions (small number of polypredicative questions: 35/582),
- rare use of relative clauses in questions,
- practical absence of questions with negated predicate,
- ...

Besides these purely syntactic observations, the corpus permitted preliminary studies on various discourse related phenomena such as: anaphorical links between answers and questions, long distance anaphora in dialogues, focus structure, dialogue structure and internal linking devices (anaphora, ellipsis, common-pattern-links, linking words).

Another practically useful result (used when designing POLINT) was the typology of observed syntactic structures (of course very much biased by the experiment setting, domain, mode etc.). The corpus attests mainly questions which require relatively little inferences. Most of them belong to the following categories (according to Arthur Graesser's taxonomy, cf. Burger et. al., 2002):

- verification,
- disjunctive,
- concept completion,
- feature specification,

[1] An early version of the system was tested as a front-end to the EXPÆRT system to store information retrieved from text documents about arts (Martinek&Vetulani, 1991).

- quantification,
- request/directive.

Within the typology of questions proposed in (Vetulani, 1989) these are mostly *basic questions,* in opposition to *non-basic questions* (cf. *compound questions* discussed in Belnap & Steel (1976)) rare in the kind of question-answering discourse oriented to the acquisition of factual, fine granulated information.

The St. Claus Corpus is supplied with rich annotations (only for questions). What follows is an annotated question-answer pair from this corpus.

Question: Co trzyma Mikolaj w prawej rece? /What is St. Claus holding in his right hand?
(1) $X_{subst,a}$;$V_{f,p(3)}$;N_n; $_{<w>}N_l$
(2) (?)[Arg$_1$: Mikolaj; Predicate: trzyma; Arg$_2$: ?; Arg$_3$: w prawej rece]
(3) [Arg$_1$: $_3$(N_n); Predicate: $_2$($V_{f,p(3)}$); Arg$_2$: $_1$($X_{subst,a}$);Arg$_3$: $_4$($_{<w>}N_1$)]
(4) Predicate=TRZYMAC-CZYMS(Arg$_1$,Arg$_2$,Arg$_3$)
Answer: Nic/Nothing

The structure (1) shows the surface linear ordering of different parameterised categories (X - interrogative phrase, N_g - noun phrase in genitive etc.); it is *called formal linear model* of the sentence. The lines (2) and (4) form the so called *predicate-argument structure* of the sentence (the line (4) describes the type of semantic requirements of the predicate). Somehow more abstract representation of the sentence is formed by (3) together with (4) (abstraction is made of surface forms, but relative position of the surface string (beginning of) is noted using the left-low numerical index, cf. for example the value 3 in $_3(N_n)$). Theoretical models for corpus annotation were introduced as an application of the unification-oriented concept of question-answer relationship (Vetulani, 1989) being inferred from the classical works by Ajdukiewicz (1965), Belnap and Steel (1976) and others.

What has appeared to be particularly useful are formal models (1) because they may be used as a skeleton of a formal grammar. Although the initial corpus is relatively small (due to time consuming and complex hand annotation procedure) it may be extended in a coherent way at any moment because the documentation of corpus generating experiment is very detailed and the collection procedure is simple (cf. Appendix 2). The corpus (now called St. Claus Corpus) and its methodology has been thoroughly described in paper publications (Vetulani,1989, 1990) and has been recently included as a basic resource in the data part of the Polish national project aiming to create NL evaluation tools for Polish (as announced at LREC1998 by Bien (1998)). Now, the St. Claus Corpus[2] is being prepared for free distribution for non-commercial purposes and will soon be available through the Internet. (This is a good reason for its presentation at the present QA Workshop.)

An important part of the above mentioned MUZARP project is based on empirical studies as well. In this project (now under development) the human user will be allowed to ask questions to virtual individuals represented in the "virtual interactive showcase". Questions will be about the "virtual showcase" world. In order to define the profile of a hypothetical user we have begun corpus collection where the (potential) human users were asked to imagine questions they *would like* to ask to the virtual scene participants *if* they were apt to do so. (The scene represents ancient country people at work.). The corpus collection is in progress and no systematic processing has started yet. It is already clear, however, that the MUZARP corpus will be substantially different from the StClaus Corpus (which is not surprising at all).

3. Generic grammatical resources of POLINT

The strength of any NL parsing (understanding, processing, etc.) system is measured by the power of its grammar and dictionary. These two modules contain the essential part of linguistic information about the language being processed but the respective role of each of them varies from case to case. In the POLINT system grammatical information is spread between rules and dictionary items, forming a lexicon-grammar. This solution will enable application of linguistically motivated heuristics to limit (at linear cost) the search during the rules-driven parsing.

