Employing Two Question Answering Systems in TREC-2005

Sanda Harabagiu, Dan Moldovan, Christine Clark, Mitchell Bowden, Andrew Hickl and Patrick Wang
Language Computer Corporation
Richardson, Texas 75080
sanda@languagecomputer.com

1 Introduction

In 2005, the TREC QA track had two separate tasks: the *main task* and the *relationship task*. To participate in TREC 2005 we employed two different QA systems. PowerAnswer-2 was used in the main task, whereas PALANTIR was used for the relationship questions. For the main task, new this year is the use of events as targets in addition to the nominal concepts used last year. Event targets ranged from a nominal event such as *"Preakness 1998"* to a description of an event as in *"Plane clips cable wires in Italian resort"*. There were 17 event targets total. Unlike nominal targets, which most often act as the topic of the subsequent questions, events provide a context for the questions. Therefore, targets representing events had questions that asked about participants in the event, about characteristics of the vent and furthermore, had temporal constraints. Also many questions referred to answers of previous questions. To complicate matters, several answers could be candidate for the anaphors used in follow-up questions, but salience mattered. This introduced new complexities for the coreference resolution. Consider the following example:

Target **136** - *Shiite*	
Q136.1	Who was the first Imam of the Shiite sect of Islam?
Q136.2	Where is his tomb?
Q136.3	What was this person's relationship to the Prophet Mohammad?
Q136.4	Who was the third Imam of Shiite Muslims?
Q136.5	When did he die?

In the above target set, questions Q136.2 and Q136.3 refer back to the answer of question Q136.1; question Q136.5 back-refers to the answer for question Q136.4. Because of this, if the QA system fails to locate the right answer for question Q136.1, the chances of getting the correct response for the following two questions are greatly decreased. Furthermore, this leads to the potential for ambiguity when the system attempts to answer question Q136.5 if the answer to the most recent question is either not found or incorrect. Compare with the situation of target 27 in TREC 2004.

Target **27** - *Jennifer Capriati*	
Q27.2	Who is her coach?
Q27.3	Where does she live?

In question Q27.3, *"she"* refers to the target, Jennifer Capriati, which is also the antecedent for the pronoun *"her"* in Q27.2. But, if the answers are also included in the candidate set for the pronouns, when processing question Q27.3, two different entities become candidates: both Jennifer Capriati and her coach - the answer to Q.27.2.

Reference resolution is not the only linguistic phenomenon that had to be tackled in TREC 2005. The main task for Q/A required various forms of inference. We describe them when we report on the PowerAnswer-2 Q/A system at TREC 2005. In the paper we also describe the approach we used to answer complex questions for the TREC 2005 Relationship Task. For that task we used the PALANTIR Q/A system

The rest of the paper is organized as follows. In Section 2 we discuss the architecture of PowerAnswer-2. In Section 3 we detail the method of exploiting redundancy on the Web. Section 4 presents the role of the logical prover in the results obtained this year. Section 5 discusses the processing of questions that have temporal constraints. Section 6 reports on the processing of "other" questions. Section 7 described the processing of relationship questions with PALANTIR, which is described in Section 8. Section 9 lists and discusses the results.

2 The PowerAnswer-2 Q/A System

As illustrated in Figure 1, The PowerAnswer-2 Q/A System has three different modules: the question processing (QP) module, the passage retrieval (PR) module and

