Exploring Document Content with XML to Answer Questions

Kenneth C. Litkowski
CL Research
9208 Gue Road
Damascus, MD 20872
ken@clres.com

Abstract

CL Research participated in the question answering track in TREC 2004, submitting runs for the main task, the document relevance task, and the relationship task. The tasks were performed using the Knowledge Management System (KMS), which provides a single interface for question answering, text summarization, information extraction, and document exploration. These tasks are based on creating and exploiting an XML representation of the texts in the AQUAINT collection. Question answering is performed directly within KMS, which answers questions either from the collection or from the Internet projected back onto the collection. For the main task, we submitted one run and our average per-series score was 0.136, with scores of 0.180 for factoid questions, 0.026 for list questions, and 0.152 for "other" questions. For the document ranking task, the average precision was 0.2253 and the R-precision was 0.2405. For the relationship task, we submitted two runs, with scores of 0.276 and 0.216, the first run was the best score on this task. We describe the overall architecture of KMS and how it permits examination of the question-answering task and strategies within TREC, but also in a real-world application in the bioterrorism domain. We also raise some issues concerning the judgments used for evaluating TREC results and their possible relevance in a wider context.

1 Introduction

In TREC 2002, CL Research examined the potential of using XML-tagged documents for question answering (Litkowski, 2003) and showed that hand-developed XPath expressions could obtain extremely good results when compared with the best sytems. In TREC 2003 (Litkowski, 2004), initial efforts at the automatic creation of XPath expressions achieved very limited results. In TREC 2004 (Litkowski, 2005), significantly better results were obtained as core XML functionality was implemented. CL Research's participation in TREC question-answering is rooted in the Knowledge Management System (KMS), which provides an integrated framework for question answering, text summarization, information extraction, and document exploration. In 2005, KMS was used in a demonstration project of question-answering and summarization in the biomedical domain, allowing the examination of various strategies for obtaining answers and document content in a wider range of non-factoid question types. This project provided some background for KMS modifications that could be employed in TREC question-answering. This paper describes extensions to the KMS architecture and how they were used in TREC 2005, particularly for the relationship task.

Section 2 presents the TREC QA task descriptions. Section 3 describes the KMS, specifically components for processing texts and for performing particular NLP tasks. Section 4 provides our question answering results, particularly our experience in handling different types of questions, our performance in the document ranking task, and our perspectives in framing the relationship task (noting its similarities to answering "other" questions in the main task and to the Document Understanding Conference's topic descriptions). Section 5 presents our overall summary and conclusions.

2 Problem Description

The TREC 2005 QA used the AQUAINT Corpus of English News Text on two CD-ROMs, about one million newswire documents from the *Associated Press Newswire*, *New York Times Newswire*, and *Xinhua News Agency*. These documents were stored with SGML formatting tags (XML compliant).

For the main task of the QA track, participants were provided with 75 targets, primarily names of people, groups, organizations, and events, viewed as entities for which definitional information was to be assembled. For each target, a few factual questions were posed, totaling 362 factoid questions for the 75 targets (e.g., for the target event "Plane clips cable wires in Italian resort", two factoid questions were "When did the accident occur?" and "How many people were killed?"). One or two list questions for each target were also posed for most of the targets (e.g., "Who were on-ground witnesses to the accident?"); there were 93 list questions. Finally, for each target, "other" information was to be provided, simulating an attempt to "define" the target. Each target was used as a search query against the AQUAINT corpus. NIST provided the full text of the top 50 documents, along with a list of the top 1000 documents.

Participants were required to answer the 362 factoid questions with a single exact answer, containing no extraneous information and supported by a document in the corpus. A valid answer could be NIL, indicating that there was no answer in the document set; NIST included 17 questions for which no answer exists in the collection. For these factoid questions, NIST evaluators judged whether an answer was correct, inexact, unsupported, or incorrect. The submissions were then scored as percent of correct answers. For the list questions, participants returned a set of answers (e.g., a list of witnesses); submissions were given F-scores, measuring recall of the possible set of answers and the precision of the answers returned. For the "other" questions, participants provided a set of answers. These answer sets were also scored with an F-score, measuring whether the answer set contained certain "vital" information and how efficiently peripheral information was captured (based on answer lengths).

