Summarization Experiments in DUC 2004

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

Abstract

CL Research's participation in the Document Understanding Conference for 2004 was primarily intended to conduct further experiments in the use of XML-tagged documents containing increasingly richer characterizations of texts. We extended the Knowledge Management System to include (1) a refined capability for identifying multiword units (phrases) for use in keyword generation, (2) the incorporation of word-sense disambiguation to tag senses and identify semantic types, and (3) the integration of question-answering functionality into the summarization framework. We did not devote much effort in refining our system to create summaries for the five tasks, but achieved reasonable levels of performance. We viewed the length restrictions imposed on the tasks as not providing sufficient flexibility to investigate different modes of summarization. We viewed the tasks of summarizing machine translations of poor quality as not very interesting. We used Tasks 1 and 3 to develop and refine a keyword generation capability, achieving levels of fourth of 18 and fourth of 10 priority 1 systems. In the more general summarization tasks, our performance was near the bottom of participating systems, but still achieved acceptable levels of performance. We performed much better on quality measures with our extraction-based summaries, with an overall level of third of 14 systems for Task 5. For several quality measures, our performance was somewhat less; these levels identify specifically those areas of summarization analysis where the use of an XML representation are particularly amenable to improvement. While we will continue to improve our summarization capability within the general guidelines, we believe that summarization is only one part of document understanding and may not represent needs of users for document exploration at a much deeper level.

1 Introduction

In the Document Understanding Conference (DUC) for 2004, CL Research primarily conducted experiments in examining documents represented in its Knowledge Management System (KMS). While we generated and submitted summaries for all five DUC tasks, we focused more on our underlying technology, rather than trying to optimize KMS to perform this year's tasks. Notwithstanding, we were able to achieve reasonable levels of performance while making some notable changes in KMS. Our performance continued to validate the approach started in DUC 2003 (Litkowski, 2003) of relying on massively XML-tagged representations of documents.

In DUC 2004, we continued the evolution of our tagging efforts by including word-sense disambiguation and semantic typing in the XML representation. We made use of this additional information in identifying biographical information for Task 5. We were able to extend this functionality to include a broader capability for examining the entities and events within and across documents. We also developed techniques for recognizing multiword units from multiple mentions of an entity and for building headlines out of document entities, relations, and events. In both cases, we were able to exploit characteristics of the expanded XML representations of texts.

Section 2 presents a description of the DUC 2004 tasks. Section 3 provides an overview of the KMS, with an emphasis on the extensions made during our preparations for DUC 2004. Section 4 describes the procedures used to perform each of the DUC tasks. Section 5 presents and analyzes the results and section 6 describes how changes made to KMS provide an improved capability for a user to examine documents from many different perspectives.

2 DUC 2004 Task Descriptions

DUC 2004 consisted of five tasks. Task 1 was to create very short summaries with a maximum length of 75 characters of 500 newspaper and newswire articles; these summaries can be construed as headlines, although participants were allowed to use any format (including keyword lists). Task 2 was to produce summaries with a maximum length of 665

characters for 50 clusters of 10 documents each; these summaries were to be general, and not focused on any particular aspect of the documents (as was the case in DUC 2003). Participants could submit up to three runs for each of these tasks, with each ranked as to priority.

Task 3 was similar to Task 1, except that the document set consisted of 24 clusters of 10 documents each. In this task, two sets of source text was provided: one consisted of machine translated texts and the other consisted of hand-generated translations of Arabic source text. For the machine translations, two sets were provided for each source document, one from the Information Sciences Institute (ISI) and one from International Business Machines (IBM). Further, participants were provided with a "best" translation and approximately 10 "variants" for each document. The "best" and the "variants" were based on a score for each sentence that was generated. CL Research used only the "best" translations from ISI and did not attempt to make use of the "variants". Task 4 used the same document set as for Task 3, except that the task was to general multidocument summaries for each of the 24 clusters. Participants were required to submit at least two runs, one using a machine translation and the other using the hand-generated translations.

Task 5 was similar to Task 2 in requiring summaries for each of 50 clusters of 10 documents. However, this task was designed to produce a biographical summary, with a particular person named for each document cluster. Thus, each document set was chosen so that all the documents contributed to answering the broad question "Who is X?", where X is the name of a person (e.g., "Stephen Hawking" and "Theodore John Kacynski").

