Available online at www.sciencedirect.com
ScienceDirect
Information Processing and Management 43 (2007) 1619–1642
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Satisfying information needs with multi-document summaries

Sanda Harabagiu [a], Andrew Hickl [b,*], Finley Lacatusu [b]

[a] Human Language Technology Research Institute, University of Texas at Dallas, Richardson, TX 75083, USA
[b] Language Computer Corporation, Richardson, TX 75080, USA

Received 17 July 2006; received in revised form 3 January 2007; accepted 8 January 2007
Available online 13 March 2007

Abstract

Generating summaries that meet the information needs of a user relies on (1) several forms of question decomposition; (2) different summarization approaches; and (3) textual inference for combining the summarization strategies. This novel framework for summarization has the advantage of producing highly responsive summaries, as indicated by the evaluation results.

Keywords: Summarization; Question decomposition; Textual entailment

1. Introduction

Multi-document summarization systems have traditionally focused on distilling the most globally relevant pieces of information contained within a collection of documents into a short passage that can be read – and digested – quickly by an end-user. While summaries can provide users with valuable information about the range of information contained in a corpus, multi-document summaries are often of limited value in an information-gathering environment where users seek particular types of information in order to perform a specific task or to achieve a research goal. Even though the content of a summary could potentially address some of the information needs of a user, many relevant information "nuggets" may not be sufficiently relevant to warrant inclusion in a "global" summary, despite having been mentioned in a collection of documents.

In order to address the shortcomings of "undirected" multi-document summaries, the recent 2005 and 2006 Document Understanding Conferences (DUC) have required participants to provide summary answers in response to research scenarios consisting of one (or more) complex questions. These types of summaries – which we will refer to as *question-directed summaries* (QDS) – require systems to generate coherent, passage-length answers that are both relevant to the topic of a collection of documents and respond the information needs of users as well.

* Corresponding author.
E-mail addresses: sanda@hlt.utdallas.edu (S. Harabagiu), andy.hickl@languagecomputer.com (A. Hickl), finley@languagecomputer.com (F. Lacatusu).

0306-4573/$ - see front matter
doi:10.1016/j.ipm.2007.01.004

In this paper, we introduce a new framework for question-directed summarization which combines the question decomposition and answer retrieval techniques pioneered for complex question-answering systems with a novel textual inference-based method for estimating the relevance of content included in a summary.

First, we show that by leveraging different combinations of question representation, passage retrieval, and relevance ranking techniques, we can generate sets of candidate summaries which address different dimensions of the user's information needs. Once these candidate summaries have been created, we then use textual inference in order to create models of the semantic content shared by these candidates in order to identify the most responsive candidate summary generated in response to a question.

Complex questions – such as those "asked" by the DUC organizers – cannot be answered with the same techniques that have so successfully been applied to answering "factoid" questions (Harabagiu et al., 2001; Harabagiu et al., 2003; Moldovan et al., 2002). Unlike informationally-simple factoid questions, complex questions seek multiple types of information simultaneously and do not presuppose that one single answer could meet all of its information needs. For example, with a factoid like "*Who controls the Falkland Islands?*", it can be safely assumed that the submitter of the question is looking for a country (or an organization) which is associated with the Falkland Islands. However, with a complex question like "*How have relations between Argentina and Great Britain developed since the war over the Falkland Islands ?*" the wider focus of the question suggests that the submitter is interested in the different types of relations between the two countries that could have been established since the conclusion of the 1982 war. In order to answer complex questions precisely, we believe that question-answering – as well as question-directed summarization – systems need to employ *question decomposition* strategies capable of understanding the range of information needs presupposed by a user's question. The question-directed summarization framework we introduce in this paper tackles the processing of complex questions with three question decomposition strategies that approximate the information sought by a complex question; these three question decomposition strategies are then used with two different passage retrieval engines, resulting in a total of six different sets of candidate passages which can be assembled into summary-length answers.

Unlike complex question-answering systems, question-directed summarization systems must do more than retrieve information relevant to a user's information needs. In order to meet the structural and length constraints imposed on summaries, QDS systems also need to include methods for estimating the inherent responsiveness of each sentence included in the summary returned to a user. Our QDS framework relies on a form of textual inference – known as *textual entailment* – in order to automatically identify the content that is most responsive to a user's information needs. In this work, we use a state-of-the-art system for recognizing textual entailment (TE) (first described in Hickl et al. (2006)) which is capable of correctly recognizing textual entailment with greater than 75% precision.[1] In this paper, we show that by considering the entailment relationships that exist between sentences originating in different summaries, we can construct hierarchical representations of relevance, similar to the hierarchical content models (or "Pyramids") first proposed in Nenkova and Passonneau (2004). Once constructed, we use these hierarchical representations in order to model the semantic content of an ideal summary response to a question and to select the most responsive summary – without the need for expensive and time-consuming human annotations.

The remainder of this paper is organized as follows. Section 2 presents an overview of GISTexter, our QDS system. Section 3 describes techniques for performing the decomposition of complex questions. Section 4 provides details of our system for recognizing textual entailment and describes how we used these types of inferential relationships to create content models that can be used in a summarization context. Section 5 details the results from evaluations performed using this framework, while Section 6 summarizes our conclusions.

2. The GISTexter system

In this section, we describe GISTexter, our *question-directed summarization* (QDS) system. (The architecture of GISTexter is provided in Fig. 1.)

[1] Systems for recognizing textual entailment have been evaluated as part of the First and Second PASCAL Recognizing Textual Entailment (RTE) Challenges in 2005 and 2006.

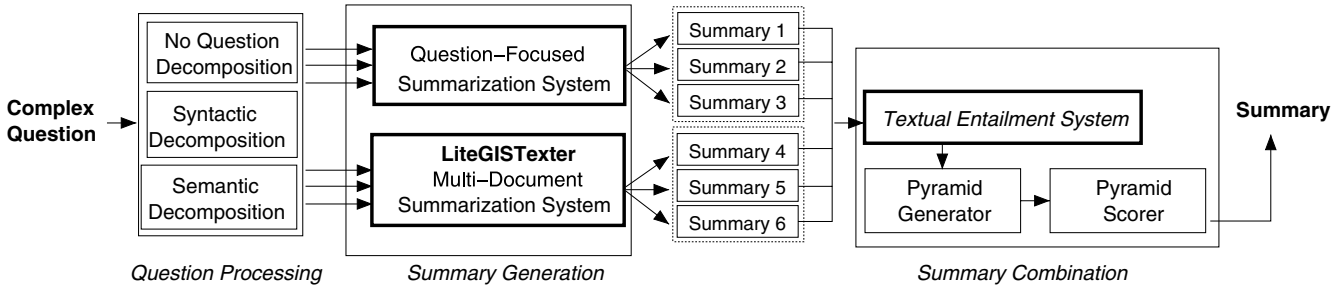

Fig. 1. Architecture of the GISTexter system.

GISTexter begins the process of QDS by submitting complex questions to a *Question Processing Module*, which employs three different types of question decomposition techniques in order to represent the types of information sought by a question. First, keywords are extracted heuristically from the question. Second, questions are decomposed *syntactically* in order to extract each overtly-mentioned query from the text of the complex question. Third (and finally), questions are decomposed *semantically* in order to identify the complete set of queries that are presupposed by the meaning of the complex question itself.

Once question processing is complete, the sets of queries (or sub-questions) generated by each of the three question decomposition strategies are then simultaneously sent to two different summarization engines: (1) a *question-focused summarization* (QFS) system and (2) a *multi-document summarization system*. While GISTexter's question-focused summarization system employs techniques inspired by the work first pioneered for textual question-answering (Q/A), its multi-document summarization system, known as Lite-GISTexter, uses a battery of lightweight natural language processing techniques for relevance estimation first developed for the Document Understanding Conference (DUC) summarization evaluations. We believe that by combining these two summarization system within the same architecture, GISTexter can retrieve more relevant sentences than frameworks that utilize only one type of retrieval engine.

The three different query representations produced by the question processing modules are then sent to each of these two summarization engines in order to retrieve a total of six different sets of relevant sentences. Each of these six sets of sentences are then sent to a *Summary Generation* module to be compiled into a fixed-length summary answer. (Table 1 lists the six different summarization strategies currently employed in GISTexter.)

GISTexter uses a framework inspired by recent approaches to evaluating multi-document summaries (Nenkova & Passonneau, 2004) in order to select the most responsive summary from the set of candidate summaries it generates. Recent work (Nenkova & Passonneau, 2004) has argued that multiple summaries created for the same topic – or in response to the same question – can be used in order to create a hierarchical model (known as a "Pyramid") of the content that an "ideal" summary should contain. Under this model, the text of a multi-document summary or summary answer is assumed to encode a set of *summary content units* (SCUs) which represent the set of propositions that the author of the summary believes to be the most relevant content contained in a set of documents. Although different authors will ultimately use different models of relevance when creating a summary, this model assumes that content that is common to multiple summaries will generally prove to be more relevant than content that is contained in one (or only a few) summaries. Given these assumptions, this approach predicts the best summaries will contain the greatest concentration of SCUs that appear multiple times in a "gold standard" set of summaries.

