LCC's GISTexter at DUC 2006: Multi-Strategy Multi-Document Summarization

Finley Lacatusu, Andrew Hickl, Kirk Roberts, Ying Shi, Jeremy Bensley, Bryan Rink, Patrick Wang, and Lara Taylor[†]

Language Computer Corporation
1701 North Collins Boulevard
Richardson, Texas 75080

[†]Department of Linguistics
University of California, San Diego
La Jolla, California 92037

finley@languagecomputer.com

Abstract

In this paper, we describe how Language Computer Corporation's GISTEXTER question-directed summarization system combines multiple strategies for question decomposition and summary generation in order to produce summary-length answers to complex questions. In addition, we introduce a novel framework for question-directed summarization that uses a state-of-the-art textual entailment system (Hickl et al., 2006) in order to select a single responsive summary answer from amongst a number of candidate summaries. We show that by considering entailment relationships between sentences extracted for a summary, we can automatically create semantic "Pyramids" that can be used to identify answer passages that are both relevant and responsive.

1 Introduction

This paper introduces a new framework for question-directed summarization (QDS) that uses textual entailment in order to select a single responsive summary-length answer from amongst a number of automatically-generated summaries. We believe that by considering the entailment relationships that exist between sentences taken from multiple summaries, we can automatically construct hierarchical representations (or "Pyramids") that can be used to either select the most responsive summary from a set of candidates or to model the semantic content of an ideal summary response to a question.

We believe that complex questions cannot be answered using the same techniques that have so successfully been applied to the answering of "factoid" questions. Unlike informationally-simple factoid questions, complex questions often seek multiple different types of information simultaneously and do not presupposed that one single answer could meet all of its information needs. For example, with a factoid question like "*What is the average age of the onset of autism?*", it can be safely assumed that the submitter of the question is looking for an age (or a range of ages) which is conventionally associated with a first diagnosis of autism. However, with complex questions like "*What is thought to be the cause of autism?*", the wider focus of this question suggests that the submitter may not have a single or well-defined information need and therefore may be amenable to receiving additional supporting information that is relevant to some (as yet) undefined informational goal.

Over the past three years, complex questions have been the focus of much attention in both the automatic question-answering (Q/A) and multi-document summarization (MDS) communities. While most current complex Q/A evaluations (including the 2004 AQUAINT Relationship Q/A Pilot, the 2005 Text Retrieval Conference (TREC) Relationship Q/A Task, and the 2006 GALE Distillation Effort) require systems to return unstructured lists of candidate answers in response to a complex question, recent MDS evaluations (including both the 2005 and 2006 Document Understanding Con-

ferences (DUC) have tasked systems with returning paragraph-length answers to complex questions that are responsive, relevant, and coherent.

Returning multiple sentence answers – whether in the form of a list or a paragraph – poses two particular problems for systems that provide answers to complex questions. First, systems must be able to decompose complex questions into a set of simpler questions before they can be submitted to either a Q/A or a MDS system. While Q/A systems can use techniques based on keyword density and topic information to find relevant answers to even the most complex of questions, we expect that by decomposing complex questions into the sets of subquestions that they entail, systems can improve the average quality of answers returned and achieve better coverage for the question as a whole. In DUC 2006, we experimented with using three different techniques for decomposing complex questions, including question decomposition based on (1) keyword extraction and expansion, (2) syntactic question decomposition, and (3) semantic question decomposition.

In addition to understanding the information need of a question, systems must be sensitive to the fact that certain types of information are more valuable to users and should be presented before other less relevant answer snippets. (Passonneau et al., 2005) has argued that the ideal answers to complex questions can be organized into hierarchical structures (or "Pyramids") that reflect the relevance of semantic content units (SCUs) to a multi-document summary or to an answer to a complex question. In this paper, we suggest that recent work in recognizing textual entailment (Haghighi et al., 2005, Hickl et al. 2006) could be used in order to construct Pyramid representations that could be used to allow MDS systems to better recognize relevant information across documents or candidate summaries. We show that by using textual entailment to identify sentences that share the same semantic content, we can identify candidate summaries that may contain the most information relevant to a complex question.

