# Question Answering with LCC's CHAUCER at TREC 2006

**Andrew Hickl, John Williams, Jeremy Bensley, Kirk Roberts, Ying Shi, and Bryan Rink**
Language Computer Corporation
1701 North Collins Boulevard
Richardson, Texas 75080
andy@languagecomputer.com

## Abstract

CHAUCER is a Q/A system developed for (a) combining several strategies for modeling the target of a series of questions and (b) optimizing the extraction of answers. Targets were modeled by (1) topic signatures; (2) semantic types; (3) lexico-semantic patterns; (4) frame dependencies; and (5) predictive questions. Several strategies for answer extraction were also tried. The best-performing strategy was based on the use of textual entailment.

## 1. Introduction

As with the TREC 2004 and TREC 2005 Question-Answering Track evaluations, the main task of the TREC 2006 evaluations required systems to answer a series of questions that sought information about a specific *target*. In order to provide information relevant to a target, two forms of semantic knowledge had to be reconciled: (1) the *expected answer types* of the questions; and (2) the *semantic signatures* of the targets. For example, in order to answer a question about the popular television show *The Daily Show* like *What was the title for The Daily Show's 2000 election coverage?*, question-answering systems need both the ability to recognize *titles* in texts as well as access to the forms of contextual knowledge to identify those titles that could potentially correspond to the names of television show segments.

**Q149.6** (*The Daily Show*) What was the title for The Daily Show's 2000 election coverage?

**Answer:** "The Daily Show" is sponsored by cool cars, cell phones and movies – and its big corporate sponsors for its **"Indecision 2000"** election coverage include Yahoo, Volkswagen and Snapple.

Table 1: Question 149.6: *The Daily Show*

Unlike our previous experience with series of questions, in which the target was processed as a pair (lexical-string, semantic-type), in CHAUCER we have developed a methodology of generating the semantic signature of the target and using interactions between this signature and the questions from a given series. In TREC 2006, we focused on the interactions between targets and only two forms of questions, namely (1) factoid questions and (2) "other" questions. In future work, we shall also consider the interaction between target signatures and list questions.

To be able to answer questions based on semantic signatures of targets, we have also considered (1) a two-tiered passage retrieval system and (2) the use of multiple answer extraction strategies. Answers were extracted by making also use of two novel approaches: (a) the automatic generation of question-answer pairs, known as *predictive questions* from texts and (b) the recognition of forms *textual entailment* between a question and a candidate answer.

The rest of this paper is organized as follows. Section 2 describes the CHAUCER Q/A system. Section 3 discusses how factoid questions were answered while Section 4 shows how we processed "other" questions. Results from CHAUCER's participation in the main task of the TREC 2006 QA track are presented in Section 5; Section 6 summarizes our conclusions.

## 2. The CHAUCER Question-Answering System

In this section, we describe the architecture of the CHAUCER question-answering system used to answer series of factoid and list questions for the TREC 2006 QA main task. The architecture of CHAUCER is presented in Figure 1.

### Target Processing

CHAUCER begins the process of providing answers to a series of questions by submitting the series' target to a *Target Processing* module. Targets are initially sent to a *Target Type Detection* module, which uses a Maximum Entropy classifier in order to associate the target with one of six different target type categories. In our TREC 2006 work, targets were classified as either (1) a PERSON (e.g. *Warren Moon*), (2) an ORGANIZATION (*American Enterprise Institute*), (3) a LOCATION (*Amazon River*), (4) an EVENT (*1991 eruption of Mount Pinatubo*), (5) an AUTHORED WORK (*The Daily Show*), or a (6) GENERIC NOUN (*avocados*). Following this classification, keywords were extracted from the target and sent to a *Document Retrieval* module in order to retrieve a set of documents relevant to the target itself. These documents are then sent to a *Topic Representation* module which employs two different statistical approaches based on the methods for computing *topic signatures* (Lin & Hovy 2000) in order to model the topic of a relevant set of documents.

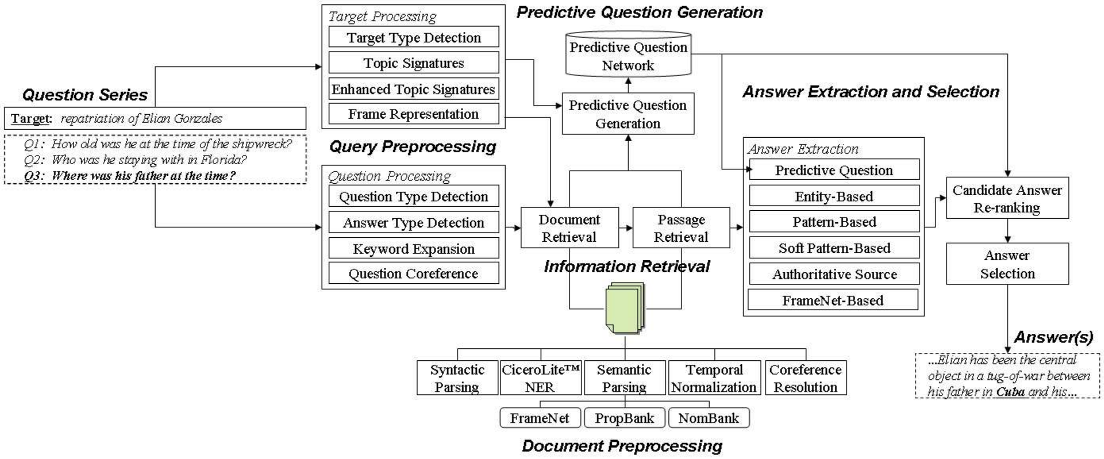

Figure 1: Architecture of the CHAUCER Question-Answering System

CHAUCER uses a subset of the text passages returned during Target Processing in order to generate a set of *predictive questions* that could potentially be asked about a given target.