Table 1
Six summarization strategies implemented in the GISTexter system

Question-focused summarization	Multi-document summarization
Strategy 1. Bag-of-words	*Strategy 4.* Bag-of-words
Strategy 2. Syntactic Question Decomposition	*Strategy 5.* Syntactic Question Decomposition
Strategy 3. Semantic Question Decomposition	*Strategy 6.* Semantic Question Decomposition

While Nenkova and Passonneau's (2004) summarization evaluation methodology depends on the use of human annotators in order to identify – and track the distribution of – SCUs across a set of human-created summaries, we believe that recent advances in the recognition of a form of textual inference – known as *textual entailment* – could be leveraged in order to (1) create "Pyramid"-like content models from a group of "peer" summaries and to (2) score summaries based on the number and type of SCUs they contain. Introduced by Dagan, Glickman, and Magnini (2005), the task of recognizing of textual entailment (TE) requires systems to determine whether the meaning of a sentence (referred to as a *hypothesis*) can be reasonably inferred from the meaning of another sentence (referred to as a *text*). While TE was not intended a measure of semantic equivalence, we believe that systems for TE can prove useful in identifying content that is common across a group of summaries.

Once GISTexter generates a set of summaries, the textual entailment system (TES) described in Hickl et al. (2006) is used in order to create a "Pyramid" model of the content common to the candidate summaries. This model is then used in conjunction with the TES to score each of the six candidate summaries using the Modified Pyramid scoring algorithm described in Passonneau, Nenkova, McKeown, and Sigelman (2005); the individual summary that receives the highest Pyramid score is then returned to the user.

2.1. The question-focused summarization system

Most current textual question-answering (Q/A) systems (Harabagiu et al., 2001; Harabagiu et al., 2003; Moldovan et al., 2002) utilize a three-stage architecture in order to return a ranked list of answers to a natural language question. First, questions undergo *Question Processing* in order to (1) extract keywords that can be used to retrieve candidate answers and (2) identify the expected answer type of the question. Second, keywords extracted from the question are submitted to a *Document Processing* module in order to retrieve a set of relevant (paragraph-length) text passages. (At this point, text passages that do not contain any entity of the same semantic class as the expected answer type of the question are generally filtered.) Third (and finally), passages are sent to an *Answer Processing* module responsible for pinpointing exact answers from the set of retrieved text passages. Lists of candidate answers are ultimately presented to a user, ranked in order of their expected relevance to the user's question.

While question-answering systems have proven successful in identifying answers to questions from individual documents (Harabagiu, Moldovan et al., 2005), we believe that this traditional Q/A architecture (illustrated in Fig. 2a) needs to be modified in order to produce the types of summary-length answers sought in question-directed summarization task. In this paper, we refer to this type of Q/A-inspired architecture for QDS as a *question-focused summarization* (QFS) system. (The architecture of the QFS system integrated into GISTexter architecture is presented in Fig. 2b.)

As with a traditional Q/A system, questions in a question-focused summarization system are initially sent to a *Question Processing* module. Queries formed during Question Processing are sent to a *Sentence Retrieval* module which retrieves (and ranks) a relevant set of sentences, based on (1) the number and proximity of question keyword terms found in each sentence and (2) the presence of entities of the same semantic class as the

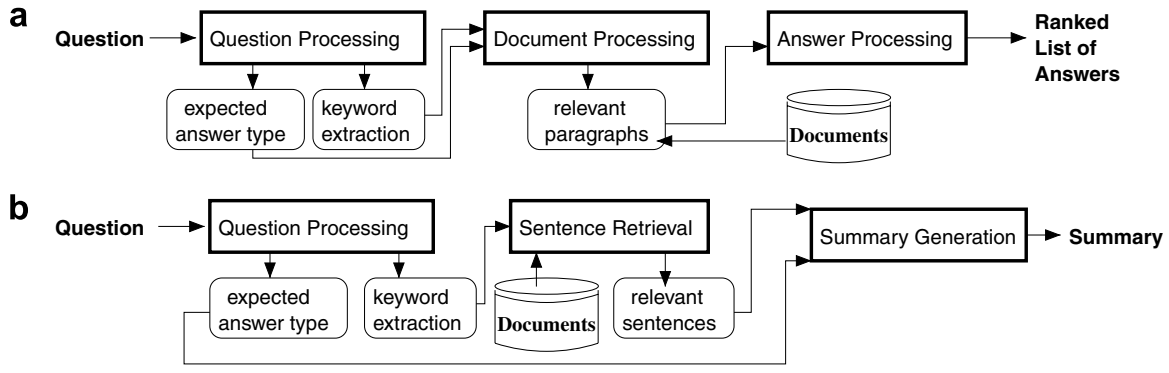

Fig. 2. (a) Traditional question-answering architecture; (b) question-focused summarization architecture.

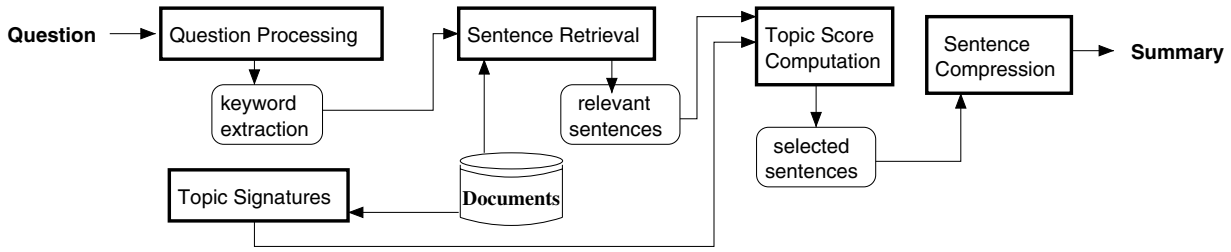

Fig. 3. Architecture of the Lite-GISTexter MDS system.

expected answer type. Retrieved sentences are sent to a *Summary Generation* module, which compiles the top-ranked sentences into a coherent, fixed-length summary.

2.2. Multi-document summarization with Lite-GISTexter

Lite-GISTexter is a stand-alone multi-document summarization (MDS) system that was evaluated in the DUC 2004 and DUC 2005 evaluations (Lacatusu, Hickl, Harabagiu, & Nezda, 2004; Lacatusu, Hickl, Aarseth, & Taylor, 2005). (The architecture of LiteGISTexter is provided in Fig. 3.)

With LiteGISTexter, keywords extracted from a question are initially used to retrieve a set of documents relevant to the question. These documents are then used to compute two different types of *topic representations*, (1) *topic signatures* (Lin & Hovy, 2000) and (2) *enhanced topic signatures* (Harabagiu, 2004). Following Lin and Hovy (2000), we assumed that the topic of a collection of documents can be represented by a topic signature $TS_1 = \langle topic, (t_1, w_1), \ldots, (t_n, w_n)\rangle$, where the terms t_i are highly correlated with the topic with association weight w_i. Term selection and weight association are determined by the use of the *likelihood ratio* λ. In addition to the term-based TS_1 representation, we also followed Harabagiu (2004) in computing enhanced topic signatures for each retrieved set of documents. With enhanced topic signatures, topics are characterized by representative relations that exist between TS_1 terms: $TS_2 = \langle topic, (r_1, w_1), \ldots, (r_m, w_m)\rangle$, where r_i is a binary relation between two topic concepts. The weights associated with TS_1 and TS_2 are used in conjunction with keywords extracted from the question in order to compute a composite *topic score* for each sentence in the document collection. Sentences are then ranked according to topic score; the top-ranked sentences are then sent to a Summary Generation module to be compiled into a fixed-length summary.

2.3. Summary generation

In our QFS-based and MDS-based systems, summaries are generated by selecting the top-ranked sentences from a list of sentences returned during Sentence Retrieval and merging them into a single paragraph of predetermined length.[2]

Two types of optimizations were 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 2.)

Second, 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 predict which pronominal mentions could be included in a summary without adversely impacting referential clarity, we can enhance both the legibility

[2] The DUC 2005 and 2006 question-directed summarization evaluations required systems to return summaries of no longer than 250 words.

Table 2
Clustering of redundant passages

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

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 74% precision. 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 occurring 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) *added* along with the immediately previous sentence found in the original document, or (3) *dropped* altogether from the summary.

3. Question processing and decomposition

We believe that the quality of question-directed summaries depends (in part) on systems' ability to interpret the information needs expressed by complex questions. In order to provide summary answers that are responsive to a user's particular information needs, we hypothesize that complex questions need to be *decomposed* into the set of simpler queries – or *sub-questions* – they either (1) mention overtly or (2) presuppose semantically. In this section, we describe two types of question decomposition that we believe must be conducted prior to beginning the process of question-directed summarization: (1) syntactic question decomposition and (2) semantic question decomposition.

3.1. Syntactic question decomposition

Complex questions often include multiple requests for information in the same sentence. In an analysis of 125 complex questions taken from the 2004 AQUAINT Relationship Q/A Pilot, the 2005 TREC Q/A Track Relationship Task and the 2005 DUC question-focused summarization task, we found that 49 questions (39%) included more than one overt, simple question.

We refer to complex questions that include the mention more than one overt question as *syntactically complex questions*. We have identified three types of syntactically complex questions: (1) questions that feature coordination (of question stems, predicates, arguments, or whole sentences), (2) questions that feature lists of arguments or clauses, and (3) questions that feature embedded or indirect questions. Examples of each of these three types are provided in Table 3.

