Proceedings of HLT-NAACL 2003
Main Papers , pp. 24-31
Edmonton, May-June 2003

In Question Answering, Two Heads Are Better Than One

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

Abstract

Motivated by the success of ensemble methods in machine learning and other areas of natural language processing, we developed a multi-strategy and multi-source approach to question answering which is based on combining the results from different answering agents searching for answers in multiple corpora. The answering agents adopt fundamentally different strategies, one utilizing primarily knowledge-based mechanisms and the other adopting statistical techniques. We present our multi-level answer resolution algorithm that combines results from the answering agents at the question, passage, and/or answer levels. Experiments evaluating the effectiveness of our answer resolution algorithm show a 35.0% relative improvement over our baseline system in the number of questions correctly answered, and a 32.8% improvement according to the average precision metric.

1 Introduction

Traditional question answering (QA) systems typically employ a pipeline approach, consisting roughly of question analysis, document/passage retrieval, and answer selection (see e.g., (Prager et al., 2000; Moldovan et al., 2000; Hovy et al., 2001; Clarke et al., 2001)). Although a typical QA system classifies questions based on expected answer types, it adopts the same strategy for locating potential answers from the same corpus regardless of the question classification. In our own earlier work, we developed a specialized mechanism called *Virtual Annotation* for handling definition questions (e.g., *"Who was Galileo?"* and *"What are antibiotics?"*) that consults, in addition to the standard reference corpus, a structured knowledge source (WordNet) for answering such questions (Prager et al., 2001). We have shown that better performance is achieved by applying Virtual Annotation and our general purpose QA strategy in parallel. In this paper, we investigate the impact of adopting such a multi-strategy and multi-source approach to QA in a more general fashion.

Our approach to question answering is additionally motivated by the success of ensemble methods in machine learning, where multiple classifiers are employed and their results are combined to produce the final output of the ensemble (for an overview, see (Dietterich, 1997)). Such ensemble methods have recently been adopted in question answering (Chu-Carroll et al., 2003b; Burger et al., 2003). In our question answering system, PIQUANT, we utilize in parallel multiple answering agents that adopt different processing strategies and consult different knowledge sources in identifying answers to given questions, and we employ resolution mechanisms to combine the results produced by the individual answering agents.

We call our approach **multi-strategy** since we combine the results from a number of independent agents implementing different answer finding strategies. We also call it **multi-source** since the different agents can search for answers in multiple knowledge sources. In this paper, we focus on two answering agents that adopt fundamentally different strategies: one agent uses predominantly knowledge-based mechanisms, whereas the other agent is based on statistical methods. Our multi-level resolution algorithm enables combination of results from each answering agent at the question, passage, and/or answer levels. Our experiments show that in most cases our multi-level resolution algorithm outperforms its components, supporting a tightly-coupled design for multi-agent QA systems. Experimental results show significant performance improvement over our single-strategy, single-source baselines, with the best performing multi-level resolution algorithm achieving a 35.0% relative improvement in the number of correct answers and a 32.8% improvement in average precision, on a previously unseen test set.

