Improving Question Answering for Reading Comprehension Tests by Combining Multiple Systems

Mihai Rotaru and Diane J. Litman

University of Pittsburgh
Department of Computer Science
Pittsburgh, PA, USA 15260
{mrotaru, litman}@cs.pitt.edu

Abstract

Most work on reading comprehension question answering systems has focused on improving performance by adding complex natural language processing (NLP) components to such systems rather than by combining the output of multiple systems. Our paper empirically evaluates whether combining the outputs of seven such systems submitted as the final projects for a graduate level class can improve over the performance of any individual system. We present several analyses of our combination experiments, including performance bounds, impact of both tie-breaking methods and ensemble size on performance, and an error analysis. Our results, replicated using two different publicly available reading test corpora, demonstrate the utility of system combination via majority voting in our restricted domain question answering task.

Introduction

This paper examines the potential for improvement offered by combining multiple reading comprehension (**RC**) question answering (**QA**) systems via majority voting as an ensemble method. The availability of RC tests as a source of training and testing data for researchers in QA has led to the implementation and comparative evaluation of a variety of QA systems in this domain (Hirschman et al. 1999; Riloff and Thelen 2000; Light et al. 2001).

QA for RC shares many similarities with QA from large text collections (as exemplified by the QA track of the Text Retrieval Conference (TREC) series). Systems in both settings are designed to answer fact-retrieval questions in any domain from arbitrary documents. However, the TREC systems must first find a document or documents, given a collection of hundreds of thousands of documents that contains the answer. In RC, the target document containing the answer is supplied with the question, obviating the need to incorporate information retrieval technology in the RC setting.

However, this difference creates additional challenges for RC QA systems. Probably the most important one is that each story can be viewed as a restricted domain: most

of the information about the domain (the story) is contained in the story text. Since in many cases the stories are fictional or about obscure events, additional information about the domain is hard to find or nonexistent. This limits the applicability of conventional QA techniques (e.g. data redundancy-based (Brill et al. 2001)). To make things even worse, the stories can discuss virtually any topic thus requiring broad-coverage NLP techniques.

On the other hand, RC tests have other properties that make RC QA tractable. Since these tests were originally designed for lower grade students, the stories have simpler structure and make use of simpler language constructs. The questions also range in difficulty from simple questions like where/when-questions to more complex questions like why-questions which require in many cases inter-sentential processing. Moreover, a lot of the words from questions' text are present in the story, making the bag-of-words approach a successful starting point (Hirschman et al. 1999).

Because RC QA systems need to answer broad-coverage questions using limited resources, improving the performance in RC comprehension systems is very hard. Previous work (Hirchman et al. 1999; Riloff and Thelen 2000) has shown that often adding complex NLP components to a RC QA system offers little or no improvement at all. Instead we propose exploiting "redundancy" across multiple systems: we believe that combining different QA systems is an underexplored alternative resource for improvement for RC and, as our results will show, there is a lot to be gained from it. Our research begins with seven RC question answering systems submitted as final projects for the graduate Natural Language Processing course at the University of Pittsburgh, and explores whether and how the use of a voting ensemble methods can improve performance.

Ensemble methods in machine learning have shown great success in improving predictive accuracy. Such systems typically employ multiple classifiers to first solve the same problem, then combine the results to provide a final ensemble answer (Dietterich 1997). Although determining exactly how to best combine individual results is still an active area of research, a variety of ensemble methods have already been shown to improve predictive

performance in various areas of natural language processing, including word sense disambiguation, parsing, and TREC style QA (Florian et al. 2002; Henderson and Brill 1999; Burger et al. 2003; Chu-Carroll et al. 2003).

In this paper we empirically evaluate the utility of employing one of the most popular ensemble methods - majority voting - to combine the outputs of seven RC QA systems. The results of our experiments show that while there is no clear "best" individual RC system (or even small set of systems), the ensemble system performs comparably to whatever the best performing system might have been in any particular experimental condition. Our experimental results also suggest that the larger the ensemble, the better the performance, and that more sophisticated methods of voting have the potential to further enhance performance.