Participants in the main task were also required to participate in the document-ranking task by submitting up to 1000 documents, ordered by score. Instead of providing an exact answer, participants were required to submit only the identifier of the document deemed to contain an answer. Document ranks were to be provided for 50 questions, with at least one document for each question. Scoring for this task used standard measures of recall (how many of the relevant documents were retrieved) and precision (how many of those retrieved were actually relevant). Summary measures are the average precision for all relevant documents and R-precision, the precision after R documents have been retrieved, where R is the number of relevant documents for the question.

For the relationship task, participants were provided with TREC-like topic statements to set a context, where the topic was specific about the type of relationship being sought (generally, the ability of one entity to influence another, including both the means to influence and the motivation for doing so). The topic ended with a question that is either a yes/no question, which is to be understood as a request for evidence supporting the answer, or a request for the evidence itself. The system response is a set of information nuggets that provides evidence for the answer. An example is shown in Figure 1 (along with a comparable topic used in DUC 2005). Answers are scored for the relationship task in the same manner as the for the "other" questions of the main task.

CL Research submitted one run for the main and document-ranking tasks and two runs for the relationship task.

Relationship Topic 11: The analyst is interested in Argentina's intentions in the Falkland Islands. Specifically, the analyst wants to know of any ongoing or planned talks between Argentina and Great Britain over the island's future.

DUC d324e: How have relations between Argentina and Great Britain developed since the 1982 war over the Falkland Islands? Have diplomatic, economic, and military relations been restored? Do differences remain over the status of the Falkland Islands?

Figure 1. Relationship Topic and DUC 2005 Topic

3 The Knowledge Management System

The CL Research KMS is a graphical interface that enables users to create repositories of files (of several file types) and to perform a variety of tasks against the files. The tasks include question answering, summarization, information extraction, document exploration, semantic category analysis, and ontology creation. The text portions of files (selected according to DTD elements) are processed into an XML representation; each task is then performed with an XML-based analysis of the texts. KMS also includes modules to perform web-based question answering (acting as a wrapper to Google) by reformulating questions into canonical forms and to search a Lucene index of document repositories by reformulating questions into a boolean search expression. These modules can be used to obtain answers to questions and to project those answers back onto document repositories, in this case, the AQUAINT collection.

KMS uses lexical resources as an integral component in performing the various tasks. Specifically, KMS employs dictionaries developed using its DIMAP dictionary maintenance programs, available for rapid lookup of lexical items. CL Research has created DIMAP dictionaries for a machine-readable version of the *Oxford Dictionary of English*, WordNet, the Unified Medical Language System (UMLS) Specialist Lexicon (which provides a considerable amount of syntactic information about general, non-medical lexical items), *The Macquarie Thesaurus*, and specialized verb and preposition dictionaries. These lexical resources are used seamlessly in a variety of ways in performing various tasks, described in more detail below.