## Question Processing

Once a set of predictive questions have been generated for a target, CHAUCER sends each question in a question series to a QUESTION PROCESSING module. Questions are initially sent to an *Annotation Module*, which uses LCC's suite of natural language processing tools to tokenize, part-of-speech tag, and syntactically parse each question. Questions are also annotated with one of over 300 different named entity classes from LCC's *CiceroLite* and are also semantically parsed using LCC's PropBank-, NomBank-, and FrameNet-based semantic parsers. Following annotation, questions are first sent to a *Question Type Detection* module, which uses a set of syntactic heuristics in order to classify individual questions as an example of a factoid, list, or "other" question. Factoid and list questions are then sent to an *Answer Type Detection* module, which follows (Li & Roth 2002) and (Chakrabarti, Krishnan, & Das 2005) in using a two-stage Maximum Entropy-based classifier in order to identify the expected answer type (EAT) of the question from LCC's answer type hierarchy. Keywords are then extracted from each question and sent to a *Keyword Expansion* module designed to identify additional key words and phrases that could be used to enhance the quality of document and passage retrieval for a particular question. CHAUCER also incorporates a *Question Coreference* module which uses a heuristic-based approach to resolve instances of pronominal and nominal coreference within a question series.

## Document Preprocessing and Retrieval

We preprocessed the AQUAINT corpus with five types of information. First, we used LCC's implementation of the Collins parser to provide a full syntactic parse for every document in the corpus. Second, we used three different semantic parsers in order to identify semantic dependencies imposed by both verbal and nominalized predicates. In addition to LCC's PropBank and NomBank parsers, we also used LCC's FrameNet-based semantic parser to identify instances FrameNet frames in natural language texts; a separate role classifier was used to identify roles associated each FrameNet frame. Third, we used LCC's CICEROLITE named entity recognition system in order to classify more than 300 different types of names found in the corpus. We also used more than 500 lexicons and gazetteers derived from web-based resources in order to tag additional name types not covered by CICEROLITE. Fourth, we used LCC's TASER temporal normalization system (Lehmann *et al.* 2005) in order to map temporal expressions found in documents to a standardized (ISO 8601) format. Finally, as with question series, we used a conservative heuristic-based approach in order to resolve instances of nominal and pronominal coreference.

Following preprocessing, the AQUAINT corpus was indexed using the Lucene Information Retrieval engine in order to allow documents to be retrieved using queries composed of either literal strings, stemmed words, or any of the entity types identified by CICEROLITE.

## Answer Extraction and Selection

CHAUCER uses a battery of six different strategies to extract answers from retrieved passages. (Each of these six strategies are described in detail in Section 3.) Following *Answer Extraction*, the top five candidate answers identified by each strategy are then sent to a *Candidate Answer Reranking* module which uses a Maximum Entropy-based reranker (based on (Ravichandran, Hovy, & Och 2003) in order to provide a single ranked list of candidate answers for a particular question. The re-ranked list of answers were then sent to a final *Answer Selection* module which uses the state-of-the-art textual entailment system described in (Hickl *et al.* 2006) in order to identify the single answer passage whose meaning is most likely to be entailed by the meaning of the original question.

### List Answer Extraction

Chaucer leverages the basic factoid question-answering (Q/A) pipeline we have described in this section in order to answer list questions from a series as well. Table 2 lists the final answers given for question 181.3 (*List the artists represented in the collection.*).

In TREC 2006, we utilized a method based on web counts from various search engines to determine how much of an association there was between the candidate answer and both the series target and answer type term (in this case *Hermitage Museum* and *artist*, respectively). The scores from these methods were then combined to give each candidate answer a final composite score. We then considered all answers above a dynamically-defined threshold.

**Q181.3** (*Hermitage Museum*) List the artists represented in the collection.

| Rank | Answer | Result |
|---|---|---|
| 1 | Vladimir Mayakovsky | Incorrect |
| 2 | Da Vinci | Correct |
| 3 | Michelangelo | Correct |
| 4 | Rembrandt | Correct |
| 5 | Poussin | Correct |
| 6 | Rubens | Correct |
| 7 | van Gogh | Correct |
| 8 | Caspar David Friedrich | Incorrect |
| 9 | Guido Reni | Incorrect |
| 10 | Parmigianino | Incorrect |

Table 2: List Extraction Example

## 3. Answering Factoid Questions

This section describes several of the novel techniques that were introduced into the Chaucer factoid Q/A pipeline for the TREC 2006 evaluations.

### Generating Predictive Questions

Following *Target Processing*, the top 50 passages taken from the set of target-relevant documents are re-ranked according to a composite score based on (1) the weights associated with $TS_1$ terms and $TS_2$ relations found in the passage, (2) the weights associated with the soft patterns and (3) manually-created patterns found in the passage, and (4) the weights assigned to any FrameNet frame detected in the passage. Following (Harabagiu, Lacatusu, & Hickl 2006), we used the output of LCC's PropBank-based semantic parser in order to generate natural language questions from each predicate found in the top-ranked passages. Given a set of semantic dependencies associated with a predicate, we used a set of heuristics in order to select a single argument from each predicate to serve as the answer of a generated "factoid" question. Features derived from LCC's CiceroLite were then used to map the argument to one of the possible wh-phrase (e.g. *Who,What,Where*) used in natural language questions. The entire passage was then submitted to a *Question Generation* module which utilized the dependency structure of the passage in order to generate a natural language question. Generated questions were then paired with their original passage-length answers and stored in a *Predictive Question Database* for later use. Table 3 provides examples of the predictive questions generated for Question 149.6, *What was the title for The Daily Show's 2000 election coverage?*.

