RC22936 (W0310-101) October 16, 2003
Computer Science

IBM Research Report

A Multi-Agent Approach to Using Redundancy and Reinforcement in Question Answering

John Prager, Jennifer Chu-Carroll, Krzysztof Czuba
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Chapter 19

A Multi-Agent Approach to using Redundancy and Reinforcement in Question Answering

John Prager, Jennifer Chu-Carroll and Krzysztof Czuba
IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598, U.S.A.
{jprager,jencc,kczuba}@us.ibm.com

Abstract

We explore how to improve the performance of our Question Answering system by using redundancy and reinforcement. We have deployed in our system a variety of agents, each of which is tuned to a different class of question types, but with considerable overlap. One source of redundancy and reinforcement is from the multiple agents: many questions give rise to two or more sets of candidate answers, which can be merged to provide better performance than any single agent. We note relative improvement of up to 16.3% using the Mean Reciprocal Rank metric, and 11.9% using the Confidence Weight Score metric. We also investigate new approaches we call QA-by-Dossier and QA-by-Dossier-with-Constraints, in which additional questions are asked to generate constraints on the answers to the original question. This can reduce the confidence of many wrong answers and reinforce many good ones, resulting in one experiment in an increase in precision from .43 to .95.

1. Introduction

As the field of Question-Answering evolves, we are learning that the one-size-fits-all approach of earlier systems is not the best way to tackle open-domain QA with a broad set of question types. We believe that just as a human may answer a question to which he does not know the answer by a text search, by calculation, by look up in a dictionary, thesaurus or other reference work, or some combination thereof, then so too can a computer adopt different strategies. Moreover, not only will the availability of different strategies allow for choosing the most appropriate one for the question in-hand, but for many question types a selection of strategies can be tried together, with the answer(s) being selected from the pooled results.

This chapter explores our use of answer redundancy and reinforcement in question answering along two dimensions. First, we discuss our multi-agent approach to QA which utilizes multiple sources and strategies for answering questions. The benefits of redundancy within a single answering agent have been reported in [Prager et al., 2004]. We have previously reported the use of structured knowledge sources on the overall performance of our QA system [Prager et al., 2001; Chu-Carroll et al., 2003b]. [Clarke et al., 2001] demonstrated the utility of redundancy by exploiting multiple answer sources (using the Web to reinforce answers found in the TREC corpus); however; they do not investigate utilizing multiple strategies. In [Chu-Carroll et al., 2003a] we reported benefits from consulting multiple corpora, as well as results of glass-box combination of two different answering agents; here we extend our earlier results to show the effects of black-box combination of five answering agents. Second, we introduce a novel approach to QA in the form of a constraint-based agent, which, while (currently) employing only a single strategy for answering questions, takes advantage of answer reinforcement by asking additional constraining questions. The answers to these constraining questions can then be used to support or refute answers proposed for the original question.

2. A Multi-Agent QA Architecture

To enable a multi-agent approach to QA, we developed a modular and extensible QA architecture as shown in Figure 1. In this architecture, a question is first processed by the question analysis module, which is based on a deep parser and a feature structure unification based rule matcher. The set of manually constructed rules allows the question analysis module to detect the required answer semantic type(s), the question type (such as definition, list, etc), the normalized keywords from the question, and a simple semantic form (single predicate with arguments) to represent the question meaning when one can be built. This information is represented in a QFrame, which is a set of question features used to activate one or more answering agents. Based on the QFrame, each agent generates its own requests to a variety of knowledge sources. This may include performing search against a text corpus and extracting answers from the resulting passages, or querying a structured knowledge source, such as