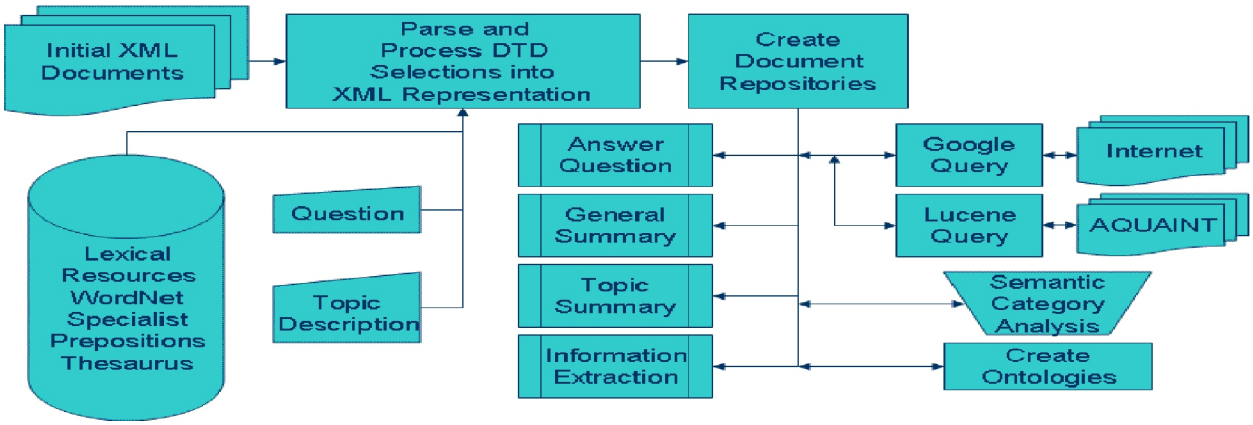

Figure 2. CL Research Knowledge Management System

KMS consists of a large number of modules. These modules are also used in two additional programs that perform more specialized processing. The DIMAP Text Parser uses the parsing and XML-generation components for background processing of large numbers of texts, such as the document sets for the QA tasks. The XML Analyzer, primarily used to enable detailed examination of XML versions of the documents for diagnostic purposes, also includes special processing routines to select elements from XML files for further processing (similar to what may be accomplished with XSLT transformations).

When performing a task with KMS, results are maintained in XML representations. For example, all answers to questions are retained and saved answers can be viewed without going through question answering again. Answers include the file they came from, the document within the file, the score, an exact answer, and the sentence from which they were extracted. (During development, as question answering routines are modified, new answer sets replace existing answer sets.) Typically, the number of answers to a question is very large, although not infrequently, no answers or only one answer is returned. For TREC tasks, these answer files are processed with Perl scripts to create the TREC submissions following the required format. For the main task, only the first answer is used, and a NIL answer is returned when no answers have been generated. For the list and "other" questions of the main task, the maximum number of answers is limited to 25 and 20, respectively. For the document-ranking task, all answers (up to a maximum of 1000) were returned with their scores. For the relationship task, up to 40 answers were returned.

Improving the performance of KMS in its various tasks requires improved characterization of the texts, development of appropriate strategies for performing the tasks, and the use of efficient mechanisms for evaluating performance. XML provides a convenient formalism for representation and analysis, but this approach involves frequent reformulations to capture and recognize large numbers of linguistic phenomena. Improvements come from first dealing with low-level phenomena, sometimes idiosyncratically, and then, as patterns emerge, from generalizing over several phenomena. In general, improved characterization of linguistic phenomena leads to improved performance.

3.1 Changes in Text Processing and XML Analyses

As described for TREC 2004 (see Litkowski, 2005 and early references for greater detail), modifications to KMS are continually being made in virtually all supporting modules. Changes are made to the parser in an attempt to ensure improved parse trees. During the last year, a significant change has been made in the dynamic creation of dictionary entries used in parsing. The parser has a built-in general-purpose dictionary. Procedures were developed to make use of the UMLS Specialist Lexicon of biomedical terminology in a project where documents pertaining to a biopathogen were being processed. Through the use of this lexicon, as well as WordNet, it was possible to create entries for phrases so that they could be recognized as single units in processing texts. Many changes were also made to the parse tree analyzer that identifies important discourse constituents (sentences and clauses, discourse entities, verbs and prepositions) and creates an XML-tagged version of the document. It is difficult to assess the benefits of changes to these major components of KMS. In evaluating improvements in question-answering performance, only a small percentage of failures are usually attributed to failures in these modules. Overall, the changes in these modules result in ever-larger size of the XML representations of the texts; the size of the XML files is roughly 10 times the size of the original documents, up from 6.7 during TREC 2003 and 10 percent

larger than for TREC 2004. Questions in the QA track are processed using the same text processing mechanisms; their XML representations are used as the basis for answering the questions.

3.2 Changes to KMS Question Answering Routines

In reporting on KMS for TREC 2004, we described in detail the major types of functions employed: XML, linguistic, dictionary, summarization, and miscellaneous string and pattern matching. In each category, the number of such functions has increased during the past year, particularly in support of other functionality in KMS. Last year, we also described our procedures for using past results as the basis for making changes in handling questions of particular types. We have continued to follow those procedures.

As described in previous years, KMS question-answering is based on identifying the question type (including modifying a question such as "What year" into a **When** question) and then using idiosyncratic routines to build an XPath expression for each question type. The XPath expression consists of two components: one part to retrieve sentences satisfying boolean-style conditions and the other part to focus on a particular element of the retrieved sentence. The focal element is either a potential answer or a pivot that can be used to examine the surrounding context for the answer. Typically, the specifications in the XPath expression are very inclusive to permit a large set of potential answers. The questions-specific routines for examining candidate answers scores the answer type and the surrounding context to exclude most candidates and to rank the remaining answers. KMS returns both an exact answer and the sentence from which it came for the convenience of the user.

