Question Answering : CNLP at the TREC-9 Question Answering Track

Anne Diekema, Xiaoyong Liu, Jiangping Chen, Hudong Wang, Nancy McCracken, Ozgur Yilmazel, and Elizabeth D. Liddy

Center for Natural Language Processing
Syracuse University, School of Information Studies
4-206 Center for Science and Technology
Syracuse, NY 1324-4100
{diekemar,xliu03,jchen06,hwang07,njm,oyilmaz,liddy}@syr.edu

Abstract
This paper describes a question answering system that automatically finds answers to questions in a large collection of documents. The prototype CNLP question answering system was developed for participation in the TREC-9 question answering track. The system uses a two-stage retrieval approach to answer finding based on keyword and named entity matching. Results indicate that the system ranks correct answers high (mostly rank 1), provided that an answer to the question was found. Performance figures and further analyses are included.

1. Introduction

Question answering is not typically found in traditional information retrieval systems. In information retrieval, the system presents the user with a list of relevant documents in response to the query. The user then reviews these documents in search of the information that prompted the original search. It is not surprising therefore that, especially for short questions, people tend to ask their peers or forego the answer rather than expending time and effort with an information retrieval system. [3] Ideally, question answering helps users in their information finding task by providing exact answers rather than a ranked list of documents that may contain the answer.

The TREC question-answering track fosters question-answering research. Question-answering systems are not as well developed as information retrieval systems, especially for domain independent questions. As first-time participants, the Center for Natural Language Processing (CNLP) developed a question- answering system to deal with domain independent questions.

The CNLP question answering system uses a two-stage retrieval approach to answer-finding based on keyword, entity, and template matching (see figure 1). In answering a question, the system first creates a logical query representation of the question that is used for the initial information retrieval step. Additional modules take the retrieved documents for further processing and answer finding. Answer finding uses two different approaches after which answer triangulation takes place to select the most likely answer. The first approach to answer finding is based on keyword and entity matching and the second on template matching. Currently only the keyword and entity matching answer-finding approach has been implemented. A detailed system overview can be found in section 3.

2. Problem description

Participants in the question-answering track were provided with 693 questions that originated from search engine logs. The initial question set of 693 questions was reduced to 682 questions after 11 questions were discarded by the National Institute for Standards and Technology (NIST). The remaining questions were mostly fact-based and required short answers only (see figure 2). The base set of questions consisted of 500 questions. For 54 questions, slight variations were

created resulting in an additional 193 questions. Answers to all 693 questions had to be retrieved automatically from approximately 3 gigabytes of data. Sources of the data were: AP newswire 1988-1990 (728 Mb), Wall Street Journal 1987-1992 (509 Mb), San Jose Mercury News 1991 (287 Mb), Financial Times 1991-1994 (564 Mb), Los Angeles Times 1989, 1990 (475 Mb), Foreign Broadcast Information Service 1996 (470 Mb).