**Q149.6** (*The Daily Show*) What was the title for The Daily Shows 2000 election coverage?

| | |
|---|---|
| $PQ_1$ | What was just the sort of piece that "The Daily Show," revels in? |
| $PQ_2$ | Who has hired Dole as a guest political commentator for its election coverage? |
| $PQ_3$ | Who best summed up Comedy Central's coverage of the 2000 Republican Convention? |
| $PQ_4$ | Who will join the cast of the "The Daily Show"? |

Table 3: Examples of the Predictive Questions Generated for Question 149.6

### Question Processing

In this section, we describe the three types of *Question Processing* Chaucer performs for each question.

**Keyword Expansion** Keywords extracted from each question were processed by a *Keyword Expansion* module that was designed to identify additional synonymous keywords that could be used to augment the query Chaucer used to retrieve documents. This module used a set of heuristics in order to append synonyms and alternate keywords from a database of similar terms developed by LCC for previous TREC QA evaluations. In addition, we used the topic representations generated by Chaucer's *Target Processing* module in two ways. First, we included as keyword expansions all $TS_1$ terms that were found either in the WordNet synsets for a particular keyword. Second, we also considered all terms found in set of the target-relevant documents that were linked to the question keyword via $TS_2$ relations with relevance scores above a fixed threshold.

**Question Coreference** We incorporated a heuristic-based *Question Coreference* module in order to resolve referring expressions found in the question series to antecedents mentioned in previous questions or in the target description. First, we used heuristics for performing name aliasing and nominal coreference from CiceroLite in order to identify the full referent for each partial name mention found in the question series. Next, we constructed an *antecedent list* from all of the named entities that occurred in the question series prior to the current question. Each potential antecedent and referring expression found in the series were then annotated with name class, gender, and number information available from CiceroLite. We then used the Hobbs Algorithm (Hobbs 1978) in order to match referring expressions to candidate antecedents. When no compatible antecedent could be identified from the antecedent list, we made no further attempt to resolve the referring expression found in the question. Table 4 presents an example where Chaucer was able to resolve the antecedent of the pronoun *it* correctly; in contrast, Table 5 presents an example where our approach is unable to correctly recognize coreference between a noun phrase from the question (*the program*) and the target phrase (*television show Cheers*).

| | |
|---|---|
| Target **143**: *American Enterprise Institute* | |
| **Q143.2**: What is the full title of the *organization*? | |
| **Q143.3**: When was *it* founded? | |

Table 4: Correctly-resolved Question Coreference

| |
|---|
| Target **150**: *television show Cheers* |
| What year was *the program* first broadcast? |

Table 5: Incorrectly-resolved Question Coreference

**Answer Type Detection** CHAUCER follows much recent work in *Answer Type Detection* (Li & Roth 2002; Chakrabarti, Krishnan, & Das 2005) in using a two-stage Maximum Entropy-based classifier in order to recognize the expected answer type of a question. CHAUCER's first answer type classifier the*coarse answer type* of the question; currently, we consider the following six coarse answer types: (1) HUMAN [1], (2) LOCATION, (3) ENTITY, (4) ABBREVIATION, (5) NUMERIC, and (6) DESCRIPTION. Once a single coarse answer type has been identified for each question, a second classifier is then used to map the question to one of the set of fine answer types associated with each coarse type. In our work, we have used a hierarchy of over 260 fine entity types derivable from the more than 300 different entity types recognized by LCC's CICEROLITE. Table 6 presents examples of the fine types we associated with each coarse answer type.

| UIUC Coarse Type | LCC Fine Types | Examples |
|---|---|---|
| ABBREVIATION | 2 | Acronym, Expanded Acronym |
| DESCRIPTION | 2 | Death Manner, Quote |
| ENTITY | 45 | Animal, Authored Work, Chemical Element |
| HUMAN | 106 | Coach, Writer, Govt Person, Medical Org |
| LOCATION | 61 | Country, Mountain, Planet, Ocean |
| NUMERIC | 46 | Age, Velocity, Money |

Table 6: Distribution of Fine Types in LCC's Answer Type Hierarchy

Our coarse answer type classifier was trained using the 5500-question UIUC Answer Type Corpus; we re-annotated this corpus with fine answer types from the LCC answer type hierarchy in order to train our fine answer type classifier. Table  and Table  presents evaluation results of our systems for answer type detection, as evaluated on the TREC 2006 questions.