The documents for Tasks 1, 2, and 5 came from the AQUAINT Corpus of English News Text on two CD-ROMs containing documents from *Associated Press Newswire*, *New York Times Newswire*, and *Xinhua News Agency*. The Arabic documents for Tasks 3 and 4 came from the Agence France Press (AFP) Arabic Newswire (1998, 2000-2001). The texts for all tasks were tagged to identify document source information and the textual elements to be processed.

Human assessors first hand-generated four summaries for each of the tasks. For Tasks 2 and 5, a single summary was deemed to be the model against which participating systems would be judged. Each of the 665-character model summaries were analyzed into "meaning units", usually corresponding to sentence clauses conveying short nuggets of information. Each of the non-selected hand summaries and each summary generated by a participating system were then scored by the assessors. (The other hand summaries were judged as well to provide an indication of the variability among human summarizers.)

As initially specified, Tasks 1 to 4 were to be scored only with an automatic (–gram) matching script, ROUGE (Recall-Oriented Understudy for Gisting Evaluation).[1] ROUGE compares a submitted summary with a manual summary, after stemming each word in the summaries, counting the proportion of words in submission with the words in the manual summaries. In addition to –gram matching, ROUGE was extended to count the "longest common substring" and to a weighted form of the longest common substring.

For Task 5, scoring involved assessors examining each "peer unit" submitted by a system (usually a full sentence). The assessor then judged which meaning units were contained in the peer unit, along with a percentage estimate of how much of the meaning unit was covered. After all peer units were judged, the mean coverage of the submission was computed as the sum of each individual meaning unit's score divided by the number of meaning units. Mean coverage (a number between 0.0 and 1.0) represents the score for each submission for a document cluster. Scores for the task was then computed as the average mean coverage over all document clusters.

To provide an opportunity for comparing human and automatic scoring, Task 2 was also scored using the same method as Task 5 and Task 5 was also scored with ROUGE.

For Tasks 5 (and then 2), overall peer quality was judged using seven quality questions, each judged on a scale from 1 to 5. The first question asks whether the summary builds from sentence to sentence to a coherent body of information about the topic. The second and third questions ask how much repetitive or useless information is in the summary. The fourth and fifth questions ask the coherence of noun phrases (whether the summaries use clear and unrepetitive references). The last two questions ask whether the sentences are grammatical or contain datelines or other information that impairs the readability of the summary.

For Task 5, the assessors rated the responsiveness of the summary to the question on a five-point scale, from "unresponsive" to "fully responsive".

Participating teams were provided with the results of the scoring for all teams, in a form suitable for further analysis. Not all teams participated in all tasks: task 1 (18), task 2 (16), task 3 (11), task 4 (11), and task 5 (14). Identities of the 20 teams were not

[1]Available from http://www.isi.edu/~cyl/ROUGE.

revealed. CL Research participated in all tasks, submitting two runs for each task, except Task 2, for which we submitted only one run.

3 System Description

CL Research's Knowledge Management System consists of three main components: (1) conversion of documents in various formats to a standard format identifying text portions; (2) parsing and processing the text into an XML-tagged representation, and (3) document querying, involving use of the XML-tagged representation for NLP applications such as text summarization, question answering, information extraction, and other analyses. The overall architecture of the system is shown in Figure 1.

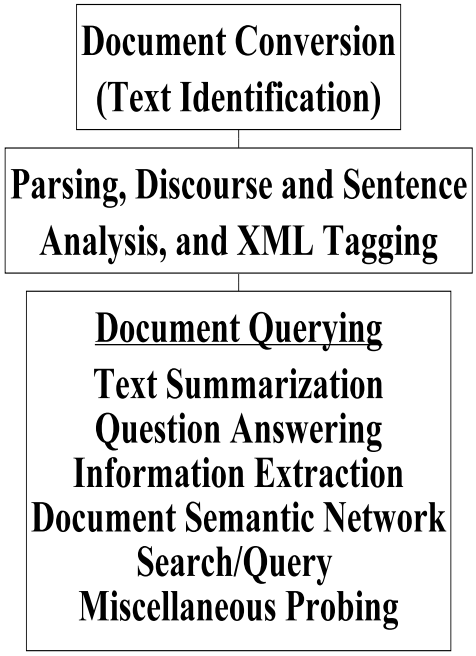

Figure 1. Architecture of Knowledge Management System

EXtensible Markup Language (XML) was chosen as the underlying representational mechanism, primarily because it provides a more natural vehicle for retaining the tree structure produced in parsing sentences. XML also provides a convenient mechanism for retaining, in attributes attached to tree nodes, annotations attached to parse tree nodes. The XML representation conveniently acts as an intermediate database of structured text, without the need to invoke the overhead of structured databases (i.e., conversion into and extraction from these databases).