Projects and Datasets

Seven QA systems were built by the class. Since students were free to choose what natural language algorithms to incorporate into their system, the choices made by each team were often quite different (e.g., only a few systems included any semantic processing, and even fewer included coreference). Even when multiple teams incorporated the same type of component (such as a part of speech tagger), they usually used a different implementation (e.g. Brill's tagger versus GATE).

The training and testing data for the QA projects came from two RC corpora available from the MITRE Corporation for research purposes. The "**Remedia** Corpus" consists of 115 short stories with 5 questions (when, where, what, why and who) following each story; these are actual RC tests (produced by Remedia Publications) given to grade school children in the United States. The "**CBC** Reading Comprehension Corpus" consists of 125 news stories from the "CBC 4 Kids" website (hosted by the Canadian Broadcasting Corporation), along with RC tests created by MITRE that were modeled after the Remedia data (but also included "how" and occasionally other question types). A sample CBC story (annotated with answers) and its RC test are shown in Figure 1.

Scientists find dinosaur bones in Antarctica
July 9, 1999

<ANSQ1>Scientists in the Antarctic have discovered some very old dinosaur bones, including partial skeletons, whole jaws and teeth.</ANSQ1>
<ANSQ2>The remains include the bones of two giant water dinosaurs: the mosasaur, a razor-toothed "duck-bill" animal with paddles, and the plesiosaur, which scientists say looks like Scotland's legendary Loch Ness monster.</ANSQ2>
American geologists made the find in January, but did not announce the discovery until this week at an international earth sciences meeting in Wellington, New Zealand.
The find shows that at least four different types of mosasaur lived in the Antarctic. One type had previously been found only in North America and Europe.

Mosasaurs have been found across the world from Sweden to New Zealand, but never in Antarctica.
<ANSQ5>The mosasaur discovery shows the Antarctic was once much warmer than it is now.</ANSQ5>
<ANSQ4> The scientists believe the creatures probably came to the Antarctic around 75 million or 80 million years ago.</ANSQ4>
Mosasaurs were around 10 metres long. <ANSQ3>Their huge skulls were filled with 10 centimetre long teeth.</ANSQ3>
A team at an American museum are studying the fossils.

<Q1> What parts of the dinosaurs did the scientists find?
<Q2> Who is the plesiosaur supposed to resemble?
<Q3> How big were the mosasaur's teeth?
<Q4> When did mosasaurs probably arrive in Antarctica?
<Q5> What does this new discovery of dinosaur bones tell us about the Antarctic?

Figure 1: A Sample Story, its RC Test and Answers (CBC Corpus - Story 1999-W28-5)

Because creating a QA system that can identify exact answers (as in the TREC competitions) is difficult, following both previous RC QA research and evaluations of classroom projects, the goal of the student QA systems is instead to identify the *sentence* in each story that best answers each question. For example, while the exact answer to <Q2> in Figure 1 (according to MITRE's answer key) is "Scotland's legendary Loch Ness monster" or "the Loch Ness monster", the answer sentence for the student project is the second sentence (shown in Figure 1 between the <ANSQ2> and </ANSQ2> tags).

We partitioned each data set in three partitions. The CBC's Training Set and Test Set 1 partitions were used by the students to develop their systems (Test Set 1 was used for a preliminary evaluation originally). To determine the project grade, each QA system was evaluated on Test Set 2 partition. To test the generality of the voting results presented here, each QA system was also evaluated using different partitions of the Remedia data; Remedia rather than CBC data was used by a previous NLP class. Note that the CBC Test Set 2 partition and all three Remedia[1] data partitions were *blind test sets* for our QA systems. The results of these evaluations are shown in Table 1. Each cell shows the percentage of questions answered correctly for each system/dataset combination. The best performing system in each column has been highlighted.