As demonstrated in TREC 2002, it is possible to develop XPath expressions by hand to obtain virtually all of the correct and exact answers. However, the automatic creation of XPath expressions is considerably more difficult and time-consuming. CL Research has used many techniques in the early TREC QA tasks when the primary mechanism was the use of database technology for storing semantic relation triples. Incorporation of these techniques into an XML-based approach is not yet complete. During preparations for TREC 2005, we were able to work in detail with only **When**, **Where**, and **HowMany** question types (slightly less than 40 percent of this year's factoid questions).

Some changes involved implementation of simple pattern recognition procedures, such as the attachment of a year to a month and day or the attachment of a state or country to a city name. Other changes involved the addition of improved and more efficient methods for manipulating and querying the XML representation. The major qualitative shift involved the representation and exploitation of the "context" of a question, that is, the text surrounding the specification in the question of the focal element or answer type. Several layers of contextual representation have been developed (such as whole noun phrases, adjective and noun constituents, and prepositional relations); these representations were variously exploited in answering different question types.

As indicated above, KMS includes a component that uses a Google search to answer a question. As with answers against a repository, web answers are maintained in their own file, in this case showing the web address of the document answering the question. KMS can use the TREC question set and have them submitted to Google, or a user can pose any question and the question is saved in a repository of questions.

Answering a question using Google (or any web-based search engine) goes through a slightly different strategy than against a repository. The main difference is how the question is analyzed and formulated for submission to Google. Given the breadth of the Internet, there is a strong likelihood

that a question can be answered using a canonical form. For example, to the "When was John Glenn born?", it is likely that "John Glenn was born in" will exist as an exact phrase and that a search for the entire phrase as given will return documents where the displayed Google results will contain the answer.

Each question must be reformulated into an exact query. KMS analyzes the question type and its constituent terms to determine what is likely to be the best form. For the most part, the key transformation involves the placement, number, and tense of the verb. For example, "How did James Dean die?" has to be transformed to "James Dean died". KMS uses the UMLS lexicon for this purpose, since it contains information about the "regularity" of a verb form and how it's various tenses are formed. In its search results, Google displays a snippet that usually consists of one or two sentences, along with some additional partial sentences. KMS examines the snippet and extracts the whole sentence containing the exact phrase in the query. This single sentence is then treated as a document and is parsed and processed into an XML representation. Since Google usually returns 10 hits per page, the corpus to answer a question usually consists of 10 single-sentence documents. After extracting these sentences, the usual KMS mechanisms are employed to analyze the questions and extract the answers, with results displayed according to the score for each sentence. When the Google answer is reliably extracted (and this occurred for 72 of the 362 questions), the answer is projected back onto the document collection and is used to "force" an answer, which may or may not be found in the repository.

When a projected answer is not found in the document set that has been used for the question (the top 50 documents provided by NIST), the major terms in the question are combined with the projected answer and the AQUAINT collection is searched using Lucene with this boolean expression to obtain a maximum of 5 more documents, which must not be in the original set. These 5 documents are then processed in the usual way and another attempt is made to find the projected answer. When found, this is then the answer that is used in the TREC submission. This method only obtained 9 additional answers in TREC 2005.

3.3 Answering Relationship Questions

As shown in Figure 1, questions for the relationship task were posed in terms describing what information was desired by an analyst. Figure 1 also showed the topic description used in DUC 2005 for one topic for which a summary was to be created. The similarity between the two is a strong suggestion that the lines between summarization and relationship questions is somewhat blurred. In addition, as specified in the task description, answers to relationship questions were to be evaluated in the same way as the "other" questions in the main task. This suggests that there is also a close tie between the "other" questions and the relationship questions. These observations influenced the strategy we used in responding to the relationship task. As will be discussed below in reporting on our results for this task, this task raised the issue of how all the different question types might fit within a larger context of question-answering strategies to server user needs.