In GISTexter, questions featuring coordination and lists were decomposed syntactically using sets of heuristics. With instances of coordination, questions were first split on the conjunction; each conjunct was then

Table 3
Syntactic question decompositions

Coordination	When and where did Fidel Castro meet the Pope?
Lists of arguments	What international aid organization operates in Afghanistan, Iraq, Somalia, and more than 100 other countries?
Embedded questions	The analyst would like to know of *any attempts by these governments to form trade or military alliances*

The analyst is concerned with a possible relationship between the Cuban and Congolese governments. Specifically, the analyst would like to know of any attempts by these governments to form trade or military alliances.

Fig. 4. Complex question.

Table 4
Example of syntactic question decomposition

(1) What attempts have been made by *the Cuban* government to form *trade* alliances?
(2) What attempts have been made by *the Cuban* government to form *military* alliances?
(3) What attempts have been made by *the Congolese* government to form *trade* alliances?
(4) What attempts have been made by *the Congolese* government to form *military* alliances?

reconstructed into a new, grammatical question using information extracted from the other conjunct. For example, a coordinated question like: "*When and where did Fidel Castro meet the Pope?*" is decomposed into: "*When did Fidel Castro meet the Pope?*" and: "*Where did Fidel Castro meet the Pope?*" by selecting a question stem and re-supplying the verb phrase found in the original question. Likewise, questions involving lists of arguments were decomposed by creating sets of new sub-questions that replaced the list in the original question with each member of the list. For example, in a question like: "*What international aid organization operates in Afghanistan, Iraq, Somalia, and more than* 100 *other countries?*", four sub-questions were generated, including: "*What international aid organization operates in Afghanistan?*" and "*What international aid organization operates in more than 100 other countries?*".

Syntactic question decomposition often requires the resolution of anaphora. We begin by identifying the antecedents of all referential expressions found in a complex question. For example, the complex question illustrated in Fig. 4 requires the resolution of the NP "*these governments*" to the set: {*the Cuban government*; *the Congolese government*}.[3] Once coreference is established, we process questions by using a set of syntactic patterns to extract embedded questions and to split questions that featured conjoined phrases or lists of terms into individual questions. For example, the complex question illustrated in Fig. 4 is decomposed syntactically in Table 4.

3.2. Semantic question decomposition

Even after syntactic question decomposition is performed, most complex questions still need to be decomposed semantically before they can be submitted to a traditional Q/A system.

We believe that the relations that exist between a question and its decompositions may be of semantic nature (e.g. definitions, generalizations) or of discourse nature (e.g. elaborations, causes, effects, parallelism). Furthermore, unlike discourse relations introduced by various coherence theories, the relations between questions have an argument. This argument may take any of the values: (1) PREDICATE, (2) EVENT, (3) ARGUMENT, (4) ATTRIBUTE, or (5) HYPERNYMY/HYPONYMY. The first four values refer to a predicate, event, argument or attribute detected in the mother-question, which will also be referred by the daughter-question. The last value (Hypernymy/Hyponymy) indicates that there is such a semantic relation (defined in WordNet) between a pair of concepts, one for the mother-question, one for the daughter-question. For example, in Fig. 5 we illustrate several relations between questions that are labeled EFFECT(Event). This type of relation indicates (1) that the expected answer type (EAT) of the decomposed question is a class of concepts that are caused by some event (here, the 1982 Falklands War between Argentina and Great Britain); and (2) the event is explicit in both questions. We have considered eight different classes of relations between questions that can be used to perform semantic question decompositions. The relations are illustrated in Appendix A of the paper.

[3] The coreference resolution algorithm we use was described in Harabagiu et al. (2001).

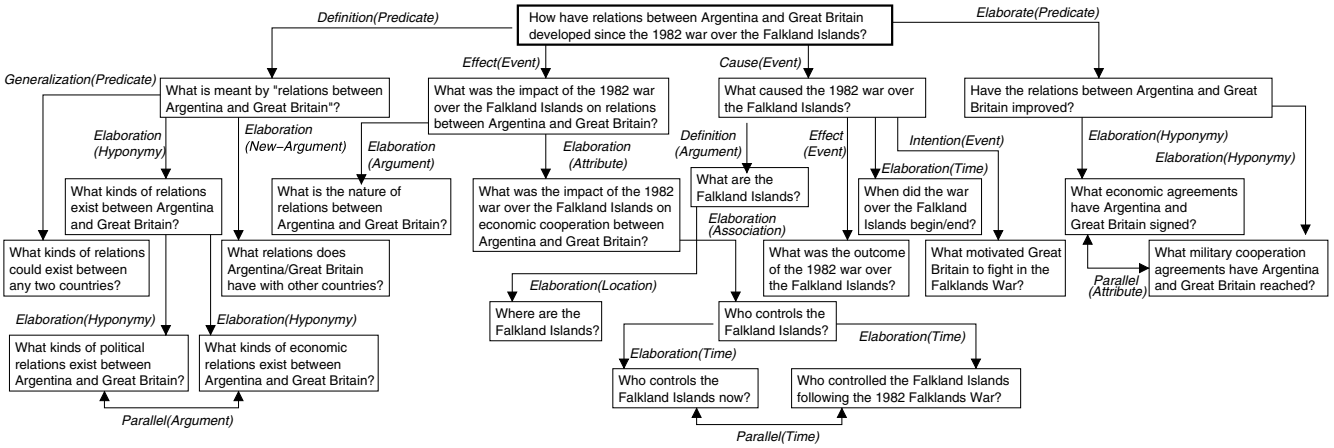

Fig. 5. Top-down question decomposition.

3.2.1. Top-down question decomposition

When decomposing complex questions in a top-down manner, systems need to have access to four forms of information:

- Semantic Dependencies (in the form of predicate-argument structures) found in the question;
- The Expected Answer Type (EAT) of the question;
- Sets of topical terms and relations derived from a collection of documents relevant to the question;
- Pragmatic associations between the question and potential decompositions;

Predicate-argument structures are provided by shallow semantic parsers trained on PropBank[4] and NomBank.[5] The EAT of the question is discovered with the technique reported in Pasca and Harabagiu (2001). The most relevant relations from the question topic are identified by the enhanced representations of topic signatures reported in Harabagiu (2004).

For each EAT, we have created a large set of 4-tuples (e_1, r, e_2, A), that we call *association vectors*. Each association vector consists of (1) e_1, the EAT of the mother-question; (2) r, the relation between the pair of questions; (3) e_2, the EAT of the daughter-question; and (4) A, the set of lexically-aligned tokens found in between two questions. (In our current work, we use the lexical alignment system developed by Hickl et al. (2006) in order to identify pairs of tokens in each pair of questions which are likely to contain corresponding semantic information.) Fig. 6 illustrates the association vectors for two questions from Fig. 5.

When generating a decomposition, we use the association information to build association rules similarly to the method introduced in Nahm and Mooney (2000). The association information is akin to the fillers of a template. Therefore, by representing it as binary features that are provided to a decision-tree classifier (C5.0, Quinlan, 1998), we generate automatically association rules from the decision rules of the classifier. In order to find the new information, that specializes the decomposed question, we select the topic relation that (a) fits best the predicate-argument structure of the decomposed question, and (b) produces similar lexical alignment while preserving grammaticality. These last two conditions must be met by the question surface realization function. The Top-Down Question Decomposition Procedure is illustrated in Fig. 7.

3.2.2. Bottom-up question decomposition

In contrast to the top-down question decomposition described in Section 2.2, complex questions can also be semantically decomposed in a bottom-up fashion by identifying the potential decomposition relations that may exist between sets of factoid questions related to the same topic.

[4] http://www.cis.upenn.edu/~mpalmer/project_pages/ACE.htm.
[5] http://nlp.cs.nyu.edu/meyers/NomBank.html.

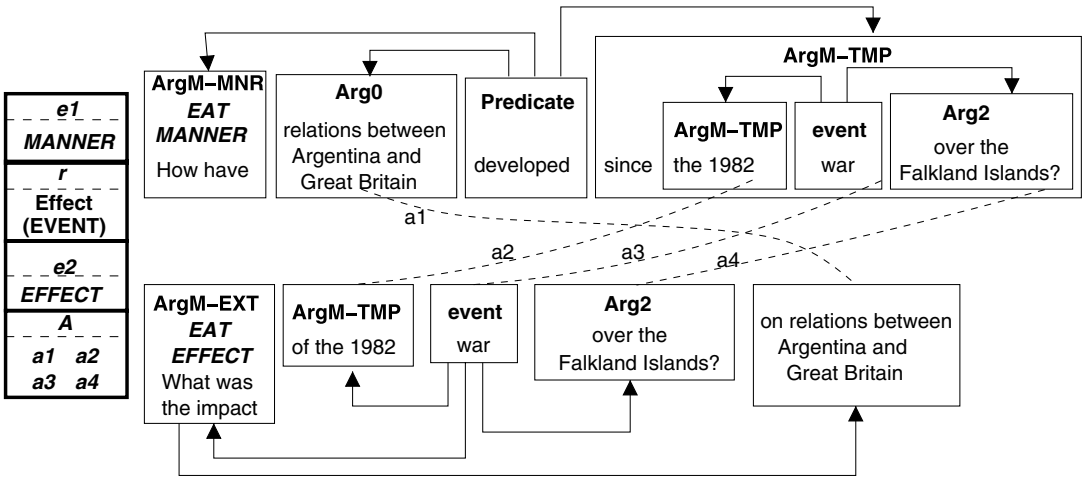

Fig. 6. Association information for two questions.