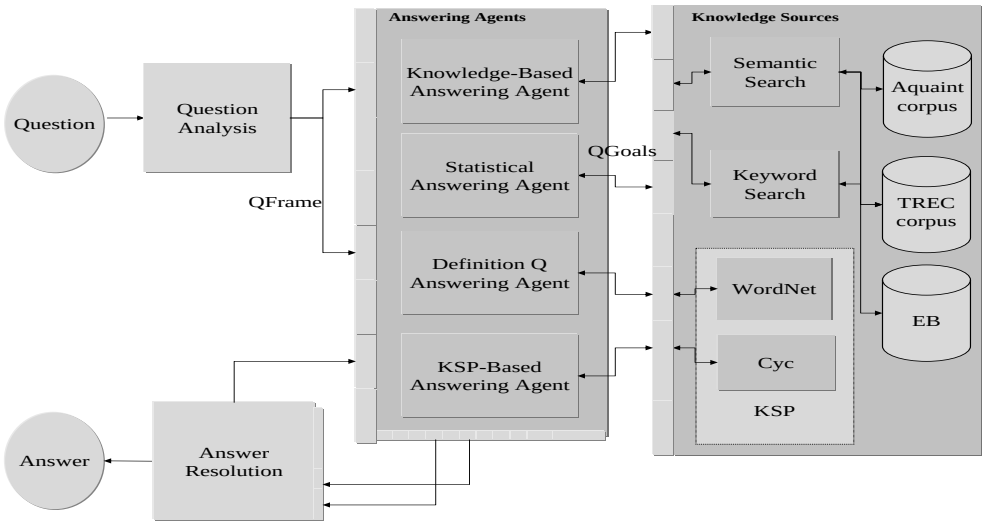

Figure 1: PIQUANT's Architecture

2 A Multi-Agent QA Architecture

In order to enable a multi-source and multi-strategy approach to question answering, we developed a modular and extensible QA architecture as shown in Figure 1 (Chu-Carroll et al., 2003a; Chu-Carroll et al., 2003b). With a consistent interface defined for each component, this architecture allows for easy plug-and-play of individual components for experimental purposes.

In our architecture, a question is first processed by the question analysis component. The analysis results are represented as a QFrame, which minimally includes a set of question features that help activate one or more answering agents. Each answering agent takes the QFrame and generates its own set of requests to a variety of knowledge sources. This may include performing search against a text corpus and extracting answers from the resulting passages, or performing a query against a structured knowledge source, such as WordNet (Miller, 1995) or Cyc (Lenat, 1995). The (intermediate) results from the individual answering agents are then passed on to the answer resolution component, which combines and resolves 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.

We have developed multiple answering agents, some general purpose and others tailored for specific question types. Figure 1 shows the answering agents currently available in PIQUANT. The knowledge-based and statistical answering agents are general-purpose agents that adopt different processing strategies and consult a number of different text resources. The definition-Q agent targets definition questions (e.g., *"What is penicillin?"* and *"Who is Picasso?"*) with a technique called *Virtual Annotation* using the external knowledge source WordNet (Prager et al., 2001). The KSP-based answering agent focuses on a subset of factoid questions with specific logical forms, such as *capital(?COUNTRY)* and *state_tree(?STATE)*. The answering agent sends requests to the KSP (Knowledge Sources Portal), which returns, if possible, an answer from a structured knowledge source (Chu-Carroll et al., 2003a).

In the rest of this paper, we briefly describe our two general-purpose answering agents. We then focus on a multi-level answer resolution algorithm, applicable at different points in the QA process of these two answering agents. Finally, we discuss experiments conducted to discover effective methods for combining results from multiple answering agents.

3 Component Answering Agents

We focus on two end-to-end answering agents designed to answer short, fact-seeking questions from a collection of text documents, as motivated by the requirements of the TREC QA track (Voorhees, 2003). Both answering agents adopt the classic pipeline architecture, consisting roughly of question analysis, passage retrieval, and answer selection components. Although the answering agents adopt fundamentally different strategies in their individual components, they have performed quite comparably in past TREC QA tracks (Voorhees, 2001; Voorhees, 2002).

3.1 Knowledge-Based Answering Agent

Our first answering agent utilizes a primarily knowledge-driven approach, based on *Predictive Annotation* (Prager et al., 2000). A key characteristic of this approach is that

potential answers, such as person names, locations, and dates, in the corpus are predictively annotated. In other words, the corpus is indexed not only with keywords, as is typical for most search engines, but also with the semantic classes of these pre-identified potential answers.

During the question analysis phase, a rule-based mechanism is employed to select one or more expected answer types, from a set of about 80 classes used in the predictive annotation process, along with a set of question keywords. A weighted search engine query is then constructed from the keywords, their morphological variations, synonyms, and the answer type(s). The search engine returns a hit list of typically 10 passages, each consisting of 1-3 sentences. The candidate answers in these passages are identified and ranked based on three criteria: 1) match in semantic type between candidate answer and expected answer, 2) match in weighted grammatical relationships between question and answer passages, and 3) frequency of answer in candidate passages (redundancy). The answering agent returns the top *n* ranked candidate answers along with a confidence score for each answer.

