Text Summarization Using XML-Tagged Documents

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

Abstract

CL Research's participation in the Document Understanding Conference extended the framework used in the TREC 2003 question-answering track, in which texts are parsed and processed into XML-tagged documents where sentence elements are marked with discourse, syntactic, and semantic attributes. This extension was made primarily to test the viability of using XML-tagged documents for summarization. The extension of the Knowledge Management System was able to take advantage of these attributes in implementing various text summarization capabilities. While implementation of these capabilities made little use of current summarization technologies, the CL Research system performed at a higher than expected level, finishing first in mean length-adjusted coverage for summaries against a provided viewpoint. The system performed less well on this measure in event summarization (tenth), novelty summarization (fifth), and headline generation (eleventh), but performed well on quality measures (finishing first among teams participating in all tasks)) and relevance (finishing first on each summarization task, with all sentences in these tasks judged relevant to the topic).

The system's performance arises primarily from the use of an "antecedent" tag attached to referring expressions (such as pronouns) within a document. In particular, when accumulating word frequencies, the antecedent was used instead of the referring expression; thus, instead of treating a pronoun as a word in the frequency count, its antecedent was used. The system's performance demonstrates the basic viability of using XML-tagged documents. Many options were explored in setting up the summarization capability, indicating considerable flexibility in examining documents from many perspectives and considerable potential in possible further improvements in the system. The system can indicate not only that a concept appears frequently within a document, but also how it is used (e.g., as subject, verb object, or prepositional object). More specifically, the availability of considerable structural information within documents permits a relatively simple examination of phenomena that have been used in text summarization, as well as the creation of a document's semantic network.

1 Introduction

In the TREC 2002 question-answering track (Litkowski 2002a), CL Research presented unofficial results which suggested that exploiting XML-tagged documents could potentially yield results equaling the best systems. It was also suggested that XML-tagged documents could be useful in such NLP applications as text summarization. The paper particularly described how the XML Analyzer provided access to the XML trees representing one or more documents, allowing examination and manipulation of low-level nodes in the trees, with utility for many NLP applications..

In response to the call for participation in the Document Understanding Conference (DUC), the XML analyzer was first extended to allow more detailed examination of word frequency and the kinds of syntactic and semantic relations present in the documents. further modifications were then made to the XML Analyzer to enable the types of summarizations called for in the DUC 2003 tasks (three of which required use of contextual information). Since CL Research had not participated in previous DUCs, it was necessary to implement more general summarization capabilities first, before responding specifically to the DUC 2003 tasks.

As a result of the changes to the XML Analyzer, the summarization functionality was embedded inside a more comprehensive processing system, known as the Knowledge Management System (KMS). This system is designed to provide a single integrated text processing capability: (1) converting arbitrary document types into an XML markup identifying text segments, (2) parsing and processing the text into XML-tagged documents, and (3) providing a single interface allowing a variety of text analysis options, including summarization, question-answering, and information extraction.

Section 2 presents a description of the DUC 2003 tasks. Section 3 provides an overview of the KMS,

with emphasis on how the XML output is generated and with a general description of the analysis modes used to examine the output. Section 4 describes the procedures used to perform each of the DUC tasks. Section 5 presents and analyzes the results and section 6 describes anticipated changes to the KMS and how these changes will provide an integrated environment that will allow a user to examine documents from many different perspectives.

2 DUC 2003 Task Descriptions

DUC 2003 consisted of four tasks. Task 1 was to create very short summaries (of approximately 10 words) of 624 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 100 word multidocument summaries focused by events for 30 clusters of documents; participants were provided with the documents and a description of a seminal event, containing what, who, where, and when statements and a topic explication (a paragraph) for each event. Task 3 was to produce 100 word multidocument summaries focused by viewpoints (a natural language string no longer than a sentence) for 30 clusters of document; participants were provided no additional information about these viewpoints. Task 4 was to produce 100 word multidocument summaries in response to a question for 30 clusters of documents; participants were provided with a question, a narrative describing relevant documents, and a set of sentences deemed relevant; the task essentially required a summary of the relevant sentences.

