Natural Language Engineering **1** (1): 000–000. Printed in the United Kingdom
© 1998 Cambridge University Press

Analyses for Elucidating Current Question Answering Technology

Marc Light
The MITRE Corporation,
202 Burlington Rd.,Bedford, MA 01730, light@mitre.org

Gideon S. Mann
Department of Computer Science, Johns Hopkins University,
Baltimore, MD 21218, gsm@cs.jhu.edu

Ellen Riloff
School of Computing, University of Utah,
Salt Lake City, UT 84112, riloff@cs.utah.edu

Eric Breck
Department of Computer Science, Cornell University,
4161 Upson Hall, Ithaca, NY 14853, ebreck@cs.cornell.edu

(*Received 19 September 2001*)

Abstract

In this paper, we take a detailed look at the performance of components of an idealized question answering system on two different tasks: the TREC Question Answering task and a set of reading comprehension exams. We carry out three types of analysis: inherent properties of the data, feature analysis, and performance bounds. Based on these analyses we explain some of the performance results of the current generation of Q/A systems and make predictions on future work. In particular, we present four findings: (1) Q/A system performance is correlated with answer repetition, (2) relative overlap scores are more effective than absolute overlap scores, (3) equivalence classes on scoring functions can be used to quantify performance bounds, and (4) perfect answer typing still leaves a great deal of ambiguity for a Q/A system because sentences often contain several items of the same type.

1 Introduction

When building a complex system to perform a task, the most important evaluation is on the end-to-end task. 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

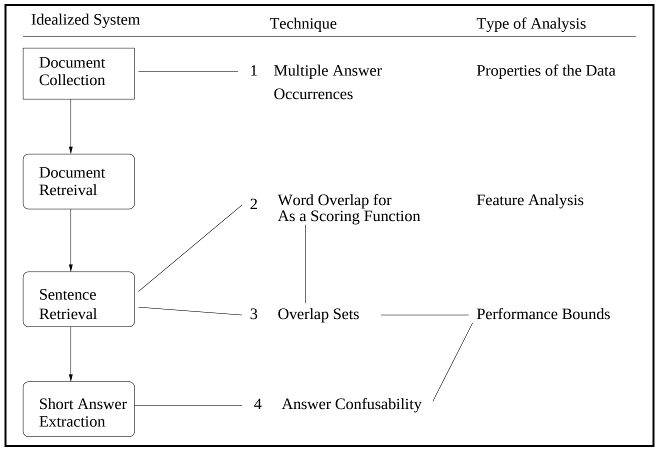

Fig. 1. Components and analyses

et al. 2000; Riloff and Thelen 2000; Wang et al. 2000). The performance statistics have been useful for determining how well techniques work.

However, while these statistics are vital, they conflate and obscure the performance of the individual components of the system and the difficulty of the task. If the score is low, we need to understand what went wrong and how to fix it. If the score is high, it is still important to understand why. With such understanding, one can hope to:

- improve the system performance,
- simplify the system (if one particular characteristic of the system is responsible for good performance and the other features are parasitic),
- predict how a system will perform on different types of questions and/or different document collections,
- satisfy scientific curiosity.

In this paper we consider an idealized Q/A system that has the system diagram shown on the left side of Figure 1. We present techniques for performing three types of analysis: inherent properties of the data, feature analysis, and performance bounds. Figure 1 shows how the four techniques we present correspond to system tasks and types of analysis. We apply these techniques to specific Q/A approaches that are currently prevalent. In many cases, the techniques are applicable to other approaches with little or no modification. Even when not directly applicable, we hope these techniques will inspire further research on analytical methods.

We first analyze the impact of having multiple answer occurrences for a question. In other words, the document collection contains the answer in multiple sentences and perhaps multiple documents. We found that TREC-8 Q/A systems performed better on questions that had multiple answer occurrences in the document collection. This suggests that redundancy in the data is important. Redundancy in a collection of documents is predictive of Q/A system performance on that collection.