A valid XML document is a tree and the entire representation can readily be designed on this tree structure. An entire collection (or any subset of documents) can be represented as one tree; the next level of the tree represents each document. At the next level, each document may be represented as a set of sentences, each of which may then be subdivided into sentence segments or clauses (elementary discourse units), which are then broken down into traditional parse trees, ending in leaf nodes corresponding to the words in the sentences. Each node in the tree may have associated attribute names and values.

A key part of the XML design philosophy is the ability to transform an XML file into usable output for display or other purposes (e.g., populating a database). This is accomplished via XML stylesheet language transformations (XSLT). XSLT is based on XPath expressions, which specify the path from the top of the XML tree to some intermediate or leaf node. Automatically generated XPath expressions are used extensively in probing documents for summarization, question answering, information extraction, general searches or queries, and overall document structure. Unlike traditional search engines, which treat text only sequentially (e.g., exact strings or proximity searches), XPath expressions combine traditional search mechanisms with structured searches. For example, in answering a *when* question, an XPath expression can look for sentences containing both the strings in the question and the elements within those sentences that have been tagged as time elements, regardless of how or where they may be expressed in the sentence.

3.1 Document Conversion

The first problem in processing documents is identifying the actual text from metadata and formatting instructions. The plethora of document formats is somewhat daunting, so an intermediate solution has been taken of converting documents in these different formats to web pages (generally in HTML format). Many major word processing software packages (Microsoft Word, WordPerfect, and freely available PDF converters) have options to convert documents to web pages. The first component of KMS converts web pages into an XML format with a document identifier and text to be processed.

Document conversion is generally quite rapid, taking only 15 or 20 seconds. KMS has been extended since DUC 2003 to include a component for interactive querying of the Google search engine. Results from a search can be selected and downloaded into repositories. Documents in the repositories can then

be processed into an XML representation (described below). Once in this format, KMS can produce keyword lists, headlines, general or focused summaries of any length within one or two minutes. In addition, KMS includes functionality to probe the contents of single or multiple documents from a variety of perspectives.

3.2 Text Parsing and Processing

The second component of KMS parses and processes text into an XML-tagged representation. This step is the most time-consuming part of KMS, although it still is quite rapid, processing in excess of 400 sentences per minute. For the processing of web pages from Google News, for example, it took longer to select desired articles than it took to process them.

The parsing and processing component consists of three modules: (1) a parser producing a parse tree containing the constituents of the sentence; (2) a parse tree analyzer that adds to a growing discourse representation of the entire text and identifies key elements of the sentence (clauses, discourse entities, verbs and prepositions) and captures various syntactic and semantic attributes of the elements (including anaphora resolution and WordNet lookup); and (3) an XML generator that uses the lists developed in the previous phase to tag each element of each sentence in creating the XML-tagged version of the document.

3.2.1 Parser

Text processing begins by splitting the text into sentences. The splitter is very efficient and accurate, particularly dealing with abbreviations and initials that frequently result in sentences being improperly split. After splitting, each sentence is submitted to the parser. The use of the Proximity parser was continued, described in more detail in (Litkowski, 2002a). As described there, the parser output consists of bracketed parse trees, with nonterminal nodes corresponding to sentence constituents such as clauses, noun phrases, and prepositional phrases, and leaf nodes describing the part of speech and root for each sentence word. Annotations, such as number and tense information and attachments points of noun and prepositional phrases, may be included at any node.

3.2.2 Discourse and Sentence Analysis

The sentence parsing in the CL Research system is part of a broader system designed to provide a discourse analysis of an entire text; this system is being used for processing encyclopedia articles, historical texts, as well as the newswire or newspaper texts in DUC, TREC, and the RST treebank (Linguistic Data Consortium, 2002).