3.2 Statistical Answering Agent

The second answering agent takes a statistical approach to question answering (Ittycheriah, 2001; Ittycheriah et al., 2001). It models the distribution $p(c|q,a)$, which measures the "correctness" (c) of an answer (a) to a question (q), by introducing a hidden variable representing the answer type (e) as follows:

$$\begin{aligned} p(c|q,a) &= \sum_e p(c,e|q,a) \\ &= \sum_e p(c|e,q,a)p(e|q,a) \end{aligned}$$

$p(e|q,a)$ is the answer type model which predicts, from the question and a proposed answer, the answer type they both satisfy. $p(c|e,q,a)$ is the answer selection model. Given a question, an answer, and the predicted answer type, it seeks to model the correctness of this configuration. These distributions are modeled using a maximum entropy formulation (Berger et al., 1996), using training data which consists of human judgments of question answer pairs. For the answer type model, 13K questions were annotated with 31 categories. For the answer selection model, 892 questions from the TREC 8 and TREC 9 QA tracks were used, along with 4K trivia questions.

During runtime, the question is first analyzed by the answer type model, which selects one out of a set of 31 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 that 1) maximize the question word match, 2) have the desired answer type, 3) minimize the dispersion of question words, and 4) have similar syntactic structures as the question. From these passages, candidate answers are extracted and ranked using the answer selection model. The top *n* candidate answers are then returned, each with an associated confidence score.

4 Answer Resolution

Given two answering agents with the same pipeline architecture, there are multiple points in the process at which (intermediate) results can be combined, as illustrated in Figure 2. More specifically, it is possible for one answering agent to provide input to the other after the question analysis, passage retrieval, and answer selection phases. In PIQUANT, the knowledge based agent may accept input from the statistical agent after each of these three phases.[1] The contributions from the statistical agent are taken into consideration by the knowledge based answering agent in a phase-dependent fashion. The rest of this section details our combination strategies for each phase.

4.1 Question-Level Combination

One of the key tasks of the question analysis component is to determine the expected answer type, such as PERSON for *"Who discovered America?"* and DATE for *"When did World War II end?"* This information is taken into account by most existing QA systems when ranking candidate answers, and can also be used in the passage retrieval process to increase the precision of candidate passages.

We seek to improve the knowledge-based agent's performance in passage retrieval and answer selection through better answer type identification by consulting the statistical agent's expected answer type. This task, however, is complicated by the fact that QA systems employ different sets of answer types, often with different granularities and/or with overlapping types. For instance, while one system may generate ROYALTY for the question *"Who was the King of France in 1702?"*, another system may produce PERSON as the most specific answer type in its repertoire. This is quite a serious problem for us as the knowledge based agent uses over 80 answer types while the statistical agent adopts only 31 categories.

In order to distinguish actual answer type discrepancies from those due to granularity differences, we first manually created a mapping between the two sets of answer types. This mapping specifies, for each answer type used by the statistical agent, a set of *possible* corresponding types used by the knowledge-based agent. For example, the GEOLOGICALOBJ class is mapped to a set of finer grained classes: RIVER, MOUNTAIN, LAKE, and OCEAN. At processing time, the statistical agent's answer type is mapped to the knowledge-based agent's classes (SA-

[1] Although it is possible for the statistical agent to receive input from the knowledge based agent as well, we have not pursued that option because of implementation issues.