The documents for the first three tasks 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*. These documents were stored with formatting tags. The documents for the fourth task came from the TREC CD-ROMs containing documents from the *Foreign Broadcast Information Service*, *Los Angeles Times*, *Financial Times*, and the *Federal Register*; each document type had its own document formatting description. Participants were provided with the 2000 documents used in DUC 2003.

Human assessors first hand-generated three summaries for each of the tasks. A single summary was deemed to be the target against which participating systems would be judged. Each of the 100-word target 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.)

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. Mean length-adjusted coverage was then computed as the mean coverage adjusted downward if a submission contained more words than the target size (by an amount equal to the target size divided by the size of the submission). Scores for each task could then be computed as the average mean coverage or the average mean length-adjusted coverage over all document clusters.

For tasks 2, 3, and 4, the quality of the summaries (i.e., summary coherence) was assessed using 12 quality questions. The assessors were asked to determine the number of instances (0, 1-5, 6-10, and >10) in a summary of confusing or disorganized text. The quality questions counted capitalization errors, incorrect word order in a sentence, number agreement between subject and verb, missing syntactic components (e.g., subject or main verb), unrelated sentence fragments joined together, articles (a, an, the) missing or used incorrectly, pronouns missing antecedents, nouns without clear referents, nouns that should have been replaced with a pronoun, dangling conjunctions, unnecessarily repeated information, and wrong sentence order. For each summary, the number of quality questions with a non-zero count was totaled.

For task 1, the assessors rated the usefulness of each headline on a five-point scale (from "no use" to "almost as good as the document"). For task 4, 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 human assessors' scoring for all 21 teams, in a form suitable for further analysis. Not all teams participated in all tasks: task 1 (13), task 2 (16), task 3 (11), task 4 (9), and all tasks (4). Identities of the 20 teams were not revealed. CL Research participated in all tasks.

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.

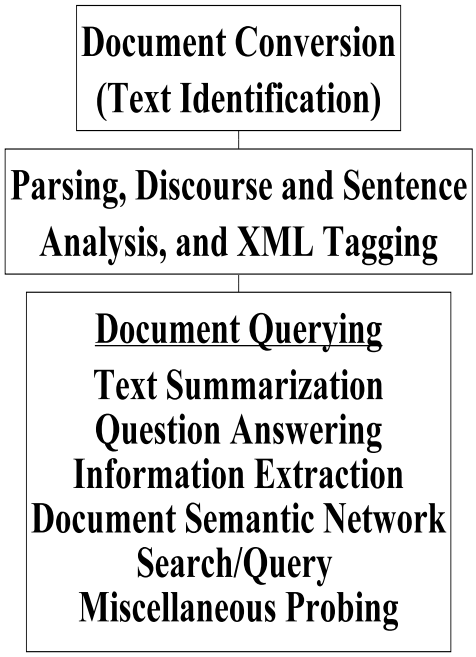

Figure 1. Architecture of Knowledge Management System

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. Sets of arbitrary web pages have been downloaded from Google News, folders containing 20 Word documents, and the PDF documents from last year's DUC proceedings. These documents have been converted , parsed and processed as described in the next section, and general and topic oriented summaries for these documents have been created in 5 to 10 minutes.

This step may be bypassed entirely when documents are already available in structured form

containing tags that adequately identify text portions to be processed. KMS enables a user to open a data type definition (DTD) file (used by SGML or XML) and specify the tags that identify individual documents in a file, document identifiers, and tags enclosing text material to be processed. This was the mechanism used in DUC 2003, since DTDs were available for the documents to be summarized.