After each sentence is parsed, its parse tree is traversed in a depth-first recursive function. During this traversal, each non-terminal and terminal node is analyzed, making use of parse tree annotations and other functions and lexical resources that provide semantic interpretations of syntactic properties and lexical information.

At the top node in the tree, prior to iteration over its immediate children, the principal discourse analysis steps are performed. Each sentence is treated as an event and added to a list of events that constitute the discourse. Data structures used for anaphora resolution are first updated. Next, a quick traversal of the parse tree is performed to identify discourse markers (e.g., subordinating conjunctions, relative clause boundaries, and discourse punctuation) and to break the sentence down into elementary discourse units. The sentence's verbs are identified and maintained at this stage, to serve as the bearers of the event for each discourse unit.

After the initial discourse analysis, the focal points in the traversal of the parse tree are the noun phrases. When a noun phrase (discourse entity) is encountered, its constituents are examined and its relationship to other sentence constituents are determined. The relationship analysis identifies the syntactic and semantic relations which characterizes the entity's role in the sentence, and a governing word to which the entity stands in the semantic relation (usually a verb or preposition, and if a preposition, where it is attached).

Each noun phrase is added to a list of discourse entities for the entire text, i.e., a "history" list. As each noun phrase is encountered, it is compared to discourse entities already on the history list. This comparison first looks for a prior mention, in whole or in part, to determine whether the new entity is a coreferent of a previous entity (particularly valuable for named entities). If the new entity is an anaphor, an anaphoric resolution module is invoked to establish the antecedent. A similar effort is made to find antecedents for definite noun phrases. The noun phrase's constituents are examined for numbers, adjective sequences, possessives (also subjected to the anaphoric resolution module), genitive determiners (made into separate discourse entities), leading noun sequences, ordinals, and time phrases.

If a noun phrase is part of a prepositional phrase, a special preposition dictionary is invoked in an attempt to disambiguate the preposition and identify its semantic type. This module identifies the attachment

point of the preposition and uses information about the syntactic and semantic characteristics of the attachment point and the prepositional object for this disambiguation. The preposition "definitions" in this dictionary are actually function calls that check for such things as literals and hypernymy relations in WordNet. A list of all prepositions encountered in the text is maintained as the text is processed. (See Litkowski (2002b) for further details.)

Just prior to our participation in DUC 2004, KMS was extended to integrate word-sense disambiguation of all content words (nouns, verbs, adjectives, and adverbs). At present, this implementation is preliminary (see Litkowski, 2004b) and uses only WordNet as the sense inventory. However, based on the disambiguation, we assign a broad semantic type to each noun and verb.

3.2.3 XML Tagging

As indicated above, the text analysis module develops four lists: (1) events (the discourse segments), (2) entities (the discourse entities), (3) verbs, and (3) semantic relations (prepositions and punctuation). These lists are used in a traversal of the entire document, tagging each sentence with information from items associated with each of its elements. Each document consists of one or more tagged segments, which may include nested segments. Each discourse entity, verb, and preposition in each segment is then tagged. A segment may also contain untagged text, such as adverbs. Each item on each list has an identification number (used in many of the functions of the text analysis module). As indicated above, each segment (and subsegment), discourse entity, verb, and preposition may have associated attributes.

For segments, the attributes include the sentence number (if the segment is the full sentence), a list of subsegments (if any), the parent segment (if a subsegment), the text of the segment, the discourse markers in the sentence, and a type (e.g., a "definition" sentence or, for nested segments, the type of clause). For discourse entities, the attributes include its segment, position in the sentence, syntactic role (subject, object, prepositional object), syntactic characteristics (number, gender, and person), type (anaphor, definite or indefinite), semantic type (such as person, location, or organization), coreferent (if it appeared earlier in the document), whether the noun phrase includes a number or an ordinal, antecedent (for definite noun phrases and anaphors), and a tag indicating the type of question it may answer (such as *who*, *when*, *where*, *how many*, and *how much*). Each noun is also tagged with its WordNet sense number. For verbs, the attributes include its segment, position in the sentence, the subcategorization type (from a set of 30 types), its arguments, its base form (when inflected), its grammatical role (when used as an adjective), and a WordNet sense number. For prepositions, the attributes include its segment, the type of semantic relation it instantiates (based on disambiguation of the preposition) and its arguments (both the prepositional object and the attachment point of the prepositional phrase).