Second, we analyze scoring functions that are used to retrieve regions of text likely to contain an answer. We focus on sentence retrieval. For example, a scoring function might assign a number to a sentence based on the number of words the sentence has in common with a question (word overlap[1]). These numbers can then be used to rank the sentences. Our analysis focuses on whether the word overlap scoring function can effectively differentiate sentences that contain an answer from those that do not. Our results show that the absolute value of an overlap score is not very meaningful, but that the *relative* value of overlap scores is valuable (i.e., all that matters is that a sentence has a score higher than competing sentences). A consequence of this result is that word overlap should not be expected to work well in Q/A scenarios where the text collection may not contain any correct answers to a question.

Third, we consider the question: if a system assigns different weights to words in the overlap, how well can it perform? We calculate upper and lower bounds on functions that use word overlap to rank sentences. To perform this analysis, we introduce the notion of an *Overlap Set* which represents an equivalence class of sentences that cannot be distinguished by the scoring function. The lower bound represents an important baseline: the percentage of questions that a system is guaranteed to answer correctly, no matter what term weights are used. The upper bound reveals the maximum performance possible if term weights are assigned optimally and ties are broken optimally. Our lower bound results show that 10-24% of questions are guaranteed to be answered correctly using word overlap as a scoring function, which is a surprisingly high baseline. On the other hand, our upper bound results show that only 65-79% of questions will be answered correctly even in the best possible circumstances. Put another way, 21-35% of questions are **impossible** to answer correctly using term overlap as a scoring function.

Finally, we look at short answer extraction, i.e., returning the exact answer as opposed to a text region containing the answer. Many systems extract a short answer from a region by looking for a specific entity type based on the question. For example, a system might look for an answer of type *Person* when processing "Who was Johnny Mathis' track coach?" Given a set of possible answer types, we analyze the ability of the answer type set to discriminate between different answers. We compute the expected score given that the tasks that precede short answer extraction are performed correctly: correct identification of the answer type for a question, correct identification of all entities of that type in answer sentences, and optimal sentence retrieval. We found that a surprising amount of ambiguity remains because sentences often contain multiple entities of the same type. For example, a sentence containing the answer to the previous question contains two person names other than "Johnny Mathis." Thus, we conjecture that grammatical or structural relations are needed to achieve high performance on short answer extraction.

[1] 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. Note: the words are *stemmed* and stop words are retained. For many tasks this has little effect as shown in (Hirschman et al. 1999)

Mars Polar Lander - Where Are You?

(January 18, 2000) After more than a month of searching for a signal from NASA's Mars Polar Lander, mission controllers have lost hope of finding it. The Mars Polar Lander was on a mission to Mars to study its atmosphere and search for water, something that could help scientists determine whether life ever existed on Mars. Polar Lander was to have touched down December 3 for a 90-day mission. It was to land near Mars' south pole. The lander was last heard from minutes before beginning its descent. The last effort to communicate with the three-legged lander ended with frustration at 8 a.m Monday. "We didn't see anything," said Richard Cook, the spacecraft's project manager at NASA's Jet Propulsion Laboratory. The failed mission to the Red Planet cost the American government more than $200 million dollars. Now, space agency scientists and engineers will try to find out what could have gone wrong. They do not want to make the same mistakes in the next mission.

- When did the mission controllers lose hope of communicating with the lander?
  (Answer: 8AM, Monday Jan. 17)
- Who is the Polar Lander's project manager?
  (Answer: Richard Cook)
- Where on Mars was the spacecraft supposed to touch down?
  (Answer: near Mars' south pole)
- What was the mission of the Mars Polar Lander?
  (Answer: to study Mars' atmosphere and search for water)

Fig. 2. Sample CBC test exam

Table 1. *Corpus statistics*

	# docs	# q/doc	#q (total)
TREC-8	500,000	N/A	198
CBC	259	≈ 9	2296

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.