Unlike the main task, no top documents were provided for the relationship task. Since CL Research has little expertise in indexing and search technology, a somewhat truncated approach was used. For the relationship subtask, we experimented with two modes of operation, both of which constitute a truncated version of the KMS document exploration functionality (which is specifically designed to examine relationships). Each relationship topic was reformulated into a simple Lucene boolean query to retrieve no more than 25 documents from the collection. In creating the search

```
Lucene Queries:
3 "Bonaire"
8 "Cuba AND Angola"
12 "FARC AND (Venezuela OR Venezuelan OR Brazil OR Brazilian)"
16 "Israel AND India AND military"
17 "Israel AND military AND China"
19 "Rifaat AND Assad AND Syria"
23 "nuclear AND proliferation AND South AND (America OR American)"

Definition Questions:
3 What is the use of Bonaire as a transit point for drugs?
8 What is the role of Cuba in Angola?
12 What are the FARC activities along the Venezuelan and Brazilian borders?
16 What are the military personnel exchanges between Israel and India?
17 What are the military ties between China and Israel?
19 What is the influence of Rifaat Assad in Syria?
23 What are the countries in South American involved in nuclear programs?
```

Figure 3. Relationship Lucene Queries and Definition Questions

query, only content words from the topic statement were used, i.e., excluding terms like "the analyst is interested in ...," none of which would generally be included in an actual search query. In several cases, the initial search query included too many AND expressions and returned no documents, in which case the query was scaled back until it returned some documents. The goal for this phase was to obtain 25 documents. The topic was then reformulated into a single question which attempted to capture the essence of what the analyst was seeking. Figure 3 shows the Lucene search queries and the associated question for seven of the relationship questions.

We did not attempt to answer the 25 relationship questions directly using the fact-based routines in KMS. Instead, we "forced" KMS to answer the questions as if they were definition questions. In KMS, special routines are used to answer the "other" portion of the main task, focusing either on **Who** or **What** as an item to be defined. These routines are specifically designed to look for definition patterns in text, such as copular verbs, appositives, and parenthetical expressions. Points are given for the presence of these constructs, matches with words defining terms as available in machine-readable dictionaries or WordNet glosses, and the presence of the term to be defined. Two runs were made. For the first run, definition-style answers were obtained with KMS definition pattern-matching routines as described. For the second run, scoring boosts to "definition" sentences were given based on the "context" analysis routines, where the question as posed was analyzed into qualifiers identifying noun, adjectives, phrases, and semantic relationships (as indicated by prepositions).

4 TREC 2005 Question-Answering Results

4.1 Main Task

CL Research submitted one run for the main task of the TREC 2005 question-answering track. All answers were generated in KMS. The question set provided by NIST was first converted into a form for parsing and processing into an XML representation. The top 50 documents for the 75 targets were also processing into an XML representation. The questions are displayed in KMS,

each with a checkbox used to select which questions are to be answered. All questions were selected and answered automatically, generating an answer file as described above. A Perl script was then used to create an answer file in the format required for a QA track submission.

Conversion of the NIST question set involved minor formatting changes (using a different tag set) and a more considerable anaphora replacement algorithm. For TREC 2005, this was performed in a Perl script (the same as used for TREC 2004), rather than attempting to use the potential capabilities of KMS for capturing anaphoric references. The Perl script identified all referring expressions in the questions, including anaphors such as *her*, *it*, and *their* and definite noun phrases such as "the organization." The script kept track of the type of anaphors so that the "other" question could be converted to either "Who is" or "What is". The revised question set was added to KMS as a question list from which question selection could be made.

Table 1 shows the summary score for the CL Research QA run, along with the median score for all participating teams, broken down by major question type. The table also shows an unofficial score based on CL Research assessment of our answers in conjunction with the factoid patterns and a subjective assessment of the results for the list and "other" questions. The table also shows an adjusted score for the factoid component based on a determination of whether a document with a known answer was present in the collection used for answering the question; this row shows the score that would result based only on the question-answering performance and not the retrieval performance. In our submission for the main task, 95 NIL answers were submitted; our precision on these answers was only 5/95 or 0.053, while our recall on NIL answers was 5/17 or 0.294. We expect that this low precision was due in large part to the unavailability of the answers in the collection that was used for each question.

