Looking Under the Hood: Tools for Diagnosing Your Question Answering Engine[1]

Eric Breck[†], Marc Light[†], Gideon S. Mann[◇], Ellen Riloff[°], Brianne Brown[‡], Pranav Anand[*], Mats Rooth[∓], Michael Thelen[°]

[†] The MITRE Corporation, 202 Burlington Rd.,Bedford, MA 01730, {ebreck,light}@mitre.org

[◇] Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, gsm@cs.jhu.edu

[°] School of Computing, University of Utah, Salt Lake City, UT 84112, {riloff,thelenm}@cs.utah.edu

[‡] Bryn Mawr College, Bryn Mawr, PA 19010, bbrown@brynmawr.edu

[*] Department of Mathematics, Harvard University, Cambridge, MA 02138, anand@fas.harvard.edu

[∓] Department of Linguistics, Cornell University, Ithaca, NY 14853, mr249@cornell.edu

Abstract

In this paper we analyze two question answering tasks : the TREC-8 question answering task and a set of reading comprehension exams. First, we show that Q/A systems perform better when there are multiple answer opportunities per question. Next, we analyze common approaches to two subproblems: term overlap for answer sentence identification, and answer typing for short answer extraction. We present general tools for analyzing the strengths and limitations of techniques for these subproblems. Our results quantify the limitations of both term overlap and answer typing to distinguish between competing answer candidates.

1 Introduction

When building a system to perform a task, the most important statistic is the performance on an end-to-end evaluation. For the task of open-domain question answering against text collections, there have been two large-scale end-to-end evaluations: (TREC-8 Proceedings, 1999) and (TREC-9 Proceedings, 2000). In addition, a number of researchers have built systems to take reading comprehension examinations designed to evaluate children's reading levels (Charniak et al., 2000; Hirschman et al., 1999; Ng et al., 2000; Riloff and Thelen, 2000; Wang et al., 2000). The performance statistics have been useful for determining how well techniques work.

However, raw performance statistics are not enough. If the score is low, we need to understand what went wrong and how to fix it. If the score is high, it is important to understand why. For example, performance may be dependent on characteristics of the current test set and would not carry over to a new domain. It would also be useful to know if there is a particular characteristic of the system that is central. If so, then the system can be streamlined and simplified.

In this paper, we explore ways of gaining insight into question answering system performance. First, we analyze the impact of having multiple answer opportunities for a question. We found that TREC-8 Q/A systems performed better on questions that had multiple answer opportunities in the document collection. Second, we present a variety of graphs to visualize and analyze functions for ranking sentences. The graphs revealed that relative score instead of absolute score is paramount. Third, we introduce bounds on functions that use term overlap[2] to rank sentences. Fourth, we compute the expected score of a hypothetical Q/A system that correctly identifies the answer type for a question and correctly identifies all entities of that type in answer sentences. We found that a surprising amount of ambiguity remains because sentences often contain multiple entities of the same type.

[1] This paper contains a revised Table 2 replacing the one appearing in the Proceedings of the Workshop on Open-Domain Question Answering, Toulouse, France 2001.

[2] Throughout the text, we use "overlap" to refer to the intersection of sets of words, most often the words in the question and the words in a sentence.

2 The data

The experiments in Sections 3, 4, and 5 were performed on two question answering data sets: (1) the TREC-8 Question Answering Track data set and (2) the CBC reading comprehension data set. We will briefly describe each of these data sets and their corresponding tasks.

The task of the TREC-8 Question Answering track was to find the answer to 198 questions using a document collection consisting of roughly 500,000 newswire documents. For each question, systems were allowed to return a ranked list of 5 short (either 50-character or 250-character) responses. As a service to track participants, AT&T provided top documents returned by their retrieval engine for each of the TREC questions. Sections 4 and 5 present analyses that use all sentences in the top 10 of these documents. Each sentence is classified as correct or incorrect automatically. This automatic classification judges a sentence to be correct if it contains at least half of the stemmed, content-words in the answer key. We have compared this automatic evaluation to the TREC-8 QA track assessors and found it to agree 93-95% of the time (Breck et al., 2000).