The resultant XML-tagged text for individual documents are combined into one overall file of documents, each with a tag for the document identifier. For DUC 2004, the document clusters for Tasks 1 and 2, Tasks 3 and 4, and Task 5 were combined into 50, 24, and 50 files each (usually containing 10 documents). These are the files used for performing the DUC tasks. Parsing and processing these 124 files (i.e., the three steps described in this section) took approximately 100 minutes in total.

3.3 Document Querying

The third component of KMS examines XML-tagged documents produced by the parsing and processing component. Broadly, this component consists of a graphical user interface that enables a user to generate summaries, answer questions, extract information, or probe the content of the documents. The XML files can be viewed (with retention of the nested structure) in Microsoft's Internet Explorer, but this does not allow any systematic examination of the data.

In KMS, a user can explore the contents of a repository along several dimensions. Initially, the KMS interface only identifies the documents contained in a repository. A usual first step in examining the documents is to create a keyword list and a headline describing each document. The user can select all documents in a repository and create these "short" summaries in about 10 seconds (for documents of the size used in DUC). KMS remembers these summaries in an XML file, so that they can be redisplayed immediately as a user switches back and forth among repositories.

The user can then explore the contents of a repository, either one document at a time or by selecting multiple or all documents. KMS includes three main methods of exploration: (1) asking fact-based questions, (2) summarizing either generally or topic-based, and (3) probing the contents by the semantic types of entities, relations, and events. Each

of these tasks is implemented by using XPath expressions to query the document (i.e., select and manipulate nodes of the XML tree).

In general, each KMS task selects particular node sets (e.g., sentences meeting particular criteria, all discourse entities labeled as persons, all discourse segments labeled as subordinate clauses, or all prepositions labeled as locational). The node sets are then subjected to analysis to produce final output corresponding to the task (e.g., summaries or answers to questions).

4 Summarization for DUC 2004

In general, all summarization in KMS begins with a frequency analysis of discourse entities. A simple XPath expression retrieves all discourse entities and these are then examined in turn to develop a frequency count of the words in them. However, the KMS method of counting is somewhat different from traditional methods used in information retrieval. First, the traditional use of the stop list is employed to remove frequent words (like articles). Next, the entity is examined to determine whether it is a referring expression, i.e., whether it has an antecedent (pronouns, co-referring expressions, or definite noun phrases). For referring expressions, the words in the antecedent are counted instead of the words in the referring expression.

All summarization also requires the specification of a summary length. While the user can specify any value, it is strictly enforced. KMS terminates the summarization when the next piece of information to be added would result in the length of the summary exceeding the target.

Except for keyword generation, summarization is based on extraction of sentences from the document cluster. Sentences for all documents are ranked, either based on the frequency analysis described above or the occurrence of words in the topic or viewpoint specification. Only checks for sentence duplication were made; methods for assessing redundancy or substituting antecedents for pronouns (or vice versa) had not been implemented. Methods used in DUC 2004 were not as elaborate as those used in DUC 2003, since we were in the process of reimplementing our summarization modules and the short length of the summaries usually meant that only 2 or 3 sentences could be used.

Summaries generated using KMS for submission usually required only a second or two. Total processing time for the entire DUC submission was about thirty minutes. As with the keyword or headline generation, summaries generated in KMS are saved in XML, along with a specification of the options in effect. The actual submission was created from these files using a Perl script.

4.1 Tasks 1 and 3: Keyword Lists and Headline Generation

Although the texts had different origins (English newswire, machine translations, or hand translations), the tasks were accomplished identically in KMS. In DUC 2003, we had not used keyword lists as document summaries. In preparation for DUC 2004, we implemented some basic methods for keyword generation using the frequency analysis. We tested the lists generated for the comparable task in DUC 2003 using ROUGE, with results considerably better than our submission last year. In comparing these results with those of other participants last year, our score would have put us in second place.