The rest of this paper is organized in the following way. Section 2 presents an overview of GISTEXTER, Section 3 presents our Results from the DUC 2006 evaluations, and Section 4 presents our conclusions.

2 System Overview

In this section, we describe the architecture of GISTEXTER, a question-directed summarization system that uses multiple question decomposition and summarization strategies in order to create a single responsive summary-length answer in response to a complex question. The architecture of GISTEXTER is presented in Figure 1.

2.1 Question Processing

Questions submitted to GISTEXTER are sent to a *Question Processing* module, which uses three different question decomposition strategies in order to represent the information need of a question. First, questions are sent to a *Keyword Extraction* module, which creates a single unstructured query from each question by removing stopwords and alternating certain query terms using a set of lexical resources assembled for LCC's automatic question-answering (Q/A) systems (Harabagiu et al., 2005a). Next, questions are processed by a *Syntactic Decomposition* module, which uses sets of heuristics (first described in (Lacatusu et al., 2005) to recognize and extract embedded questions from each complex question. In addition to embedded questions, complex questions that feature conjoined phrases or lists of arguments are also broken down into individual questions that contain each of the conjuncts or arguments. Finally, questions, are submitted to a *Semantic Decomposition* module in order to identify subquestions that represent a different dimension of the information need encoded by a complex question. Unlike syntactic decomposition, which only extracts overtly-mentioned questions from a complex question, the process of semantic decomposition seeks to generate the set of informationally-simple questions that are entailed by a complex question. In previous work (Harabagiu et al., 2006), we described how complex questions can be decomposed by performing a random walk over a bipartite graph of sentences and relations derived from a collection of documents relevant to the complex question. Once a relation is identified in a complex question, a sentence is selected (at random) from the graph that also contains that relation. This sentence is then sent to an *Automatic Question Generation* module (first described in (Harabagiu et al., 2005b)) in order to pro-

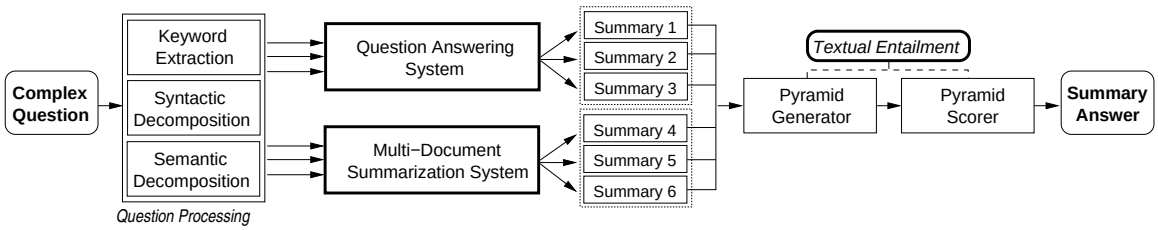

Figure 1: Architecture of GISTEXTER.

duce a well-formed natural language question that could be later submitted to a Q/A or a MDS system. Relations from the newly-generated subquestion are then extracted and used to identify other relevant sentences (and relations) from the graph which could be (in turn) to create new sets of decompositions. This process continues until there are no more nodes in the graph that can be traversed or until some pre-specified termination condition is met. [1] Table 1 illustrates examples of some of the syntactic and semantic decompositions produced for complex questions considered in DUC 2006.

Original Question	
Discuss measures that schools and school districts have taken to prevent violent occurrences and shootings, such as those in Littleton, Colorado and Jonesboro, Arkansas.	
Syntactic Decomposition	
$SynQD_1$	What measures have schools taken to prevent violent occurrences?
$SynQD_2$	What measures have school districts taken to prevent violent occurrences?
$SynQD_3$	What measures have schools taken to prevent shootings?
$SynQD_4$	What measures have school districts taken to prevent shootings?
Semantic Decomposition	
$SemQD_1$	What U.S. schools have had to perform a security lockdown?
$SemQD_2$	Who conducted drills over the summer in which police SWAT teams entered schools to put down school shootings?
$SemQD_3$	Which school district faced legal action for not preventing violence at its schools?
$SemQD_4$	Which schools now use metal detectors at special events?
$SemQD_5$	Who pulled a fire alarm and shot students filing out of Westside Middle School?
$SemQD_6$	Which Columbine High School student killed twelve other students?
$SemQD_7$	How many students were suspended after threatening to take over their school?
$SemQD_8$	How are school districts carefully regulating who can be school grounds?
$SemQD_9$	Why have local secondary schools beefed up security measures?
$SemQD_{10}$	What organization has held public hearings to prepare a report on improving school safety?