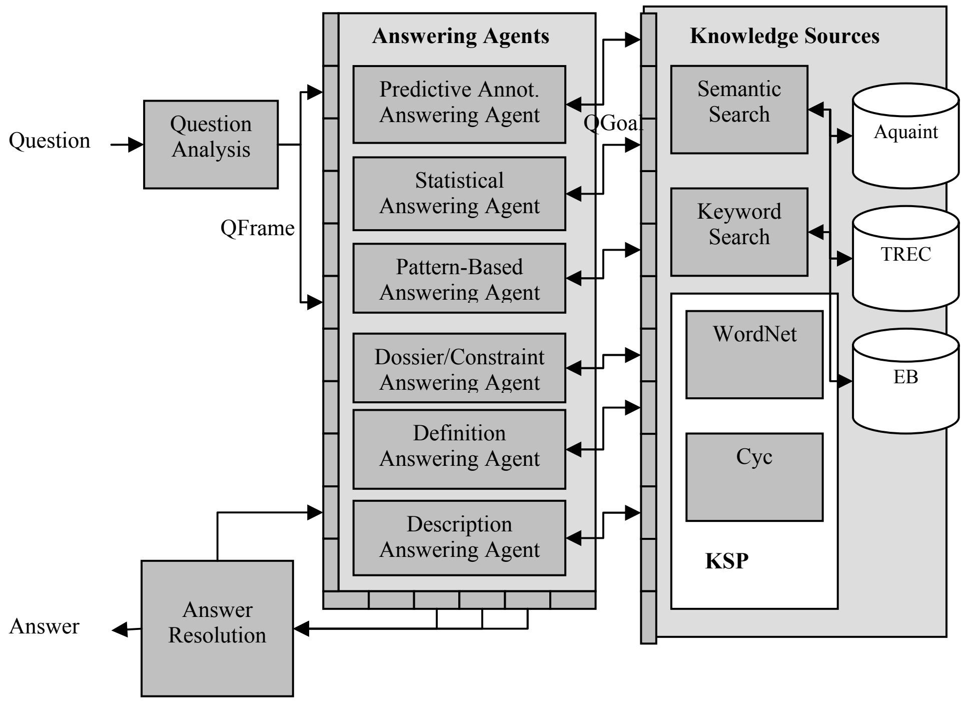
Question
Question Analysis
QFrame
Answering Agents
Predictive Annot. Answering Agent
Statistical Answering Agent
Pattern-Based Answering Agent
Dossier/Constraint Answering Agent
Definition Answering Agent
Description Answering Agent
QGoal
Knowledge Sources
Semantic Search
Keyword Search
WordNet
Cyc
KSP
Aquaint
TREC
EB
Answer
Answer Resolution

Figure 1 PIQUANT's Architecture

WordNet [Miller, 1995] or Cyc [Lenat, 1995]. The (intermediate) results from the individual agents are then passed on to the Answer Resolution component, which combines and reconciles the set of results, and either produces the system's final answers or feeds the intermediate results back to the answering agents for further processing.

Our system currently employs several answering agents, as shown in Figure 1. Some agents are general purpose, such as the Predictive Annotation Agent [Prager et al., 2000] and the Statistical Agent [Ittycheriah et al., 2001], while others are designed to handle specific question types. Note that given a question, one or more answering agents will be activated, allowing the system to take into account the redundancy factor [Clarke et al., 2001] for answer reinforcement.

In the rest of this chapter, we describe our various agents, and focus on two kinds of evaluation experiments, one of which is an ablation study on the impact of adopting multiple strategies in question answering, and the other a pair of evaluations of the performance of our Constraint Agent.

3. Answering Agents

3.1 Predictive Annotation Agent

The Predictive Annotation Agent ([Prager et al., 2000, 2004]), takes as input the QFrame produced by the question analysis module. It issues a query to the search engine based on the answer type and the question keywords, and receives answer passages which are then processed by an answer selection module to locate and rank candidate answers. The key feature of Predictive Annotation is that before the text corpora are indexed, they are processed by our semantic annotator, which recognizes approximately 100 semantic types. The identified semantic labels are then indexed along with the text. The search engine query contains the answer type, thus guaranteeing that each returned passage contains at least one candidate answer of the right type. Answer selection evaluates candidate answers according to a number of text-based features. The output of answer selection is a ranked list of answer candidates with confidences.

3.2 Statistical Agent

This agent is based on a statistical approach to QA ([Ittycheriah et al, 2001]). Because of the specific approach, it does not use the information in the QFrame as the answer type assignment is statistically modeled. The agent models the "correctness" of an answer to a question using a maximum entropy formalism [Berger et al., 1996]. At runtime, the question is first analyzed by the answer type model, which selects one of 31 possible answer types for use by the answer selection model. Simultaneously, the question is expanded using local context analysis [Xu and Croft, 1996] with an encyclopedia, and the top 1000 documents are retrieved by the search engine. From these documents, the top 100 passages are chosen based on lexical and syntactic features. From the passages, candidate answers are extracted and ranked using the answer selection model. The top candidate answers are returned, each with an associated confidence score.