Step 1: Generate predicate-argument structures for complex question.
Step 2: Find the EAT of the question.
Step 3: Find association rules from the classification of association information.
Step 4: Use the association rules to have access to most likely
(a) discourse relation to decomposed question
(b) EAT of decomposed question
(c) lexical alignment between complex question and decomposed question
Step 5: Select most likely topic relations that fit the association information.
Step 6: Produce surface realization of decomposed question.

Fig. 7. Top-down question decomposition procedure.

In previous work (Harabagiu, Hickl, Lehmann, & Moldovan, 2005) we have described an approach that used syntactic patterns – in conjunction with semantic dependency and named entity information – to generate factoid questions automatically from large text corpora. This approach is illustrated in Fig. 9. Fig. 8 illustrates a bottom-up decomposition produced by the procedure from Fig. 9. In Fig. 8, all dashed arrows correspond to further produced decompositions that are not distancing themselves semantically from the complex question (Step 10 in Fig. 9).

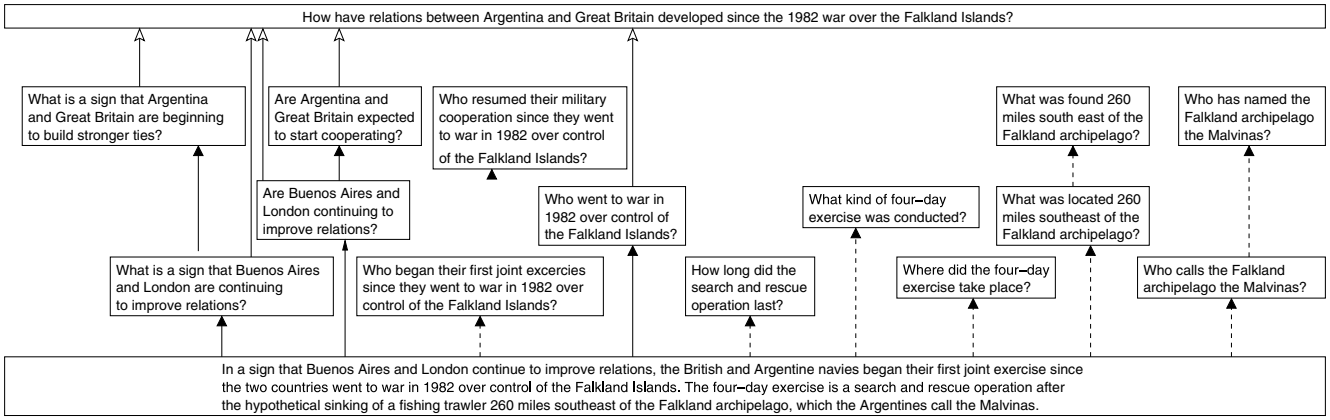

Fig. 8. Bottom-up question decomposition.

Step 1:	Text passages corresponding to candidate answers to factoid questions are identified and extracted from text, using techniques first developed for answer type detection for factoid Q/A (Harabagiu, Moldovan et al., 2005)
Step 2:	Once these answers were identified, we used a pattern specification language to generate a factoid-style natural language generator from each answer's sentence. Examples of these automatically-generated questions are presented in Figure 8.
Step 3:	In order to measure the coverage of the set of questions generated from a text, we used a paraphrase acquisition system (similar to the method proposed in (Shinyama et al., 2002)) to generate additional questions that could be associated with each identified answer. Under this approach, each of the generated questions were parsed using a semantic parser trained on PropBank. Pairs of entities assigned a semantic role by the same predicate were then selected and used to generate a web query that returned the top 500 documents containing both entities. Sentences containing both terms were then extracted, and a method described in (Clifton & Teahan, 2004) was used to extract the intervening text (or paraphrase) that occurred between the terms. In order to ensure that only semantically equivalent paraphrases were used to create new questions, the set of extracted paraphrases were clustered (using a complete-link clustering algorithm introduced in (Barzilay and Lee, 2003); only clusters containing text passages extracted from the original question were considered to be viable paraphrases.
Step 4:	Once a set of new questions are generated, a concept similarity function = (2× *Nr of Alignments*) / (*Nr of Predicates and Arguments in both Questions*) was used to calculate the similarity between each pair of questions in the collection. This score was then used in a KNN clustering algorithm to cluster questions into sets which were assumed to seek similar types of information.
Step 5:	For each question cluster, select it's centroid question.
Step 6:	Find association rules from the classification of association information when considering the centroid question.
Step 7:	Use association rules to have access to the most likely (a) discourse relation to a more complex question, (b) EAT of the more complex question, (c) lexical alignment between the complex and the more complex question that is proposed.
Step 8:	Use the alignment information to select the most likely topic relations that fit the association information.
Step 9:	Produce surface realization of the more complex question.
Step 10:	Measure the distance to the original question, by using the concept similarity described in Step 4. If the distance increased, STOP.
Step 11:	If more than one complex question was produced, GO TO Step 4.
Step 12:	When the conceptual similarity is above a threshold, STOP the decomposition.

Fig. 9. Bottom-up question decomposition algorithm. (Barzilay & Lee, 2003; Clifton & Teahan, 2004; Harabagiu, Moldovan et al., 2005; Shinyama, Sekine, Sudo, & Grishman, 2002).

4. Textual entailment between summaries

In this section, we describe how we leveraged a state-of-the-art system for textual entailment system (TES) in order to select amongst each of the six candidate summaries generated by GISTexter. In Section 4.1, we provide an overview of the TES we first developed for the 2006 Second PASCAL Recognizing Textual Entailment Challenge (RTE-2) (Hickl et al., 2006). In Section 4.2, we describe how we used TES output was used to automatically generate content models from the content of the candidate summaries generated by GISTexter. Finally, in Section 4.3, we describe how we again employed TES output in order to score each candidate summary against a content model in order to identify the candidate summary which best meets the user's information needs.

4.1. The textual entailment system

In this section, we provide a brief overview of the system for recognizing textual entailment first described in Hickl et al. (2006). The architecture of this TES is presented in Fig. 10.

Following Dagan et al. (2005) and Bar-Haim et al. (2006), we consider that a sentence S_1 (conventionally referred to as a *text*) *textually entails* a second sentence S_2 (referred to as a *hypothesis*) if the meaning of S_2 can be reasonably inferred from the meaning of S_1. For example, in the positive example of TE in Table 5, the

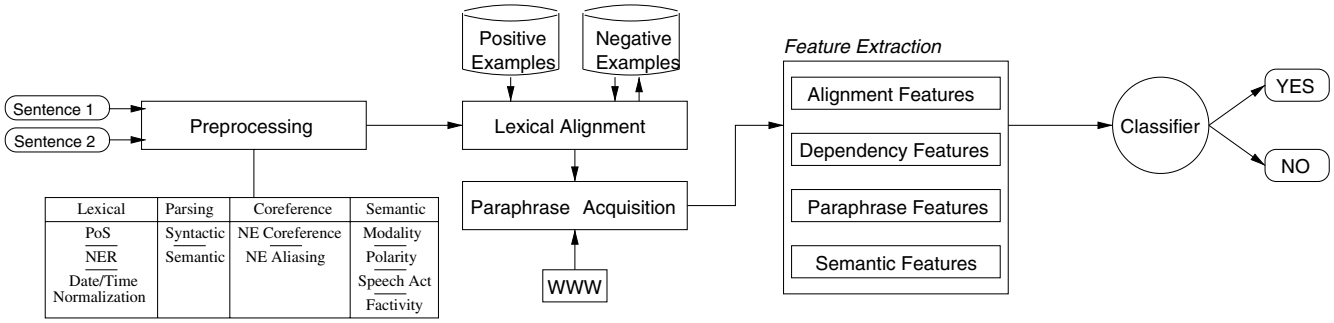

Fig. 10. Architecture of the Hickl et al. (2006) TES system.

Table 5
Example 139

Judgment	Example
YES	Text: The Bills now appear ready to hand the reins over to one of their two-top picks from a year ago in quarterback J.P. Losman, who missed most of last season with a broken leg Hypothesis: The Bills plan to give the starting job to J.P. Losman

hypothesis is considered to be textually entailed by the text, since under most normal readings, the meaning of a statement such as "*planning to give the starting job to*" someone is seen as compatible with a statement of being "*ready to hand the reins over to*" (the same) someone.

In order to acquire the necessary linguistic information needed to recognize textual entailment, text-hypothesis sentence pairs are first sent to a *Text Preprocessing* module. Here, sentences are syntactically parsed, semantic dependencies are identified using a semantic parser trained on predicate-argument annotations derived from PropBank, entities are associated with named entity information from LCC's CiceroLite Named Entity Recognition system, time and space expressions are normalized (using a method first described in Lehmann, Aarseth, Nezda, Deligonul, & Hickl (2005)), coreferential entities are detected and resolved, and predicates are annotated with semantic features such as polarity and modality.