Table 1: Question Decompositions.

2.2 Sentence Retrieval and Ranking

Once question processing is complete, the output of each of these three question decomposition strategies are sent to both a (1) automatic question-answering (Q/A) system and a (2) multi-document summarization (MDS) system in order to generate a total of six candidate summaries for each individual complex question.

Question-Answering	Multi-Document Summarization
Strategy 1. Bag-of-Words	**Strategy 4.** Bag-of-Words
Strategy 2. Syntactic QD	**Strategy 5.** Syntactic QD
Strategy 3. Semantic QD	**Strategy 6.** Semantic QD

Table 2: Six Summarization Strategies.

GISTEXTER's *Question Answering* module uses keyword-based techniques developed for LCC's PALANTIR Q/A system (Harabagiu et al., 2005a) in order to retrieve sets of relevant sentences for each subquestion. Keywords are first extracted from each subquestion and are then sent to a sentence retrieval engine, which returns and ranks a list of sentences based on the number and proximity of keyword terms in each sentence. In contrast, GISTEXTER's *Multi-Document Summarization* module uses keywords extracted from each subquestion in conjunction with sets of relevant terms and relations derived from automatically-computed topic representations in order to retrieve a set of sentences for each candidate summary. As with our DUC 2004 and DUC 2005 systems (Lacatusu et al., 2004; Lacatusu et al., 2005), we followed (Lin and Hovy, 2000) in computing a weighted set of terms – known as topic signatures (TS_1) – based on the relative frequency of a given term in a relevant set of documents. In addition, we also followed (Harabagiu, 2004) in computing a weighted set of topic-relevant relations – known as *enhanced topic signatures* (TS_2) that identified relevant syntactic-based relations (e.g. noun-noun, adjective-noun, verb-noun) that exist between sets of topic signature terms. As in our 2005 DUC system (Lacatusu et al., 2005), weights associated with TS_1 terms and TS_2 relations were used to compute a composite *topic score* for each sentence in the document collection. Keywords were then extracted

[1] In our work, only the first 10 questions generated by the system were considered as potential semantic decompositions.

from each subquestion (as before) and used to retrieve a set of sentences; sentences were re-ranked based on their *topic score* before being incorporated into a candidate summary.

2.3 Summary Generation

A total of six candidate summaries were then generated by merging the top-ranked sentences retrieved by each summarization strategy into a single paragraph. Two types of optimizations were then performed in order to enhance the overall linguistic quality of summaries.

First, in order to reduce the likelihood that redundant information would be included in a summary, sentences selected for a candidate summary were clustered using k-Nearest Neighbor clustering based on cosine similarity. Following clustering, only the top-ranked sentence from each cluster was included in the summary. (An example of a cluster can be found in Table 3.)

D0641E	The dominant view is that the surface warming is at least partly attributable to emissions of heat-trapping waste industrial gases like carbon dioxide, a product of the burning of fossil fuels like coal, oil and natural gas.
	Greenhouse gas emissions - including carbon dioxide created by the burning of coal, gas and oil, are believed by most atmospheric scientists to cause the warming of the Earth's surface and a change in the global climate.
	Global warming is the change in the climate thought to occur because human activities add to a buildup of greenhouse gases such as carbon dioxide, methane and nitrous oxide.

Table 3: Clustering of Redundant Passages

In addition, we sought to enhance the referential clarity of summaries by developing a set of heuristics that would allow a system to automatically predict whether the antecedent of a pronoun could be (1) found in the current sentence, (2) found in the preceding sentence, or (3) not found without the use of a pronoun resolution system. While we are still committed to integrating a state-of-the-art coreference resolution system into the architecture of GISTEXTER, we believe that by being able to predicting which pronominal mentions could be included in a summary without adversely impacting referential clarity, we could enhance both the legibility and coverage of multi-document summaries without significant increasing the overhead required by a summarization system.