Table 1. Summary Scores for the Main Task (with median)

	Factoid (0.152)	"Other" (0.156)	List (0.053)	Overall
Official	0.180	0.152	0.026	0.135
Unofficial	0.282	0.260	0.049	0.218
Adjusted	0.374	0.260	0.049	0.264

As can be seen, CL Research scored somewhat higher than the median for the factoid component and this was somewhat higher than our performance last year. In reviewing the official results, however, we observed a very high number of "inexact" judgments and looked at these answers more closely in relation to the answer patterns. In at least 17 cases (0.049), we were perplexed by the judgments. In some cases, our answers were identical to what had been scored as correct for other submissions. In other cases, e.g., "How many openings ..." with our answer of "17 openings", the inexact judgment seems to distort what KMS was actually returning. We include our assessments for the list and "other" questions without yet having assessed these answers in detail, based on the experience with the factoid judgments and with our scores for these components last year and the results of our performance on the relationship task (described below) which seem somewhat at odds with the official score.

Tables 2 shows the unofficial and adjusted scores for the factoid questions by question type. These scores reflect the focus described above for the question types we were able to examine in detail. In particular, these detailed results show that we have considerably improved our performance on **When**, **Where**, and **HowMany** questions, although only a little more effort was spent on **HowMany** questions. This table also shows where considerably more work is required. In particular, the poor performance on the **WhatIs** and **WhatNP** questions reflects, in considerable measure, the

fact that we have not yet incorporated the necessary routines for taking advantage of WordNet hierarchies and thesaurus groupings into the XML processing that we used in earlier years. The effect of this poor performance also carries over to our performance on the list questions, all of which essentially require the identification of a hierarchical relationship.

Table 2. Factoid Questions (Estimated, with Adjustment for Unprocessed Documents)					
Question Type	Number	No Document	Correct	Accuracy	Adjusted
How	4	1	3	0.750	1.000
HowMany	40	8	11	0.275	0.344
HowMeas	18	4	2	0.111	0.143
HowMuch	1	1		0.000	0.000
Name	3	1		0.000	0.000
What	3		2	0.667	0.667
WhatIs	61	18	6	0.098	0.140
WhatName	11	2		0.000	0.000
WhatNP	49	17	5	0.102	0.156
WhatVP	6	1		0.000	0.000
When	67	12	37	0.552	0.673
Where	43	11	21	0.488	0.656
WhoIs	35	6	10	0.286	0.345
WhoVP	19	5	5	0.263	0.357
Why	2	2		0.000	0.000
Total	362	89	102	0.282	0.374

4.2 Document Ranking Task

For the document ranking task, CL Research's average precision was 0.2253 and R-precision was 0.2405. These results were slightly below the results shown for the best runs from the top 13 groups (0.2445 and 0.2596 for average precision and R-precision, respectively). These results are consistent with our position for the main task. While document ranking is not a primary criterion that CL Research will use for assessing question-answering performance, this task is useful primarily for identifying relevant documents and will be used in diagnosing problems with our performance.

4.3 Relationship Task

For the relationship task, we submitted two runs (clr05r1 and clr05r2), with scores of 0.276 and 0.216. The first run, which used the second method described above, was the best score on this task; the second run was the 4th best score among 11 submissions. The results by question number are shown in Table 3. The table shows that KMS performed better with each method for different questions, suggesting that a mixture of strategies is appropriate. The table also highlights questions for which one of the runs achieved the best score. For clr05r1, the best score was achieved on 6 of the questions; for clr05r2, the best score was achieved on 2 of the questions.

The table also shows how many documents were present in the set that was used to respond to the relationship questions, identified as either present or not present. These columns are based on the judgment set for the relationship questions from all participating teams. While this judgment set is known to be an incomplete representation, since not all nuggets were found, it does suggest that CL Research's retrieval performance also had a significant effect on the overall score.