Once preprocessing is complete, sentence pairs are sent to a *Lexical Alignment* module which uses a Maximum Entropy-based classifier in order to determine the likelihood that single tokens (or phrases) selected from each text and hypothesis corresponding or otherwise equivalent information. (An example of the alignments computed for the TE pair in Table 5 is depicted in Fig. 11.)

Information from *Lexical Alignment* was then used in conjunction with a *Paraphrase Acquisition* module in order to identify sets of phrase-level alternations – or *paraphrases* – which could be used to relate the semantic content of the two sentences being compared. Since paraphrases generally predicate about the same sets of individuals, we assume that pairs of entities which receive high-confidence lexical alignment scores can be used to construct queries to retrieve sets of sentences which may be paraphrases – or near paraphrases – of the semantic content encoded in either text or the hypothesis. For example, the given the high-confidence aligned

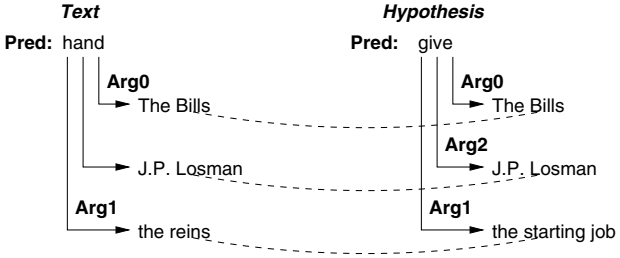

Fig. 11. Lexical alignment graph.

Table 6
Automatically-generated paraphrases

Judgment	Paraphrase
YES	*The Bills* have gone with quarterback *J.P. Losman*
YES	*The Bills* decided to put their trust in *J.P. Losman*
YES	*The Bills* benched Bledsoe in favor of *Losman*
YES	*The Bills* are now molding their quarterback-of-the-future *J.P. Losman*
YES	*Bills*' coach Mike Mularkey will start *J.P. Losman*
YES	*The Bills* have turned over the keys to the offense to *J.P. Losman*
NO	*The Bills*' 2005 season hinges on the performance of quarterback *J.P. Losman*
NO	*The Bills* gave away to acquire *J.P. Losman*

tokens from the hypothesis in Table 5, "*The Bills*" and "*J.P. Losman*", the eight sentences in Table 6 were retrieved from a standard *Google* WWW search.[6]

In our system, we use the two highest-confidence aligned tokens from each text-hypothesis pair to construct a search query; sentences containing both terms within a context window are returned from the top 500 documents returned retrieved for this query. Since not all retrieved sentences will be synonymous with either the text or hypothesis, a complete-link clustering algorithm (similar to Barzilay & Lee, 2003) was used to cluster paraphrases into sets that are presumed to convey the same content.

Semantic information – including features derived from Text Processing, Lexical Alignment, and Paraphrase Generation – is then combined into an *Entailment Classifier* which determines the likelihood that a textual entailment relationship exists for a particular pair of sentences. Four classes of features are extracted: (1) *alignment features*, which compare properties of aligned constituents, (2) *dependency features*, which compare entities and predicates using dependencies identified by a semantic parser, (3) *paraphrase features*, which determine whether passages extracted from the two sentences match acquired paraphrases, and (4) *semantic features*, which contrast semantic values assigned to predicates in each example sentence. Based on these features, the Entailment Classifier outputs both an entailment classification (either `YES` or `NO`) and a confidence value.

4.2. Automatic pyramid generation

Once a complete set of six candidate summaries have been generated, we used our TE system described in Section 4.1 in order to select the candidate summary that best met the expected information need of the complex question.

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. $\langle S_1, S_2 \rangle$) 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).[7]

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,

[6] In this case, six out of the eight candidate paraphrases were deemed by human annotators to be "acceptable" paraphrases of the text in Table 5. Results from an evaluation of the Paraphrase Acquisition module are presented in Section 5.4.3.

[7] For the purposes of this work, we have defined the scope of a SCU to be a single sentence; this represents a slight departure from the work of Passonneau et al. (2005), who consider the scope of an SCU to be either a clause or a sub-clause within a sentence.

Table 7
Textual entailment rules

Entailment judgments	Action
$S_1 \models S_2$ and $S_2 \models S_1$	Add S_2 to the cluster containing S1; drop all other pairs containing S_2
$S_1 \models S_2$ and $S_2 \not\models S_1$	Add S_2 to cluster containing S_1
$S_1 \not\models S_2$ and $S_2 \not\models S_1$	Do not add S_2 to cluster containing S_1

Table 8
Example pyramid clusters

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 5500 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

sentences that did not exhibit any entailment relationship with any other sentence were assigned a weight of 1. Table 7 provides a synopsis of the rules used to construct Pyramids from entailment judgments.

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 8.

In Table 8, 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.

4.3. Pyramid scoring of summaries

Each of the six candidate summaries was assigned a Pyramid score using the Modified Pyramid scoring algorithm described in Passonneau et al. (2005). First, each sentence in each candidate summary was assigned an "SCU score" based on the SCU weight assigned to its cluster. Sentences associated with an SCU cluster of weight $w > 1$ received a score equal to their weight; sentences associated with an SCU cluster of weight $w = 1$ received a zero SCU score. Next, in order to ensure that sentences assigned to large SCU clusters represented "true" entailments of the SCU underlying the cluster, we used TE to filter all sentences that were not textually entailed by the sentence from the cluster that was assigned the highest passage retrieval score by either the

question-focused summarization or the multi-document summarization sentence retrieval engines. Sentences that passed textual entailment retained their original SCU score; sentences that were not textually entailed received a zero score. Finally, a composite Modified Pyramid score was computed for the summary, and the top-scoring summary was returned to the user.

5. Experimental methods

In this section, we present results from an evaluation of the performance of our overall approach to question-directed summarization. Section 5.1 describes the test data that was used to create QDS along with the data that was used to train and to test our system for recognizing textual entailment. Section 5.2 provides quantitative results from three different evaluations of QDS. Section 5.3 details methods for evaluating the output of a semantic question decomposition module, while Section 5.4 describes how we evaluated each of the components of our system for recognizing textual entailment. Finally, in Section 5.5, we discuss the performance of our Pyramid-based summary selection module for selecting the most responsive summary from among a set of candidate summaries.

5.1. Test data

Over the past three years, the answering of complex questions has been the focus of much attention in both the automatic question-answering (Q/A) and the 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 the 2005 and 2006 Document Understanding Conference (DUC) have tasked systems with returning paragraph-length answers to complex questions that are responsive, relevant, and coherent.

We conducted the evaluations of the GISTexter QDS system on a total of 100 different topics taken from the Document Understanding Conference (DUC) question-directed summarization shared task from 2005 and 2006. In this task, systems were presented with a complex question (generally known as a ''DUC topic'') and a set of approximately 25 newswire documents that were assumed to contain all of the relevant information needed to construct a perfectly responsive summary answer to the complex question. Summaries were required to be less than 250 words in both DUC 2005 and DUC 2006.

Human assessors from the National Institute of Standards and Technology (NIST) were used to create the ''topics'' – and to assemble the document sets – used in the DUC evaluations. Once a set of topics had been agreed upon for an evaluation, assessors were then tasked with hand-creating a set of ''model'' summaries that could be used in evaluating machine-generated summaries. In DUC 2005 and DUC 2006, another set of annotators followed the techniques outlined in Nenkova and Passonneau (2004) in order to convert the model summaries created by the NIST assessors into ''model pyramids'' that could be used for the manual Pyramid scoring of automatically-generated summaries.

We conducted the evaluation of our textual entailment system using the set of 3200 entailment sentence pairs compiled for the 2005 and 2006 PASCAL Recognizing Textual Entailment Challenges (Bar-Haim et al., 2006; Dagan et al., 2005). This set consisted of pairs of sentences derived from the output of a number of natural language processing applications (including automatic question-answering systems, multi-document summarization systems, information retrieval systems, and information extraction systems); approximately 50% of these examples were deemed by human annotators to be positive instances of textual entailment, while another 50% were deemed by annotators to be negative instances of textual entailment.

5.2. Evaluation of question-directed summarization

In this section, we present results from three different types of evaluations of question-directed summaries. In Section 5.2.1, we discuss two types of subjective measures designed to evaluate the responsiveness of summary answers. In Section 5.2.2, we present results from two different automatic summary scoring algorithms: ROUGE (Lin, 2004) and Basic Elements (BE) (Hovy, Lin, & Zhou, 2005). Finally, since overall readability

and coherence are an important part of any summary generation task, we discuss results from 5 different linguistic quality metrics that have been used to evaluate question-directed summaries.

5.2.1. Evaluating the responsiveness of summaries

Evaluating the responsiveness of summary-length answers to complex questions has traditionally represented more of a challenge than evaluating answers to the types of "factoid" style questions typically asked in the annual Text Retrieval Conference (TREC) Question-Answering Evaluations. Unlike informationally-simple "factoid" questions, complex questions often seek multiple different types of information simultaneously and do not presuppose that one single entity or proposition could meet all of the information needs implied by the question itself. 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 range 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 broader focus of this question may suggest that the submitter may not have a single or well-defined information need and therefore may be amenable to receiving additional information that is relevant to an (as yet) undefined information goal.