In a pilot study using a decision tree-based classifier, we found that we could predict both the form and location of the antecedent of pronouns occurring in subject position with approximately 85% F-measure. [2] We trained two classifiers using newspaper texts annotated with coreference information. For each instance of a pronoun, the first classifier learned whether an antecedent could be found in (1) the current sentence, (2) the preceding sentence, or (3) not in either the current or immediately preceding sentence. When antecedents were classified as occuring in either the current or preceding sentences, a second classifier was used to determine whether the candidate antecedent was (1) a full NP or (2) another pronoun.

We transformed the decision-tree rules into a set of heuristics for examining all the pronouns in a given sentence: for each pronoun contained in a summary sentence S_1, the heuristics determined whether S_1 should be (1) *kept* in the summary, (2) *add* the previous sentence, or (3) *drop* the sentence.

Since reflexive pronouns almost always co-occur with a non-pronominal antecedents in the same sentence, sentences containing reflexive pronouns were always kept in summaries. For all other third person pronouns, we developed two sets of procedures, depending on whether the pronoun occurred as subject (or contained inside the subject) of the main verb in the sentence.

When pronouns occur in non-subject position, our study showed that antecedents are often found in the same sentence as the pronoun. In order to ensure that the antecedent was not also a pronoun, we checked to see if the sentence contained other pronouns that matched in number and gender feature with the pronoun under consideration. If no other pronouns with these features could be found in the current, the sentence was kept; otherwise, the sentence was dropped from the summary.

When non-reflexive pronouns occurred in subject position, we knew the antecedent was likely to be in the previous sentence, although it could occur in another previous sentence in the discourse. In this case, if no pronoun could be found in the previous sentence that shared number and gender features with the pronoun under consideration, both the current sentence and a sentence immediately preceding the current sentence were added to the summary; if

[2] Precision: 74%, Recall: 100%

these conditions could not be met, the sentence was again dropped from the summary.

2.4 Automatic Pyramid Creation

Once a complete set of six candidate summaries have been generated, we used a state-of-the-art textual entailment (TE) system (described in (Hickl et al., 2006)) in order to select the candidate summary that best met the expected information need of the complex question.

Much recent work (Haghighi et al., 2005, Hickl et al. 2006, Harabagiu and Hickl, 2006) has demonstrated the viability of supervised machine learning-based approaches to the acquisition of robust forms of textual inference such as textual entailment or textual contradiction.

A text passage t is said to *textually entail* a hypothesis h whenever the meaning of t can be inferred from the meaning of p. In most TE systems, textual entailment is recognized using a classifier which evaluates the probability that a particular inferential relationship exists between two text passages using models based on a variety of statistical, lexico-semantic, or structural features. Even though most machine learning-based textual inference (TI) systems have not yet incorporated the forms of structured world knowledge featured in many logic-based TI systems, classification-based systems have consistently been among the top-performing systems in the PASCAL 2005 and 2006 Recognizing Textual Entailment Challenges (Bar-Haim, et al. 2006), with the best systems (such as (Hickl et al., 2006)) correctly identifying instances of textual entailment more than 75% of the time. In order to create a model Pyramid from the candidate summaries, each sentence from each of the six summaries were paired with every other sentence taken from the remaining summaries. Sentence pairs (e.g. $< S_1, S_2 >$) were then submitted to the TE system, which returned a judgment – either *yes* or *no* – depending on whether the semantic content of S_1 could be considered to entail the content of S_2. Entailment judgments output for each sentence pair were then used to group sentences into clusters that, when taken together, were expected to represent the content of a potential semantic content unit (or SCU). When a sentence S_1 was judged to entail a sentence S_2, S_2 was added to the cluster associated with the entailing sentence S_1 and the index associated with the cluster (assumed to be equal to the SCU weight) was incremented by 1. If entailment was judged to be bidirectional – that is, S_1 entailed S_2 and S_2 entailed S_1 – the two sentences were considered to convey roughly the same semantic content, and all sentence pairs containing S_2 were dropped from further consideration. When entailment could only be established in one direction – i.e. S_1 entailed S_2 but S_2 did not entail S_1, S_2 was considered to convey additional information not strictly found in S_1 and was permitted to create a cluster corresponding to a separate SCU. Finally, sentences that did not exhibit any entailment relationship with any other sentence were assigned a weight of 1. Table 4 provides a synopsis of the rules used to construct Pyramids from entailment judgments.