In generating the keywords, we observed several characteristics which made those lists less than ideal. In particular, we noticed that several of the most frequent words originated from a single phrase or multiword unit. After determining an initial keyword list, we performed a second analysis to identify phrases. To do this, we used the initial list and identified all the discourse entities (or their antecedents) containing these words and determined when multiple words from the initial list occurred. We then reordered the keyword list, putting together whatever phrases we could find. In general, the results looked intuitively correct and we used this method in our official submissions for the Priority 1 run of Task 1 and both runs of Task 3.

For the Priority 2 run of Task 2, we created headlines for each of the newswire documents. In DUC 2003, we had tried to construct a headline by creating a sentence out of the syntactic role of the most frequent words (i.e., as subject, verb, object, or prepositional phrases). In DUC 2004, we started instead by using the extraction method described above to select a sentence having the highest score in matching the frequency analysis for the document. Using the XML tagging for this sentence, we attempted to construct a new sentence based on the main syntactic elements, looking for the subject of the main clause, the main verb, the object of the main verb, and prepositional phrases attached to the verb or object. We used the maximum length of 75 characters before cutting off the construction. As a fallback, we simply cut off the sentence at 75 characters.

4.2 Task 2 and 4: General Summaries

In these two tasks, we used the general frequency-based method for ranking sentences across multiple documents. We used these sentences in turn to construct the overall summary for the document cluster. As indicated, we made only one test in selecting sentences, avoiding complete duplicates (not uncommon since many of the document clusters contained virtually identical texts, when a document was merely a slightly later version of an earlier newswire report). We included sentences until adding the next sentence would exceed the 665-character restriction.

For Task 2, we submitted only one run. For Task 3, we submitted two runs, as specified. The Priority 1 run was a summary based on the machine translation output; the Priority 2 run was based on the hand translation.

4.3 Task 5: Summaries Focused by "Who Is X?"

In this task, we used the KMS functionality for topical focus to generate a summary. The topical focus in this case was simply X, the person's name. In this respect, this task was identical to last year's tasks where summaries were focused by events or viewpoints. That is, sentences were extracted based on scoring them for inclusion of the person's name. A sentence would be scored more highly if it contained coreferring expressions (such as anaphors or use of a reduced expression such as the person's last name). These summaries constituted our Priority 1 submission. For our Priority 2 submission, we used the general summary functionality, i.e., using the frequency analysis for the entire document cluster.

As indicated earlier, we had recently extended the KMS functionality to enable document exploration by entity. In particular, we had initially added this capability to look specifically at discourse entities that had been tagged as being a person, either directly or by use of the WordNet-based semantic tagging. We combined this functionality with the question-answering functionality for answering *who* questions in TREC (Litkowski, 2004a). In this combination, the selection of sentences is first focused by topic (i.e., the person's name). Then, all discourse entities that include the person's name (or an antecedent containing the name) are identified. For example, discourse entities such as "Professor Hawking" and "Dr. Hawking" would be identified as referring to the same person. Finally, each sentence is evaluated according to whether it contains a "definitional" pattern. Thus, sentences containing phrases like "Hawking, the Lucasian professor", "Hawking, who is the Lucasian professor", or "Hawking is the Lucasian professor" would be given higher scores. Due to time constraints, we were unable to create summaries based on these rankings. However, inspection of the rankings seem intuitively correct, ranking definitional sentences higher than non-definitional sentences (e.g., "He said that he ...").

5 Results and Analysis

The results on the five tasks as scored by ROUGE are shown in Table 1. The results show only CL Research's rank in comparison to other systems having the same priority. If all runs were combined for each task, the CL Research rank would be lower; the combined would be more difficult to compare and discuss.

Table 1. DUC 2004 Task ROUGE-1 Results

Task	Priority	Score	Rank	Range
1 (18)	1	0.212	4	0.121 - 0.250
2 (16)	1	0.303	14	0.242 - 0.382
3 (11)	1	0.182	9	0.165 - 0.236
3 (10)	2	0.237	4	0.149 - 0.259
4 (11)	1	0.301	8	0.189 - 0.388
4 (11)	2	0.346	9	0.234 - 0.416
5 (14)	1	0.297	13	0.263 - 0.355

The results for Tasks 1 and 3 indicate that the KMS generation of keywords does reasonably well. Several opportunities for improvement have been identified and would suggest that these keyword lists might be quite satisfactory as providing an initial characterization of a document. The lower performance on generating keywords for the machine translations of documents (Task 3, Priority 1) suggests our system does not cope as well with these renditions.