3.1. Grammar rules

The POLINT grammar is composed of DCG-like rules. (It is implemented in PROLOG, but the parsing technique is much more sophisticated then the standard parsing algorithm inherent to PROLOG). That means that they have context free shape and allow arbitrary terms, including variables, as parameters. As POLINT was conceived as NL understanding system, the main goal of syntactic rules is to result in sentence segmentation useful for further (or parallel) semantic evaluation. The chosen theoretical model is the predicate-argument model, as described in (Vetulani, 1989). As a linguistic grammatical background we use the traditional phrasal approach, with some simplifications when compared to the traditional syntactic categorisations. For example we have removed some classical, very common but for us superfluous categories, as e.g. *subject phrase* or *direct/ indirect complement.* Instead, in both cases, we are using the category *noun phrase* to denote the sentence phrase which function will be specified by values of morpho-syntactic parameters (as, e.g., case: genitive). In principle, we have assumed that a sentence is composed of one or more noun phrases (arguments) and just one verbal phrase (predicate) in an order which is highly free in Polish. In practice, because of over-generation of rules, the initial (generic DCG-like) grammar has been transformed into more effective one, based on a "new" category of *sentence_segment* (*sentence_segment* is composed *of noun_phrase/verbal_phrase* + *sentence_segment*). This solution involving recursion will permit to control

[2] Its substantial enlargement is being planned for the nearest future.

effectively parsing by involving special control parameters (cf. Vetulani 1997) which function as heuristics calculated at the pre-analysis stage. (A "normal" grammar, i.e. grammar not involving rules engaging the category *sentence_segment* may easily be obtained from the POLINT grammar.)

At present, the POLINT system is based on ca 150 grammar rules encoded in PROLOG (ca. 60KB of the source ASCII code). These rules may be grouped as follows:

? sentence level rules: ca 35

? argument level rules: ca 45

? predicate level rules: ca 20

? other and auxiliary rules: rest

What follows is an example of a relatively simple rule encoded in PROLOG. These rules recognise the kernel of the verb group, based on a non-transitive finite verb, possibly reflexive or/and negated and optionally complemented by an adverb.

```
gv0(A,gv0_1(M),[[Ro,Li,Os,Cz],[R,L,mian,T],Rel],
[czas_0,0],[Tz,Neg,[Wcz,[Ro,Li,Os,Cz]]],
[[W,N]|X0],X4) :-
 neg_pred(A,Neg,[[W,N]|X0],X1),
 slo9(0,czas_0,[[Ro,Li,Os,Cz],Rel0,[R,L,mian,T]],
 X1,X2),
 eqw(X1,[[Wcz,_]|_]),eqw(Ro,R),eqw(Li,L),
 pron_refl_1(A,Rel0,X2,X3),
 eqw(X3,[[_,K]|_]),
 case([adv_poss(1,K) ->
gr_adv(A,1,M,S,X3,X4),
adv_poss(2,K) ->
gr_adv(A,2,M,S,X3,X4)|
gr_adv(A,3,M,S,X3,X4)]),
  sem(A,[gv_0(Neg),Tz,Rel0,S,Rel]).
```

3.2. Dictionary

The POLINT grammar requires a dictionary of the kind of lexicon-grammar, i.e. a lexicon where predicative words are supplied with syntactic information. At the pre-analysis stage the sentence is being scanned word by word for all predicative words, the syntactic requirements are read out from the dictionary and compared to properties of surrounding words. This observation usually permits formulation of a plausible hypothesis about syntax of the considered sentence in form of an expected configuration of sentence arguments. Such configurations are used as input parameter to the parsing module in order to make parsing more deterministic. This method proved particularly efficient while analysing sentences of medium size and medium complexity. The POLINT grammar has been tested with a dictionary containing ca 3000 dictionary entries (one word form per entry). Now, work is in progress to generate automatically (or semi-automatically) the system's dictionaries. The following resources are being tested as possible support of automatic dictionary generation: the morphological analyser LEM by Vetulani and Obrebski, cf. (Hajnicz, Kupsc, 2001), the resources of POLEX, GRAMLEX and CEGLEX projects (reported at LREC 2000 (Vetulani, 2000a)).

At present, the grammar is being translated into our new formalism FROG based on DCG-like rules well suited for free order languages with frequent discontinuity phenomena (Vetulani, 2002). This is a preparative step for further enhancement of the system's grammatical coverage to fully include discontinuous constructions. In this form, free distribution for non-commercial purposes is planned.