In the DUC 2006 Question-Directed Summarization Evaluations, human assessors were tasked with providing two different subjective estimates of how well they felt a question-directed summary responded to the information need of a complex question. In the first measure, known as *content responsiveness*, annotators indicated the degree to which the information contained in the QDS satisfied the information need of the summary. In the second measure, known as *overall responsiveness*, annotators scored summaries based on both their information content and their overall readability. Annotators scored both measures on a 5-point scale, with 1 representing the lowest level of satisfaction and 5 representing the highest.

In order to evaluate the relative responsiveness of the different question-directed summarization strategies described in this paper, we tasked a team of 5 human annotators to evaluate each of the 6 candidate summaries generated for each of the 50 DUC 2006 QDS topics for both *content responsiveness* and *overall responsiveness*. Results from these evaluations are presented along with GISTexter's official DUC 2006 results in Table 9.

5.2.2. Automatic evaluations (Rouge+BE)

While subjective measures of responsiveness are an effective way to gauge the quality of a multi-document or a question-directed summary, gathering human judgments can often be too time-consuming and/or expensive to be a useful source of feedback for developers of summarization systems. In addition to evaluating GISTexter's candidate summaries in terms of *content responsiveness* and *overall responsiveness*, we evaluated each of the summaries generated for the 50 DUC 2006 topics using two different automatic summarization scoring systems: ROUGE and Basic Elements (BE).

Introduced in Lin and Hovy (2003), ROUGE (Recall-Oriented Understudy for Gisting Evaluation) automatically compares machine-generated candidate summaries against a human-created model by counting the number of *n*-grams, word sequences, or word pairs that the two summaries have in common. Features common to both summaries are then used to compute a score which approximates the degree of correspondence between the machine-generated summary and the human-created model summary. While ROUGE has been

Table 9
GISTexter responsiveness scores: internal evaluation and DUC 2006

Strategy	Content responsiveness	Overall responsiveness
Strategy 1	2.20	2.96
Strategy 2	2.56	3.04
Strategy 3	2.73	3.56
Strategy 4	1.97	2.10
Strategy 5	2.30	2.64
Strategy 6	2.75	3.32
GISTexter (rank)	3.08 (1)	2.84 (1)
Best DUC06	3.08	2.84

Table 10
GISTexter ROUGE and BE scores: DUC 2006

Strategy	ROUGE-2	ROUGE-SU4	BE
Strategy 1	0.0739	0.1319	0.0381
Strategy 2	0.0746	0.1316	0.0389
Strategy 3	0.0802	0.1352	0.0413
Strategy 4	0.0725	0.1299	0.0377
Strategy 5	0.0728	0.1311	0.0380
Strategy 6	0.0790	0.1349	0.0411
GISTexter (rank)	0.0809 (11)	0.1359 (15)	0.0419 (12)
Best DUC06	0.0951	0.1547	0.0508

one of the ''official scores'' used by the DUC organizers to evaluate system submissions since DUC 2004, ROUGE's dependence on token overlap often limits in its effectiveness, as summaries can receive ROUGE scores that say more about the number of words or phrases in common between the two summaries than the amount of shared semantic content they contain. In order to account for some of the shortcomings of ROUGE's term-based approach, Hovy et al. (2005) suggested that automatic summary scoring should be performed using text fragments (known as basic elements (BEs)) which express syntactic or semantic dependencies. Unlike ROUGE-based scoring metrics, which treated individual summaries as bags-of-words, Hovy et al. argued that a BE-based scoring metric could be used identify correspondences between summaries based on phrase-level constituents that were derivable automatically from the output of syntactic or semantic parsers.

Average results from two different ROUGE metrics – ROUGE-2 and ROUGE SU-4 – and from BE are provided in Table 10 for each strategy. (Results from GISTexter's DUC 2006 submission are provided as well.)

5.2.3. Linguistic quality of summaries

In addition to content-based evaluation, machine-generated summaries in DUC have traditionally also been evaluated with regards to a number of linguistic factors designed to gauge the overall clarity, coherence, and readability of multi-document summary or question-directed summary answer.

We used a team of 5 human annotators to evaluate each of the candidate summaries generated for the 50 DUC 2006 topics along 5 different sets of criteria: (1) *grammaticality*, a rough measure of the well-formedness (in terms of grammar and orthographic case) of sentences in the summary; (2) *non-redundancy*, corresponding to the amount of repetition or redundant information contained in the summary text; (3) *referential clarity*, a measure of how often referring expressions (such as pronouns, names, or definite NPs) could be associated with their antecedents; (4) *focus*, corresponding to the degree to which a summary discussed a single topic or theme; and (5), *structure and coherence*, a measure corresponding to the level of organization and structure found in the summary. As with responsiveness, annotators were asked to rate each summary on a 5-point scale, with 1 representing a poorly-formed summary and 5 representing an well-formed summary. Table 11 presents results for these five linguistic criteria from our internal evaluations as well as the official DUC 2006 results for GISTexter.

Table 11
Linguistic quality results for the six strategies

Strategy	Grammaticality	Non-redundancy	Referential clarity	Focus	Coherence
Strategy 1	4.62	3.98	3.24	3.14	2.64
Strategy 2	4.80	4.10	3.68	3.32	2.60
Strategy 3	4.80	4.08	3.48	3.56	3.02
Strategy 4	3.82	3.18	2.82	2.52	2.22
Strategy 5	4.04	3.56	3.00	2.84	2.48
Strategy 6	4.44	4.00	3.36	3.26	2.68
GISTexter (rank)	4.62 (1)	4.50 (4)	3.72 (4)	4.28 (1)	3.28 (1)
Best DUC06	4.62	4.66	4.00	4.28	3.28

Since GISTexter's approach to QDS treats the acquisition of relevant content separately from summary generation, it is not surprising that there are not significant differences between types of candidate summaries in terms of grammaticality, non-redundancy, and referential clarity, and coherence. Only focus – a measure of how well the information in a summary pertains to the same topic – appears to provide any meaningful variation, as summaries constructed from semantically decomposed questions were judged to have a slightly higher focus score than summaries constructed from either syntactically-decomposed questions or sets of keywords.

5.3. Evaluation results of question decompositions

In this section, we present results from experiments targeting both (1) the evaluation of the decomposed questions and (2) the evaluation of the impact of the decomposed questions on the quality of answer summaries.

5.3.1. Intrinsic evaluations

The evaluation of the decomposed questions was performed in two ways. First, the decomposed questions were evaluated against decompositions created by humans. Second, question decompositions were evaluated against questions generated from the answer summaries. The second evaluation was also compared against an evaluation involving only human-generated questions, both from the complex question and from the answer summaries. The evaluation was performed against 8 complex questions that were asked as part of the DUC 2005 question-directed summarization task. The questions correspond to the topics listed in Table 12.

We had 4 human annotators perform manual question decomposition based solely on the complex questions themselves. Annotators were asked to decompose each complex question into the set of sub-questions they felt needed to be answered in order to assemble a satisfactory answer to the question. (For ease of reference, we will refer to this set of question decompositions as QD_{human}.) The sub-questions generated by the annotators were then compiled into a "pyramid" structure similar to the ones proposed in Nenkova and Passonneau (2004). In order to create pyramids, humans first identified sub-questions that sought the same information (or were reasonable paraphrases of each other) and then assigned each unique question a score equal to the number of times it appeared in the question decompositions produced by all annotators.

Next, we used the top-down question decomposition model described in Section 3.2.1 in order to generate a second set of question decompositions ($QD_{top\text{-}down}$). Finally, we used the bottom-up question decomposition algorithm presented in Section 3.2.2 in order to generate a third set of question decompositions($QD_{bottom\text{-}up}$) As with QD_{human}, the sub-questions generated for $QD_{top\text{-}down}$ and $QD_{bottom\text{-}up}$ were combined into pyramid structures by human annotators.

Each of these three sets of question decompositions were then compared against a set of "gold standard" decompositions created by another team of 4 human annotators from the 4 "model summaries" prepared by

Table 12
Pyramid coverage of question decompositions

Topic description	Pyramid score for question decompositions		
	$QD_{top\text{-}down}$	$QD_{bottom\text{-}up}$	QD_{human}
Falkland Islands	0.2012	0.3202	0.3889
Tourist Attacks	0.2317	0.3745	0.5000
Drug Development	0.3114	0.5195	0.6744
Amazon Rainforest	0.2500	0.4091	0.6000
Welsh Government	0.2931	0.4873	0.5091
Robot Technology	0.2268	0.4421	0.6222
UK Tourism	0.0196	0.3917	0.4035
Czechoslovakia	0.2301	0.3116	0.3836
Average	0.2205	0.4070	0.5000

NIST annotators as "gold standard" answers to the 8 complex questions. Each of the three question decompositions described above (i.e. QD_{human}, $QD_{top\text{-}down}$, and $QD_{bottom\text{-}up}$) were then scored against the corresponding "model" question decomposition pyramid using the technique outlined in Nenkova and Passonneau (2004). Table 12 illustrates the Pyramid coverage for $QD_{top\text{-}down}$, $QD_{bottom\text{-}up}$, and QD_{human}. It is to be noted that although the QD_{human} captured 45% of the questions contained in the "model" pyramids, the high average Pyramid score (0.5000) suggests that human question decompositions typically included questions that corresponded to the most vital information identified by the authors of the "model" summaries.