The CBC data set was created for the Johns Hopkins Summer 2000 Workshop on Reading Comprehension. Texts were collected from the Canadian Broadcasting Corporation web page for kids (http://cbc4kids.ca/). They are an average of 24 sentences long. The stories were adapted from newswire texts to be appropriate for adolescent children, and most fall into the following domains: politics, health, education, science, human interest, disaster, sports, business, crime, war, entertainment, and environment. For each CBC story, 8-12 questions and an answer key were generated.[3] We used a 650 question subset of the data and their corresponding 75 stories. The answer candidates for each question in this data set were all sentences in the document. The sentences were scored against the answer key by the automatic method described previously.

[3]This work was performed by Lisa Ferro and Tim Bevins of the MITRE Corporation. Dr. Ferro has professional experience writing questions for reading comprehension exams and led the question writing effort.

3 Analyzing the number of answer opportunities per question

In this section we explore the impact of multiple answer opportunities on end-to-end system performance. A question may have multiple answers for two reasons: (1) there is more than one different answer to the question, and (2) there may be multiple instances of each answer. For example, *"What does the Peugeot company manufacture?"* can be answered by *trucks*, *cars*, or *motors* and each of these answers may occur in many sentences that provide enough context to answer the question. The table insert in Figure 1 shows that, on average, there are 7 answer occurrences per question in the TREC-8 collection.[4] In contrast, there are only 1.25 answer occurrences in a CBC document. The number of answer occurrences varies widely, as illustrated by the standard deviations. The median shows an answer frequency of 3 for TREC and 1 for CBC, which perhaps gives a more realistic sense of the degree of answer frequency for most questions.

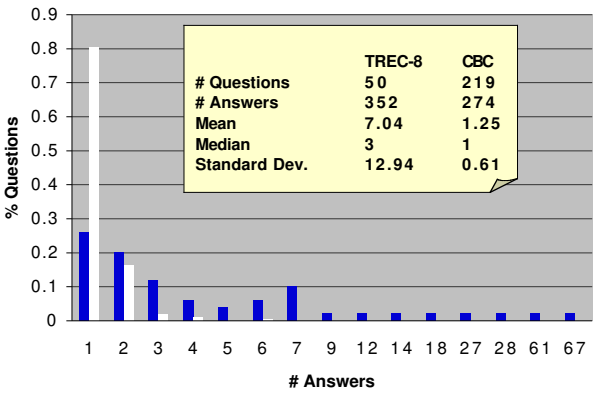

Figure 1: Frequency of answers in the TREC-8 (black bars) and CBC (white bars) data sets

To gather this data we manually reviewed 50 randomly chosen TREC-8 questions and identified all answers to these questions in our text collection. We defined an "answer" as a text fragment that contains the answer string in a context sufficient to answer the question. Figure 1 shows the resulting graph. The x-axis displays the number of answer occurrences found in the text collection per question and the y-axis shows the per-

[4]We would like to thank John Burger and John Aberdeen for help preparing Figure 1.

centage of questions that had x answers. For example, 26% of the TREC-8 questions had only 1 answer occurrence, and 20% of the TREC-8 questions had exactly 2 answer occurrences (the black bars). The most prolific question had 67 answer occurrences (the Peugeot example mentioned above). Figure 1 also shows the analysis of 219 CBC questions. In contrast, 80% of the CBC questions had only 1 answer occurrence in the targeted document, and 16% had exactly 2 answer occurrences.

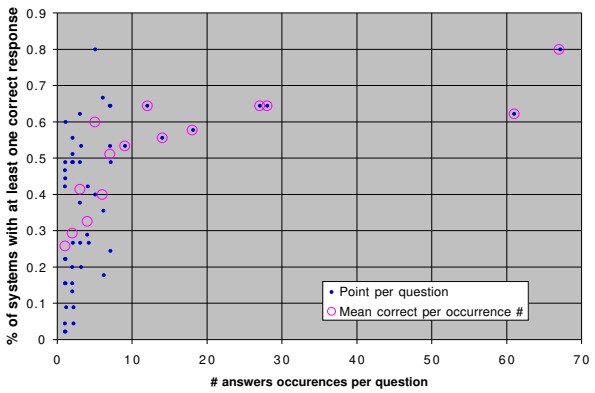

Figure 2: Answer repetition vs. system response correctness for TREC-8

Figure 2 shows the effect that multiple answer opportunities had on the performance of TREC-8 systems. Each solid dot in the scatter plot represents one of the 50 questions we examined.[5] The x-axis shows the number of answer opportunities for the question, and the y-axis represents the percentage of systems that generated a correct answer[6] for the question. E.g., for the question with 67 answer occurrences, 80% of the systems produced a correct answer. In contrast, many questions had a single answer occurrence and the percentage of systems that got those correct varied from about 2% to 60%.