3.3 Definition Agent

The Definition Agent ([Prager et al, 2001]) is used to answer definitional questions, such as "*What is X?*" and "*Who is X?*", whose answer type typically cannot be determined from the question itself. The agent employs an ontology provided by an external, structured resource such as WordNet to look up candidate answers by examining the superclasses (hypernyms) of the term to be defined. These candidates are evaluated using a collocation method based on the reference corpus, thus allowing the system to determine the most natural granularity level for the answer. The reference corpus is then consulted to select passages that define the term in question as the preferred answer.

3.4 Description Agent

The Description Agent ([Prager et al, 2004]) targets a subset of questions handled by the Definition Agent. However, instead of using an external knowledge source for answers as in the Definition Agent, the Description Agent attempts to identify answers that satisfy predetermined syntactic constructs used commonly for definitions, including appositions, pre- and post-modifiers, and relative clauses. This approach is at a high level similar to that of Blair-Goldensohn et al., in Chapter 13. The Description Agent first retrieves passages containing the term to be defined from the reference corpus. Descriptive constructs related to the question term are then dynamically annotated and the answer selection component evaluates and ranks all such identified descriptions.

3.5 Pattern Agent

The Pattern Agent [Czuba et al., forthcoming] addresses question types unsuitable for a general purpose strategy because of the potentially low precision or an unknown answer type. It extends approaches such as [Soubbotin et al., 2002] and [Hermjakob et al., 2002] by basing patterns on the deeper representation of parse trees. It leverages a broad-coverage parser based on the English Slot Grammar [McCord, 1990] formalism to create parse trees for answer passages on which pattern matching is performed. Its patterns are parse tree fragments representing contexts in which an answer is likely to be found. The agent utilizes patterns from two sources. The first is a repository of question type specific patterns, such as a person's birthday, birthplace, cause of death, etc., which were manually created.[1] These question semantic types are usually parameterized with an argument, such as *birthday(Robert Frost)* for "*When was Robert Frost born?*". The repository contains a set of patterns that match the parse trees for the typical contexts in which someone's birthday can occur in text and these patterns are retrieved based on the predicate (*birthday*). They are then instantiated with the argument (*Robert Frost*) and passed on to the matcher.

The second source is dynamic patterns generated from the parse tree of the question. They are useful for questions for which the predicate-argument structure is clear enough to allow for precise matching. For instance, for "*What does Knight Ridder publish?*", since the question is a simple transitive verb construct, a pattern that matches clauses whose subject is *Knight Ridder*, and whose main verb is *publish* is generated. The content of the obligatory object slot in the parse tree is considered to be an answer candidate.

3.6 Dossier Agent

The Dossier Agent addresses definitional questions using a new approach called QA-by-Dossier by issuing *auxiliary* questions intended to build a profile/dossier on the question subject. For people, it asks for their birthdate, birthplace, deathdate, profession, college attended, and so on. For organizations, it asks for their headquarters, CEO, and their line of business. For things, it provides a definition by invoking the Definition Agent described in Section 3.3.

Currently these auxiliary questions are manually constructed. We are developing an ontology of interesting properties, which not only will provide a source of these questions, but also of follow-up questions determined by first-round answers. For example, depending on a person's profession, it will ask what they wrote, performed in, explored, invented, etc. We currently ask these questions by default for all people, on the assumption that inappropriate questions will generate low confidence answers which will be rejected.

[1] In future work, we plan to investigate automatic acquisition of these patterns.

3.7 Constraint Agent

The Constraint Agent employs a novel approach to improving QA precision by building upon QA-by-Dossier. QA-by-Dossier-with-Constraints (QDC) exploits answer reinforcement and is based on the idea that answers to related questions often constrain one another and thus allow a system to more confidently select the correct answer to each. We illustrate this idea by way of the example, "*When did Leonardo paint the Mona Lisa?*" Columns 2 and 3 in Table 1 shows the Predictive Annotation Agent's top answers to this question, with associated scores in the range 0-1.