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, 2002c). 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, scientific articles[1], as well as the newswire or newspaper texts in DUC, TREC, and the RST treebank (Linguistic Data Consortium, 2002). Frequently, the input has already been tagged (e.g., in SGML) and the processing may result in additional tagging.

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

[1] See http://www.clres.com/sa-articles.xml.

separate discourse entities), leading noun sequences, ordinals, and time phrases. Finally, an attempt is made to assign a semantic type to the head noun of the phrase using WordNet or an integrated machine-readable dictionary or thesaurus.

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.)

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). 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), and its grammatical role (when used as an adjective). 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, the document clusters for tasks 2, 3, and 4 were combined into 30 files each (usually containing 10 to 25 documents). For task 4, the source files rather than the provided relevant sentences were used, so that the relevant sentences could be examined within their original context. The files for tasks 2 and 3 were then used for these tasks and for task 1 as well. These are the files used for performing the DUC tasks. Parsing and processing these 90 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, known as the XML Analyzer, 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. The XML Analyzer loads the XML-tagged document (in one step) where it can then be probed using one of several analysis modes.[2]

Conventionally, those working with XML files develop XML stylesheets (XSLT) for portraying the data, perhaps embedded in interactive browser web pages. XSLT involves the use of XPath expressions to query the document (i.e., select and manipulate nodes of the XML tree). These XPath expressions underlie each of the analysis modes, including summarization. The detailed use of these expressions is not described in this section (see below for their use in summarization and see Litkowski (2002a) for details

[2]The graphical user interface can be seen at http://www.clres.com/kms.html.

on question answering). In general, each analysis mode 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 analysis mode (e.g., summaries or answers to questions). The development environment provides powerful tools for low-level access to the XML data.

The XML Analyzer also links various lexical resources into the examination of documents. This includes the use of a stop list containing words that occur very frequently (for some types of analysis), machine-readable dictionaries, and WordNet. The Analyzer also links back to the parser so that the user can enter new questions to be posed to the text or new points of view to slant a summary.

4 Summarization for DUC 2003

In general, all summarization in KMS begins with a frequency analysis of discourse entities. A simple XPath expression retrieves all discourse entities and then examines each in term 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, it is ascertained whether the discourse entity is a referring expression, i.e., whether it has an antecedent (pronouns, co-referring expressions, or definite noun phrases), and 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. The XML Analyzer terminates the summarization when the next piece of information to be added would result in the length of the summary exceeding the target.

At the time when the DUC submission was generated, the implementation was essentially based on extraction of key sentences. Only checks for sentence duplication were made; methods for assessing redundancy or substituting antecedents for pronouns (or vice versa) had not been implemented . Further, methods reflecting current research in summarization have not been implemented .

Summaries generated using KMS for submission usually required only a second or two. The total processing time for the entire DUC submission was about two hours.

4.1 Task 1: Headline Generation

For this task, which required a headline for each document in the file, an option in the XML Analyzer was invoked to produce automatically a summary less than or equal to 10 words for each document. The top 10 words of the frequency list were used to examine the discourse entities in each sentence (removing stop words), counting the occurrences of these words in the sentence. the sentence with the most hits was then selected and its length examined. If it was less than 11, the full sentence was submitted . If not, subsegments (clauses) were removed one at a time, attempting to reach the cutoff; if it did, the reduced sentence was submitted. If this process still left more than 10 words, the last words were removed down to a total of 10.

4.2 Task 2: Summaries Focused by Events

In this task, only the what statement was extracted from the event description and this statement was processed into an XML representation using the second component of KMS. The who, where, and when statements and the topic explication for each event were not used. A single file was created containing only the topic number and the what statements (e.g., “Kofi Annan visits Libya to appeal for surrender of PanAm bombing suspects”). This file was parsed and processed into an XML representation of each event characterization.