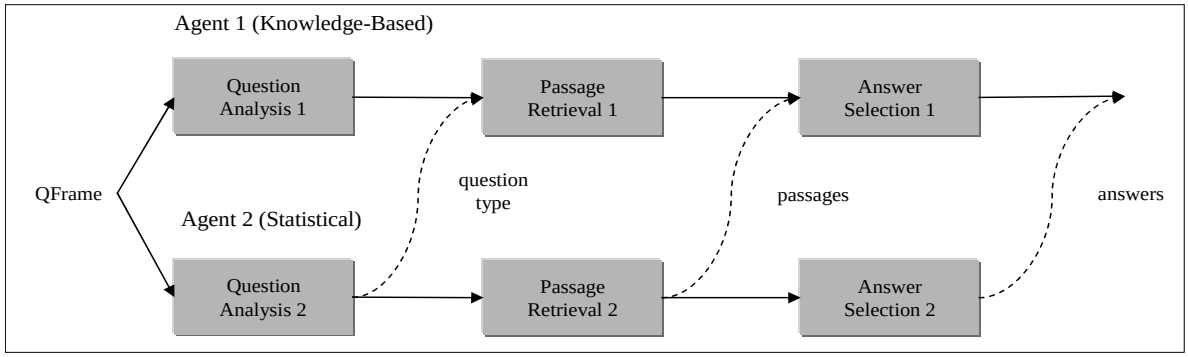

Figure 2: Answer Resolution Strategies

types), which are then merged with the answer type(s) selected by the knowledge-based agent itself (KBA-types) as follows:

1. If the intersection of KBA-types and SA-types is non-null, i.e., the two agents produced consistent answer types, then the merged set is KBA-types.

2. Otherwise, the two sets of answer types are truly in disagreement, and the merged set is the union of KBA-types and SA-types.

The merged answer types are then used by the knowledge-based agent in further processing.

4.2 Passage-Level Combination

The passage retrieval component selects, from a large text corpus, a small number of short passages from which answers are identified. Oftentimes, multiple passages that answer a question are retrieved. Some of these passages may be better suited than others for the answer selection algorithm employed downstream. For example, consider *"When was Benjamin Disraeli prime minister?"*, whose answer can be found in both passages below:

1. Benjamin Disraeli, who had become prime minister in 1868, was born into Judaism but was baptized a Christian at the age of 12.

2. France had a Jewish prime minister in 1936, England in 1868, and Spain, of all countries, in 1835, but none of them, Leon Blum, Benjamin Disraeli or Juan Alvarez Mendizabel, were devoutly observant, as Lieberman is.

Although the correct answer, 1868, is present in both passages, it is substantially easier to identify the answer from the first passage, where it is directly stated, than from the second passage, where recognition of parallel constructs is needed to identify the correct answer.

Because of strategic differences in question analysis and passage retrieval, our two answering agents often retrieve different passages for the same question. Thus, we perform passage-level combination to make a wider variety of passages available to the answer selection component, as shown in Figure 2. The potential advantages are threefold. First, passages from agent 2 may contain answers absent in passages retrieved by agent 1. Second, agent 2 may have retrieved passages better suited for the downstream answer selection algorithm than those retrieved by agent 1. Third, passages from agent 2 may contain additional occurrences of the correct answer, which boosts the system's confidence in the answer through the redundancy measure.[2]

Our passage-level combination algorithm adds to the passages extracted by the knowledge-based agent the top-ranked passages from the statistical agent that contain candidate answers of the right type. More specifically, the statistical agent's passages are semantically annotated and the top 10 passages containing at least one candidate of the expected answer type(s) are selected.[3]

4.3 Answer-Level Combination

The answer selection component identifies, from a set of passages, the top *n* answers for the given question, with their associated confidence scores. An answer-level combination algorithm takes the top answer(s) from the individual answering agents and determines the overall best answer(s). Of our three combination algorithms, this most closely resembles traditional ensemble methods, as voting takes place among the end results of individual an-

[2]On the other hand, such redundancy may result in error compounding, as discussed in Section 5.3.

[3]We selected the top 10 passages so that the same number of passages are considered from both answering agents.

swering agents to determine the final output of the ensemble.