The circles in Figure 2 represent the average percentage of systems that answered questions correctly for all questions with the same number of answer occurrences. For example, on average about 27% of the systems produced a correct answer for questions that had exactly one answer occurrence, but about 50% of the systems produced a correct answer for questions with 7 answer opportunities. Overall, a clear pattern emerges: the performance of TREC-8 systems was strongly correlated with the number of answer opportunities present in the document collection.

[5]We would like to thank Lynette Hirschman for suggesting the analysis behind Figure 2 and John Burger for help with the analysis and presentation.

[6]For this analysis, we say that a system generated a correct answer if a correct answer was in its response set.

4 Graphs for analyzing scoring functions of answer candidates

Most question answering systems generate several answer candidates and rank them by defining a scoring function that maps answer candidates to a range of numbers. In this section, we analyze one particular scoring function: *term overlap* between the question and answer candidate. The techniques we use can be easily applied to other scoring functions as well (e.g., weighted term overlap, partial unification of sentence parses, weighted abduction score, etc.). The answer candidates we consider are the sentences from the documents.

The expected performance of a system that ranks all sentences using term overlap is 35% for the TREC-8 data. This number is an expected score because of ties: correct and incorrect candidates may have the same term overlap score. If ties are broken optimally, the best possible score (*maximum*) would be 54%. If ties are broken maximally suboptimally, the worst possible score (*minimum*) would be 24%. The corresponding scores on the CBC data are 58% expected, 69% maximum, and 51% minimum. We would like to understand why the term overlap scoring function works as well as it does and what can be done to improve it.

Figures 3 and 4 compare correct candidates and incorrect candidates with respect to the scoring function. The x-axis plots the range of the scoring function, i.e., the amount of overlap. The y-axis represents **Pr(overlap=x | correct)** and **Pr(overlap=x | incorrect)**, where separate curves are plotted for correct and incorrect candidates. The probabilities are generated by normalizing the number of correct/incorrect answer candidates with a particular overlap score by the total number of correct/incorrect candidates, respectively.

Figure 3 illustrates that the correct candidates for TREC-8 have term overlap scores distributed between 0 and 10 with a peak of 24% at an over-

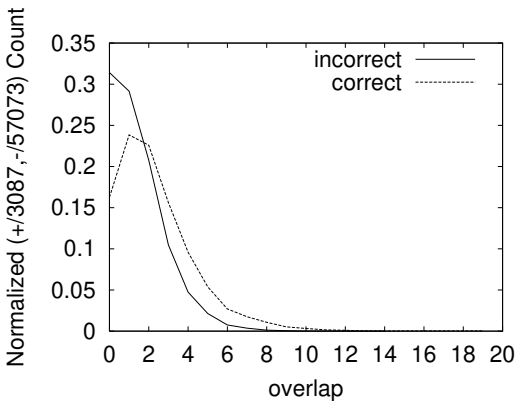

Figure 3: Pr(overlap=x|[in]correct) for TREC-8

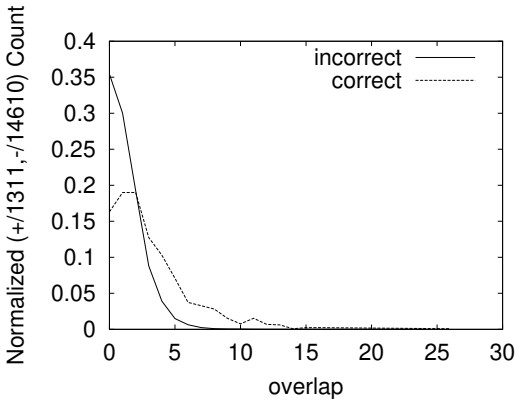

Figure 4: Pr(overlap=x|[in]correct) for CBC

lap of 2. However, the incorrect candidates have a similar distribution between 0 and 8 with a peak of 32% at an overlap of 0. The similarity of the curves illustrates that it is unclear how to use the score to decide if a candidate is correct or not. Certainly no static threshold above which a candidate is deemed correct will work. Yet the expected score of our TREC term overlap system was 35%, which is much higher than a random baseline which would get an expected score of less than 3% because there are over 40 sentences on average in newswire documents.[7]

After inspecting some of the data directly, we posited that it was not the absolute term overlap that was important for judging candidate but how the overlap score compares to the scores of other candidates. To visualize this, we generated new graphs by plotting the rank of a candidate's score on the x-axis. For example, the candidate with the highest score would be ranked first, the candidate with the second highest score would be ranked second, etc. Figures 5 and 6 show these graphs, which display **Pr(rank=x | correct)** and **Pr(rank=x | incorrect)** on the y-axis. The top-ranked candidate has rank=0.