2.1 TREC Question Answering Track

The task of the TREC-8 Question Answering Track was to find the answers to 198 questions using a document collection consisting of roughly 500,000 newswire documents. The questions were back-generated by participants from answers they found in the collection. These back-generated questions were then collected and sent out by NIST. For each question, systems were allowed to return a ranked list of 5 short (either 50-character or 250-character) responses. Documents supporting the character strings as answers were also a required part of each response. TREC-8 Question Answering Track assessors then judged each response as correct or incorrect in the context of the document provided. The analysis in Section 3 makes use of the documents from which a correct answer was extracted. Section 6 also makes use of such documents but from the TREC-9 Question Answering Track. The TREC-9 Question Answering evaluation was very similar to TREC-8 with the notable improvement that the questions were not back-generated but created independent of the documents.

As a service to track participants, AT&T provided top documents returned by their retrieval engine for each of the TREC questions. In Sections 4 and 5, our analyses use all sentences in the top 10 of these documents. We classified each sentence as correct or incorrect automatically. Our scoring program judged a sentence to be correct if it contained at least half of the stemmed, content-words in an answer key.[2] We have compared this automatic scoring method with the manual judgments of the TREC-8 Question Answering track assessors and found it to agree 93-95% of the time (Breck et al. 2000).

2.2 CBC Reading Comprehension Data Set

The texts for these reading comprehension tests were collected from the Canadian Broadcasting Corporation web page for kids (http://cbc4kids.ca/). The CBC has been publishing five current-event stories a week for over two years. They seem to be aimed at elementary and middle school students (eight to thirteen year olds). On average, they contain 450 words, 24 sentences, and have a Flesch Reading Ease score (Flesch 1943) of 80. The higher the number, the more people who can read it. For comparison, 91.2 is the score for the Remedia 5W's exams (Hirschman et al. 1999) and 43.9 for AP Newswire.[3] The stories are often based on newswire articles and mostly fall into the following domains: politics, health, education, science, human interest, disaster, sports, business, crime, war, entertainment, environment (in descending order of frequency).

We compiled 259 CBC stories and asked two people to create 8-12 questions and an answer key for each story.[4] See Figure 2 for an example story with corresponding questions. This data set is freely available for others to use. In some cases, the answer key allows for several acceptable answers. For example, varying levels of granularity (e.g., "Toronto, Ontario" vs. "Toronto"), varying amounts of information (e.g., "he died" vs. "he died in his sleep of natural causes"), paraphrases (e.g., "Human Immunodeficiency Virus" vs. "HIV"), or occasionally different interpretations of

[2] This answer key was prepared by Lisa Ferro without knowledge of the design of the experiments described in this paper.

[3] Lisa Ferro performed these calculations.

[4] This work was performed by Lisa Ferro and Tim Bevins of The MITRE Corporation. Neither was directly involved in the experiments described in this paper. Lisa Ferro has professional experience writing questions for reading comprehension exams and she led the question writing effort.

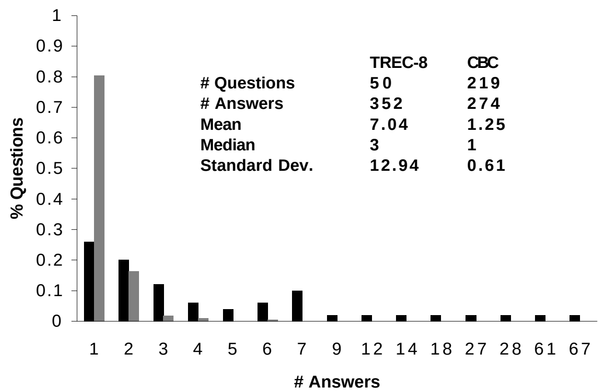

Fig. 3. Frequency of answers in the TREC-8 (black bars) and CBC (grey bars) data sets

the question (e.g., Where did the boys learn how to survive a storm? “camping tips from a friend” vs. “their backyard”).

3 Analyzing the number of answer occurrences per question

In this section we present a study of an inherent property of the data. We explore the impact of multiple answer occurrences 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.