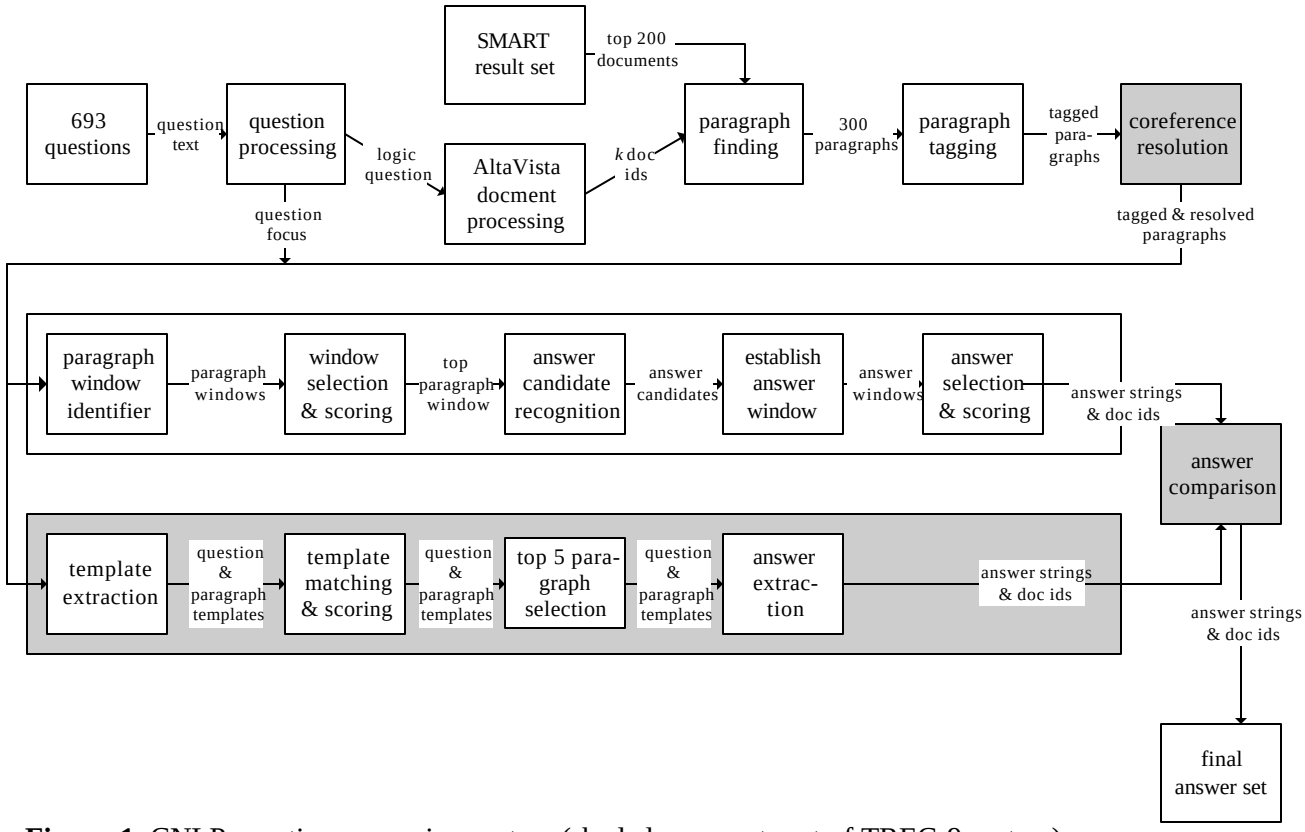

Figure 1. CNLP question answering system (shaded areas not part of TREC-9 system).

For each question, up to five ranked answer submissions were permitted, with the system producing the most likely answer ranked first. The maximum length of the answer string for a retrieval run was either 50 bytes or 250 bytes. An response to a question consisted of the question number, the document ID of the document containing the answer, rank, run name, and the answer string itself. The submitted answer strings were evaluated by NIST's human assessors for correctness. [6]

	TREC-9 question answering questions
Base question	419: Who was Jane Goodall?
Question variants	746: What is Jane Goodall famous for? 747: What is Jane Goodall known for? 748: Why is Jane Goodall famous? 749: What made Jane Goodall famous?
Answer string (50 bytes)	748 AP880225-0129 1 80.90 SUT9p2c3c050 for her 28 years of chimpanzee research

Figure 2. Examples of TREC-9 questions.

3. System overview

The prototype of the CNLP question-answering system consists of four different processes: question processing, document processing, paragraph finding, and keyword and entity based answer finding. Each of the processes is described in detail below.

3.1 Question processing

During question processing, the system converts the question into a logical query representation used for first stage information retrieval and the system determines the focus of each question used for answer finding. Question processing takes place in our Language-to-Logic or L2L module. The L2L process for the question-answering track is optimized for retrieval using the AltaVista search engine (see section 3.2), and includes a focus recognizer. For example, the question "What was the monetary value of the Nobel Peace Prize in 1989?" results in the following output:

AltaVista query:	"monetary value*" +"Nobel Peace Prize*" 1989*
Question focus:	money|numb

The L2L module converts a natural language query or question into a generic logical representation, which can be interpreted by different search engines. The conversion from language to logic takes place based on an internal query sublanguage grammar, which has been developed by CNLP. Prior to conversion, query processing such as stemming, stopword removal, and phrase and Named Entity recognition take place. We experimented with query expansion for first stage retrieval but experienced a slight drop in the results. Based on these results query expansion was left out of the TREC-9 question-answering system.