Table 3. TREC-14 Relationship Task Details							
Question	clr05r1	clr05r2	Present	Not Present	Best	Median	Worst
1	0.212	0.238	5	16	0.460	0.263	0.082
2	0.000	0.000	0	1	0.000	0.000	0.000
3	0.253	0.000	1	0	0.253	0.000	0.000
4	0.443	0.415	8	7	0.523	0.336	0.000
5	0.000	0.000	3	2	0.316	0.188	0.000
6	0.000	0.000	1	6	0.297	0.000	0.000
7	0.000	0.000	0	4	0.288	0.168	0.000
8	0.646	0.257	6	1	0.646	0.000	0.000
9	0.218	0.196	5	0	0.516	0.000	0.000
10	0.000	0.000	3	2	0.000	0.000	0.000
11	0.427	0.427	9	7	0.638	0.427	0.000
12	0.240	0.249	7	7	0.249	0.000	0.000
13	0.585	0.170	7	0	0.585	0.338	0.131
14	0.440	0.440	8	6	0.649	0.187	0.000
15	0.350	0.161	9	8	0.478	0.161	0.000
16	0.566	0.308	7	0	0.566	0.005	0.000
17	0.495	0.137	10	9	0.495	0.000	0.000
18	0.291	0.299	6	4	0.345	0.169	0.000
19	0.404	0.000	10	3	0.404	0.000	0.000
20	0.000	0.000	0	1	0.000	0.000	0.000
21	0.502	0.436	3	2	0.731	0.436	0.000
22	0.351	0.423	5	12	0.431	0.180	0.000
23	0.123	0.808	1	2	0.808	0.000	0.000
24	0.118	0.000	1	2	0.567	0.000	0.000
25	0.233	0.435	4	3	0.626	0.233	0.000
Overall	0.276	0.216					

To examine our performance in this task in more detail, we asked several questions:

- What is the effect of qualifiers in answering relationship questions?
- What was the effect of expanding the document set?
- What was the effect of changing the number of answers submitted?
- Are there other ways to obtain answers to these questions?

Table 4 summarizes our official and unofficial results, identifying measures of performance for the two official runs (clr05r1 and clr05r2) and two unofficial runs (clr05r3 and clr05r4).

Table 4. TREC-14 Relationship Task Summary				
	Official		Unofficial	
Measure	clr05r1	clr05r2	clr05r3	ckr05r4
Answers generated	2849	801	3455	3549
Answers submitted	948	638	951	951
Recall	0.457	0.345	0.414	0.407
Precision	0.074	0.074	0.069	0.068
F-Score	0.276	0.216	0.244	0.241

As indicated above, the main difference between our official runs was the use of strict definition question criteria (clr05r2) or the expansion to include consideration of contextual information (clr05r1). This difference shows up in the number of answers that are generated. As can be seen, use of contextual information returned a considerably larger number of sentences for

consideration. This suggests that casting a wider net (i.e., essentially using fewer terms for retrieval), combined with useful criteria for ranking the sentences was successful. Even though a higher recall was achieved and more sentences submitted (with a maximum of 40 sentences for any individual question), the precision was not compromised.

To examine the question of expanding the document collection, we examined how much was lost by our "naive" boolean searches. By examining the judgment set for the all runs submitted in the relationship task, we were able to determine that 105 documents were identified by other teams that were not included in the maximum of 25 documents we included in the set we processed. This is an average of four documents per question. When we added these documents to our collection, resulting in the unofficial run clr05r3 shown in Table 4, an additional 600 sentences were generated in our answer set. However, this resulted in only 3 additional submissions under our criterion of a maximum of 40 answers per question. As shown in Table 4, adding these documents did not improve our score, but rather resulted in a significant drop in recall and a smaller drop in precision, with an overall significant drop in F-score.

Next, we identified documents based on official nuggets which no submission found. To do this, we used keywords in the nugget with Lucene to identify documents in which the answers might be found. We were able to identify 40 documents that contained a sentence that we judged to be an appropriate answer for the nugget. However, after completing this effort, we found that 15 of the documents were already in our collection, but we had not returned the sentence. The remaining 25 new documents thus added only one additional document per question to our collection (clr05r4). Adding these documents again did not improve our score, but rather led to another small drop in recall, precision, and F-score.

In both clr05r3 and clr05r4, the effect of adding documents and retrieving more candidate sentences resulted in bumping down good answers to lower positions in the ranking and had the effect of removing some entirely, thus explaining the lower scores. This results suggests once again that better methods for assessing individual sentences is the key to improving our results in this task.