4. Coverage

In order to characterise the grammatical (and functional) coverage of the system we have listed a number of problems covered by POLINT:

- confirmation questions ("Czy" + affirmative sentence?)
- questions about arguments ("Kto/Who...?", "Co/What...?", "Z kim/With whom...?", etc.)
- questions concerning place ("Gdzie znajduje sie...?" / "Where is...?")
- questions concerning time ("Kiedy...?" / "When...?")
- questions concerning existence (Kogo nie ma...?/Who is absent...?)
- questions concerning name ("Jak nazywa sie...?" / "What is the name of...?")
- concerning type, position in a hierarchy ("Kim jest...?" / "Who is...?")
- about complement ("Czyim bratem jest...?" / "Whose brother is ....?")

At the predicate-argument level the word order is arbitrary (the system ignores differences in the degree of pragmatic markedness depending on the order of arguments).

The system recognises correctly a large class of nominal constructions. The following are the main types of noun phrases the system understands: proper names, complex proper names, complex noun group, common names, pronouns, genitive (possessive-like) constructions, complement nominal constructions. The nominal groups may be also completed with relative clauses (with possible iteration or embedding), adjectives etc. The predicates may take one, two or tree referential or locative arguments (Vetulani 1989). The predicate group may be, e.g.: personal forms of verbs, constructions with the auxiliary "byc" ("to be"), construction with noun in the instrumental case or with an adjectival group, constructions based on a supporting verb, construction with negation. The POLINT grammar was tested against the StClaus Corpus with satisfactory result (80% syntactic coverage for non-polypredicative, non-elliptical questions).

5. Efficiency

Contrarily to the most of NL systems written in PROLOG, the parser of POLINT is a real time system. This effect is difficult to reach for languages with flexible word order, like Polish, because of intensive and costly backtracking if grammar rules observe traditional grammar encoding procedures (various rules for various surface orderings, each rule reflecting the surface ordering of words). The main idea applied in the POLINT system to improve efficiency was to precede application of the grammar rules by a pre-analysis module. The pre-analysis was based on the concept of "lexical witness" for syntactic phenomena and on systematic usage made of lexicon grammar dictionary. A lexical witness (as, e.g., relative pronoun for relative clause) may help to select appropriate grammar rule in a deterministic way. Exploration of syntactic or/and semantic requirements may help to limit the grammatical search space up to making the search deterministic in many cases. This additional information may be obtained from the dictionary when reading-in the sentence (in linear time).

6. Acknowledgements

The Author wishes to thank Alain Colmerauer for having hosted him as a young research fellow at GIA in Marseille, many years ago. This exercise permitted him to discover the charm of the Computer Question Answering Challenge.

7. References

Ajdukiewicz, K., 1965. *Logika Pragmatyczna (Pragmatic Logic)*. Warszawa: PWN.

Belnap, N.D.Jr., Steel, T.B.Jr., 1984. *The Logic of Questions and Answers*. New Haven: Yale Univ. Press.

Bien, J., 1998. Evaluating Analysers of Polish. In A. Rubio et al. (eds.), *First International Conference on Language Resources and Evaluation, Granada, Spain, 28.05.-30.05.1998, (Proceedings)*. Paris: ELRA. 951-955.

Burger, J. et al., 2002. Issues, Tasks and Program structures to Roadmap Research in Question & Answering (Q&A). (www-nlpir.nist.gov/projects'duc/papers/qa.Roadmap-paper_v2.doc)

Hajnicz, E. and A. Kupsc, 2001. A survey of morphological analysers for the Polish language (in Polish). Prace IPI PAN (ICS PAS REPORTS), No 937. Warszawa: IPI PAN.

Martinek, J. and Z. Vetulani, 1991. An Expert system for Art History Data and Documents. In J. Banczerowski (ed.). *The Application of Microcomputers in Humanities*, *Adam Mickiewicz* University Press, Poznan, 63-74.

Vetulani, Z., 1989. *Linguistic problems in the theory of man-machine communication in natural language. A study of consultative question answering dialogues. Empirical approach.* Bochum: Brockmeyer.

Vetulani, Z., 1990. *Corpus of consultative dialogues. Experimentally collected source data for AI applications.* Wyd. Nauk. UAM (Adam Mickiewicz University Press), Poznan.

Vetulani, Z., 1997. A system for Computer Understanding of Texts. In R. Murawski and J.Pogonowski (eds), *Euphony and Logos* (Poznan Studies in the Philosophy of the Sciences and the Humanities, vol. 57). Amsterdam-Atlanta: Rodopi. 387-416.

Vetulani, Z., 2000a. Electronic Language Resources for POLISH: POLEX, CEGLEX and GRAMLEX. In: M. Gavrilidou et al. (eds.), *Second International Conference on Language Resources and Evaluation*, Athens, Greece, 30.05.-2.06.2000, (Proceedings), ELRA, 367-374.