Question focus recognition aims to determine the expected answer by analyzing the question. For example, consider the question: "What is the monetary value of the Nobel Peace Prize in 1989?" The questioner is obviously looking for a monetary value and that is the focus of the question. Determining the question focus (also referred to as question type, answer type or asking point) helps to narrow the possible answers for which the system will look.

The system uses two strategies to determine the question focus: the question, and, if that strategy fails, the CNLP Named Entity hierarchy. The first strategy tries to find the focus of the question based on clues found directly in the question itself. If the beginning of a question resembles any of a set of clues it is clear what focus is intended. For example, if a question contains the words "which capital city" then the focus is "city". However, it is impossible to predict all possible questions and to have a program that deals with any question. If the system cannot assign a focus to a question using example question phrases, the system then moves to the Named Entity hierarchy clues. The system incorporates one or more clue words for each of the hierarchy classes. For example, the words *hurricane* or *storm* in a question might indicate that the questioner is looking for a weather event. The "why" focus is an exceptional case since it does not indicate a particular topic but rather a place in the sentence where an answer might be found (e.g. after the word "because"). The performance of the focus recognition capability is analyzed in section 5.3.

3.2 Document processing for first stage retrieval

We used two different retrieval approaches for first stage retrieval: Boolean and probabilistic. The entire TREC-9 question answering document collection has been indexed using AltaVista Search Engine 3.0, which is a modified version of the software that runs the search engine at http://www.altavista.com. [2] AltaVista 3.0 indexes all words and does not use stemming. The document collection consisted of 978,952 documents with the average number of words per

document being less than 500. Indexing this collection took approximately 6 days using a Dual Pentium III, 550Mhz, with 512MB ram, running Windows 2000 server. AltaVista also provides the Search Developer's Kit (SDK). The SDK's Interoperability API allows programs to read data from indexes created by the search engine. A batch process takes the L2L query representations and the index directory of document collection as input. For each question, the program returns up to 32,000 documents.

For our probabilistic runs we used the SMART retrieval runs as provided by NIST. The SMART information retrieval system, originally developed by Salton, uses the vector-space model of information retrieval that represents query and documents as term vectors. [5] All vectors have *t* components where *t* is the number of unique terms (or stems) in the collection. A comparison of the Boolean and probabilistic first stage retrieval approaches can be found in section 5.1.

3.3 Paragraph finding

The system uses paragraphs rather than documents for its second stage retrieval. Based on the TREC question-answering guidelines and last year's questions, we assumed that the desired answers were going to be short and factual (less than 50 bytes long). Also, the answer context, which identifies an answer as belonging to a certain question, is usually a small part of the original document. [4] Paragraphs, which are much shorter than documents, have the added benefit of cutting down costly processing time. Paragraph detection is based on text indentations.

889: What is the highest mountain in the world?
Question focus: mnt (mountain)
<NC cat=numb> two|CD </NC> <CN> three-person|JJ team|NNS </CN> of|IN <NP cat=geoadj id=3> American|NP </NP> ,|, <NP cat=geoadj id=0> Soviet|NP </NP> and|CC <CN> <NP cat=geoadj id=1> Chinese|NP </NP> climber|NNS </CN> will|MD attempt|VB to|TO reach|VB the|DT top|NN of|IN <NC cat=dist> 29,028-foot|JJ </NC> <NP cat=mnt id=2> Mount|NP Everest|NP </NP> ,|, the|DT world|NN 's|POS <CN> highest|JJS mountain|NN </CN> ,|, on|IN <NC cat=time> May|NP 6|CD </NC> .|.

Figure 3. Example of tagged paragraph (AP900429-0033) with answer "Mount Everest."