We developed two answer-level combination algorithms, both utilizing a simple confidence-based voting mechanism, based on the premise that answers selected by both agents with high confidence are more likely to be correct than those identified by only one agent.[4] In both algorithms, named entity normalization is first performed on all candidate answers considered. In the first algorithm, only the top answer from each agent is taken into account. If the two top answers are equivalent, the answer is selected with the combined confidence from both agents; otherwise, the more confident answer is selected.[5] In the second algorithm, the top 5 answers from each agent are allowed to participate in the voting process. Each instance of an answer votes with a weight equal to its confidence value and the weights of equivalent answers are again summed. The answer with the highest weight, or confidence value, is selected as the system's final answer. Since in our evaluation, the second algorithm uniformly outperforms the first, it is adopted as our answer-level combination algorithm in the rest of the paper.

5 Performance Evaluation

5.1 Experimental Setup

To assess the effectiveness of our multi-level answer resolution algorithm, we devised experiments to evaluate the impact of the question, passage, and answer-level combination algorithms described in the previous section.

The baseline systems are the knowledge-based and statistical agents performing individually against a single reference corpus. In addition, our earlier experiments showed that when employing a single answer finding strategy, consulting multiple text corpora yielded better performance than using a single corpus. We thus configured a version of our knowledge-based agent to make use of three available text corpora,[6] the AQUAINT corpus (news articles from 1998-2000), the TREC corpus (news articles from 1988-1994),[7] and a subset of the Encyclopedia Britannica. This multi-source version of the knowledge-based agent will be used in all answer resolution experiments in conjunction with the statistical agent.

We configured multiple versions of PIQUANT to evaluate our question, passage, and answer-level combination algorithms individually and cumulatively. For cumulative effects, we 1) combined the algorithms pair-wise, and 2) employed all three algorithms together. The two test sets were selected from the TREC 10 and 11 QA track questions (Voorhees, 2002; Voorhees, 2003). For both test sets, we eliminated those questions that did not have known answers in the reference corpus. Furthermore, from the TREC 10 test set, we discarded all definition questions,[8] since the knowledge-based agent adopts a specialized strategy for handling definition questions which greatly reduces potential contributions from other answering agents. This results in a TREC 10 test set of 313 questions and a TREC 11 test set of 453 questions.

5.2 Experimental Results

We ran each of the baseline and combined systems on the two test sets. For each run, the system outputs its top answer and its confidence score for each question. All answers for a run are then sorted in descending order of the confidence scores. Two established TREC QA evaluation metrics are adopted to assess the results for each run as follows:

1. **% Correct**: Percentage of correct answers.
2. **Average Precision**: A confidence-weighted score that rewards systems with high confidence in correct answers as follows, where N is the number of questions:

$$\frac{1}{N}\sum_{i=1}^{N} \#\text{ correct up to question i}/i$$

Table 1 shows our experimental results. The top section shows the comparable baseline results from the statistical agent (SA-SS) and the single-source knowledge-based agent (KBA-SS). It also includes results for the multi-source knowledge-based agent (KBA-MS), which improve upon those for its single-source counterpart (KBA-SS).

The middle section of the table shows the answer resolution results, including applying the question, passage, and answer-level combination algorithms individually (Q, P, and A, respectively), applying them pair-wise (Q+P, P+A, and Q+A), and employing all three algorithms (Q+P+A). Finally, the last row of the table shows the relative improvement by comparing the best performing system configuration (highlighted in boldface) with the better performing single-source, single-strategy baseline system (SA-SS or KBA-SS, in italics).

Overall, PIQUANT's multi-strategy and multi-source approach achieved a 35.0% relative improvement in the

[4] In future work we will be investigating weighted voting schemes based on question features.

[5] The confidence values from both answering agents are normalized to be between 0 and 1.

[6] The statistical agent is currently unable to consult multiple corpora.

[7] Both the AQUAINT and TREC corpora are available from the Linguistics Data Consortium, http://www.ldc.org.

[8] Definition questions were intentionally excluded by the track coordinator in the TREC 11 test set.