5.3.2. Impact of questions decompositions on QDS

We evaluated the impact of question decompositions by using the responsiveness score. As in Section 5.2.1, the responsiveness score was assessed by a team of five annotators who selected an integer value between 1 and 5 to assess their satisfaction with the information contained in the summary as an answer to the question. In order to better evaluate the impact of question decomposition on the quality of QDS, we separated the "semantic QD" strategy used in Strategies 3 and 6 into 3 separate strategies: (1) a top-down QD strategy, (2) a bottom-up QD strategy, and (3) the "normal" hybrid QD strategy which contains decomposed questions from both semantic QD strategies.

To be able to evaluate the impact of question decomposition on multi-document summarization, we created eight different summaries for each of the 50 DUC 2006 topics and had human annotators evaluate them in terms of overall responsiveness. Results from these experiments are listed in Table 13.

When we wanted to measure the impact of question decomposition on multi-document summarization, we compared the results of the experiments listed in Table 13 against the two baseline strategies (Strategies 1 and 4) in which no question decomposition is available. By computing the difference in responsiveness score between the results obtained in the experiments listed in Table 13, and the baseline experiments, we have found that the largest impact of question decomposition for MDS was obtained in experiment E_4 (Strategy 3). The least impact was obtained in experiment E_5.

5.4. Evaluating textual entailment

In this section, we discuss how we used data taken from the PASCAL Second Recognizing Textual Entailment Challenge (Bar-Haim et al., 2006) in order to evaluate the performance of our textual entailment system.

5.4.1. Textual entailment evaluation

The system for recognizing textual entailment was evaluated as part of the Second PASCAL Recognizing Textual Entailment (RTE) Challenge (Bar-Haim et al., 2006). Systems were evaluated in two ways. First, systems received an *accuracy* score equal to the number of examples from the PASCAL RTE-2 test set where the existence (or non-existence) of entailment was identified correctly. Systems also received an *average precision score* which evaluated systems' ability to rank sentence pairs in order of confidence. Following Voorhees

Table 13
Description of QD evaluation experiments

Experiment	Summarization method	QD method	Overall responsiveness
E_1	QFS	Bag-of-words	2.96
E_2	QFS	Syntactic	3.04
E_3	QFS	Top-down	3.15
E_4	QFS	Bottom-up	3.39
E_5	QFS	Top-down and bottom-up	3.56
E_6	MDS	Bag-of-words	2.10
E_7	MDS	Syntactic	2.64
E_8	MDS	Top-down	2.87
E_9	MDS	Bottom-up	3.12
E_{10}	MDS	Top-down and bottom-up	3.32

Table 14
Accuracy on the 2006 RTE test set

Training data Number of examples		Development set 800		Additional corpora 201,000	
Evaluation metric		Accuracy	Avg. precision	Accuracy	Avg. precision
Task	QA-test	0.5750	0.6135	0.6950	0.8237
	IE-test	0.6450	0.6553	0.7300	0.8351
	IR-test	0.6200	0.6295	0.7450	0.7774
	SUM-test	0.7700	0.8011	0.8450	0.8343
Overall		0.6525	0.6636	0.7538	0.8082

Table 15
Performance of alignment classifier

Classifier	Training set	Precision	Recall	*F*-measure
Linear	10K pairs	0.837	0.774	0.804
Maximum Entropy	10K pairs	0.881	0.851	0.866
Maximum Entropy	450K pairs	0.902	0.944	0.922

and Harman (1999), average precision was computed as the average of the system's precision values at all points in the ranked list in which recall increases. Results from the 2006 RTE-2 Challenge are provided in Table 14.

As noted in Hickl et al. (2006), access to additional sources of training data significantly ($p < 0.05$) enhanced the performance of our system to correctly identify instances of textual entailment. When the system was trained on an additional 201,000 entailment pairs (101,000 positive instances, 100,000 negative instances), performance increased by over 10% overall.

5.4.2. Lexical alignment

The performance of the textual entailment system's *Lexical Alignment* module was evaluated on a set of 1000 phrase pairs extracted from the 800 textual entailment sentence pairs assembled for the 2006 PASCAL RTE-2 Training Set.[8]

Three versions of the alignment module were evaluated. In the first version, a set of 10,000 human-annotated alignment token pairs were used to train a hill-climber-based alignment classifier; the second version used the same set of annotated data to train a Maximum Entropy-based classifier. In the third version, the hill-climber classifier was used to annotate a set of 450,000 alignment pairs extracted from more than 200,000 additional entailment sentence pairs compiled by Hickl et al. (2006); these machine-generated annotations were then used to train a new version of the Maximum Entropy-based alignment classifier. Performance of each of these classifiers is presented below in Table 15.

While alignment classifiers trained on human-annotated data performed admirably (with *F*-scores over 0.8), access to a large amount of machine-annotated training data resulted in substantial boosts in both precision and recall.

5.4.3. Evaluations of the quality of paraphrases

We evaluated the paraphrases generated for each sentence in an entailment pair in two ways. First, we had two human annotators judge whether an automatic-generated paraphrase approximated the semantic content

[8] Training data from the RTE-1 and RTE -2 Challenges are available for download at: http://www.pascal-network.org/Challenges/RTE2/Datasets/.

Table 16
Evaluation of the quality of paraphrases

Document corpus	# Sentences	# Candidate paraphrases	Accuracy (%)	Paraphrase quality
Newswire collection	350	5502	22.4	4.05
World Wide Web	350	10,106	49.3	3.03
Overall	700	15,608	39.8	3.23

of the original passage. If the paraphrase was deemed to be acceptable by both judges, another pair of annotators were asked to subjectively grade the quality of the paraphrase on a 5-point scale, with a score of 5 being equal to a "perfect" paraphrase, and 1 being equal to a "marginally acceptable" paraphrase of the original text. No additional instruction was given to annotators on how to score paraphrases. All paraphrases which received annotator scores that differed by more than 3 points were excluded from our results.

In order to perform this subjective evaluation of paraphrase quality, we used our *Paraphrase Generation* module to generate (as many as) the top 250 paraphrases for 350 (randomly-selected) sentences taken from the PASCAL RTE-2 Test Set. (As in Hickl et al. (2006), paraphrases were generated for the text span that occurred between the top pair of aligned tokens identified by the *Lexical Alignment* module for each sentence.) Paraphrases were generated using two different corpora: (1) a large, 1 million-document newswire corpus and (2) the top 500 documents retrieved from the World Wide Web [9] We then selected one candidate paraphrase from the top 10 newswire- and WWW-based paraphrases generated for each of the 350 original sentences; these text-paraphrase pairs were then pseudo-randomized and submitted to annotators for judgment.

Table 16 presents results from this evaluation. Annotators found 49.3% of the text-paraphrase pairs generated from WWW documents to be at least "marginally acceptable"; this number dipped to 22.4% when paraphrases generated from newswire documents were considered. Despite the lower overall accuracy of the newswire-based paraphrases, annotators generally considered these paraphrases to be of higher quality: newswire-based paraphrases received an average 4.05 "paraphrase quality" score, while WWW-based paraphrases received only a 3.03 average score.

In addition to measuring the accuracy and quality of generated paraphrases, we also evaluated the impact that access to paraphrases had on the performance of our system on the 2006 PASCAL RTE-2 Test Set. In order to perform this evaluation, we used our *Paraphrase Generation* module in order to generate (as many as) the top 250 candidate paraphrases for the 1600 sentences used in the PASCAL RTE-2 Test Set. (A total of 340,402 candidate paraphrases were generated from these 1600 sentences.) Access to this collection of boosted the overall accuracy of the system by 4.13%, increasing from 71.25% accuracy to 75.38% overall.

5.5. Evaluations of the summary selections

In this section, we evaluate the performance of our Pyramid-based automatic summarization selection system using data from the 2006 DUC Question-Directed Summarization Task.

Table 17 presents the output of the Pyramid-based summary selection module for the 50 topics featured in the 2006 DUC evaluations.

Although summaries based on semantic question decomposition received the highest automatically-computed Pyramid score for 32 of the 50 topics (64%), the average overall responsiveness (as determined by NIST assessors) 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 our QDS system's question-answering based strategies (56%) than its traditional summarization-based strategies (44%), the average overall responsiveness remained relatively constant for both types of summaries: Q/A-based summaries received an average overall responsiveness score of 2.875, while MDS-based summaries scored 2.828.

[9] We used the *Google* API in order to retrieve these WWW documents.

Table 17
Output of pyramid-based summary selector

Strategy	# Summaries	LCC resp	Avg resp
Strategy 1	4	2.50	2.36
Strategy 2	5	3.40	2.22
Strategy 3	19	2.79	2.10
Strategy 4	5	2.60	1.84
Strategy 5	4	3.00	2.32
Strategy 6	13	2.85	2.34
Total	50	2.86	2.20

Table 18
Evaluation of the summary selection

Strategy	# Selections	# Most Responsive	Accuracy (%)
Strategy 1	4	5	60
Strategy 2	5	6	83
Strategy 3	19	17	94
Strategy 4	5	5	80
Strategy 5	4	4	75
Strategy 6	13	13	92
Total	50	50	86

We used the content responsiveness scores compiled by our own annotators (as described in Section 5.2.1) in order to determine how frequently the Pyramid-based summary selection module selected the most responsive of the 6 candidate summaries generated for each topic. Table 18 details the accuracy of the Pyramid-based summary selection module for each of the 6 strategies.