## Document Retrieval

In CHAUCER, we experimented with a novel two-tiered approach approach to *document retrieval* which used a conservative entity-based answer extraction strategy in order to augment traditional keyword- and entity-based retrieval queries. First, the top 200 documents were retrieved using an expanded keyword query; these documents were then passed to a *Passage Retrieval* module which was used to extract the most relevant text passages from each document. Next, the top 500 retrieved passages were then sent to an entity-type based *Answer Extraction* module, which re-ranked passages according to the distribution of (1) entities matching the EAT of the question, (2) topic signature terms and relations identified during *Target Processing*, and (3) keywords extracted from the original question. The original set of 200 retrieved documents were then re-ranked based on the distribution of these top-ranked passages; only the top 50 documents were then considered by later *Answer Extraction* and *Answer Selection* modules.

| | Score |
|---|---|
| Coarse | 92.9 |
| Fine | 84.0 |

Table 7: Answer Type Detection on TREC 2006 Questions

| Coarse Type | Total Questions | Score |
|---|---|---|
| ENTITY | 31 | 67.7 |
| DESCRIPTION | 2 | 100 |
| LOCATION | 66 | 93.9 |
| NUMERIC | 179 | 91.6 |
| ABBREVIATION | 3 | 100 |
| HUMAN | 84 | 92.9 |

Table 8: Fine Answer Type Detection on TREC 2006 Questions

By approximating incorporating the entity constraints on candidate answerhood into its *Document Retrieval* engine, CHAUCER is able to eliminate documents that may be keyword-dense but may not contain any relevant candidate answers. This approach also enables CHAUCER retain a high level of precision in answering factoid questions while processing fewer documents. In our experiments using the factoid questions from TREC 2005, we found no appreciable improvement in the precision or the coverage of CHAUCER's answers when more than 50 documents were retrieved.

## Answer Extraction

CHAUCER uses a total of six different *answer extraction* strategies in order to identify exact answers from a set of retrieved passages.

**Entity-Based Answer Extraction** CHAUCER's *entity-based* answer extraction strategy takes advantage of the large number of entity types recognized by LCC's CICEROLITE named entity recognition system in order to identify candidate answers to individual questions. Under this approach, only passages that contain entity types associated with the question's expected answer type are considered as candidate answers; remaining passages are then re-ranked based on the distribution and density of question keywords discovered in each passage. The wide coverage of LCC's CICEROLITE allows this strategy to retrieve a surprising number of exact answers, even without incorporating additional lexico-semantic features. With a question like Q181.3 (*Who is the*

[1] The HUMAN coarse answer type encompasses the more familiar PERSON and ORGANIZATION entity types.

*manager of Manchester United?*), CHAUCER's *Answer Type Detection* module associates the EAT with a number of entity types related to people and organizations – including POLITICIAN, NOBILITY, and COACH. Here, it is able to return the correct answer without the need for patterns or other types of semantic information.

| **Q181.3** (*Manchester United Football Club*) Who is the manager of Manchester United? | |
|---|---|
| Fine Answer Type | COACH |
| Answer | Manchester United soccer club manager **Alex Ferguson** gave his qualified backing to quotas on foreign players in English football, hours after scooping a top book prize for his autobiography Managing My Life on Friday. |

Table 9: Question 183.1: Correct Entity Detection

Of course, the performance of this strategy is also ultimately limited by the coverage and quality of CICEROLITE, as well. In TREC 2006, we failed to retrieve an answer to question Q157.5 (*Who was the President of the U.N. Security Council for August 1999?*): even though CHAUCER correctly identified the expected answer type of the question as a type of PERSON, this strategy failed to extract the correct answer *Martin Andjaba* because it was not tagged with one of the entity types associated with this question's EAT.

| **Q157.5** (*United Nations (U.N.)*) Who was the President of the U.N. Security Council for August 1999? | |
|---|---|
| Fine Answer Type | GOVT PERSON |
| Answer | **Martin Andjaba**, Namibian ambassador to the United Nations, will succeed Hasmy Agam of Malaysia as the president of the U.N. Security Council as of August 1. |

Table 10: Question 157.5: Failure of Entity Detection

**Pattern-Based Answer Extraction** CHAUCER utilizes two different pattern-based approaches in in order to identify answers to a small set of question types. Hand-crafted extraction patterns are first used to extract answers to the question types frequently asked in past TREC evaluations from the AQUAINT corpus. In addition, we have experimented with using structured web-based sources of information related to people, places, and authored works (e.g. *imdb.com*, *nndb.com*, *iplpotus.com*) in order to answer other specific types of questions.

**Soft Pattern-based Answer Extraction** Following (Cui, Kan, & Chua 2004), we used a *soft pattern* matching approach in order to automatically generate additional patterns that could be used to extract exact answers to different types of factoid questions. Under this approach, we first organized questions taken from the previous TREC QA evaluations into a set of 30 different categories, based on expected answer type. Once this classification was in place, we used the set of "gold" answer sentences associated with each question in a category in order to train a bigram soft pattern model for each question category. As with (Cui, Kan, & Chua 2004), we then used this soft pattern model in order to compute the percentage match between the set of training examples and the the candidate answers retrieved for a question.