	Score	**Painting Date**		Score	**Born**		Score	**Died**
1	.64	2000		.66	1452		.99	1519
2	.43	1988		.12	1519		.98	1989
3	.34	1911		.04	1920		.96	1452
4	.31	1503		.04	1987		.60	1988
5	.31	1490		.04	1501		.60	1990

Table 1. Answers for "*When did Leonardo paint the Mona Lisa?", "When was Leonardo born?"* and *"When did Leonardo die?"*

The correct answer is "1503", which is in 4th place with a low score. Using QDC, we ask two related questions "*When was Leonardo born?*" and "*When did Leonardo die?*" The answers to these auxiliary questions are shown in columns 4-7 in Table 1.

Given common knowledge about a person's life cycle, the best proposed dates consistent with one another are that Leonardo was born in 1452, died in 1519, and painted the Mona Lisa in 1503. This example illustrates how the use of auxiliary questions helps constrain answers to the original question, and promotes correct answers with initial low confidence scores. Note that by using QDC, not only have we correctly answered the original question, but we have built a simple "dossier" on Leonardo, with three important dates in his life-cycle.

3.7.1 Applying QA-by-Dossier

To automatically apply QA-by-Dossier (with or without constraints) during question answering, several problems need to be addressed. First, we must develop criteria for invocation for the process. Second, we must identify question types suitable for this approach, and, for each question type, develop auxiliary questions and constraints among the answers. Third, for each question type, we must determine how the results of applying constraints should be utilized.

The first problem is addressed by a cursory examination of the scores associated with answers to the original question. The process is invoked when either the top score or the difference between the top two scores is below a threshold. The second issue is currently addressed by manually constructing auxiliary questions and general, reusable constraints among answers for select question. For instance, biographical data is constrained by life-cycle parameters, geographical data by 2-d spatial geometry, and kinship data by reciprocal relationships. As mentioned earlier, we are currently investigating developing an ontology from which such questions and constraints can be derived. The third issue concerns the (general) problem of when more than one combination of answers satisfies our constraints. In that case we combine the first-round scores of the individual answers to provide a score for the dossier. There are several ways to do this, and we found experimentally that it does not appear critical exactly how this is done. In the examples in the evaluation we mention one particular combination algorithm.

4. Answer Resolution

Our earlier experiments showed that although the greatest improvement in performance was obtained by combining agents at multiple levels (after question analysis, passage retrieval, and answer selection), near optimal results can be achieved by treating each agent as a black box and performing answer resolution only on the ranked candidate answers. Thus, in this work, we investigate combining candidate answers from multiple answering agents and examine how answer redundancy and reinforcement results in overall performance improvement.

PIQUANT's current answer resolution component takes as input the top *n* (=5) answers for a question from each agent, with their associated confidence scores, and selects the overall best answer(s). It first determines semantic equivalence of answers using named entity normalization to equate proper names (e.g., “Clinton”, “Bill Clinton”, and “President Clinton”) [Byrd and Ravin, 1999]. Semantically equivalent answers are then combined using a simple voting procedure where each instance of an answer votes with a weight equal to its confidence score and the weights of equivalent answers are summed. The answers are sorted based on the combined score, and the top answer(s) selected as the system’s final answer(s). In this fashion, answer resolution exploits answer redundancy and reinforcement by allowing equivalent answers proposed by different agents to provide support for one another. In the next section, we describe experiments conducted to measure performance improvement using this multi-agent question answering paradigm.

5. Impact of Answering Redundancy and Reinforcement

We describe two sets of experiments intended to evaluate the impact of answer redundancy and reinforcement on overall system performance. One is an ablation study for evaluating the effectiveness of employing multiple agents for QA, and the other is designed to assess the contributions of the Dossier and Constraint Agents.

5.1 Experimental Results – Ablation study

For our ablation study, we selected a test set of 500 questions from the TREC 10 QA track [Voorhees, 2002]. We eliminated questions for which there is no known answer in the reference corpus, which resulted in 440 questions in the final test set.

To evaluate the impact of individual answering agents on a multi-agent QA system, we combined each specialized agent with the Predictive Annotation Agent as well as evaluated the combined performance of all available agents. We then compared the combined performance with the best performing agent. For each run, the Mean Reciprocal Rank (MRR) [Voorhees, 2000] of the top 5 answers was computed. Furthermore, the top answer for each question was selected for each question and all answers sorted based on the overall confidence scores and evaluated using the Confidence Weighted Score (CWS) metric [Voorhees, 2003].