We hypothesized that Q/A systems perform better on questions that have many answer occurrences than on questions that have fewer answer occurrences. We investigated this hypothesis empirically by examining both the TREC-8 Q/A task and the CBC data set for multiple answer occurrences. We manually reviewed 50 randomly chosen TREC-8 questions and identified all answer occurrences to these questions in the documents judged to contain correct answers by the TREC assessors. We defined an “answer” as a text fragment that contains the answer string in a context sufficient to answer the question. We performed a similar analysis of 219 questions in the CBC development set. It should be noted that for any given TREC question, the number of documents collected as described above is a lower bound on the number of documents containing an answer since other such answer documents may have been overlooked by the systems that competed in TREC-8.

Figure 3 shows that, on average, there are 7 answer occurrences per question in the TREC-8 collection. In contrast, there are only 1.25 answer occurrences in a CBC document. The number of answer occurrences varies widely. 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 3 shows the percentage of questions having each exact number of answer occurrences. The x-axis represents the number of answer occurrences found in the

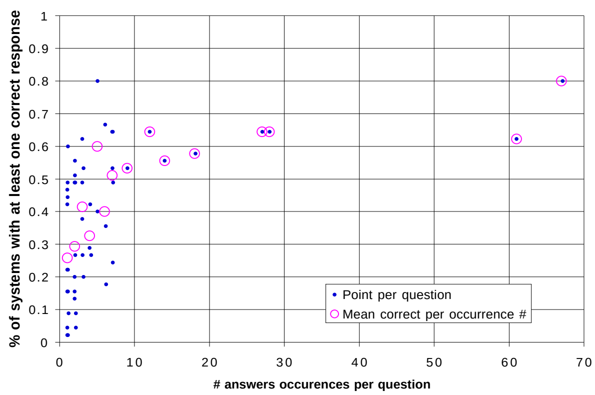

Fig. 4. Answer repetition vs. system response correctness for TREC-8

text collection and the y-axis shows the percentage of questions that had x answers. For example, 26% of the TREC-8 questions had only 1 answer occurrence in the text collection, while 80% of the CBC questions had exactly 1 answer occurrence in the targeted document. The most prolific TREC question had 67 answer occurrences (the Peugeot example mentioned previously), while the most prolific CBC question had 6 answer occurrences.

Figure 4 shows the effect that multiple answer occurrences had on the performance of TREC-8 systems. Each solid dot in the scatter plot represents one of the 50 questions we examined. The x-axis shows the number of answer occurrences a question had, and the y-axis represents the percentage of systems that generated a correct answer[5] for the question. For example, 80% of the systems produced a correct answer for the question with 67 answer occurrences. In contrast, many questions had exactly one answer occurrence and system performance varied widely on these questions: 2%-60% of systems got these questions correct.

Each circle in Figure 4 represents the average percentage of systems that correctly answered all questions with x answer occurrences. For example, on average about 27% of the systems produced a correct answer for questions with exactly one answer occurrence, while about 50% of the systems produced a correct answer for questions with 7 answer occurrences. Overall, a clear pattern emerges: the performance of TREC-8 systems was strongly correlated with the number of answer occurrences present in the document collection.

One way to use this result is to help predict the performance of a Q/A system on a new set of questions and/or documents: a high average number of answer occurrences bodes well for system performance.

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

4 Analyzing scoring functions of answer candidates

Many 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: *word overlap* between the question and answer candidate. The answer candidates we consider are the sentences from the documents. The techniques we use can be easily applied to other scoring functions as well such as weighted word overlap, partial unification of sentence parses, weighted abduction score, etc.

Word overlap is an important scoring function because systems based on it do surprisingly well at ranking at least one answer highly. For example, if one starts with the top 10 documents from the AT&T search engine and ranks each sentence by the number of words that overlap with the question, the expected performance is 35% for the TREC-8 data. This number is an expected score because of ties: correct and incorrect candidates may have the same word overlap score. If ties are broken optimally, the best possible score (*maximum*) would be 54%. If ties are broken pessimally (maximally suboptimally), the worst possible score (*minimum*) would be 24%. The expected performance is not necessarily the mean of the best and worst possible scores, since the number of sentences with the highest word overlap varies significantly. Since the expected performance (35%) is less than the mean (39%), that indicates that the number of incorrect answers is slightly greater on average than the number of correct answers. The random baseline is an expected score of less than 0.25% , since there are over 40 sentences on average in newswire documents. The corresponding scores on the CBC data are 58% expected, 69% maximum, and 51% minimum with a random baseline of 4%. We would like to understand why the word overlap scoring function works as well as it does and what can be done to improve it. Again, other scoring functions can be analyzed in a similar fashion.