To create a summary for a particular topic, the topic number is selected and a reference list is generated containing all words in all discourse entities in that topic. Then, the document file corresponding to the topic number is loaded into the XML Analyzer. Each sentence in each document is assessed against the reference list and the number of words in the discourse entities of the sentence determined. The number of hits characterizes the importance of the sentence. Those with the highest number of hits are selected for inclusion in the summary. A simple check was made among the sentences to eliminate full duplicates (a common occurrence in the DUC documents) and then the sentences selected were ordered by document date and position within its document. This creates a minimal ordering of the sentences intended to maintain the sequential ordering of the event.

4.3 Task 3: Summaries Focused by Viewpoints

In this task, the viewpoint was parsed and processed into an XML representation using the

second component of KMS. A single file was created containing the document cluster number and the viewpoint (e.g., "The banks are using all tactics to keep the profitable ATM non-customer charge"). This file was parsed and processed into an XML representation of each viewpoint.

Summaries were created using the same steps as Task 2: selecting a cluster number, forming a reference set of discourse entities, examining each sentence for the number of hits of the reference entities, and picking those with the highest number for inclusion in the summary.

4.4 Task 4: Summaries Focused by a Question

This task was performed in essentially the same way as Tasks 2 and 3. As with Task 2, only the question was used; the narrative was not. A single file was created containing the questions (e.g., What drugs are being used in the treatment of Alzheimer's Disease and how successful are they?) and parsed and processed into an XML representation.

Because a set of relevant sentences from the documents was provided, some artificial processing had to be introduced into the KMS. The full text from which these sentences were taken was processed so that they were within their proper context. To accomplish this, the system had to be forced to select the provided relevant sentences and then use these sentences as the basis for creating the summary. (The question answering component of the KMS was not used.) These sentences were treated as the text to be summarized and a frequency count of the words in the discourse entities of these sentences was created .Those sentences containing the highest number of hits of the most frequent words were selected for the summary.

5 Results and Analysis

The results on the four tasks are shown in Table 1. In the table, the first column identifies the task number and the number of participating teams. The second column (MLAC) is the average mean length-adjusted coverage in the results provided by the assessors for 624 headlines and 30 summaries for each of the other tasks; this score gives a penalty for summaries that exceed the target. The third column is a variation on MLAC (MLAC') that gives a bonus for summaries that are less than the target size. This variation indicates the density of the coverage (i.e., per unit of text). The fourth column shows the CL Research rank among the participating systems. The fifth column shows the range of the variation in the MLAC' scores. (Ranks are similar for MLAC and MLAC'. CL Research's rank is identical for Tasks 2 and 3, and one position lower for the other tasks. The change in ranks for Task 2 is more significant for some systems.)

Table 1. Mean Length-Adjusted Coverage

Task	MLAC	MLAC'	Rank	Range
1 (13)	0.112	0.169	11	0.144 - 0.385
2 (16)	0.145	0.236	10	0.082 - 0.328
3 (11)	0.128	0.213	1	0.108 - 0.213
4 (9)	0.104	0.168	5	0.074 - 0.215

Table 2 shows the system's performance and rank on the other two primary measures: the number of non-zero quality counts in the summaries (each out of 360) and the number of sentences in the summaries that were judged not related to the topic (these judgments were not made for Task 1). Among the four systems performing all three tasks, the total of 128 was the lowest (compared to 144, 175, and 231). No other system had zero non-relevant sentences for all tasks (generally, all systems performed well on this measure). For Task 4, the system was ranked 5th of 9 on responsiveness of the summary to the question. For Task 1, the headlines were judged the least useful.

Table 2. Quality and Relevance

Task	Quality Count	Non-Relevant
2	55 (7)	0 (1)
3	41 (2)	0 (1)
4	32 (1)	0 (1)

For Task 1, an experiment used the most frequent discourse entities in a document and the most frequent verb for which these discourse entities was the subject. The intent was to search for appropriate prepositional phrases that might go along with the subject and main verb, but appropriate criteria were not developed at the time of submission. In retrospect, a semicolon delimited list of keywords based on the overall document frequencies could easily have been submitted. Headline creation seems to be a somewhat artificial task. Automatic keyword generation is more likely to convey the substance of the text.