Table 2 summarizes the results of our ablation study. For each run, we show 1) the number of questions for which answers were returned, 2) the number of questions for which an answer was returned by all agents involved, 3) the

MRR/CWS for the questions attempted, and 4) the MRR/CWS computed over the whole test set. The last row in Table 2 shows the relative improvement of our full multi-agent system over the Statistical Agent, the best "overall" agent.

Table 2 shows that the Predictive Annotation Agent answers a large subset of questions, while the Statistical Agent answers all questions. The Description and Definition Agents, both address primarily definition questions, are triggered in rough one third of the questions. The Pattern Agent achieves the highest precision of all our agents; however, it was applicable in only a small number of questions.

	# Q's	#Q's all agents	Subclass MRR	Overall MRR	Subclass CWS	Overall CWS
Predictive Annotation (PA)	304	304	0.445	0.307	0.600	0.510
Description (DS)	94	94	0.254	0.054	0.230	0.119
Definition (DF)	133	133	0.402	0.121	0.464	0.278
Pattern (PT)	16	16	0.812	0.029	0.779	0.118
Statistical (ST)	440	440	0.424	0.423	0.540	0.539
PA+DS	396	2	0.399	0.358	0.530	0.509
PA+DF	400	37	0.435	0.395	0.556	0.538
PA+PT	304	16	0.449	0.310	0.624	0.527
PA+DS+DF	421	2	0.430	0.410	0.542	0.533
PA+DS+DF+PT	421	0	0.434	0.414	0.572	0.562
All	440	0	0.493	0.492	0.604	0.603
Improvement over ST				16.3%		11.9%

Table 2. Ablation Study Results

Our results show that in all but one case, multi-agent systems outperformed their components. The Predictive Annotation and Description Agents answered nearly disjoint sets of questions, and their combined MRR (PA+DS) was nearly the sum of the component MRRs. The Definition Agent also provided fairly substantial gain over the Predictive Annotation agent alone (PA+DF) using both metrics. Although the high-precision Pattern Agent, which answered a subset of questions addressed by the Predictive Annotation Agent, yielded only a small improvement in MRR (PA+PT vs. PA, and PA+DS+DF+PT vs. PA+DS+DF), the corresponding gain in CWS is much greater. This indicates that while the Pattern Agent did not add many new correct answers, it enabled the system to be more confident about previously found correct answers, resulting in a higher CWS. The Predictive

Annotation Agent and all specialized agents (PA+DS+DF+PT) again outperformed the pair-wise combination. Finally, the addition of a second general purpose agent (ST) yielded the overall best performance, achieving a 16.3% relative improvement in MRR and 11.9% in CWS over the performance of the Statistical Agent.

5.2 Experimental Results – Constraint Agent

We performed two evaluations of the Constraint Agent. The first involved 40 questions about celebrity marriages. The original question was "*Who is X married to?*", and for the auxiliary question we asked the same of the candidate answers. The constraint was simply that the second-round answer be the same as the person in the original question. Thus, for each person X, we asked "Who is X married to" and collected the top 5 answers $\{X_i\}$. For baseline performance we examined the top answer X_1. We then generated 5 auxiliary questions and collected 5 answers for each, $\{X_{ij}\}$. We then assigned each X_i the average of the *supported score* of the sum of the scores of the X_{ij} that matched X, and the original score of X_i. The X_i with the greatest resulting score was output as the QDC answer.

We compare the baseline v. QDC answers using three metrics. The first is percent-correct. The second is the Confidence-Weighted Score (CWS). Unfortunately, at relatively high accuracy and ranking ability, CWS fails to make necessary distinctions ([Chu-Carroll et al. 2003c]). Thus we introduce a third metric called the Risk-Based Score, where an answer score *p* (in the range 0-1) is treated as a bet: if the answer is correct, the overall score is increased by *p*; otherwise, it is decreased by *p*. The resulting totals are normalized into the range 0-1. The Risk-Based Score is intuitively appealing, has a basis in both gambling and multiple-choice test scoring, and overcomes many deficiencies of the CWS.