Examination of Table 1 shows that there is no clear "best" QA system. Within both corpora, the best system varies with data partition. For example, on the Remedia data, Team 5 is the best of the seven systems on Set 3 but the worst on Set 2! Across corpora, the systems that rank

---

[1] Surprisingly the performance figures for Remedia were quite comparable to the figures from the previous class, even though those systems were actually trained on Remedia data. Note that the much lower accuracy figures for the Remedia datasets suggest that Remedia is a more difficult corpus than CBC for QA technology. A similar difference between CBC and Remedia was also noted in the studies of (Dalmas et al. 2003).

1st are disjoint. Furthermore, while the ranking of Team 3 ranges between 1st and 3rd in the CBC evaluations, it only ranges between 3rd and 6th in the Remedia evaluations. We also found that the "best" system also differs by question type. This variety among systems with respect to performance motivates our research on combining system outputs as one method for increasing QA accuracy.

# of questions	CBC			Remedia		
	324	310	479	150	150	275
System	Training set	Test Set 1	Test Set 2	Set 1	Set 2	Set 3
Team 1	56.79%	53.23%	62.21%	30.00%	29.33%	32.36%
Team 2	58.95%	52.90%	64.30%	32.67%	34.67%	33.09%
Team 3	63.89%	60.00%	67.43%	30.00%	30.00%	32.00%
Team 4	53.09%	53.23%	59.71%	27.33%	28.00%	29.09%
Team 5	60.49%	60.97%	61.17%	30.00%	26.00%	36.73%
Team 6	61.11%	54.19%	61.80%	29.33%	30.67%	32.73%
Team 7	61.73%	63.87%	70.15%	30.67%	29.33%	36.36%
Intersection	30.86%	28.71%	37.37%	10.00%	9.33%	9.09%
Union	80.56%	78.71%	82.88%	54.00%	54.00%	57.45%

Table 1. Evaluation Results (% correct)

Combining the QA Systems

We first investigate what the upper and lower bounds of ensemble methods for combining our existing systems might be, to demonstrate that an ideal ensemble technique offers enough performance improvement to make our own investigations worthwhile. We then apply a simple yet very popular ensemble technique – majority voting – to our systems and data, and present our results. Finally, we study the effect of ensemble size on voting performance, and conclude with an analysis of how and where majority voting can likely be improved.

Upper and lower bounds

The last two rows of Table 1 present some bounds on the type of performance figures that could be expected by combining our seven systems. An ensemble that uses *Intersection* to combine multiple answers counts an answer as correct only if all seven systems answer the question correctly; this can be viewed as a lower bound. *Union*, on the other hand, requires only at least one of the seven systems to answer the question correctly, and thus provides an upper bound. The Union results for the CBC data, for example, indicate that none of our systems return the correct answer for approximately 20% of the questions; thus, it is not possible for an ensemble of only our seven systems to perform better than approximately 80%. Furthermore, note that it is hard to achieve this upper bound in practice because we need to know how to pick the correct answer, which is the problem we are trying to solve.

Further analysis of the last two rows of Table 1 offers other interesting insights about our QA systems' coverage. While there are relatively few cases where all seven systems give correct answers (the intersection values are

around 33% for CBC and 9% for Remedia), their abilities to return the correct answer are divergent and seem to complement one another. This is supported by the fact that union values from Table 1 are much bigger than the best performer (the highlighted cells) in each column. The large increase in accuracy offered by union (about 15% absolute improvement for CBC and 20% for Remedia) suggests that each individual QA system has a certain area of competence different from the others.

Moreover, the amount of correlation between our systems is relatively low (a low correlation is a prerequisite for successful ensemble methods). To measure the correlation, we use the interagreement metric used in (Florian et al. 2002). Interagreement is defined as the percentage of time two QA systems provide the same answer. The average interagreement was around 68% (std. deviation: 4%) for CBC partitions and 50% (std. deviation: 6%) for Remedia. These numbers together with the variability in systems' performance and the high values of ensemble upper bound suggest that system combination has the potential for a lot of improvement.