In the paragraph finding stage, we aimed to select the most relevant paragraphs from the retrieved documents from the first stage retrieval step. Paragraph selection was based on keyword occurrences in the paragraphs. The top 300 most relevant paragraphs were selected for each question. After selection, the paragraphs were part of speech tagged and categorized by <!**meta**Marker>™ using CNLP's categorization rules (see figure 3).[1] The quality of selected paragraphs and the system's categorization capabilities directly impact later processing such as co-reference resolution (currently not implemented), and answer finding.

3.4 Keyword and entity based answer finding

The keyword and entity based answer finding process took the tagged paragraphs from the paragraph finding stage and identified different paragraph windows within each paragraph. A weighting scheme was used to identify the most promising paragraph window for each paragraph. These paragraph windows were then used to find answer candidates based on the question focus. All answer candidates were weighted and the top 5 were selected.

3.4. 1. Paragraph-window identification and selection

Paragraph windows were selected by examining each occurrence of a question keyword in a paragraph. Each occurrence of a keyword in relation to the other question keywords was considered to be a paragraph window. A keyword that occurred multiple times thus resulted in multiple paragraph windows, one for each occurrence. A weight for each window was determined by the position of the keywords in the window and the distance between them. An alternative

weighting formula was used for single-word questions. The window with the highest score was selected to represent that paragraph. The process was repeated for all 300 paragraphs resulting in an ordered list of paragraph windows - all potentially containing the answer to the question.

3.4.2 Answer candidate identification

Answer candidates were identified in each paragraph window based on the question focus. Each paragraph window can have multiple answer candidates. If the question focus matched any of the categorized named entities, complex nominals, or numeric concepts in the window, they were considered to be answer candidates. If none of the categorized entities matched the question focus, the system translated the focus into a more general tag. For example, if the question focus called for a city and the paragraph did not have a city tag, the system then looked for a named entity in that paragraph. Naturally these matches received lower weights than entities that directly matched the question tag. If there was no question focus assigned to the question, the system reverted to an alternative strategy and picked the sentence with the largest number of question keywords and looked for named entities. In identifying the different answer candidates, the required window sizes of 50 or 250 bytes were also generated.

3.4.3 Answer-candidate scoring and answer selection

The system used a weighting scheme to assign a weight to each answer candidate. The weight was based on the keywords (presence, order, and distance), whether the answer candidate matched the question focus, and punctuation near the answer candidate. This resulted in a pool of at least 300 candidates for each query. The 5 highest scoring answer candidates were selected as the final answers for each question. The answer strings were formatted according to NIST specifications of either 50 bytes or 250 bytes depending on the run. This process was repeated for all 693 questions resulting in an answer file of 4815 (693x5) lines that were submitted to NIST.

4. Results

Our submission for the question-answering track consisted of four different runs. The SUT9bn3c runs use our L2L module (see section 3.1) with the AltaVista retrieval system for the first-stage retrieval, whereas the SUT9p2c3c runs used the SMART (provided by NIST). Each of these runs had a 50 byte as well as a 250 byte answer string submission. A system bug caused our 250 byte answers to be about 50 bytes shorter (see table 1), which caused a slight drop in results. The program only extended the number of answer bytes on the right-hand side of the answer string but failed to do so on the left-hand side.

Averages over 682 questions (strict evaluation):	**SUT9 bn3c050**	**SUT9 p2c3c050**	**SUT9 bn3c250**	**SUT9 p2c3c250**
Allowed answer length in bytes	50	50	250	250
Average response length in bytes	49.68	49.65	203.24	198.62
Mean reciprocal rank (682 questions)	0.247	0.249	0.365	0.385
Questions with no answer found	436 (63.9%)	439 (64.4%)	334 (49.0%)	319 (46.8%)
Questions above the median[1]	191 (28.0%)	190 (27.86%)	202 (29.62%)	198 (29.03%)
Questions on the median	427 (62.61%)	450 (65.98%)	351 (51.47%)	358 (55.64%)
Questions below the median	64 (9.48%)	42 (6.16%)	129 (18.91%)	99 (14.52%)

Table 1. Question answering results for all four runs.

[1] The median is the middle score (or the average of the two middle scores in case of an even number of scores) for each question after the answer scores for all participants have been put in rank order. 33 groups submitted a 50 byte runs, 42 groups submitted a 250 byte run.