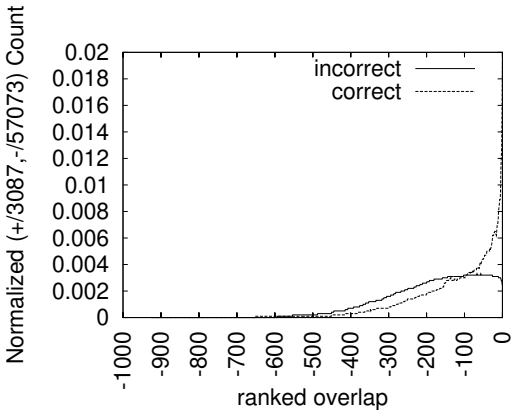

Figure 5: Pr(rank=x | [in]correct) for TREC-8

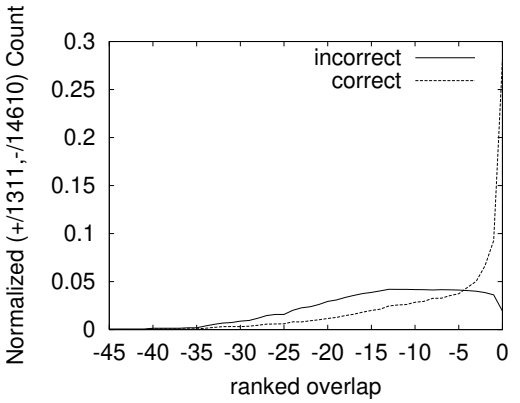

Figure 6: Pr(rank=x | [in]correct) for CBC

The ranked graphs are more revealing than the graphs of absolute scores: the probability of a high rank is greater for correct answers than incorrect ones. Now we can begin to understand why the term overlap scoring function worked as well as it did. We see that, unlike classification tasks, there is no good threshold for our scoring function. Instead relative score is paramount. Systems such as (Ng et al., 2000) make explicit use of relative rank in their algorithms and now we understand why this is effective.

Before we leave the topic of graphing scoring functions, we want to introduce one other view of the data. Figure 7 plots term overlap scores on

[7]We also tried dividing the term overlap score by the length of the question to normalize for query length but did not find that the graph was any more helpful.

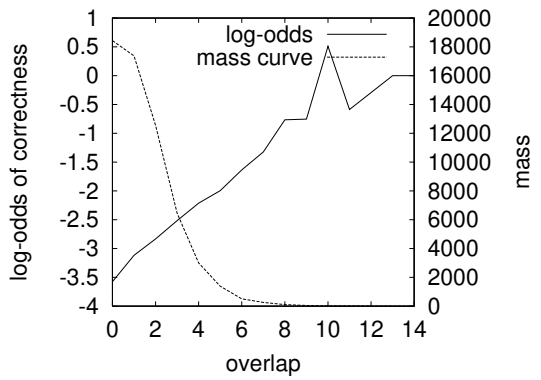

Figure 7: TREC-8 log odds correct given overlap

the x-axis and the log odds of being correct given a score on the y-axis. The log odds formula is:

$$\log \frac{Pr(correct|overlap)}{Pr(incorrect|overlap)}$$

Intuitively, this graph shows how much more likely a sentence is to be correct versus incorrect given a particular score. A second curve, labeled "mass," plots the number of answer candidates with each score. Figure 7 shows that the odds of being correct are negative until an overlap of 10, but the mass curve reveals that few answer candidates have an overlap score greater than 6.

5 Bounds on scoring functions that use term overlap

The scoring function used in the previous section simply counts the number of terms shared by a question and a sentence. One obvious modification is to weight some terms more heavily than others. We tried using inverse document frequence based (IDF) term weighting on the CBC data but found that it did not improve performance. The graph analogous to Figure 6 but with IDF term weighting was virtually identical.