Before applying the majority voting ensemble technique we wanted to know if the improvement offered by Union is due to systems' complementarity or randomness. That is, the more answers we are allowed to provide the higher the chance to report at least one correct answer even if the new answers are not well-informed. To test this hypothesis, we combined the best performing QA with six random systems (systems that report random answers for each question) and computed the Union for this random ensemble. We found that Union's performance on our 7 systems is better than the random ensemble on all our sets.

Majority voting results

Because (to our knowledge) we are the first to apply ensemble methods to the area of RC QA, we chose to experiment with a simple but popular combination technique that has been shown to be robust across many other types of natural language applications: majority voting. All seven of our QA systems were run in parallel, and for each answer, we counted how many QA systems produced that answer. The answer with the highest count (number of votes) was selected as the voting answer.

Please note that even though we have an odd number of systems, ties are possible. For example, 3 systems might output A, the next 3 might answer B while the last one might answer with C. We solved the ties by randomly picking an answer among the answers with the highest count. However, because more complicated tie breaking techniques could have been employed, we also investigated the upper bound (always pick the correct answer among the ties - if any) and lower bound (always pick the wrong answer among the ties) of tie-breaking. We will call these three tie-breaking strategies *voting random*, *voting upper* and *voting lower*.

Table 2 reports, for every set, the improvements over the best performing QA system in that set for different ensemble techniques: always selecting the worst

performer, average performer (performance computed as the average accuracy of all seven QA systems), the three voting schemes for majority voting and union. We can observe that if we are unfortunate to always select the worst performer, the loss in performance over the best performer is very large. Not surprisingly, the average performance, which is the average performance one might expect when *randomly* choosing a QA system, reduces the loss in half.

Difference	CBC			Remedia		
	Training set	Test Set 1	Test Set 2	Set 1	Set 2	Set 3
Worst perf.	-10.80%	-10.97%	-10.44%	-5.3%	-8.6%	-7.6%
Avg. perf.	-4.45%	-6.96%	-6.32%	-2.67%	-4.95%	-3.53%
Vot. lower	-0.31%	-2.26%	-1.04%	0.67%	-6.00%	-0.36%
Vot. random	0.62%	-0.97%	1.25%	3.33%	-2.67%	1.09%
Vot. upper	2.78%	1.94%	2.09%	6.00%	2.67%	2.55%
Union	16.67%	14.84%	12.73%	21.33%	19.33%	20.73%

Table 2. Improvement (in absolute percentage) over the best performer

In contrast, we find that voting upper always offers an improvement over the best performer and that in many cases (4 out of 6), voting random performs better than the best performer. However, we want to stress again that since there is no global "best" QA system, we have no way of knowing in advance who the best performer will be; thus, in practice, average performance and voting random are likely to be achieved, while best performance is not. Finally, the loss in accuracy by the lower bound voting is very small and much better than the loss from average performance (with the exception of Remedia Set 2). The improvement over the best performer offered by union is much larger than voting upper.

To summarize, our results suggest that, even by employing this simple voting technique, we will do better or comparable with the best performer (whoever it might be) in practice. We can say that majority voting *relieves* us from the task of selecting the best performer.

The effect of ensemble size on voting

We also wanted to investigate the effect of the number of QA systems on voting. For that, we performed the voting experiments discussed above for each subset of individual QA systems, for ensemble sizes 1 through 7. Figure 2 shows the average accuracy and the standard deviation for each ensemble size and tie breaking method, for the Test Set 2 partition of the CBC corpus. We can observe that the accuracy of voting upper increases very fast and then, from ensemble size of 3, starts to flatten. On the other hand, voting random and voting lower have a slow start but exhibit a steady increase, managing to almost catch up with voting upper as the ensemble size increases. Since the plots for other dataset-corpus combinations are similar, we can hypothesize that having more QA systems helps voting by increasing the accuracy of the lower bound as well as by reducing the number of ties that need to be solved.