The measure used for evaluation in the question-answering track is the mean reciprocal answer rank. For each question, a reciprocal answer rank is determined by evaluating the top five ranked answers starting with one. The reciprocal answer rank is the reciprocal of the rank of the first correct answer. If there is no correct answer among the top five, the reciprocal rank is zero. Since there are only five possible ranks, the mean reciprocal answer ranks can be 1, 0.5, 0.33, 0.25, 0.2, or 0. The mean reciprocal answer ranks for all the questions are summed together and divided by the total number of questions to get the mean reciprocal rank for each system run.

As is to be expected, the 50 byte runs have a much larger number of questions without an answer than the 250 byte runs. In all four runs, for most questions the system performance equaled the median reciprocal rank of all runs. The majority of the remaining questions were placed above the median.

Answer ranks	**SUT9 bn3c050**	**SUT9 p2c3c050**	**SUT9 bn3c250**	**SUT9 p2c3c250**
Correct answer ranked 1	126 (18.48%)	128 (18.77%)	193 (28.30%)	208 (30.50%)
Correct answer ranked 2	43 (6.30%)	42 (6.16%)	59 (8.65%)	52 (7.62%)
Correct answer ranked 3	35 (5.13%)	37 (5.43%)	45 (6.60%)	46 (6.74%)
Correct answer ranked 4	21 (3.08%)	22 (3.23%)	28 (4.11%)	31 (4.55%)
Correct answer ranked 5	21 (3.08%)	14 (2.05%)	23 (3.37%)	26 (3.81%)
No correct answer found (rank 0)	436 (63.93%)	439 (64.37%)	334 (48.97%)	319 (46.77%)
Total	682	682	682	682

Table 2. Answer rank distribution of question answering results.

The strength of our system lies in answer ranking. Consistently across all four runs, the majority of the correct answers were ranked first. Unfortunately, in all four runs we had trouble locating the answers to the questions.

5. Analysis

This section examines retrieval performance of first stage retrieval, the Language-to-Logic module, and question focus assignment as well as exact answer finding and the effect of question variants on system performance. Overall analysis based on the probabilistic 50 byte run (SUT9p2c3c050) shows that the system retrieves at least one relevant document for each of 625 questions. In the paragraph finding stage we extract paragraphs from 609 of these documents. Out of these 609 paragraphs, 578 paragraphs contain a possible correct answer. However, for only 243 questions we find that correct answer in these paragraphs. Thus, it appears that the answer scoring mechanism and entity tagging, need further refinement.

5.1 First stage retrieval

The analysis of the first stage retrieval was based on the list of relevant documents provided by NIST. We used two different first stage retrieval approaches, a Boolean approach using our L2L module with AltaVista, and a probabilistic approach using the SMART runs (see section 3.1).

Analysis shows that the retrieval performance of both systems is very similar except for the retrieved number of relevant documents, which is larger for SMART (see table 3). This difference is probably caused by a number of AltaVista query representations that had a large number of mandatory terms and failed to retrieve a single document.

Although the SMART retrieval system retrieves more relevant documents, the performance of the two first-stage retrieval models in question answering is very similar. SMART performed slightly better in the 250 byte runs (see table 1).

	Boolean	Probabilistic
Questions without any retrieved documents	3	0
Questions without any relevant retrieved documents	50	48
Questions for which relevant documents are unknown[2]	20	20
Questions with relevant retrieved documents	620	625
Total number of questions for first stage retrieval	693	693
Total number of documents retrieved	111,530	134,600
Number of known relevant documents	7,963	7,963
Total number of relevant documents retrieved	5,579	6,014
Average Precision[3]	0.2766	0.2870

Table 3. First stage retrieval performance.

5.2 Question representation

Logical question representations are one of the things created in the question processing stage (see section 3.1). The question representation analysis is based on the probabilistic 50 byte run (SUT9p2c3c050). A close examination of the question representations created by our Language-to-Logic module showed that for 539 (78.89%) questions, the representation was correct, although 64 (9.38%) representations could stand to be improved. 144 (21.11%) question representations had one or more problems. The most frequently occurring problems were: part-of-speech tagging errors; difficulties with query length (single word questions and very long questions), and; keyword selection problems (see figure 4).