Entailment Judges	Action
$S_1 \vDash S_2$ and $S_2 \vDash S_1$	Add S_2 to the cluster containing S1; drop all other pairs containing S_2.
$S_1 \vDash S_2$ and $S_2 \nvDash S_1$	Add S_2 to cluster containing S_1.
$S_1 \nvDash S_2$ and $S_2 \nvDash S_1$	Do not add S_2 to cluster containing S_1.

Table 4: Textual Entailment Rules.

When the identification of TE is complete, sentence clusters were assembled into a model Pyramid based on their SCU weights. An example of the top levels of an automatically-generated Pyramid is presented in Table 5.

In Table 5, the original sentence used to construct each Pyramid cluster is presented along with its weight. While each node in an automatically-constructed Pyramid may contain multiple SCUs, every sentence added to a Pyramid cluster is expected to be textually entailed by the original sentence. While this may lead to situations where a sentence added to a cluster may only be entailed by a portion of the original sentence, we expect that, when taken together, each cluster will approximate a content unit that should be included in a summary answer.

2.5 Pyramid Scoring

TE was then used to score each of the six candidate summaries using the Modified Pyramid scoring algorithm described in (Passonneau et al., 2005). In order to begin this process, each SCU cluster was replaced by the sentence from the cluster that was assigned the highest retrieval score by either the Question-Answering or the Multi-Document Sum-

Weight	Clustering Sentence
4	Some districts, by contrast, have rejected the idea of detection systems, finding them an affront to educational openness, and are concentrating instead on a drumbeat of programs to make students feel more responsible for their school's safety and less reluctant to report violations.
4	Everyone at the ceremony will have to pass through a metal detector, and the crowd will be peppered with plainclothes officers on the lookout for a potential killer.
4	But this summer, even small districts that felt most distant from urban violence have been trying to find out which 'best practices' are affordable, he said.
3	There were bomb threats in New York area schools after the April 20 shootings at Columbine High School, but no serious incidents were reported.
3	While school safety measures such as metal detectors or additional security officers can be expensive, Stone estimated the cost of the software to be less than $2 per student.
3	Either way, the potential for violence at schools has weighed heavily on many administrators and has generated forums for public discussion in New York, New Jersey and Connecticut, which have been spared school shootings but not the heightened alarm.
3	'The date has them worried about a lot of copycats or kids who may try to send a very, very strong message,' said Curt Lavarello, executive director of the National Association of School Resource Officers, a group of K-12 school officers that has nearly doubled to 5,500 members in the last year.
3	In the most recent gun-related expulsions, 61 percent involved a handgun, 7 percent a rifle or shotgun, and the remaining 32 percent another type of firearm or explosives.Corresponds to: 17: Students with guns in school are required to leave school
3	Reacting to Columbine, New York state officials are requiring districts to report major violent acts - bomb threats, explosions, shootings - to the state Department of Education within 48 hours of the incident.

Table 5: Example Pyramid Clusters

marization modules, given the set of keywords extracted by the Keyword Extraction module from the complex question. Each SCU of weight w ¿ 1 was paired with each sentence from each of the six candidate summaries and submitted to the TE system. Positive instances of entailment (i.e. SCU_n entails S_1) were interpreted as representing a semantic match between the SCU cluster and the summary sentence; negative instances of entailment were taken as indicating no match between the two sentences. Each summary was then assigned an Modified Pyramid score. The top-scoring summaries were then included as part of our final submission.

Table 6 presents a breakdown of the number of times that each strategy was selected over the 50 topics evaluated in the DUC 2006 evaluations.