Figures 5 and 6 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 calculated as:

$$\mathrm{Pr}(\mathrm{overlap} = x|\mathrm{correct}) = \frac{c(\mathrm{overlap} = x, \mathrm{correct})}{c(\mathrm{correct})}$$

where c is a count function. Probability functions for incorrect answers are computed in a similar manner.

Figure 5 illustrates that the correct candidates for TREC-8 have word overlap scores distributed between 0 and 10 with a peak of 24% at an overlap 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. For example, if the graph had produced curves showing that the probability of an overlap score $\geq X$ was high for correct sentences but low for incorrect sentences, then we could set a threshold at X to identify the correct candidates. Figures 5 and 6 show that no

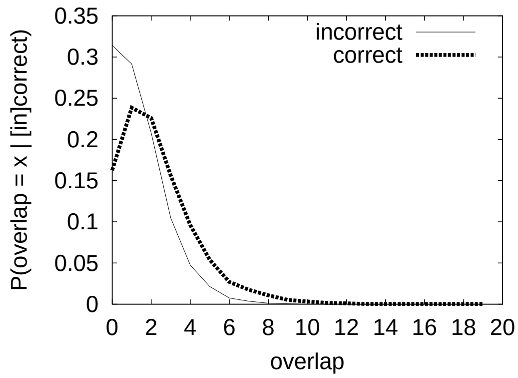

Fig. 5. Pr(overlap=x|[in]correct) for TREC-8

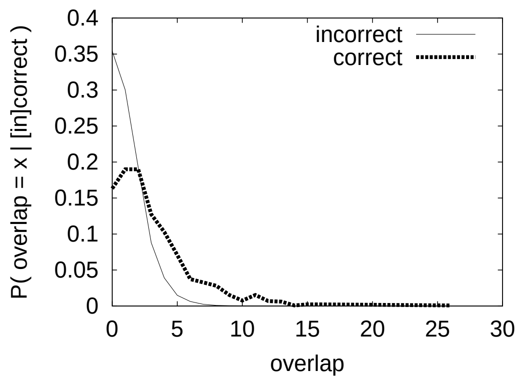

Fig. 6. Pr(overlap=x|[in]correct) for CBC

such threshold exists for word overlap scores.[6] Both correct and incorrect sentences often have low overlap scores, and high overlap scores are relatively rare but present in both groups.

Yet the expected score of our TREC word overlap system was 35%, much higher than the random baseline. After inspecting some of the data directly, we posited that it is not the absolute word overlap that was important for judging candidates 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 7 and 8 show these graphs, which display **Pr(rank=x | correct)** and **Pr(rank=x | incorrect)** on the y-axis. The top-ranked candidate has rank 1.