	TREC 10 (313)		TREC 11 (453)	
	% Corr	Avg Prec	% Corr	Avg Prec
SA-SS	36.7%	0.569	*32.9%*	*0.534*
KBA-SS	*39.6%*	*0.595*	32.5%	0.531
KBA-MS	43.8%	0.641	38.2%	0.622
Q	44.7%	0.647	38.9%	0.632
P	49.5%	0.661	40.0%	0.627
A	49.5%	0.712	43.5%	0.704
Q+P	48.9%	0.656	41.1%	0.640
P+A	**51.1%**	0.711	44.2%	0.686
Q+A	49.8%	**0.716**	43.9%	**0.709**
Q+P+A	50.8%	0.706	**44.4%**	0.690
rel. improv.	29.0%	20.3%	35.0%	32.8%

Table 1: Experimental Results

number of correct answers and a 32.8% improvement in average precision on the TREC 11 data set. Of the combined improvement, approximately half was achieved by the multi-source aspect of PIQUANT, while the other half was obtained by PIQUANT's multi-strategy feature. Although the absolute average precision values are comparable on both test sets and the absolute percentage of correct answers is lower on the TREC 11 data, the improvement is greater on TREC 11 in both cases. This is because the TREC 10 questions were taken into account for manual rule refinement in the knowledge-based agent, resulting in higher baselines on the TREC 10 test set. We believe that the larger improvement on the previously unseen TREC 11 data is a more reliable estimate of PIQUANT's performance on future test sets.

We applied an earlier version of our combination algorithms, which performed between our current P and P+A algorithms, in our submission to the TREC 11 QA track. Using the average precision metric, that version of PIQUANT was among the top 5 best performing systems out of 67 runs submitted by 34 groups.

5.3 Discussion and Analysis

A cursory examination of the results in Table 1 allows us to draw two general conclusions about PIQUANT's performance. First, all three combination algorithms applied individually improved upon the baseline using both evaluation metrics on both test sets. In addition, overall performance is generally better the later in the process the combination occurs, i.e., the answer-level combination algorithm outperformed the passage-level combination algorithm, which in turn outperformed the question-level combination algorithm. Second, the cumulative improvement from multiple combination algorithms is in general greater than that from the components. For instance, the Q+A algorithm uniformly outperformed the Q and A algorithms alone. Note, however, that the Q+P+A algorithm achieved the highest performance only on the TREC 11 test set using the % correct metric. We believe

		KBA			
		TREC 10 (313)		TREC 11 (453)	
		+	-	+	-
SA	+	185	**43**	254	**58**
	-	24	61	41	100

Table 2: Passage Retrieval Analysis

that this is because of compounding errors that occurred during the multiple combination process.

In ensemble methods, the individual components must make *different* mistakes in order for the combined system to potentially perform better than the component systems (Dietterich, 1997). We examined the differences in results between the two answering agents from their question analysis, passage retrieval, and answer selection components. We focused our analysis on the potential gain/loss from incorporating contributions from the statistical agent, and how the potential was realized as actual performance gain/loss in our end-to-end system.

At the question level, we examined those questions for which the two agents proposed incompatible answer types. On the TREC 10 test set, the statistical agent introduced correct answer types in 6 cases and incorrect answer types in 9 cases. As a result, in some cases the question-level combination algorithm improved system performance (comparing A and Q+A) and in others it degraded performance (comparing P and Q+P). On the other hand, on the TREC 11 test set, the statistical agent introduced correct and incorrect answer types in 15 and 6 cases, respectively. As a result, in most cases performance improved when the question-level combination algorithm was invoked. The difference in question analysis performance again reflects the fact that TREC 10 questions were used in question analysis rule refinement in the knowledge-based agent.