To score these two unofficial runs automatically, we made use of two criteria. The first criterion was exact: an answer that had been judged correct officially was merely moved to a different position in the ranking. The second criterion was more subjective. We attempted to measure the overlap between the words in the official nugget and the candidate answer. We eliminated stop words from this assessment and then used the criterion that 40 percent of the words had to be present in a candidate answer. The 40 percent criterion was reached after some experimentation (higher percentages eliminated many correct answers, while smaller percentages returned too many). This 40 percent criterion may have some utility in assessing answers for the "other" questions on the main task.

Table 5 shows the effect of the number of answers submitted. To obtain the score for 39 answers, we first removed the last answer and rescored the results based on the official judgment. We continued in this way down to our results with a maximum of one answer submitted. These results show that our selection of 40 answers was quite fortuitous and resulted in almost the maximum score that our system could have obtained. Thus, any further improvements in our system would have to result from a better selection and ranking of candidate sentences.

Table 5. Effect of Answers Submitted in Relationship Task			
clr05r1		clr05r2	
Number of Answers	F-Score	Number of Answers	clr05r3
40	0.276	40	0.216
39	0.278	39	0.216
38	0.281	38	0.216
37	0.284	37	0.217
36	0.276	36	0.213
...			
1	0.059	1	0.007

The above analyses suggest that we can obtain further improvements in our system. For this task, we can expect that summarization techniques for measuring overlap or redundancy may be useful. In addition, improvements may be made in better evaluations of answers, perhaps attempting to tailor the system to spheres of influence. It is also possible that better assessment of paraphrases and recognition of hyponymic sentences would benefit the task. However, it seems that completeness is lost and making such improvements is very difficult. This raises the question of whether alternative methods might be useful for answering relationship questions.

In a parallel project in the bioterrorism domain, we experimented with the use of information extraction techniques to identify pertinent pieces of information. Within the KMS framework, we found that it was relatively easier to construct XPath expressions (using string, syntactic, and semantic criteria) to identify key items (such as people, organizations, and countries) and their relations. An important aspect of this search approach is that results are not affected by growing documents sets where it is necessary to rank candidate answers.

To examine this approach in the relationship task, using the first question about the al-Qaeda network, we used KMS to identify all mentions of "persons". This resulted in 2967 hits in KMS (easily scrollable). This search identified many people that were part of the al-Qaeda network, but that had not been identified in the official nugget set. These results also identified many groups that were part of al-Qaeda, none of which were included in the official nugget set, even though this was a significant part of the topic description. Once having identified a list of people, it was then possible to focus a search in more detail for specific individuals. Thus, we were able to identify 40 sentences that referred to a specific person, including 59 anaphoric references to this individual, in 11 documents.

5 Summary and Conclusions

Our participation in TREC 2005 question-answering track, particularly in the relationship task, raises several issues. Although the track has maintained a focus on answering factoid questions, the attempt to broaden the scope, first with the "other" questions and now with the relationship questions, seems to be moving into areas where it has considerable overlap with other NLP technologies. This is most evident with the similarity between the questions in the relationship task and the topic descriptions used in the Document Understanding Conference for producing topic-based summaries.

The similarities among the tasks (factoid questions, list questions, "other" questions, and relationship questions) suggests the need for a still broader conceptualization of the question-answering track. Suggestions have been made that all of the information included in the answers to

these various types of questions might fit within a templatized notion of the question-answering task. In this conceptualization, it may be desirable that the main task should be the automatic development and completion of templates pertaining to a target of interest.

References

Litkowski, K. C. (2003). Question Answering Using XML-Tagged Documents. In E. M. Voorhees & L. P. Buckland (eds.), *The Eleventh Text Retrieval Conference (TREC 2002)*. NIST Special Publication 500-251. Gaithersburg, MD., 122-131.

Litkowski, K. C. (2004). Use of Metadata for Question Answering and Novelty Tasks. In E. M. Voorhees & L. P. Buckland (eds.), *The Twelfth Text Retrieval Conference (TREC 2003)*. NIST Special Publication 500-255. Gaithersburg, MD., 161-170.

Litkowski, K. C. (2005). Evolving XML and Dictionary Strategies for Question Answering and Novelty Tasks. In E. M. Voorhees & L. P. Buckland (eds.), *The Twelfth Text Retrieval Conference (TREC 2004)*. NIST Special Publication 500-261. Gaithersburg, MD., TREC 2004 Proceedings CD. (Available: http://trec.nist.gov/pubs/trec13/t13_proceedings.html)