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

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

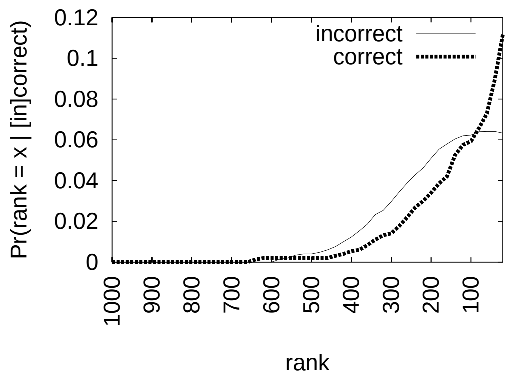

Fig. 7. Pr(rank=x | [in]correct) for TREC-8

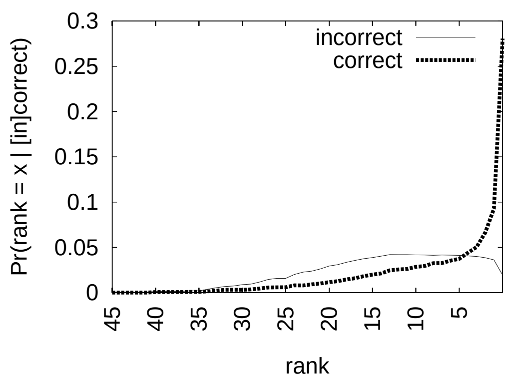

Fig. 8. Pr(rank=x | [in]correct) for CBC

we can begin to understand why the word 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.

An interesting observation based on this analysis is that systems that use word overlap may have difficulty judging if an answer to a question exists in the document collection. If word overlap scores are only useful for ranking candidates, how can we judge the absolute quality of a candidate? This problem does not arise in the CBC data since each question has an answer in its corresponding document, and it was not a factor in the TREC Q/A tasks because questions in TREC-8 and TREC-9 were guaranteed to have answers in the document collection. However, this problem must be addressed if we expect Q/A systems to operate in real scenarios where questions may be posed that do not have answers in the targeted collection.

Before we leave the topic of analyzing scoring functions, we want to introduce one other view of the data. Figure 9 plots word overlap scores on the x-axis and

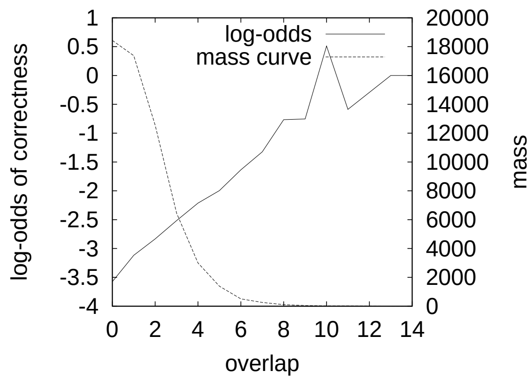

Fig. 9. TREC-8 log odds correct given overlap

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

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

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 9 shows that the log 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 word overlap

The scoring function used in the previous section simply counts the number of words shared by a question and a sentence. One obvious modification is to weight some words more heavily than others. We tried using inverse document frequency based (IDF) word weighting on the CBC data but found that it did not improve performance. The graph analogous to Figure 8 but with IDF word weighting was virtually identical.

Could another weighting scheme perform better? How well could an optimal weighting scheme do? How poorly would the pessimal scheme do? The analysis in this section addresses these questions. First, we make the observation that many candidate answers have exactly the same set of words overlapping with the question (e.g., they both share words w_1 and w_2 in common with the question). We can put these candidates in an equivalence class, since they will be assigned exactly the same score no matter what word weights are used. Many candidates often belong to the same equivalence class because questions and candidate answers are typically short, limiting the number of words they can have in common. In addition, subset relations often hold between overlap sets — a candidate whose overlap is a subset of a second candidate cannot receive a higher score, regardless of the weighting scheme.[7] We formalize these relations among sentences based on the words in their

[7] Assuming that all word weights are positive.

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.

Maximum Overlap Sets: ({S2, S4}, {S3})

Fig. 10. Example of Overlap Sets from CBC

overlap sets and then calculate statistics for the CBC and TREC data based on these overlap sets.

We now introduce the notion of an *overlap set* which contains sentences as elements. Figure 10 presents an example from the CBC data. The four overlap sets are

- $\{S1\}$ based on the word "basketball,"
- $\{S2, S4\}$ based on the words "Babe" and "Belanger,"
- $\{S3\}$ based on the words "play," "amateur," and "basketball,"
- $\{S5\}$ based on the word "Babe."