Problem count	Problems with description
76	part-of-speech errors: wrong tags lead to bad phrases and non-content words being added to query
49	query length: single word queries provide little information for answer finding, long queries with many mandatory terms hinder retrieval
6	misplaced wildcards: wildcards placed on final terms of multi-word terms only, or in the wrong place of single terms creating bad stems
10	keyword selection problems: content words such as numbers erroneously filtered out

Figure 4. Question representation problems.

It is clear that the part-of speech tagger had trouble dealing with the unusual phrase structure presented by questions. Other problems, such as the single word queries, are a direct result of the phrasing of the original question. Question expansion for second-stage retrieval might be a solution for this problem. Keyword selection is an L2L problem that needs to be adjusted to keep numbers, and possibly adjectives, that specify the answer (i.e. Who was the first Russian astronaut to walk in space?).

The query representation problems were expected to have a negative impact on answer finding but further analysis showed that this was not the case (see table 4). Even with a problematic question representation, the system was still able to find answers for 77 questions while for 276 questions that did have correct query representations, no correct answers were found. This means that query representation alone only accounts for part of the error.

[2] Number includes the 11 questions discarded by NIST and 9 questions for which no relevance judgments were available.

[3] Average precision over all relevant documents, non-interpolated.

	Correct representation	Problematic representation
Answer correct	166 (37.56%)	77 (32.08%)
Answer incorrect	276 (62.44%)	163 (67.92%)
Total	442	240

Table 4. Question representation correctness and question answering ability.

5.3 Question focus

As described in section 3.1, we determined the focus based on the question clues or Named Entity Hierarchy clues. The question focus analysis is based on the probabilistic 50 byte run (SUT9p2c3c050). Out of 682 answerable questions, our system determined a question focus for 434 (63.64%) of the questions. Out of these 434 questions, 348 questions (80.18%) had a correct focus, and 86 questions (19.82%) had an incorrect focus. For 248 (36.36%) questions, our system could not determine a focus.

	Correct question focus	**Incorrect question focus**	**No determinable question focus**
Rank 1	97 (27.87%	5 (5.81%	26 (10.48%)
Rank 2	22 (6.32%	4 (4.65%	16 (6.45%)
Rank 3	19 (5.46%	2 (2.33%	16 (6.45%)
Rank 4	9 (2.59%	1 (1.16%	12 (4.84%)
Rank 5	7 (2.01%	1 (1.16%	6 (2.42%)
Rank 0	194 (55.75%	73 (84.88%)	172 (69.35%)
Total	348	86	248

Table 5. Answer rank distribution of question focus status.

Out of all the questions that ranked the correct answer first, 97 questions (75.78%) had a correct question focus. It appears that a correct focus aids in answer ranking. When looking at the questions with an incorrect query focus (86) we see that most of these questions (73, or 84.88%) failed to retrieve an answer at all. We can conclude that it pays to have a determinable focus as long as this focus is correct. However, finding the correct query focus is not a guarantee for finding the answer since 194 questions (55.75%) with a correct focus did not retrieve a correct answer.

A closer examination of questions with an incorrect question focus shows that 40 of these questions are erroneously assigned a “person” focus. 17 of the erroneous person focus questions are of the “who is Colin Powell” type. Unlike questions such as “who created the Muppets?” the answer to “who is <person name>” questions is not a person’s name but rather a description of that person. Additional problems with the person focus were questions looking for groups of people (i.e. cultures, sports teams) rather than individual persons, or other entities than persons (i.e. companies, cartoon characters).

5.4 Question variants

As described in section 2, NIST included 193 question variants which are re-wordings of a set of 54 questions (see figure 2). The question variants analysis is based on the probabilistic 50 byte run (SUT9p2c3c050). These question variants allowed us to study the effect of question formulation on system performance. For 25 out of the 54 question sets, the query variation caused no difference in performance. The majority of these questions did not retrieve correct answers no matter how the questions were posed to the system.