When judged over the 50 topics in DUC 2006, our summary selection module selected the most responsive summary (as judged by our human annotators) 86% of the time (43/50). This a particularly encouraging result, as it suggests that Pyramid creation using TE is sufficiently discriminative to identify differences even among sets of highly responsive and similar summaries.

6. Conclusions

In this paper we have described GISTexter, a summarization system that was designed to satisfy the user information needs expressed by a complex question. GISTexter processes complex questions by performing syntactic and semantic decompositions. Additionally, two approaches for generating summaries are used, which enable six different summarization strategies. To combine the summarization strategies, we have used textual entailment in two ways: (1) for selecting information; and (2) for scoring the summaries based on pyramid-based measures. We have shown that these methods have allowed us to automatically generate question-directed summaries that are not only coherent and readable but that are highly responsive to the information needs of users.

Acknowledgement

This material is based upon work funded in whole or in part by the U.S. Government, and any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US Government.

Appendix A. Relations between decomposed questions

Relation	Example
DEFINITION Relations	
DEFINITION(predicate)	What are the procedures for generating new drugs?
DEFINITION(argument)	What is a drug?
ELABORATION relations	
ELABORATION(Hyponymy)	Which new drugs are being produced?
ELABORATION(Number)	How many new drugs are being produced?
ELABORATION(Time)	When are new drugs being produced?
ELABORATION(Location)	Where are new drugs being produced?
ELABORATION(Manner)	How do pharma companies produce new drugs?
ELABORATION(Quantity)	How many new drugs did pharma companies produce?
ELABORATION(Rate)	What was the greatest number of new drugs that pharma companies produced?
ELABORATION(Duration)	How long will pharma companies produce new drugs?
ELABORATION(Trend)	How much has pharma companies' production of new drugs increased/decreased?
ELABORATION(Inchoative)	When did pharma companies begin producing new drugs?
ELABORATION(Terminative)	When did pharma companies stop producing new drugs?
ELABORATION(Subjective)	How beneficial/detrimental was pharma companies' production of new drugs?
EVENT–EVENT relations	
CAUSE(Event)	What steps did pharma companies take to produce new drugs?
INTENTION(Event)	Why did pharma companies produce new drugs?
EFFECT(Event)	What happened because pharma companies produced new drugs?
RESULT(Event)	What advantages resulted pharma companies producing new drugs?
OUTCOME(Event)	What profits did pharma companies take from producing new drugs?
TEMPORAL(Event)	What happened after/before pharma companies produced new drugs?
RELATIONSHIP(Event)	What is the connection between pharma companies producing new drugs and the higher incidence of autism in the US?
GENERALIZATION/SPECIALIZATION relations	
SPECIALIZATION(Predicate)	What kind of activities are involved in the creation of new drugs?
GENERALIZATION (Predicate)	What commercialization effects are typical in the drugs industry?
COUNTERFACTUAL relations	
NEGATION(Predicate)	What pharma companies don't produce new drugs?
NEGATION(Argument)	What pharma companies have produced no (new) drugs?
NEGATION(Attribute)	What pharma companies produce only existing drugs?
EXCEPTIVE(Argument)	What pharma companies are producing new drugs other than MAOI inhibitors?
CONTRARY(Fact)	Despite the FDA's ban on new drug development, which pharma companies are producing new drugs?
ANSWER RESTRICTING relations	
RESTRICT(Location)	What pharma companies are producing new drugs in the U.S.?
RESTRICT(Temporal)	What pharma companies are producing new drugs in 2006?
RESTRICT(Attribute)	What up-and-coming pharma companies are producing new drugs?
EPISTEMIC relations	
EPISTEMIC(Event)	Are pharma companies producing new drugs?

Appendix A (*continued*)

Relation	Example
EPISTEMIC-CONDITIONAL(Event)	Is it known if pharma companies are producing new drugs?
EPISTEMIC-EVIDENTIAL(Event)	Is there evidence that pharma companies are producing new drugs?
EPISTEMIC-REPORTED(Event)	Does anyone believe that pharma companies are producing new drugs?
EPISTEMIC-ALTERNATIVE(Event)	Do U.S. pharma companies produce new drugs or research new drugs?
EPISTEMIC-ELABORATION(Event)	Do pharma companies produce new drugs [with the help of foreign labs]?
PARALLEL relations	
PARALLEL(Predicate)	What pharma companies work with infectious agents?
PARALLEL(Predicate)	What pharma companies research new drugs?
PARALLEL(Argument)	What pharma companies produce vaccines?
PARALLEL(Attribute)	What pharma companies produce affordable drugs?

References

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., et al. (2006). The second PASCAL recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*.

Barzilay, R., & Lee, L. (2003). Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *HLT-NAACL*.

Clifton, T., & Teahan, W. (2004). Bangor at TREC 2004: question answering track. In *Proceedings of the thirteenth text retrieval conference*.

Dagan, I., Glickman, O., & Magnini, B. (2005). The PASCAL recognizing textual entailment challenge. In *Proceedings of the PASCAL challenges workshop*.

Harabagiu, S. (2004). Incremental topic representations. In *Proceedings of the 20th COLING conference, Geneva, Switzerland.*

Harabagiu, S., Moldovan, D., Pasca, M., Surdeanu, M., Mihalcea, R., Girju, R., et al. (2001). Answering complex, list and context questions with LCC's question-answering server. In *Proceedings of the tenth text retrieval conference*.

Harabagiu, S., Moldovan, D., Clark, C., Bowden, M., Williams, J., Bensley, J. (2003). Answer mining by combining extraction techniques with abductive reasoning. In *Proceedings of the twelfth text retrieval conference*.

Harabagiu, S., Hickl, A., Lehmann, J., & Moldovan, D. (2005). Experiments with interactive question-answering. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*.

Harabagiu, S., Moldovan, D., Clark, C., Bowden, M., Hickl, A., & Wang, P. (2005). Employing two question answering systems in TREC 2005. In *Proceedings of the fourteenth text retrieval conference*.

Hickl, A., Williams, J., Bensley, J., Roberts, K., Rink, B., & Shi, Y. (2006). Recognizing textual entailment with LCC's groundhog system. In *Proceedings of the second PASCAL recognizing textual entailment challenge, Venice, Italy*.

Hovy, E., Lin, C.-Y., & Zhou, L. (2005). Evaluating DUC 2005 using basic elements. In *Proceedings of the document understanding workshop (DUC-2005)*.

Lacatusu, F., Hickl, A., Harabagiu, S., & Nezda, L. (2004). Lite-GISTexter at DUC 2004. In *Proceedings of the document understanding workshop (DUC-2004), Boston, MA*.

Lacatusu, F., Hickl, A., Aarseth, P., & Taylor, L. (2005). Lite-GISTexter at DUC 2005. In *Proceedings of the document understanding workshop (DUC-2005) presented at the HLT/EMNLP annual meeting*.

Lehmann, J., Aarseth, P., Nezda, L., Deligonul, M., & Hickl, A. (2005). TASER: a temporal and spatial expression recognition and normalization system. In *Proceedings of the 2005 automatic content extraction conference, Gaithersburg, MD*.

Lin, C.-Y. (2004). ROUGE: a package for automatic evaluation of summaries. In *Proceedings of the workshop on text summarization branches out (WAS 2004), Barcelona, Spain*.

Lin, C.-Y., & Hovy, E. (2000). The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th COLING conference, Saarbrücken, Germany*.

Lin, C.-Y., & Hovy, E. (2003). Automatic evaluation of summaries using *n*-gram cooccurrence statistics. In *Proceedings of 2003 language technology conference (HLTNAACL 2003)*.

Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., et al. (2002). LCC tools for question answering. In *The eleventh text retrieval conference.*

Nahm, U. Y., & Mooney, R. J. (2000). A mutually beneficial integration of data mining and information extraction. In *Proceedings of the seventeenth national conference on artificial intelligence*.

Nenkova, A., & Passonneau, R. (2004). Evaluating content selection in summarization: the pyramid method. In *HLT-NAACL 2004, Boston, MA*.

Pasca, M., & Harabagiu, S. (2001). High performance question/answering. In *Proceedings of the 24th annual international ACM SIGIR conference*.

Passonneau, R., Nenkova, A., McKeown, K., & Sigelman, S. (2005). Applying the pyramid method in DUC 2005. In *Proceeding of the document understanding workshop (DUC'05)*.

Quinlan, R. (1998). C5.0: An Informal Tutorial. RuleQuest. http://www.rulequest.com/see5-unix.html.

Shinyama, Y., Sekine, S., Sudo, K., & Grishman, R. (2002). Automatic paraphrase acquisition from news articles. In *Proceedings of human language technology conference, San Diego, CA*.

Voorhees, E., & Harman, D. (1999). Overview of the eighth text retrieval conference (TREC-8). In *Proceedings of the 8th text retrieval conference*.