In any word weighting scheme, a sentence containing the words "Babe Belanger" {S2, S4} will have a higher score than sentences containing just "Babe" {S5}, and sentences with "play amateur basketball" {S3} will have a higher score than those with just "basketball" {S1}. However, we cannot generalize with respect to the relative scores of sentences containing "Babe Belanger" and those containing "play amateur basketball" because some words 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 overlapping words are a subset of the overlapping words of competing overlap sets. The correct answer is S2. An optimal weighting scheme has a 50% chance of ranking S2 first if it correctly selects the set $\{S2, S4\}$ (by weighting "Babe Belanger" higher than "play amateur basketball") and then randomly chooses between S2 and S4. A pessimal weighting scheme could rank S2 no lower than third.

We will formalize these concepts using the following variables:

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. We will refer to a maximal overlap set as a *MaxOset*.

Table 2. *Maximum overlap analysis of scores*

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

$o_{w,q} = \{s | s \cap q = w\}$
$\Omega_q =$ all unique overlap sets for q
$maximal(o_{w,q})$ if $\forall o_{v,q} \in \Omega_q, w \not\subset v$
$M_q = \{o_{w,q} \in \Omega_q \ \ | \ maximal(o_{w,q})\}$
$C_q = \{s | s$ correctly answers $q\}$

We can use these definitions to give upper and lower bounds on the performance of word weighting functions on our two data sets. Table 2 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 *expected max* 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. The *expected max* statistic captures the hope that a good word weighting scheme could identify the best MaxOset, but choosing among its members will necessarily be random (since they all have exactly the same overlapping words). Formally, the statistics for a set of questions Q are computed as:

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

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

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

Table 2 displays the results for these statistics on both the TREC-8 and CBC data sets. 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.

The *max* results show that high performance is possible using word overlap as a scoring function: 79% of CBC questions and 65% of TREC-8 questions can be answered correctly. However, these same numbers can be turned around to reveal an inherent limitation of word overlap: 21% of CBC questions and 35% of TREC-8 questions are *impossible* to answer correctly, even when making perfect choices.

Table 3. *Maximal Overlap Set analysis for CBC data*

	number of questions	percentage of questions
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%
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%
Impossible to get it wrong ($\forall o_w \in M_q, \forall s \in o_w, s \in C_q$)	159	24%
There is no chance to get it right ($\forall o_w \in M_q, \forall s \in o_w, s \notin C_q$)	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%

This result illustrates the benefit of using the MaxOset formalism: MaxOsets allow us to identify the answer candidates that are impossible to find because they will always be ranked lower than incorrect candidates, no matter what weighting scheme is used.

Table 2 also shows the *min* and *expected max* results. The lower bound is 24% for the CBC data and 10% for the TREC-8 data, which tells us the percentage of questions that are trivially easy to answer using the word overlap scoring function (i.e., they will always be ranked higher than incorrect candidates). The *expected max* results are much higher for CBC than TREC-8, suggesting that a good term weighting scheme can produce good performance on the CBC data but that substantial random tie-breaking will still be necessary on the TREC-8 data.

In Table 3, 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 weighted word overlap 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 word 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 on performance for sentence detection using word weighting schemes is quite low and the lower performance 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 word 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 Q/A systems: they do not have features that can prefer one entity over another of the same type in the same sentence (with the notable exception of (Harabagiu et al. 2000)).

We manually annotated data for 165 TREC-9 questions and 186 CBC questions with perfect question typing, perfect answer sentence identification, and perfect semantic tagging. Using these annotations, we measured the "answer confusability": the expected score 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 4 shows the answer types that we used. Most of the types are fairly standard, except for the *Default NP* and *Default VP* which are default tags for questions that desire a noun phrase or verb phrase but cannot be more precisely typed.

We computed the answer confusability 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 (only one of which is correct), 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 score across all questions for each answer type.

Table 4 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 expected score of 25% because there were four noun phrases per sentence on average. Some common types were particularly problematic. *Agent* questions (most *Who* questions) had an expected score of 0.63, while *Quantity* questions had an expected score of 0.58.