Although summaries based on semantic question decomposition received the highest Pyramid score for 32 out of 50 topics (64%), average overall responsiveness did not degrade significantly ($p < 0.05$) when other types of summaries were selected. In addition, even though slightly more summaries were created using sentences derived from GISTEXTER's Q/A module (56%) than its MDS module, average overall responsiveness remained relatively constant for both types of summaries: Q/A-based summaries received an average responsiveness score of 2.875, while MDS-based summaries scored 2.828.

Strategy	# Summaries	LCC Resp	Avg Resp
Semantic QD + Q/A	19	2.79	2.10
Semantic QD + MDS	13	2.85	2.34
Syntactic QD + Q/A	5	3.40	2.22
Bag-of-Words + MDS	5	2.60	1.84
Bag-of-Words + Q/A	4	2.50	2.36
Syntactic QD + MDS	4	3.00	2.32
Total	**50**	**2.86**	**2.20**

Table 6: Output of Pyramid-Based Summary Selector.

Although it is difficult to evaluate the effectiveness of our Pyramid-based summary selection algorithm without evaluating each of the 6 summaries that the system returns, we believe that GISTEXTER's ability to return summaries that are consistently more responsive than the mean for each topic – regardless of the summary strategy employed – suggests that this method is able to find a highly-responsive summary among the candidate summaries with some consistency.

3 Results

In this section, we present results from GISTEXTER's participation in the DUC 2006 evaluations. In addition to scoring amongst the top five systems for each of the linguistic quality questions, our system also returned very competitive results for metrics – including *overall responsiveness*, *content responsiveness*, and *Modified Pyramid* – that evaluated the content of summary answers. Table 7 presents a summary of our system's results for 8 metrics considered in DUC 2006.

Metric	Score	Rank
Overall Responsiveness	2.84	1
Content Responsiveness	3.08	1
Modified Pyramid	0.21	4
LQ1: Grammaticality	4.62	1
LQ2: Non-redundancy	4.60	5
LQ3: Referential clarity	3.72	4
LQ4: Focus	4.28	2
LQ5: Structure and Coherence	3.28	2

Table 7: DUC 2006 Results.

GISTEXTER ranked first overall on both the *overall responsiveness* and *content responsiveness* metrics. While our system's results continued to lag behind the responsiveness scores assigned to the human summaries, we believe the multi-strategy ap-

proach to question-directed summarization that we implemented this year allowed us to create – and select – summaries that best approximated the information needs of complex questions. Complete results from both responsiveness metrics are presented in Table 8. Even though the methods used to eval-

	Overall Responsiveness		Content Responsiveness	
Summarizer	Score	Rank	Content	Rank
Human Avg	4.74	–	4.75	–
27	**2.84**	**1**	**3.08**	**1**
23	2.76	2	3	2
31	2.6	3	2.86	6
2	2.46	4	2.54	20
24	2.44	5	2.88	5
5	2.42	6	2.76	9
14	2.42	8	2.82	7
28	2.42	7	2.78	8
6	2.36	9	2.62	11
13	2.36	10	2.7	10
20	2.28	12	2.52	21
33	2.28	11	2.58	16
34	2.24	13	2.24	32
3	2.22	14	2.6	12
12	2.22	16	2.92	4
30	2.22	15	2.58	17
35	2.2	17	2.42	25
4	2.18	18	2.54	19
10	2.16	19	2.94	3
9	2.12	20	2.36	27
22	2.12	21	2.56	18
7	2.08	22	2.5	22
21	2.08	25	2.36	28
29	2.08	24	2.44	24
32	2.08	23	2.6	13
15	2.06	27	2.48	23
25	2.06	26	2.34	29
1	2	28	2.04	34
16	1.98	31	2.3	31
18	1.98	30	2.32	30
19	1.98	29	2.6	14
8	1.96	32	2.58	15
17	1.88	33	2.38	26
26	1.68	34	2.06	33
11	1.34	35	1.68	35

Table 8: Results for Responsiveness metrics.

uate responsiveness did change between DUC 2005 and DUC 2006, we feel that our 2006 system significantly outperforms our 2005 system (which ranked tenth overall amongst systems in (overall) responsiveness in the DUC 2005 evaluations) in providing responsive summary answers to users' questions.