The plot shows an anomaly at ensemble size 2: the voting upper accuracy is bigger than the accuracy for all other ensemble sizes. This is caused by the fact that there are only two QA systems used when voting and, except for the cases where the systems output the same answer, there are always ties. Since voting upper always selects the correct answer between the ties, it means that for ensemble size 2, voting upper acts like the union from Table 1. If we were to plot the same curve for union, the values for ensemble sizes 1 and 2 would be the same as the ones for voting upper. The union curve would then continue its increasing trend until it reaches the accuracy value from Table 1.
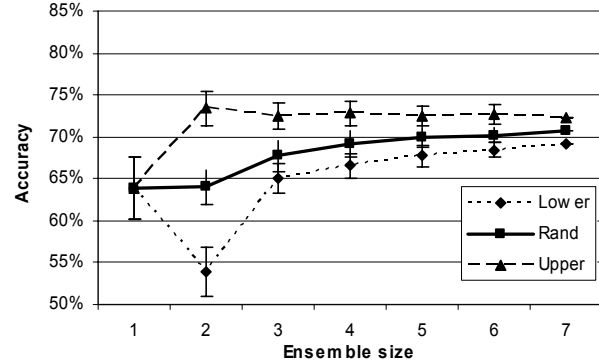

Figure 2. Influence of the ensemble size on voting (CBC – Test set 2)

Improving over majority voting

Even though majority voting relieves us from the task of selecting the best performer, and with optimal tie-breaking offers a small absolute improvement over it, there is still a long way to go before we achieve the upper bound offered by union (recall Table 2). To analyze the errors made by the voting algorithm, we investigated how far the voting process was from selecting the correct answer. Figure 3 plots the percentage of questions where the correct answer was (from left to right) 1, 2, 3, 4 and 5 votes away from the majority vote answer, for each partitioning of the CBC corpus. We can observe that for Test Set 2, in about 4% of the cases, the correct answer had only 1 vote less than the majority answer (for example 3 votes for answer A and 2 for correct answer B and another 2 for answer C); at the other extreme, in 0.63% cases the correct answer was 5 votes away (6 votes for answer A and only 1 vote for correct answer B). Please note that the entire gain from union is accumulated in the five bars for each partition. Again, although not shown, plots for the Remedia corpus exhibit a similar pattern.

The decreasing trend of the percentages with distance leads to the hypothesis that a smart weighting of each QA system vote might improve performance compared to our current results based on un-weighted majority voting. Weighting the votes of the systems might move some of the questions from bar 1 and 2 (which are close to being answered correctly) into the correctly answered question set, thus improving the accuracy. Of course, weighting

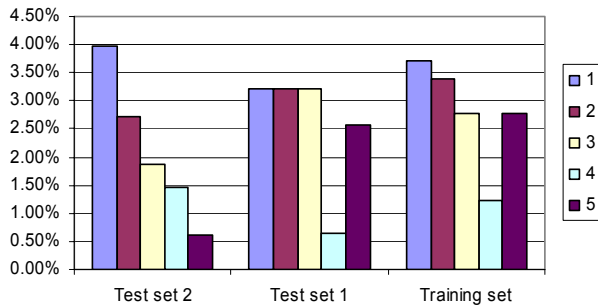might have the opposite effect of providing incorrect answers to questions correctly answered previously.

Figure 3. % of questions where the correct answer was 1, 2, 3, 4 and 5 votes away from majority answer (CBC)

We are currently experimenting with an algorithm for computing the ideal weights given a training set and test their performance on a test set. Our preliminary results indicate that the ideal weights offer a limited improvement over the best performer (a maximum of 6-7% absolute improvement for CBC). This suggests question dependent weights as a direction for future work. Our ultimate goal is to automatically learn the area of competence of each individual QA system, by treating the system as a black box and examining its answers on a training set.