Table 4. *Expected scores and frequencies for each answer type*

	TREC		**CBC**	
Answer Type	*Score*	*Freq*	*Score*	*Freq*
defaultNP	0.33	47	0.25	28
organization	0.50	1	0.72	3
length	0.50	1	0.75	2
thingName	0.58	14	0.50	1
quantity	0.58	13	0.77	14
agent	0.63	19	0.40	23
location	0.70	24	0.68	29
personName	0.72	11	0.83	13
city	0.73	3	n/a	0
defaultVP	0.75	2	0.42	15
temporal	0.78	16	0.75	26
personNoun	0.79	7	0.53	5
duration	1.00	3	0.67	4
province	1.00	2	1.00	2
area	1.00	1	n/a	0
day	1.00	1	n/a	0
title	n/a	0	0.50	1
person	n/a	0	0.67	3
money	n/a	0	0.88	8
ambigSize	n/a	0	0.88	4
age	n/a	0	1.00	2
comparison	n/a	0	1.00	1
mass	n/a	0	1.00	1
measure	n/a	0	1.00	1
Overall	0.59	165	0.61	186
Overall w/o Defaults	0.69	116	0.70	143

The CBC data showed a similar level of answer confusion, with an expected score of 61%, although the answer confusability of particular 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 w/o Defaults** row of Table 4 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 answer 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.

Figure 11 shows the fundamental problem behind answer confusability. Many sentences contain multiple instances of the same type, such as lists and ranges. For example, dates are often mentioned in pairs, such as “Fred Smith lived from 1823 to 1897.” Question Q2 is clearly asking for a city, but that still only narrows down the options to five: Boston, Dartmouth, San Diego, Montreal, and Columbia.

To achieve better performance, Q/A systems need to use features that can more precisely pinpoint an answer, e.g., grammatical or semantic relations.

Q1: *When was Fred Smith born?*
S1: Fred Smith lived from **1823** to 1897.

Q2: *What city is Massachusetts General Hospital located in?*
S2: 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.

Fig. 11. A sentence with multiple items of the same type

7 Conclusion

In this paper we have demonstrated the utility of analyzing the subcomponents of a complex system, as a complement to end-to-end evaluation. As a means to performing this analysis, we developed four new evaluation tools. We looked at multiple answer occurrences in the data and found that they are strongly tied to system performance. We analyzed word overlap for sentence identification and showed that relative overlap is more effective than absolute overlap. We further investigated word overlap and introduced the notion of an *overlap set*. This tool allowed us to give tight bounds on the performance of sentence detection using word overlap with differing weighting schemes. Finally, we tested the performance of the answer type set in isolation and suggested that using answer types alone may not be sufficient: some kind of structural information must also be applied.

These tools present examples of the kinds of analyses we feel are relevant. Performance bounds, feature analysis, and data analysis are general techniques that have been applied to other complicated tasks and can be applied to question answering systems as well. Any systems that use a scoring function to rank answers can do the types of analyses presented in Section 5. The notion of using equivalence classes for estimating performance bounds is important for understanding the limits of a tagging scheme. Data analysis is useful for predicting performance on untested domains.

Further work could include ablation experiments, where one component or subcomponent is removed. In addition, we have only examined non-statistical discriminative processes. How would these kinds of analyses extend to purely statistical systems? Finally, we have demonstrated that answer confusability is useful for assessing the performance of the current tag set. This measure is also useful for comparing tag sets, in order to understand which tag set results in the smallest answer confusability.

8 Acknowledgements

We would like to thank John Burger and John Aberdeen for help preparing Figure 3, Lynette Hirschman for many helpful comments and for suggesting the analysis behind Figure 4, and John Burger again for help with Figure 4's analysis and presentation. We also thank Pranav Anand, Brianne Brown, Mats Rooth, and Michael Thelen for help generating some of the data used in Sections 3 and 6. Finally, this work was initiated while the authors were at the 2000 NSF Summer Workshop at the Center for Language and Speech Processing, Johns Hopkins University and we would like to thank the staff and sponsors for the opportunity to participate.

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.*

R. Flesch. 1943. *Marks of Readable Writing.* Ph.D. thesis.

S.M. Harabagiu, M.A. Pasca, and S.J. Maiorano. 2000. Experiments with Open-Domain textual Question Answering. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING 2000).*

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.*