At the passage level, we examined, for each question, whether the candidate passages contained the correct answer. Table 2 shows the distribution of questions for which correct answers were (+) and were not (-) present in the passages for both agents. The boldfaced cells represent questions for which the statistical agent retrieved passages with correct answers while the knowledge-based agent did not. There were 43 and 58 such questions in the TREC 10 and TREC 11 test sets, respectively, and employing the passage-level combination algorithm resulted only in an additional 18 and 8 correct answers on each test set. This is because the statistical agent's proposes in its 10 passages, on average, 29 candidate answers, most of which are incorrect, of the proper semantic type per question. As the downstream answer selection component takes redundancy into account in answer ranking, incorrect answers may reinforce one another and become top ranked answers. This suggests that

		KBA					
		TREC 10 (313)			TREC 11 (453)		
		1st	2-5th	none	1st	2-5th	none
SA	1st	**66**	**22**	*26*	**93**	**21**	*35*
	2-5th	**26**	*9*	13	**29**	*19*	22
	none	*45*	14	92	*51*	21	162

Table 3: Answer Voting Analysis

the relative contributions of our answer selection features may not be optimally tuned for our multi-agent approach to QA. We plan to investigate this issue in future work.

At the answer level, we analyzed each agent's top 5 answers, used in the combination algorithm's voting process. Table 3 shows the distribution of questions for which an answer was found in 1st place, in 2nd-5th place, and not found in top 5. Since we employ a linear voting strategy based on confidence scores, we classify the cells in Table 3 as follows based on the perceived likelihood that the correct answers for questions in each cell wins in the voting process. The boldfaced and underlined cells contain *highly likely* candidates, since a correct answer was found in 1st place by both agents.[9] The boldfaced cells consist of *likely* candidates, since a 1st place correct answer was supported by a 2nd-5th place answer. The italicized and underlined cells contain *possible* candidates, while the rest of the cells cannot produce correct 1st place answers using our current voting algorithm. On TREC 10 data, 194 questions fall into the *highly likely*, *likely*, and *possible* categories, out of which the voting algorithm successfully selected 155 correct answers in 1st place. On TREC 11 data, 197 correct answers were selected out of 248 questions that fall into these categories. These results represent success rates of 79.9% and 79.4% for our answer-level combination algorithm on the two test sets.

6 Related Work

There has been much work in employing ensemble methods to increase system performance in machine learning. In NLP, such methods have been applied to tasks such as POS tagging (Brill and Wu, 1998), word sense disambiguation (Pedersen, 2000), parsing (Henderson and Brill, 1999), and machine translation (Frederking and Nirenburg, 1994).

In question answering, 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 question answering. However, the approaches taken by both these systems differ from ours in that they enforce an order between the two strategies by attempting to locate answers in structured databases first for select question types and falling back to unstructured text when the former fails, while we explore both options in parallel and *combine* the results from multiple answering agents.

The multi-agent approach to question answering 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(s). Furthermore, their approach differs from ours in that they focused on combining the end results of a large number of systems, while we investigated a tightly-coupled design for combining two answering agents.

7 Conclusions

In this paper, we introduced a multi-strategy and multi-source approach to question answering that enables combination of answering agents adopting different strategies and consulting multiple knowledge sources. In particular, we focused on two answering agents, one adopting a knowledge-based approach and one using statistical methods. We discussed our answer resolution component which employs a multi-level combination algorithm that allows for resolution at the question, passage, and answer levels. Best performance using the % correct metric was achieved by the three-level algorithm that combines after each stage, while highest average precision was obtained by a two-level algorithm merging at the question and answer levels, supporting a tightly-coupled design for multi-agent question answering. Our experiments showed that our best performing algorithms achieved a 35.0% relative improvement in the number of correct answers and a 32.8% improvement in average precision on a previously unseen test set.

Acknowledgments

We would like to thank Dave Ferrucci, Chris Welty, and Salim Roukos for helpful discussions, Diane Litman and the anonymous reviewers for their comments on an earlier draft of this paper. 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] These cells are not marked as *definite* because in a small number of cases, the two answers are not equivalent. For example, for the TREC 9 question, *"Who is the emperor of Japan?"*, Hirohito, Akihito, and Taisho are all considered correct answers based on the reference corpus.