**FrameNet-Based Answer Extraction** We leveraged semantic dependency information from LCC's FrameNet-based parser in order to extract candidate answers from the set of top-ranked passages retrieved for a question. Under this approach, we used LCC's FrameNet parser in order to recognize a set of semantic frame dependencies for each question. Passages retrieved for each question were then ranked based on (1) the distribution of semantic frames detected in each passage and (2) the parser's estimation of the confidence of the frame assignment. For example, as depicted in Table 11, LCC's FrameNet parser identifies two FrameNet frames for a question like Q199.4 (*How old was Padre Pio when he died?*): (1) a AGE frame, used to encode the *age* of an *entity* and (2) a DEATH frame, used to encode information about an event in which a *protagonist* dies.

| **Q199.4** (*Padre Pio*) How old was Padre Pio when he died? | |
|---|---|
| Age | Entity: Padre Pio |
| Death | Protagonist: Padre Pio |
| **Answer:** Padre Pio, who died in 1968 at the age of 81, was right. | |
| Age | Entity: Padre Pio |
| | Age: 81 |
| Death | Protagonist: Padre Pio |
| | Time: 1968 |

Table 11: Correct Answer based on FrameNet Matching

| **Q141.2** (*Warren Moon*) Where did Moon play in college? | |
|---|---|
| Being Located | Location: Where |
| Competition | Participant: Moon |
| **Answer:** Warren Moon did not play at all for Kansas City as coach Gunther Cunningham tried to protect his 43-year-old backup quarterback from a banged-up offensive line. | |
| Being Located | Location: Kansas City |
| Competition | Participant: Warren Moon |
| | Location: Kansas City |

Table 12: Incorrect Answer based on FrameNet Matching

A FrameNet parse of the top-ranked passage (*Padre Pio, who died in 1968 at the age of 81, was right.*), also includes the same two FrameNet frames detected in the question. By aligning the frame slots associated with the AGE frame both found in the question and the answer, we found compelling evidence which could be used to identify to this candidate answer being as the right answer.

However, the alignment of frames does not always point to the right answer. In an example like question Q141.2 (*Where did Moon play in college?*), both the question and the top-ranked candidate answer are associated with both a BEING-LOCATED and a COMPETITION frame, yet the *location* argument identified in the answer points to a location other than the answer to the question.

**Predictive Question-Based Answer Extraction** Finally, CHAUCER uses the set of *predictive questions* generated as a part of *Target Processing* in order to provide an additional source of candidate answers. Following *Predictive Question Generation*, CHAUCER uses the question similarity metrics described in (Harabagiu *et al.* 2005) in order to select the top 50 most similar predictive question-answer pairs stored in the *Predictive Question Network*. These question-answer pairs are then ranked based on (1) their overall similarity to the original question, (2) the presence of entity types corresponding to the EAT of the original question, and (3) the distribution of question keywords. After re-ranking, the top 25 question-answer pairs are then sent to the *Answer Ranking* and *Answer Selection* modules.

### Answer Ranking

Following *Answer Extraction*, CHAUCER uses a Maximum Entropy-based re-ranker (similar to (Ravichandran, Hovy, & Och 2003)) in order to compile answers from each of the six answer extraction strategies into a single ranked list. This re-ranker was trained on the top ten answers returned by each of CHAUCER's answer extraction strategies for each of the questions taken from the TREC 2004 and TREC 2005 datasets. (Answers were keyed automatically using "gold" answer patterns made available by the TREC organizers and other participating teams.) Five sets of features were used in this re-ranker: (1) the strategy used to extract the answer, (2) the EAT of the original question, (3) the entity type associated with the exact answer, (4) the redundancy of the answer across the top-ranked answers, and (5) the confidence assigned to the answer by each answer extraction strategy.

### Answer Selection

Once a ranking of candidate answers is performed, the top 25 answers were then sent to an *Answer Selection* module which leverages LCC's state-of-the-art textual entailment system in order to identify the answer which best approximates the semantic content of the original question. Popularized by the recent PASCAL Recognizing Textual Entailment (RTE) Challenges (Dagan, Glickman, & Magnini 2005), textual entailment systems seek to identify whether the meaning of a *hypothesis* can be reasonably inferred from the meaning of a corresponding *text*. While the RTE Challenges have focused to-date only on the computation of entailment relationships between sentence-length texts and hypotheses, our recent work (Harabagiu & Hickl 2006) has shown that current systems for recognizing TE can be leveraged to accurately identify entailment relationships between questions and answers – or even questions and other questions.

CHAUCER uses the entailment system described in (Hickl *et al.* 2006) in order to estimate the likelihood that a question entails either (1) a candidate answer extracted by one of CHAUCER's six answer extraction strategies or (2) a predictive question generated by the *Predictive Question Generation* module. Following (Harabagiu & Hickl 2006), we first filtered all candidate answers that were not entailed by the original questions. The remaining candidate answers (including any remaining predictive question-answer pairs) were re-ranked based on the entailment confidence output by the RTE system. The top-ranked answer was then returned as our submitted answer.

In our TREC 2006 experiments, we found that the establishment of textual entailment between a question and a predictive question or an answer passage to be a powerful tool for the validation of candidate answers.

In Table 13, we present an example where the top-ranked predictive question generated from text is entailed by the original question; in this case, the correct answer associated with the predictive question is also the answer to the original question.

| **Q142.3** (*LPGA*) How many events are part of the LPGA tour? | |
|---|---|
| $PQ_1$ | How many events did the LPGA expand its schedule to? |
| Answer$_1$ | Under Ritts, the LPGA expanded its schedule from 36 to **43** events; increased purses to $36.2 million |

Table 13: Correct Answer: Entailed Predictive Question

However, this is not always the case: in Table 14, the predictive question (*Where was the 82nd Airborne Division formed?*) is incorrectly classified as being entailed by the original question (*Where in the US is the [82nd Airborne Division] based?*). Here, the failure of the TE system to distinguish between the implications of the verbs *formed* and *based* results in the selection of an incorrect answer.

| **Q144.1** (*82nd Airborne Division*) Where in the US is the division based? | |
|---|---|
| $PQ_1$ | Where was 82nd Airborne Division formed? |
| Answer$_1$ | The 82nd Airborne Division was formed in 1917 at **Camp Gordon, Ga**. |