For the more general summarizations (Tasks 2, 4 (2), and 5), our scores suggest that KMS is performing at a level not too different from the average. Considering the relatively limited implementation in which very little analysis is performed on the overlap between sentences, the simple extraction process produces results better than we had anticipated.

Table 2 shows our results for Tasks 2 and 5 based on human assessor's judgments of the mean coverage of the summaries against model units. Although the ranks are somewhat similar to those in the ROUGE scoring, the absolute scores are somewhat different.

This suggests that the correspondence between the automatic scoring program and human assessment is not quite the same.

Table 2. Mean Coverage of Model Units			
Task	**MC**	**Rank**	**Range**
2 (16)	0.216	11	0.049 - 0.303
5 (14)	0.158	13	0.145 - 0.241

Table 3 shows our results for Task 5 on the quality questions. As indicated, these questions were addressed for each summary and a score was given between 1 and 5, with 1 being the best. The table shows CL Research's average over all questions, our rank out of 14 systems, and the range of averages for the participating systems.

Table 3. Quality Questions (Task 5)		
Question	**Quality**	**Range**
1 (Coherence)	3.12 (3)	2.90 - 4.52
2 (Uselessness)	2.70 (4)	2.32 - 4.04
3 (Repetitive)	1.58 (7)	1.16 - 1.98
4 (Referents)	2.04 (8)	1.24 - 2.96
5 (Same entities)	1.70 (8)	1.18 - 2.42
6 (Ungrammatical)	1.36 (4)	1.22 - 2.76
7 (Formatting)	1.46 (5)	1.30 - 2.54
Overall	1.99 (3)	1.82 - 2.78

These results provide a reasonable overall picture of how KMS performs with its extractive summaries. In general, and overall, KMS performs better than most systems. Question 1 shows that most systems did not provide a very coherent biographical picture; question 2 indicates that a lot of information was not useful or relevant to a short biography. For questions 3, 4, and 5, KMS' performance slipped somewhat, reflecting the fact that we had not performed any intersentential analysis. For questions 6 and 7, our average scores and ranks indicate that using extraction minimized the problems with grammaticality and formatting. The overall score indicates that the extraction mechanism, based on frequency analysis, performs better than most systems. In general, these results suggest that further analysis of the sentential components can provide significant improvements.

6 Discussion and Future Developments

In participating in DUC 2004, we conducted some further experiments in using an underlying XML representation. We did not pursue the tasks as vigorously as possible, in part because we did not view them as extrinsically interesting. We were concerned about the use of the automatic scoring program and intend to make further analyses of its applicability and possible extension.

In preparing for our participation with the Knowledge Management System, we became concerned that the kinds of summaries being produced may not reflect the needs of users for deeper document understanding. As a result, our focus was on the development of deeper document analysis and querying functionality.

References

Linguistic Data Consortium (2002). The Rhetorical Structure Theory Discourse Treebank. ISBN 21-58563-223-6. Philadelphia, PA.

Litkowski, K. C. (2002a) "CL Research Experiments in TREC-10 Question Answering", in Voorhees, E. M. and Harman, D. K. (eds) *Information Technology: The Tenth Text REtrieval Conferenence (TREC 2001)*, NIST Special Publication 500-250. Gaithersburg, MD: National Institute of Standards and Technology, pp. 122-31.

Litkowski, K. C. (2002b). Digraph Analysis of Dictionary Preposition Definitions. Word Sense Disambiguation: Recent Success and Future Directions. Philadelphia, PA: Association for Computational Linguistics.

Litkowski, K. C. (2003). Text Summarization Using XML-Tagged Documents. Available: http://www-nlpir.nist.gov/projects/duc/pubs.html.

Litkowski, K. C. (2004a, in press). "Use of Metadata for Question Answering and Novelty Tasks. In E. M. Voorhees & L. P. Buckland (Eds.), *Information Technology: The Twelfth Text REtrieval Conferenence (TREC 2003)*, NIST Special Publication. Gaithersburg, MD: National Institute of Standards and Technology.

Litkowski, K. C. (2004b, in press). Explorations in Disambiguation Using XML Text Representation. Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text. Barcelona, Spain.