Could another weighting scheme perform better? How well could an optimal weighting scheme do? How poorly would the maximally suboptimal scheme do? The analysis in this section addresses these questions. In essence the answer is the following: the question and the candidate answers are typically short and thus the number of overlapping terms is small – consequently, many candidate answers have exactly the same overlapping terms and no weighting scheme could differentiate them. In addition, subset relations often hold between overlaps. A candidate whose overlap is a subset of a second candidate cannot score higher regardless of the weighting scheme.[8] We formalize these overlap set relations and then calculate statistics based on them for the CBC and TREC data.

Question: How much was Babe Belanger paid to play amateur basketball?

S1: She was a member of the winningest **basketball** team Canada ever had.
S2: **Babe Belanger** never made a cent for her skills.
S3: They were just a group of young women from the same school who liked to **play amateur basketball**.
S4: **Babe Belanger** played with the Grads from 1929 to 1937.
S5: **Babe** never talked about her fabulous career.

MaxOsets : ({S2, S4}, {S3})

Figure 8: Example of Overlap Sets from CBC

Figure 8 presents an example from the CBC data. The four overlap sets are (i) *Babe Belanger*, (ii) *basketball*, (iii) *play amateur basketball*, and (iv) *Babe*. In any term-weighting scheme with positive weights, a sentence containing the words *Babe Belanger* will have a higher score than sentences containing just *Babe*, and sentences with *play amateur basketball* will have a higher score than those with just *basketball*. However, we cannot generalize with respect to the relative scores of sentences containing *Babe Belanger* and those containing *play amateur basketball* because some terms may have higher weights than others.

The most we can say is that the highest scoring candidate must be a member of $\{S2, S4\}$ or $\{S3\}$. S5 and S1 cannot be ranked highest because their overlap sets are a proper subset of competing overlap sets. The correct answer is S2 so an optimal weighting scheme would have a 50% chance of ranking S2 first, assuming that it identified the correct overlap set $\{S2, S4\}$ and then randomly chose between S2 and S4. A maximally suboptimal weighting scheme could rank S2 no lower than third.

We will formalize these concepts using the following variables:

[8] Assuming that all term weights are positive.

q: a question (a set of words)
s: a sentence (a set of words)
w,v: sets of intersecting words

We define an *overlap set* ($o_{w,q}$) to be a set of sentences (answer candidates) that have the same words overlapping with the question. We define a *maximal overlap set* (M_q) as an overlap set that is not a subset of any other overlap set for the question. For simplicity, we will refer to a maximal overlap set as a *MaxOset*.

$$o_{w,q} = \{s | s \cap q = w\}$$

$$\Omega_q = \text{all unique overlap sets for } q$$

$$maximal(o_{w,q}) \text{ if } \forall o_{v,q} \in \Omega_q, w \not\subset v$$

$$M_q = \{o_{w,q} \in \Omega_q \mid maximal(o_{w,q})\}$$

$$C_q = \{s | s \text{ correctly answers } q\}$$

We can use these definitions to give upper and lower bounds on the performance of term-weighting functions on our two data sets. Table 1 shows the results. The max statistic is the percentage of questions for which at least one member of its MaxOsets is correct. The min statistic is the percentage of questions for which all candidates of all of its MaxOsets are correct (i.e., there is no way to pick a wrong answer). Finally the $expectedmax$ is a slightly more realistic upper bound. It is equivalent to randomly choosing among members of the "best" maximal overlap set, i.e., the MaxOset that has the highest percentage of correct members. Formally, the statistics for a set of questions Q are computed as:

$$\text{max} = \frac{|\{q | \exists o \in M_q, \exists s \in o \text{ s.t. } s \in C_q\}|}{|Q|}$$

$$\text{min} = \frac{|\{q | \forall o \in M_q, \forall s \in o \quad s \in C_q\}|}{|Q|}$$

$$\text{exp. max} = \frac{1}{|Q|} * \sum_{q \in Q} \max_{o \in M_q} \frac{|\{s \in o \text{ and } s \in C_q\}|}{|o|}$$

The results for the TREC data are considerably lower than the results for the CBC data. One explanation may be that in the CBC data, only sentences from one document containing the answer are considered. In the TREC data, as in the TREC task, it is not known beforehand which documents contain answers, so irrelevant documents may contain high-scoring sentences that distract from the correct sentences.