References

Adam L. Berger, Vincent Della Pietra, and Stephen Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Eric Brill and Jun Wu. 1998. Classifier combination for improved lexical disambiguation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 191–195.

John D. Burger, Lisa Ferro, Warren Greiff, John Henderson, Marc Light, and Scott Mardis. 2003. MITRE's Qanda at TREC-11. In *Proceedings of the Eleventh Text Retrieval Conference*. To appear.

Jennifer Chu-Carroll, David Ferrucci, John Prager, and Christopher Welty. 2003a. Hybridization in question answering systems. In *Working Notes of the AAAI Spring Symposium on New Directions in Question Answering*, pages 116–121.

Jennifer Chu-Carroll, John Prager, Christopher Welty, Krzysztof Czuba, and David Ferrucci. 2003b. A multi-strategy and multi-source approach to question answering. In *Proceedings of the Eleventh Text Retrieval Conference*. To appear.

Charles Clarke, Gordon Cormack, and Thomas Lynam. 2001. Exploiting redundancy in question answering. In *Proceedings of the 24th SIGIR Conference*, pages 358–365.

C.L.A. Clarke, G.V. Cormack, G. Kemkes, M. Laszlo, T.R. Lynam, E.L. Terra, and P.L. Tilker. 2003. Statistical selection of exact answers. In *Proceedings of the Eleventh Text Retrieval Conference*. To appear.

Thomas G. Dietterich. 1997. Machine learning research: Four current directions. *AI Magazine*, 18(4):97–136.

Robert Frederking and Sergei Nirenburg. 1994. Three heads are better than one. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*.

John C. Henderson and Eric Brill. 1999. Exploiting diversity in natural language processing: Combining parsers. In *Proceedings of the 4th Conference on Empirical Methods in Natural Language Processing*.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, and Chin-Yew Lin. 2001. Question answering in Webclopedia. In *Proceedings of the Ninth Text REtrieval Conference*, pages 655–664.

Abraham Ittycheriah, Martin Franz, Wei-Jing Zhu, and Adwait Ratnaparkhi. 2001. Question answering using maximum entropy components. In *Proceedings of the 2nd Conference of the North American Chapter of the Association for Computational Linguistics*, pages 33–39.

Abraham Ittycheriah. 2001. *Trainable Question Answering Systems*. Ph.D. thesis, Rutgers - The State University of New Jersey.

Douglas B. Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11).

Jimmy Lin, Aaron Fernandes, Boris Katz, Gregory Marton, and Stefanie Tellex. 2003. Extracting answers from the web using knowledge annotation and knowledge mining techniques. In *Proceedings of the Eleventh Text Retrieval Conference*. To appear.

George Miller. 1995. Wordnet: A lexical database for English. *Communications of the ACM*, 38(11).

Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. 2000. The structure and performance of an open-domain question answering system. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 563–570.

Ted Pedersen. 2000. A simple approach to building ensembles of naive Bayesian classifiers for word sense disambiguation. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, pages 63–69.

John Prager, Eric Brown, Anni Coden, and Dragomir Radev. 2000. Question-answering by predictive annotation. In *Proceedings of the 23rd SIGIR Conference*, pages 184–191.

John Prager, Dragomir Radev, and Krzysztof Czuba. 2001. Answering what-is questions by virtual annotation. In *Proceedings of Human Language Technologies Conference*, pages 26–30.

Ellen M. Voorhees. 2001. Overview of the TREC-9 question answering track. In *Proceedings of the 9th Text Retrieval Conference*, pages 71–80.

Ellen M. Voorhees. 2002. Overview of the TREC 2001 question answering track. In *Proceedings of the 10th Text Retrieval Conference*, pages 42–51.

Ellen M. Voorhees. 2003. Overview of the TREC 2002 question answering track. In *Proceedings of the Eleventh Text Retrieval Conference*. To appear.

Jinxi Xu and W. Bruce Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th SIGIR Conference*, pages 4–11.