For Tasks 2 and 4, both of which require delving down from a general statement of a topic to more specific instances, the lower performance was due to an inability to create summaries that generalized relevant sentences. That is, given a set of relevant sentences, some abstraction away from the details is required. The summarization only extracted detailed sentences and no procedures were in place to generalize from these specifics. On the other hand, the

scores would likely have improved by implementing procedures that simply removed redundant sentence components, enabling more sentences to be added to the summaries.

In Task 4, a degenerate XML tagging was obtained for one question. To handle this case, as well as to generalize the ability of the KMS to provide different slants on a set of documents, a capability was implemented to allow a user to enter arbitrary text and to use that text as the reference set, without parsing and creation of an XML representation. To create this reference set, each word of the reference text (removing stop words) is used to determine the number of hits in the documents' sentences. This method may work as well as the XML representation of the query. This result seems to stem from the fact that when only the discourse entities in the sentences are examined, they do not contain verbs, prepositions, and adverbs in the query, effectively negating their use in searching.

This functionality was also effectively used in examining the other topical explications and narratives provided with Tasks 2 and 4. That is, sentences or phrases from this additional text could be entered for use as reference text and could (virtually instantaneously) create summaries with a different slant, thus enabling a capability for examining a set of documents from many perspective. Sentences not included in the summaries could also be used as the reference text, allowing further document probes.

In working with the questions in Task 4, the reference text approach also identifies sentences answering the fact-based questions posed in the TREC question-answering track. Although this method doesn't extract the exact answer from the sentence, it may prove to be a useful adjunct to question answering by providing a candidate set of sentences.

6 Future Developments

As alluded to in the above discussions, the availability of XML-tagged representations of documents provides a significant set of opportunities for further improvements. The exploitation of the data has clearly only begun. Closer examination of the summarization literature is expected to provide many insights that can be effectively exploited with the XML representations. The KMS provides the opportunity to make these investigations much easier.

In addition, a capability for a more general examination of document structure has now been implemented in KMS . WordNet has been integrated in KMS so that a document's semantic network can be examined. This is accomplished by identifying the nouns and verbs in a document and hierarchizing them using WordNet as a reference point. The nouns and verbs are first collected according to their WordNet synsets and then a determination is made of what other WordNet relations are instantiated in the document. Thus, for example, the document contains an object noun and also a noun that is a part of that object, the WordNet relation is instantiated. The net effect of this process is the carving out a sub-WordNet for each document, thus constituting the document's semantic network. (See also Litkowski & Harris (1997) and Surdeanu & Harabagiu (2002).)

7 Summary

The performance of CL Research's Knowledge Management System in DUC 2003 corroborates the conjecture in Litkowski (2002a) that XML-tagged documents provide a useful basis for text summarization. The KMS looks forward to reading and summarizing the DUC 2003 proceedings.

References

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

Litkowski, K. C. (2002a). Question Answering Using XML-Tagged Documents. In E. M. Voorhees & D. K. Harman (eds.), *The Eleventh Text Retrieval Conference (TREC 2002)*. Proceedings of the Conference. Gaithersburg, MD., 328-337.

Litkowski, K. C. (2002b). Digraph Analysis of Dictionary Preposition Definitions. *Proceedings of the ACL SIGLEX Workshop: Word Sense Disambiguation*. Philadelphia, PA., 9-16.

Litkowski, K. C. (2002c) "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. and M. D. Harris (1997) *Category Development Using Complete Semantic Networks*, Technical Report 97-01. Gaithersburg, MD: CL Research.

Surdeanu, M. and Harabagiu, S. (2002). Infrastructure for Open-Domain Information Extraction. *Proceedings of DUC 2002 in Human Language Technology*. San Diego, California.