Related Work

Ensemble methods have only recently been applied to combining the results of multiple QA systems, and only in the context of TREC rather than RC systems. (Chu-Carroll et al. 2003) demonstrate performance improvements using a tightly-coupled ensemble of a knowledge-based and a statistical QA system. In contrast to our work which focuses on combining the end results of a number of systems designed by others, their work focuses on combining the internal results of two of their own systems. More similar to our research is that of (Burger et al. 2003), which combines the outputs of the 67 QA systems evaluated in TREC 11 via voting. However, because TREC systems return "answer phrases" rather than "answer sentences" (as in our and most other RC work), a centroid method of voting is used to exploit partial matches across systems; this method reduces to majority voting in the case of exact matches. So far this use of centroid voting does not improve performance compared to the top scoring TREC systems. Similarly, (Jijkoun and de Rijke 2004) combine six radically different QA strategies in the TREC setting. They investigate the impact of various weighted voting techniques (including question type dependent) but no comparison with individual systems is provided. Finally, (Brill et al. 2001) present a method for automatically learning a function for combining QA system results; their method addresses the issue of partial matches, and leads to measurable performance improvements. Like (Chu-Carroll et al.

2003), however, this research has to date only been used to combine two semi-independent versions of a single group's own QA system. Unlike all of the above approaches, only our QA ensemble research has investigated the impact of ensemble size on performance.

Conclusions

Since the restricted domain nature of the RC tests limits the applicability of conventional QA techniques (e.g. data redundancy-based), we exploit system redundancy by combining multiple RC QA systems via ensemble methods. Our QA systems were the final projects of a NLP class, and were quite diverse in character. In fact, none of these systems was globally optimal: the "best" performing system varied both across and within corpora, and also by question type. Our ensemble experiments showed the utility of majority voting as a method for combining the output of such systems: the ensemble could in general match and even exceed the performance of each (locally) best individual system, without knowing how to actually choose it. Further analyses showed that performance increased with ensemble size, and that the choice of tie-breaking method became less important as ensemble size increased. Finally, we conducted an error analysis suggesting ways to improve over majority voting, to more fully reach the potential of ensemble methods.

References

Brill, E.; Lin, J.; Banko, M.; Dumais, S.; and Ng, A. 2001. Data Intensive Question Answering. In *Proc. of TREC*.

Burger, J. D.; Ferro, L.; Greiff, W.; Henderson, J.; Light, M.; Mardis, S.; and Morgan, A. 2003. MITRE's Qanda at TREC-11. In *Proc. of TREC-11*

Chu-Carroll, J.; Czuba, K.; Prager, J.; and Ittycheriah, A. 2003. In Question Answering, Two Heads are Better Than One. In *Proc. of HLT-NAACL*.

Dalmas, T.; Leidner, J. L.; Webber, B.; Grover C.; and Bos, J. 2003. Generating Annotated Corpora for Reading Comprehension and Question Answering Evaluation. In *Proc. of EACL, Question Answering Workshop*.

Dietterich, T. G. 1997. Machine Learning Research: Four Current Directions. *AI Magazine* 18(4)

Florian, R.; Cucerzan, S.; Schafer, C.; and Yarowsky, D. 2002. Combining Classifiers for Word Sense Disambiguation. *Natural Language Engineering* 8(4).

Henderson, J.; and Brill, E. 1999. Exploiting Diversity for Natural Language Processing: Combining Parsers. *Empirical Methods in Natural Language Processing/Very Large Corpora.*

Hirschman, L.; Light, M.; Breck, E.; and Burger, J.D. 1999. Deep Read: A Reading Comprehension System. In *Proc. of ACL*.

Jijkoun, V.; and de Rijke, M. 2004. Answer Selection in a Multi-Stream Open Domain Question Answering System. In *Proc. of European Conference on Information Retrieval*.

Light, M.; Mann, G. S.; Riloff, E.; and Breck, E. 2001. Analyses for elucidating current question answering technology. *Natural Language Engineering,* 7(4).

Riloff, E.; and Thelen, M. 2000. A Rule-based Question Answering System for Reading Comprehension Tests. *ANLP-NAACL Workshop on Reading Comprehension Tests as Evaluation for Comp.-Based Language Understanding Systems.*