Vetulani, Z., 2000b. Understanding Human Language by Computers: Projects in Artificial Intelligence and Language Technology. In Yosiho Hamamatsu et al. (eds). Formal Methods and Intelligent Techniques in Control, Decision Making, Multimedia and Robotics. Proceedings of the 2nd International Conference, Polish-Japanese Institute of Information Technology, Warsaw, October 2000, 218-229.

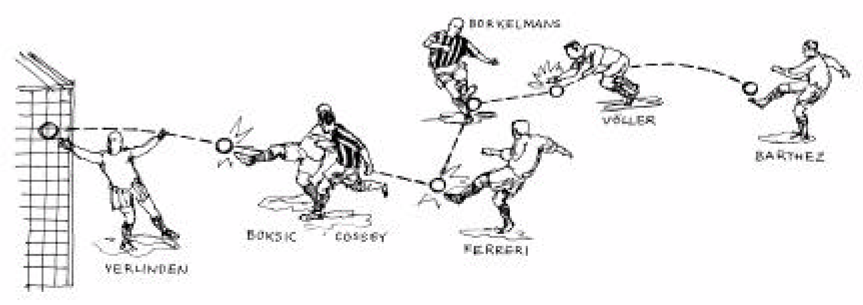
Vetulani, Z., 2002. A reinterpretation of the Definite Clause Grammar: Free Order DCG (FROG) (typescript, to appear).

Vetulani, Z. and Marciniak, J., 2000. Corpus Based Methodology in the Study and Design of Systems with Emulated Linguistic Competence. In Dimitris N. Christodoulakis (ed.). *Natural Language Processing - NLP 2000, Lecture Notes in Artificial Intelligence, no 1835*. Springer. 346-357.

Appendices

Appendix 1. Example of soccer game scene askable in POLINT

Information represented picturally in Figure 1 is encoded in the form of PROLOG predicates and accessible through POLINT.



User: - Jak nazywa sie pilkarz, który strzelil bramke? /What is the name of the player who scored?/
System: - Boksic.

Figure 1. Episode represented in the data-base[3] and a question-answer exchange.

Appendix 2. Experiment design

We are presenting here a detailed description of the St. Claus experiment setting.

1. Participants: A and B.

2. The scene (S) is represented by a complete picture (P) and an incomplete picture (P') (see below).

3. The participant A has the picture P'.

4. The participant B has the picture P.

5. Goal for A: to complete his knowledge about S.

6. Scenario for A: to ask questions to B (in writing).

7. Scenario for B: to answer the question (in writing).

8. Both A and B control (see) all previous questions and answers.

9. Restrictions:

- a single answer follows a single question (but no restrictions on the form of questions and answers),
- A and B are not permitted any form of communication (oral, gesture),
- dialogues are limited to 20 question-answer cycles (which corresponds to 30 min.-1h. sessions),
- a human supervisor is present during the session.

10. It is implicitly suggested to the participants that the experiment is a part of psychological research.

11. Instructions are read by the supervisor at the beginning of the dialogue session and no other explanations are allowed; the instructions are, however, available to participants (in writing) during the session.

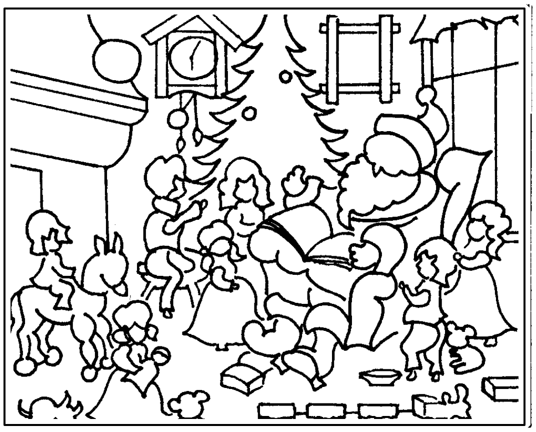


Figure 2. Complete picture (P)

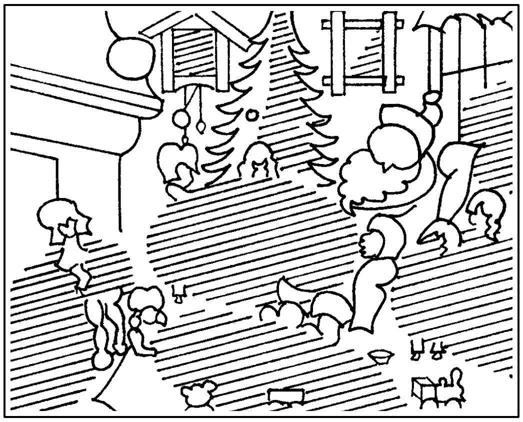


Figure 3. Incomplete picture (P')