Table 14: Incorrect Answer: Entailed Predictive Question

A similar phenomenon is seen when comparing entailment relationships between questions and answer passages. While both candidate answers are entailed by Q197.1 (*What animal was the first mammal successfully cloned from adult cells?*), only the first candidate answer passage is correct.

| **Q197.1** (*cloning of mammals*) What animal was the first mammal successfully cloned from adult cells? | |
|---|---|
| Correct | **Dolly** the sheep, the world's first clone of an adult mammal, has made history again to become a mother, its creator, Scotland's Roslin Institute, said Thursday. |
| Incorrect | The University of Hawaii scientists, reporting in Thursday's issue of the journal Nature, describe their work as "the first reproducible cloning of a **mammal** from adult cells" extending at least three generations. |

Table 15: Entailment Example

## 4. Processing "Other" Questions

In this section, we describe the approach used to provide answers to the "other" questions associated with each question series in the TREC 2006 Main QA Task. The architecture of the CHAUCER system for answering "other" questions is presented in Figure 2.

CHAUCER begins the processing of answering "other" questions by submitting a question series target to the same *Target Processing* module used in the factoid and list question-answering pipelines depicted in Figure 1. As with factoid and list questions, the process of answering "other" questions begins by categorizing targets using the *Target Type Detection* module. In addition, topic representations (including topic signatures and enhanced topic signatures) are also computed from the top 100 target-relevant documents retrieved from the AQUAINT corpus.

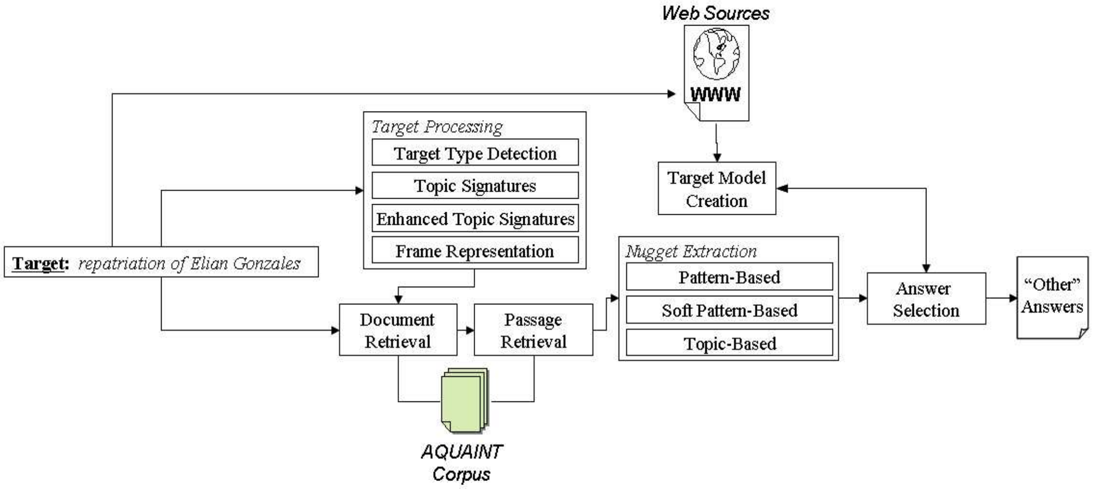

Figure 2: Architecture of the Chaucer "Other" Q/A System

| Weight | Name | Rule |
|---|---|---|
| 0.97 | Lifespan | TARGET (DATE - DATE) |
| 0.85 | Alias | TARGET, *(also)? known as* NP |
| 0.81 | Achievement | TARGET BE *(one of)? the* |
| 0.78 | Famous | TARGET, *who is famous for* |
| 0.65 | Membership | TARGET *of* NP |
| 0.52 | Appositive | TARGET, *(who BE)?* NP, |
| 0.46 | Copular | TARGET BE NP |
| 0.23 | Parenthetical | TARGET (NP) |
| 0.17 | Number | TARGET, NUMBER, |
| 0.09 | DoubleDash | TARGET – |

Table 16: Top PERSON Patterns

## Nugget Extraction

We used three different strategies to extract relevant nuggets from the documents retrieved from the AQUAINT corpus. First, we extracted nuggets for each target using libraries of high-precision patterns developed for each of our six different target types. Second, we used our own implementation of the algorithm for automatically generating *soft patterns* introduced in (Cui, Kan, & Chua 2004; 2005) in order to identify an additional set of patterns that could be used to extract relevant information for a particular target type. Third, we used information derived from the the two different *topic representations* generated during *Target Processing* in order to identify sets of sentences that contained information relevant to the topic denoted by the target itself.

**Pattern-Based Nugget Extraction** In Chaucer's *pattern-based* nugget extraction strategy, nuggets were extracted if the target appeared in any of a fixed set of extraction patterns that were defined for a particular target type. While extraction patterns based on the recognition of appositives, relative clauses, parentheticals, and copular constructions were used for each of the six target types, we developed specific patterns (when possible) for each individual target type. Table 16 provides examples of the types of patterns used to extract nuggets for targets classified as PERSON.