Table 3 shows that by using QDC, a few more questions were answered correctly, but the Risk-Based Score shows a dramatic improvement. This is because a principal contribution of QDC is to provide more accurate confidence scores. As with the Pattern Agent performance discussed in Section 5.1, we expect that a major contribution of QDC when used with other agents will be to improve the system's overall confidence of its answers, and to improve the other measures by promoting the correct answer to the top.

	No. Correct	CWS	Risk-Based Score
Baseline	32 (80%)	0.95	0.698
QDC	36 (90%)	0.98	0.900

Table 3. Results of Performance Evaluation of QA-by-Dossier-with-Constraints

The second evaluation was based on the people *definition* questions of TREC12 ("*Who was X?*"). We evaluate an auxiliary question "*What compositions did X have?*" The system was asked to return as many answers or *nuggets* as possible, based on developed rejection criteria. For each C_i returned by the auxiliary question, we asked the reciprocal constraining question "*Who had C_i?*", and looked at the top 5 answers. We boosted the original subject X's score by the confidence of the reciprocal answer that was equal to X, if it existed. Row 1 in Table 4 shows the evaluation of the plain Dossier Agent, which correctly found 60 nuggets out of 140, a precision of 0.43. Using QDC, 78 incorrect and 18 correct nuggets were thrown out, giving a new precision of 0.95 – a relative increase of 123%. We cannot calculate absolute recall figures because we don't have a list of all possible correct nuggets; if we consider the baseline recall to be 1.0 (because in this set-up, QDC cannot introduce new answers), the resulting recall is .70. This gives a change of balanced F-measure from .60 to .81.

	Total	No.	No.	Precisio	Recall (see	F
Baseline	140	60	80	0.43	1.0	.60
QDC	44	42	2	0.95	.70	.81

Table 4. Results of Performance Evaluation of QA-by-Dossier-with-Constraints

6. Related Work

A number of researchers have investigated federated systems for identifying answers to questions. For example, [Clarke et al., 2003] and [Lin et al., 2003] employ techniques for utilizing both unstructured text and structured databases for QA. However, their approaches differ from ours in that they enforce an order between the two strategies by locating answers in structured databases first and falling back to unstructured text when the former fails, while we explore multiple options in parallel and *combine* the results from all answering agents.

The multi-agent approach to QA most similar to ours is that by Burger *et al.* [2003]. They applied ensemble methods to combine the 67 runs submitted to the TREC 11 QA track, using an unweighted centroid method for selecting among the 67 proposed answers for each question. However, their combined system did not outperform the top scoring system. Furthermore, their experiments focused on exploiting a large number of general purpose answering agents, while we studied the impact of several specialized agents on overall system performance.

The precision-oriented approach of QA-by-Dossier-with-Constraints can be contrasted with the work of Weischedel et al. in Chapter 14, who use a general IR query and then IE techniques to find information nuggets, and that of Blair-Goldensohn et al. in Chapter 13, who search for instances of a set of general *definitional predicates*. These latter approaches stand to be more recall-oriented; a combination experiment would be a very interesting exercise.

7. Conclusions

We have described two approaches we are exploring simultaneously to improve the performance of our QA system. We use a variety of agents with different intrinsic coverage and precision to tackle more question types than any individual agent can handle well. Where the agents' coverage overlap, we use redundancy to adjust answers and their confidences. In addition, we developed a constraint-based agent, which uses answers to auxiliary questions to greatly increase the confidence of mutually reinforcing answers. Using multiple agents we have shown a 16.3% and 11.9% relative improvement in MRR and CWS scores, respectively. Our experiments show that while the major benefit of some agents is that they contribute more correct answers, others reinforce existing correct answers and thus improve confidence scores. The Constraint Agent is not yet fully integrated, but a separate evaluation showed very promising results, including a relative increase in precision of 123% .

8. Acknowledgments

We would like to thank Dave Ferrucci and Chris Welty for helpful discussions, and Abe Ittycheriah and Salim Roukos for making their system available for experimentation. This work was supported in part by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program under contract number MDA904-01-C-0988.

9. References

Berger A., V. Della Pietra, and S. Della Pietra, 1996, "A maximum entropy approach to natural language processing", *Computational Linguistics*, 22(1):39-71.