The scatterplots in Figure 2 and Figure 3 can be used to compare GISTEXTER's responsiveness against the average responsiveness scores received by all other participating systems.

GISTEXTER scored at or above the mean content responsiveness score for 36 out of 50 topics (72%) and outperformed the mean overall content responsiveness score for 39/50 topics (78%).

GISTEXTER's Modified Pyramid score was 0.2097, which ranked fourth amongst systems participating in the Pyramid evaluations. (Table 9

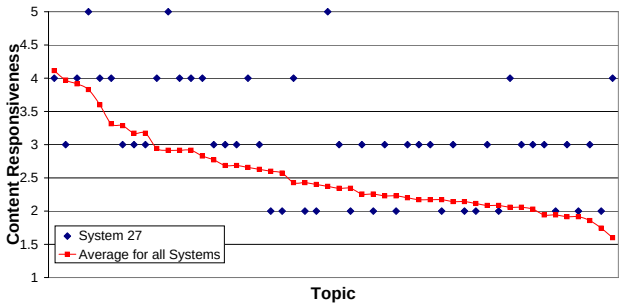

Figure 2: Content Responsiveness vs. Mean.

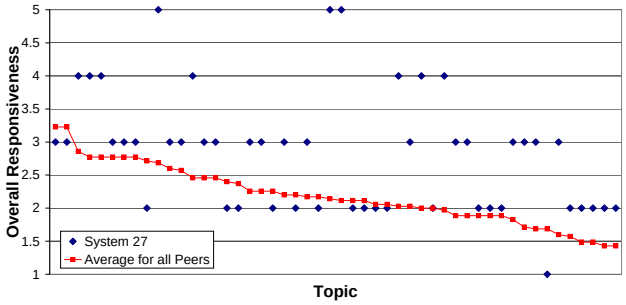

Figure 3: Overall Responsiveness vs. Mean.

presents results from the 2006 Pyramid evaluations.)

Even though our system uses automatically-generated Pyramids as part of its summarization pipeline, the Pyramids GISTEXTER creates are used to identify the semantic content common to a set of machine-generated candidate summaries and not to model the expected content of an "ideal" or perfectly responsive set of human-generated summaries. Although we expect that Pyramids generated from multiple candidate summaries can prove useful in selecting a highly-responsive candidate summary, we do not necessarily expect the introduction of Pyramids (or Pyramid-based) techniques to automatically improve the Modified Pyramid score of our system.

GISTEXTER also received high marks for a number of the DUC "linguistic quality" questions. Heuristics implemented to determine whether the antecedent of a pronoun was contained in the current (or previous) sentence resulted in a 3.72 score for "Referential Clarity", while our clustering-based approach to enhancing the coherence of a summary returned a 3.28 score (good enough for second overall) for the "Structure and Coherence" of our summaries.