In a departure from previous pattern-based approaches to nugget extraction (Xu, Licuanan, & Weischedel 2003), we used a large corpus of definitions, descriptions, and biographies extracted from the Web in order to assign weights to each of the extraction patterns associated with each target type. Weights were computed for each individual extraction pattern associated with a target type based on the frequency that the pattern occurred in the corpus of descriptions assembled for each target type. Sentences were then extracted from the set of documents retrieved for the target based on a composite score equal to the sum of the weights of all of the patterns that were extracted from a sentence. Since this strategy necessarily favors precision over recall, all sentences that were assigned a non-zero weight were considered during *Answer Selection*.

**Soft Pattern-Based Nugget Extraction** In addition to hand-crafted extraction patterns, we also experimented with using the probabilistic soft matching techniques first described in (Cui, Kan, & Chua 2004) in order to identify additional patterns that could be used to extract nuggets for a particular target type. As with the soft pattern-based answer extraction strategy used in Chaucer's factoid Q/A pipeline, we followed (Cui, Kan, & Chua 2004) in developing a bigram soft pattern model in order to identify potential matches between a set of training sentences and each of the sentences extracted for a particular target. Training sentences were derived for each target type from two different sources: (1) the collection of "gold" nuggets identified for the TREC 2005 "other questions" and a collection of 5,000 biographies, descriptions, and encyclopedia articles that were downloaded from *wikipedia.org*, *s9.com*, and *biography.com*. We used the probablity that a passage was matched by an soft pattern in order to assign confidence weights to each of the sentences retrieved for a target; only the top 50 sentences were considered during *Answer Selection*.

**Topic-Based Nugget Extraction** Following work done by (Lacatusu *et al.* 2006) for question-focused summarization, we used weights associated with TS$_1$ terms and TS$_2$ relations to compute a composite *topic score* for each sentence in the set of documents retrieved for a target. Sentences were re-ranked based on their *topic score* before being submitted to the *Answer Selection* module. As with the soft pattern nugget extraction strategy, only the top 50 passages were considered during *Answer Selection*.

### Answer Selection

Recent work in summarization (Nenkova & Passonneau 2004) has benefited from the use of content models in selecting a set of relevant sentences for inclusion in a multi-document or question-focused summary. As with summaries, we believe that the set of answers returned in response to an "other" question can be modeled using techniques which are able to evaluate the relevance of each candidate passage (or "nugget") against some approximation of the content a user is seeking when asking this type of question.

In order to select amongst the set of candidate nuggets identified by our three nugget extraction strategies, we constructed a model of the idealized content of a set of answers to an "other" question based on passages extracted from a set of documents retrieved from number of authoritative sources found on the World Wide Web. (In our TREC 2006 work, we experimented with documents from three web sources: *wikipedia.org*, *s9.com*, and *biography.com*.) The top 10 documents from each site were retrieved with a simple web query, using only stemmed keywords extracted from the series target. Relevant passages were extracted from these downloaded documents by selecting passages that contained target keywords and topic signature ($TS_1$) terms. In order to acquire a set of passages that most closely resembled the the types of nuggets we hoped to select for our final answer submission, we discarded any sentence that contained fewer than 5 tokens or more than 150 tokens. After the model sentences were selected, we discarded remaining stop words, stemmed the remaining words, and built a term vector based on the *tf.idf* value computed for each word.

We then used a greedy search algorithm in order to identify the set of extracted nuggets that most closely resembles the content of the relevant passages extracted from the set of Web documents downloaded for that target. We defined an *answer submission* as any non-zero set of candidate nuggets identified from the set of candidate nuggets sent to the *Answer Selection* module. Each possible answer submission was then turned into a *tf.idf* term vector (using the same process as was used in processing passages included in the model). The resulting answer submission vector was then scored against the model using cosine similarity, defined as $Sim(\vec{x}_1, \vec{x}_2) = \frac{\vec{x}_1 \cdot \vec{x}_2}{|\vec{x}_1||\vec{x}_2|}$.

After creating an empty answer submission, the greedy search algorithm considers each candidate nugget is considered in turn; candidate nuggets is added to the answer set only if it would increase the answer submission's similarity when compared to the model. Search halts after a single pass through the nuggets. To prevent the inclusion of a large number of redundant nuggets, we use a heuristic to limit to size of the answer set. Each time a nugget was added to the answer set, we recorded the factor by which the similarity score was increased. Rather than searching until all of the candidate nuggets were considered, the search was terminated when the average of the last 10 score increases fell below a threshold.

| **Q199: Padre Pio** | |
|---|---|
| Soft Pattern | Padre Pio's followers still credit him with miracles, intercessions and supernatural powers |
| Pattern | Padre Pio was a Capuchin monk who skyrocketed to fame in 1918 when he began to bleed from his hands, feet and side, the first priest in centuries to show signs of the stigmata. |
| Topic | Padre Pio, who was born Francesco Forgione, the son of impoverished farm workers, was a sickly, deeply pious child. |

Table 17: "Other" Answers to Q199: Padre Pio

## 5. Evaluation Results

Table 18 presentsCHAUCER's performance on the TREC 2006 factoid Q/A task. We were encouraged by the overall performance of our system, as it suggests that current systems for textual entailment can be used effectively in order to select amongst the output of a multi-strategy approach to factoid Q/A.

| Judgment | Percent |
|---|---|
| Wrong | 37.5% |
| Unsupported | 2.7% |
| Inexact | 4.7% |
| Locally Correct | 1.2% |
| Globally Right | 53.8% |

Table 18: TREC 2006 Factoid Q/A Results

While we experimented with a single novel strategy for answering list questions, the bulk of our team's efforts were spent decidedly on factoid questions. Table 19 details the CHAUCER's performance on list questions.

| Metric | Score |
|---|---|
| Recall | 0.187 |
| Precision | 0.162 |
| F($\beta$=1) | 0.148 |

Table 19: TREC 2006 List Q/A Results

Finally, Table 20 shows our precision, recall, and F-Score for Other questions.