Burger, J. D., Ferro, L., Greiff, W., Henderson, J., Light, M., and Mardis, S., 2003, "MITRE's Qanda at TREC-11", *Proceedings of the 11th Text Retrieval Conference*.

Byrd, R. and Ravin,, Y,, 1999, "Idenifying and Extracting Relations in Text", *Proceedings of NLDB*.

Chu-Carroll, J., Czuba, K., Prager, J., and Ittycheriah, A., 2003, "In Question Answering, Two Heads Are Better Than One", *Proceedings of the Joint HLT/NAACL Conference*.

Chu-Carroll, J., Ferrucci, D., Prager, J., and Welty, C., 2003, "Hybridization in Question Answering Systems", *Working Notes of the AAAI Spring Symposium on New Directions in Question Answering*, pp. 116-121.

Chu-Carroll, J., Prager, J., Welty, C., Czuba, K., and Ferrucci, D., 2003, "A Multi-Strategy and Multi-Source Approach to Question Answering", *Proceedings of the 11th Text Retrieval Conference*.

Clarke, C.L.A., Cormack, G.V. and Lynam, T.R. "Exploiting Redundancy in Question Answering", *Proceedings of the 24th SIGIR Conference*, pp. 358-365, New Orleans, Sept. 2001.

Clarke, C. L. A., Cormack, G. V., Kemkes, G., Laszlo, M., Lynam, T. R., Terra, E. L., and Tilker, P. L., 2003, "Statistical Selection of Exact Answers", *Proceedings of the 11th Text Retrieval Conference*.

Czuba, K, Prager J., forthcoming, "Exploiting Grammatical Structure in Question Answering".

Hermjakob, U., Echihabi A., Marcu, D., 2002, "Natural Language Based Reformulation Resource and Web Exploitation for Question Answering", *TREC'02 Notebook Proceedings*, pp. 734—742.

Ittycheriah A., M. Franz, Wei-Jing Zhu, and A. Ratnaparki, 2001, “Question answering using maximum entropy components”. *Proceedings of NAACL’01*, pp. 33—39.

Lenat, D. B. 1995. "Cyc: A Large-Scale Investment in Knowledge Infrastructure." Communications of the ACM 38, no. 11.

Lin, J., Fernandes, A., Katz, B., Marton, G., and Tellex, S., 2003, “Extracting Answers from the Web Using Knowledge Annotation and Knowledge Mining Techniques”, *Proceedings of the 11th Text Retrieval Conference*.

McCord, M.C., 1990, “Slot grammar: A system for simpler construction of practical natural language grammars”, In R. Studer, editor, *International Symposium on Natural Language and Logic*, Springer Verlag, 1990.

Miller, G. “WordNet: A Lexical Database for English”, *Communications of the ACM* 38(11) pp. 39-41, 1995.

Prager, J.M., Brown, E.W., Coden, A. and Radev, D. "Question-Answering by Predictive Annotation”. In *Proceedings of SIGIR 2000*, pp. 184-191, Athens, Greece.

Prager, J.M., Radev, D.R. and Czuba, K. “Answering What-Is Questions by Virtual Annotation”. In *Proceedings of Human Language Technologies Conference*, San Diego CA, March 2001.

Prager, J.M., Chu-Carroll, J., Brown, E.W. and Czuba, K. "Question Answering by Predictive Annotation”, in “Advances in Open-Domain Question-Answering", Strzalkowski, T. and Harabagiu, S. Eds., to appear.

Soubbotin M. M. and S. M. Soubbotin (2002), “Use of Patterns for Detection of Likely Answer Strings: A Systematic Approach”, *TREC’02 Notebook Proceedings*, pp. 134—143.

Voorhees, E. M., 2000, “The TREC-8 Question Answering Track Report”, *Proceedings of the 8th Text Retrieval Conference*, pp. 77-82.

Voorhees, E. M., 2002, “Overview of the TREC 2001 Question Answering Track”, *Proceedings of the 10th Text Retrieval Conference*, pp. 42-51.

Voorhees, E. M., 2003, “Overview of the TREC 2002 Question Answering Track”, *Proceedings of the 11th Text Retrieval Conference*.

Xu, J. and W.B. Croft, 1996, "Query expansion using local and global document analysis", *Proceedings of the 19th SIGIR Conference*, pp. 4—11.