	Score								Rank							
Peer	Mod-Pyr	L1	L2	L3	L4	L5	Cont	Resp	Mod Pyr	L1	L2	L3	L4	L5	Cont	Resp
23	0.2420	3.90	3.85	3.95	3.65	2.40	3.00	2.65	1	8	18	2	4	11	3	1
10	0.2414	3.15	4.30	2.55	2.95	1.80	3.10	2.20	2	20	7	22	22	20	2	11
8	0.2139	3.20	3.70	3.15	3.30	1.80	2.60	2.05	3	17	21	12	15	20	13	15
27	**0.2097**	**4.75**	**4.60**	**3.65**	**4.45**	**3.50**	**3.35**	**2.65**	**4**	**1**	**2**	**3**	**1**	**2**	**1**	**1**
28	0.2049	4.25	3.95	3.40	3.30	2.15	2.65	2.40	5	3	17	6	15	15	9	3
15	0.2002	3.35	3.75	2.80	3.35	2.05	2.50	2.05	6	14	20	18	11	18	15	15
2	0.1993	3.50	4.30	3.55	3.65	2.50	2.45	2.35	7	13	7	4	4	5	18	4
6	0.1980	3.10	3.85	3.25	3.35	2.10	2.50	2.20	8	21	18	10	11	17	15	11
3	0.1950	3.70	4.40	2.95	3.30	2.25	2.50	2.20	9	10	6	14	15	13	15	11
24	0.1919	3.35	4.15	3.55	3.65	2.70	2.85	2.25	10	14	14	4	4	3	4	9
33	0.1822	3.30	4.05	2.85	3.50	2.45	2.85	2.35	11	16	15	16	8	7	4	4
5	0.1787	3.80	4.60	3.20	3.65	2.45	2.70	2.30	12	9	2	11	4	7	7	7
19	0.1763	3.20	4.05	2.85	3.30	2.00	2.55	1.85	13	17	15	16	15	19	14	19
14	0.1729	3.70	4.25	3.40	3.40	2.50	2.65	2.35	14	10	10	6	9	5	9	4
22	0.1690	4.15	4.30	2.65	3.30	2.45	2.80	2.15	15	4	7	21	15	7	6	14
32	0.1671	2.90	3.60	2.70	3.05	1.80	2.65	1.80	16	22	22	20	21	20	9	21
29	0.1632	4.05	4.20	2.90	3.20	2.15	2.70	2.25	17	6	13	15	20	15	7	9
25	0.1504	3.20	4.25	2.80	3.40	2.20	2.40	1.95	18	17	10	18	9	14	19	18
18	0.1352	4.15	4.50	3.35	3.35	2.45	2.20	1.85	19	4	4	8	11	7	21	19
17	0.1313	3.60	4.25	3.10	3.35	2.40	2.40	1.60	20	12	10	13	11	11	19	22
35	0.1291	4.60	4.65	3.35	3.70	2.65	2.65	2.30	21	2	1	8	3	4	9	7
1	0.1134	3.95	4.45	4.70	4.40	4.20	1.90	2.00	22	7	5	1	2	1	22	17

Table 9: Pyramid Evaluation

4 Conclusions

In this paper, we presented a novel framework for question-directed summarization that uses a state-of-the-art textual entailment system in order select a single responsive summary answer from amongst a number of automatically summaries. By combining techniques for modeling the information needs of complex questions with Pyramid-based techniques for evaluating the relevance of answers, we believe that GISTEXTER can produce summary-length answers to a wide variety of complex questions that are both responsive and informative.

In future work, we will experiment with new ways that textual entailment can be used to combine the output of multiple candidate summaries. Since model Pyramids can be created automatically from the output of different summarization modules, we expect that highly-responsive candidate summaries could also be generated by distilling the content of the most relevant portions of a Pyramid.

References

S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, A. Hickl, and P. Wang. 2005a. Employing Two Question Answering Systems in TREC 2005. In *Proceedings of the Fourteenth Text REtrieval Conference*.

Sanda Harabagiu, Andrew Hickl, John Lehmann, and Dan Moldovan. 2005b. Experiments with Interactive Question-Answering. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*.

Sanda Harabagiu, Finley Lacatusu, and Andrew Hickl. 2006. Answering Complex Questions with Random Walk Models. In *Proceedings of the 29th Annual International ACM SIGIR*.

Sanda Harabagiu. 2004. Incremental Topic Representations. In *Proceedings of the 20th COLING Conference*, Geneva, Switzerland.

Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing Textual Entailment with LCC's Groundhog System. In *Proceedings of the Second PASCAL Challenges Workshop (to appear)*.

Finley Lacatusu, Andrew Hickl, Sanda Harabagiu, and Luke Nezda. 2004. Lite-GISTexter at DUC 2004. In *DUC 2004*, Boston, MA.

F. Lacatusu, A. Hickl, P. Aarseth, and L. Taylor. 2005. Lite-GISTexter at DUC 2005. In *Proceedings of the Document Understanding Workshop (DUC-2005) Presented at the HLT/EMNLP Annual Meeting*.

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th COLING Conference*, Saarbrücken, Germany.

R.J. Passonneau, A. Nenkova, K. McKeown, and S. Sigelman. 2005. Applying the Pyramid Method in DUC 2005. In *In Proceeding of the Document Understanding Workshop (DUC '05)*.