29 question sets did show differences in retrieval performance. For 12 sets, the performance differences originated entirely in additional question terms being either present or missing. For 7 sets, the differences in performance were partially due to divergence of question terms. Some question terms would guarantee a correct answer, whereas others would throw the results off. The majority of the questions are rather short, so each question term has a relatively large influence on finding the answer. The query variant results indicate that query expansion could have a large impact on system performance. Although we experimented with query expansion for first stage retrieval, we did not have enough time to explore it in the answer-finding stage.

For 14 sets, some of the differences in system performance appear to be caused by a different or missing question focus. In eight question sets, some of the differences in performance were caused by the question focus being incorrect. Additional question words mislead the system in choosing the wrong question focus. In sets where the question focus is either missing, different, or incorrect, the well-performing counterpart questions did have the correct or more exact focus, and the variant questions, without the exact clue, experienced a drop in rank or a failed attempt to find the answer. These findings indicate that having a correct question focus is of importance, which supports findings of the question focus analysis (section 5.3).

In seven question sets, some of the differences were caused by inconsistencies in the answer judgments. Certain answers would be judged to be correct for some questions, whereas for others the same answer would be judged to be incorrect.

5.5 Exact answer finding

Although plans for an "exact answer" run were abandoned by NIST, we examined the system's exact answer-finding capabilities for the probabilistic 50 byte run (SUT9p2c3c050). The majority of the exact answers that our system produced were judged correct (197 or 81.07%), and only 46 (18.93%) of the answers were produced by the context of the answer window (see table 6). This indicates that our system had quite a high answer-finding accuracy when a correct answer was contained in the retrieved document.

Question answered at rank …	Number of Q. judged correct	Exact correct answer string found	Answer produced by context words in the 50-byte window
Rank 1	128	112	16
Rank 2	42	31	11
Rank 3	37	27	10
Rank 4	22	15	7
Rank 5	14	12	2
Total	243	197 (81.07%)	46 (18.93%)

Table 6. Rank distribution of correctly answered questions and our system performance

6. Conclusions and future research

The performance of the CNLP question answering system is highly encouraging. The majority of the correct answers are ranked first and the majority of question representations and assigned question foci were accurate. The prototype system also does well at exact answer finding. However, for a large number of questions no correct answers are found. It appears that the current system does not capitalize on the large number of relevant documents found in the first retrieval stage.

Further research is needed to refine the weighting in the paragraph selection and answer finding stages, and to improve the query sublanguage grammar to increase question focus assignment robustness. In addition, a new morphological analyzer needs to be implemented and the part-of-speech tagger needs to be trained on question phrase structure, to improve question representations. A more detailed study of the categorization performance and coverage is also in order. Time also needs to be spent on researching and implementing a second approach to answer finding based on template matching.

Acknowledgements
We would like to thank Catie Christiaanse, Michelle Monsour, and Eileen Allen for creating the necessary categorization rules in a very limited time frame. We would also like to thank Hassan Bolut, and Wen-Yuan Hsiao for their engineering support. Lastly we would like to thank Lois Elmore for keeping us all in line.

References

[1] <!metaMarker>™ . http:// http://www.solutions-united.com/products_information.html

[2] AltaVista Search Engine 3.0. http://solutions.altavista.com/downloads/downloads.html

[3] Cooper, William S. (1978) The "Why Bother?" Theory of Information Usage. *Journal of Informatics,* 2, p. 2-5.

[4] Prager, John; Brown, Eric; Coden, Anni. (2000). Question-Answering by Predictive Annotation. In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval,* July 24 - 28, 2000, Athens, Greece.

[5] Salton, G. Ed. (1971) *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc. Englewood Cliffs, NJ. 556p.

[6] Voorhees, Ellen M.; Tice, Dawn M. (1999). The TREC-8 Question Answering Track Evaluation. In: E.M. Voorhees and D.K. Harman (Eds.) *The Eighth Text REtrieval Conference (TREC-8)*. 1999, November 17-19; National Institute of Standards and Technology (NIST), Gaithersburg, MD.