	exp. max	max	min
CBC training	72.7%	79.0%	24.4%
TREC-8	48.8%	64.7%	10.1%

Table 1: Maximum overlap analysis of scores

In Table 2, we present a detailed breakdown of the MaxOset results for the CBC data. (Note that the classifications overlap, e.g., questions that are in "there is always a chance to get it right" are also in the class "there may be a chance to get it right.") 21% of the questions are literally impossible to get right using only term weighting because none of the correct sentences are in the MaxOsets. This result illustrates that maximal overlap sets can identify the limitations of a scoring function by recognizing that some candidates will always be ranked higher than others. Although our analysis only considered term overlap as a scoring function, maximal overlap sets could be used to evaluate other scoring functions as well, for example overlap sets based on semantic classes rather than lexical items.

In sum, the upper bound for term weighting schemes is quite low and the lower bound is quite high. These results suggest that methods such as query expansion are essential to increase the feature sets used to score answer candidates. Richer feature sets could distinguish candidates that would otherwise be represented by the same features and therefore would inevitably receive the same score.

6 Analyzing the effect of multiple answer type occurrences in a sentence

In this section, we analyze the problem of extracting short answers from a sentence. Many Q/A systems first decide what answer type a question expects and then identify instances of that type in sentences. A scoring function ranks the possible answers using additional criteria, which may include features of the surrounding sentence such as term overlap with the question.

For our analysis, we will assume that two short answers that have the same answer type and come from the same sentence are indistinguishable to the system. This assumption is made by many

	number of questions	percentage of questions
Impossible to get it wrong ($\forall o_w \in M_q, \forall s \in o_w, s \in C_q$)	159	24%
There is always a chance to get it right ($\forall o_w \in M_q, \exists s \in o_w$ s.t. $s \in C_q$)	204	31%
There may be a chance to get it right ($\exists o_w \in M_q$ s.t. $\exists s \in o_w$ s.t. $s \in C_q$)	514	79%
The wrong answers will always be weighted too highly ($\forall o_w \in M_q, \forall s \in o_w, s \notin C_q$)	137	21%
There are no correct answers with any overlap with Q ($\forall s \in d$, s is incorrect or s has 0 overlap)	66	10%
There are no correct answers (auto scoring error) ($\forall s \in d$, s is incorrect)	12	2%

Table 2: Maximal Overlap Set Analysis for CBC data

Q/A systems: they do not have features that can prefer one entity over another of the same type in the same sentence.

We manually annotated data for 165 TREC-9 questions and 186 CBC questions to indicate perfect question typing, perfect answer sentence identification, and perfect semantic tagging. Using these annotations, we measured how much "answer confusion" remains if an oracle gives you the correct question type, a sentence containing the answer, and correctly tags all entities in the sentence that match the question type. For example, the oracle tells you that the question expects a person, gives you a sentence containing the correct person, and tags all person entities in that sentence. The one thing the oracle does not tell you is *which* person is the correct one.

Table 3 shows the answer types that we used. Most of the types are fairly standard, except for the *Defaultnp* and *Defaultvp* which are default tags for questions that desire a noun phrase or verb phrase but cannot be more precisely typed.

We computed an expected score for this hypothetical system as follows: for each question, we divided the number of correct candidates (usually one) by the total number of candidates of the same answer type in the sentence. For example, if a question expects a *Location* as an answer and the sentence contains three locations, then the expected accuracy of the system would be 1/3 because the system must choose among the locations randomly. When multiple sentences contain a correct answer, we aggregated the sentences. Finally, we averaged this expected accuracy across all questions for each answer type.

	TREC		**CBC**	
Answer Type	*Score*	*Freq*	*Score*	*Freq*
defaultnp	.33	47	.25	28
organization	.50	1	.72	3
length	.50	1	.75	2
thingname	.58	14	.50	1
quantity	.58	13	.77	14
agent	.63	19	.40	23
location	.70	24	.68	29
personname	.72	11	.83	13
city	.73	3	n/a	0
defaultvp	.75	2	.42	15
temporal	.78	16	.75	26
personnoun	.79	7	.53	5
duration	1.0	3	.67	4
province	1.0	2	1.0	2
area	1.0	1	n/a	0
day	1.0	1	n/a	0
title	n/a	0	.50	1
person	n/a	0	.67	3
money	n/a	0	.88	8
ambigbig	n/a	0	.88	4
age	n/a	0	1.0	2
comparison	n/a	0	1.0	1
mass	n/a	0	1.0	1
measure	n/a	0	1.0	1
Overall	.59	165	.61	186
Overall-dflts	.69	116	.70	143