A breakdown of the number of questions lost at each stage of CHAUCER's factoid Q/A processing is provided in Table 21.

Despite using over 260 fine answer types, CHAUCER only assigns a spurious expected answer type to approximately 10% of the factoid questions. While we would predict that using a coarser answer type hierarchy would reduce some of this loss at both the question analysis and answer extraction stages, we would anticipate that reducing the number of

| Metric | Score |
|---|---|
| Recall | 0.143800 |
| Precision | 0.079760 |
| F($\beta$=3) | 0.108387 |

Table 20: TREC 2006 Other Q/A Results

| Component | Accuracy | Loss |
|---|---|---|
| Question Analysis | 89.6 | 10.4% |
| Document Retrieval | 86.1 | 3.4% |
| Answer Extraction | 75.9 | 10.2% |
| Answer Ranking | 53.8 | 22.1% |

Table 21: Component Analysis of Chaucer on TREC 2006 Factoid

entity types considered by Chaucer would make the tasks of *Answer Extraction* and *Answer Ranking* sufficiently more difficult. In addition, we believe that the relatively small number of questions lost at the level of *Document Retrieval* suggests that our approaches to keyword expansion and passage are well-suite for the factoid Q/A task.

## 6. Conclusions

In this paper, we described Chaucer, the new automatic question-answering system developed at LCC for the TREC 2006 QA evaluations. This system is notable in that it utilizes four new retrieval and answer detection techniques in order to better retrieve passages and extract exact answers from natural language texts. First, Chaucer features a novel query expansion process which leverages automatically-generated topic representations created specifically for each question series target to identify new keywords for each question. Second, higher precision passage retrieval was achieved for factoid and list questions through a two-phase approach to information retrieval which uses topic signatures to select better candidate passages for answer extraction. Performance of Chaucer's retrieval components was further enhanced by combining keyword queries with entity types selected from the set of over 300 types recognized by LCC's CiceroLite named entity recognition system. Third, Chaucer's system for answering ”other” questions exploits a new retrieval approach which exploits language models computed from collections of topical web documents in order to select relevant passages from 5 competing answer extraction modules. Finally, instead of adopting the abductive reasoning framework utilized by several of LCC's past TREC QA submissions, Chaucer exploits a mechanism for answer validation that incorporates forms of textual inference from a state-of-the-art textual entailment in order to retrieve and select answers to both factoid and list questions.

## Acknowledgments

The authors would like to thank Sanda Harabagiu, Tobias Jungen, and Cosmin Adrian Bejan for their assistance with this work. 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 U.S. Government.

## References

Chakrabarti, S.; Krishnan, V.; and Das, S. 2005. Enhanced answer type inference from questions using sequential models. In *Proceedings of EMNLP*.

Cui, H.; Kan, M.-Y.; and Chua, T.-S. 2004. Unsupervised Learning of Soft Patterns for Definitional Question Answering. In *Proceedings of the Thirteenth World Wide Web conference (WWW 2004)*, 90–99.

Cui, H.; Kan, M.-Y.; and Chua, T.-S. 2005. Generic Soft Pattern Models for Definitional Question Answering. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR 2005)*.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The pascal recognizing textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop*.

Harabagiu, S., and Hickl, A. 2006. Methods for Using Textual Entailment in Open-Domain Question Answering. In *Proceedings of COLING-ACL*.

Harabagiu, S.; Hickl, A.; Lehmann, J.; and 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.; Lacatusu, F.; and Hickl, A. 2006. Answering complex questions with random walk models. In *Proceedings of SIGIR-06*.

Hickl, A.; Williams, J.; Bensley, J.; Roberts, K.; Rink, B.; and Shi, Y. 2006. Recognizing Textual Entailment with LCC's Groundhog System. In *Proceedings of the Second PASCAL Challenges Workshop (to appear)*.

Lacatusu, F.; Hickl, A.; Roberts, K.; Shi, Y.; Bensley, J.; Rink, B.; Wang, P.; and Taylor, L. 2006. Lcc's gistexter at duc 2006: Multi-strategy multi-document summarization. In *Proceedings of DUC 2006*.

Lehmann, J.; Aarseth, P.; Nezda, L.; Deligonul, M.; and Hickl, A. 2005. TASER: A Temporal and Spatial Expression Recognition and Normalization System. In *Proceedings of the 2005 Automatic Content Extraction Conference*.

Li, X., and Roth, D. 2002. Learning question classifiers. In *Proc. the International Conference on Computational Linguistics (COLING)*.

Lin, C.-Y., and Hovy, E. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics*, 495–501. Morristown, NJ, USA: Association for Computational Linguistics.

Nenkova, A., and Passonneau, R. 2004. Evaluating content selection in summarization: the pyramid method. In *Proceedings of NAACL-HLT 2004*.

Ravichandran, D.; Hovy, E.; and Och, F. 2003. Statistical qa - classifier vs re-ranker: What's the difference? In *Proceedings of the ACL Workshop on Multilingual Summarization and Question Answering*.

Xu, J.; Licuanan, A.; and Weischedel, R. 2003. Trec 2003 qa at bbn: Answering definitional questions. In *Proceedings of TREC 2003*.