Table 3: Expected scores and frequencies for each answer type

Table 3 shows that a system with perfect question typing, perfect answer sentence identification, and perfect semantic tagging would still achieve only 59% accuracy on the TREC-9 data. These results reveal that there are often multiple candidates of the same type in a sentence. For example, *Temporal* questions received an expected score of 78% because there was usually only one date expression per sentence (the correct one), while *Default NP* questions yielded an ex-

pected score of 25% because there were four noun phrases per question on average. Some common types were particularly problematic. *Agent* questions (most *Who* questions) had an answer confusability of 0.63, while *Quantity* questions had a confusability of 0.58.

The CBC data showed a similar level of answer confusion, with an expected score of 61%, although the confusability of individual answer types varied from TREC. For example, *Agent* questions were even more difficult, receiving a score of 40%, but *Quantity* questions were easier receiving a score of 77%.

Perhaps a better question analyzer could assign more specific types to the *Default NP* and *Default VP* questions, which skew the results. The **Overall-dflts** row of Table 3 shows the expected scores without these types, which is still about 70% so a great deal of answer confusion remains even without those questions. The confusability analysis provides insight into the limitations of the answer type set, and may be useful for comparing the effectiveness of different answer type sets (somewhat analogous to the use of grammar perplexity in speech research).

Q1: *What city is Massachusetts General Hospital located in?*
A1: It was conducted by a cooperative group of oncologists from Hoag, Massachusetts General Hospital in **Boston**, Dartmouth College in New Hampshire, UC San Diego Medical Center, McGill University in Montreal and the University of Missouri in Columbia.

Q2: *When was Nostradamus born?*
A2: Mosley said followers of Nostradamus, who lived from **1503** to 1566, have claimed ...

Figure 9: Sentences with Multiple Items of the Same Type

However, Figure 9 shows the fundamental problem behind answer confusability. Many sentences contain multiple instances of the same type, such as lists and ranges. In Q1, recognizing that the question expects a city rather than a general location is still not enough because several cities are in the answer sentence. To achieve better performance, Q/A systems need use features that can more precisely target an answer.

7 Conclusion

In this paper we have presented four analyses of question answering system performance involving: multiple answer occurence, relative score for candidate ranking, bounds on term overlap performance, and limitations of answer typing for short answer extraction. We hope that both the results *and* the tools we describe will be useful to others. In general, we feel that analysis of good performance is nearly as important as the performance itself and that the analysis of bad performance can be equally important.

References

E.J. Breck, J.D. Burger, L. Ferro, L. Hirschman, D. House, M. Light, and I. Mani. 2000. How to Evaluate your Question Answering System Every Day and Still Get Real Work Done. In *Proceedings of the Second Conference on Language Resources and Evaluation (LREC-2000)*.

E. Charniak, Y. Altun, R. de Salvo Braz, B. Garrett, M. Kosmala, T. Moscovich, L. Pang, C. Pyo, Y. Sun, W. Wy, Z. Yang, S. Zeller, and L. Zorn. 2000. Reading Comprehension Programs in a Statistical-Language-Processing Class. In *ANLP/NAACL Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.

L. Hirschman, M. Light, E. Breck, and J. Burger. 1999. Deep Read: A Reading Comprehension System. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

H.T. Ng, L.H. Teo, and J.L.P. Kwan. 2000. A Machine Learning Approach to Answering Questions for Reading Comprehension Tests. In *Proceedings of EMNLP/VLC-2000 at ACL-2000*.

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

TREC-8 Proceedings. 1999. *Proceedings of the Eighth Text Retrieval Conference (TREC8)*. National Institute of Standards and Technology, Special Publication 500-246, Gaithersburg, MD.

TREC-9 Proceedings. 2000. *Proceedings of the Ninth Text Retrieval Conference (forthcoming)*. National Institute of Standards and Technology, Special Publication 500-XXX, Gaithersburg, MD.

W. Wang, Auer J., R. Parasuraman, I. Zubarev, D. Brandyberry, and M.P. Harper. 2000. A Question Answering System Developed as a Project in a Natural Language Processing Course. In *ANLP/NAACL Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems*.