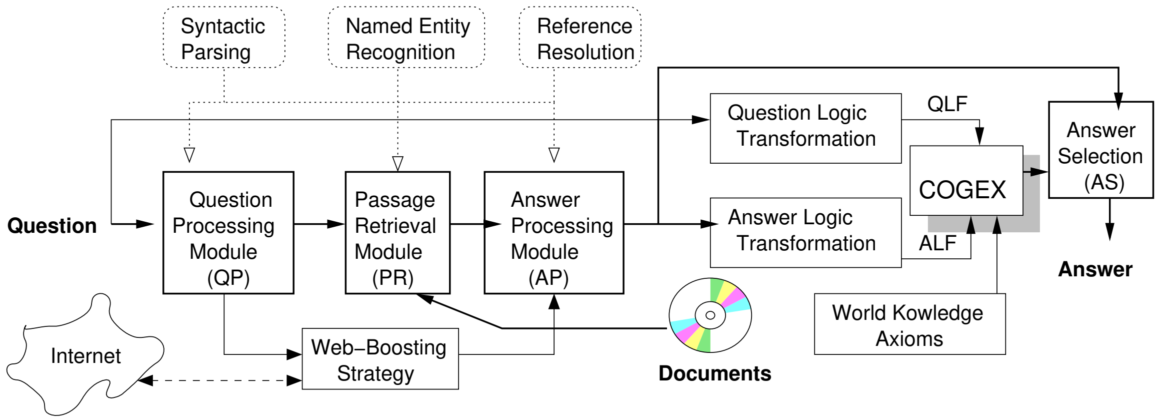

Figure 1: Architecture of PowerAnswer-2

the answer processing (AP) module. The role of the QP module is to determine (1) the expected answer type and (2) to select the keywords used in retrieving relevant passages. The PR module ranks passages that are retrieved, while the AP determines the extraction of the candidate answers. All modules have access to a syntactic parser, a named entity recognizer and a reference resolution system. To improve the statistical methods used for answer selection, we took advantage of redundancy in large corpora, specifically in this case, the Internet. As the size of a document collection grows, a question answering system is more likely to pinpoint a candidate answer that closely resembles the surface structure of the question. Such an intuition has been verified by (Breck, et al., 2001) and empirically re-enforced by several QA systems. The Web-Boosting Strategy module uses features which are described in Section 3. These features have the role of correcting the errors in answer processing that are produced by the selection of keywords, by syntactic and semantic processing and by the absence of pragmatic information. The ultimate decision for selecting answers is based on logical proofs.

Before selecting the answer, an abductive proof of its correctness is performed by using the COGEX logical prover. Details of the operation of COGEX were presented in (Moldovan et al., 2003) and (Moldovan et al. 2005). To perform the abductive inference, the question and each candidate answer need to be transformed in logical representations, which rely on the syntactic, semantic and reference resolution information already used in the QA modules. The process of translating a question or an answer in logical transformations is illustrated in Figure 2. As illustrated in Figure 1, the inputs to COGEX consist of the question logical form (QLF), the answer logical form (ALF) and a set of axioms modeling world knowledge.

The QLF and the ALF are produced in a three-layered

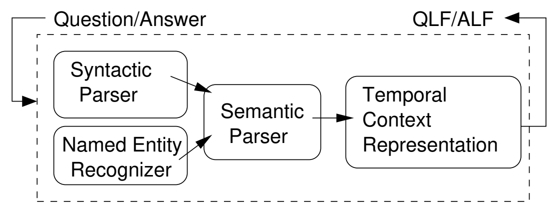

Figure 2: Logical Form Transformations

first order logic representation as illustrated in Figure 2. The first layer relies on the syntactic parse and the named entity recognition, available also to the QA modules, as was illustrated in Figure 1. The second layer relies on the recognition of semantic relations processed by the semantic parser reported in (Bixler et al. 2005). The third layer represents temporal contextual information that is produced by temporal ordering of events, anchoring events in time intervals and normalizing temporal expressions. The temporal context representation was introduced in (Moldovan et al. 2005).

The QLF and the ALF are not the only inputs to COGEX. As illustrated in Figure 1 a set of world axioms also participate in the proof of the answer. There are five sources of world knowledge axioms: (1) axioms derived from the eXtended WordNet, available from *http://xwn.hlt.utdallas.edu*; (2) ontological axioms generated by the JAGUAR knowledge acquisition tool described in (Bixler et al. 2005); (3) Linguistic Axioms handcrafted to account for several linguistic phenomena, e.g. possessives, appositions, nominal coreference; (4) a semantic calculus described in (Moldovan et al. 2005); and (5) temporal reasoning axioms available from the SUMO knowledge base. The SUMO knowledge base was described in (Niles and Pease 2001).

The proof of the abduction performed by COGEX scores each candidate answer, thus allowing the answer selection (AS) module to chose the exact answers when high confidence is given to the abductive proof. When the abductions fail or are obtained with low confidence, the AS module selects the highest-ranking answer provided by the AP module.

In 2005, we focused on advanced textual inference techniques that involve temporal inference. Furthermore, we continued development on automatic axiom generation for linguistic entailment as motivated by the RTE Pascal Challenge (Fowler, et al., 2005). To address the "OTHER" question type, we developed new methods of extracting novel, interesting nuggets through the use of the Suggested Upper Ontology (Niles and Pease 2001), templates and entity associations as well as patterns.

3 Exploiting Answer Redundancy on the Web

The "web-boosting" features are based on a web strategy that utilizes general linguistic patterns in order to construct a series of search engine queries. The answers from the web documents are extracted by considering their redundancy. Furthermore, the most redundant answer is added to the keyword features used in the AP module to extract answers. For example, for the question **Q124.5**, the illustrated web query allows to find redundant answers which leads to another ranking of the most relevant answers produced by the AP module.

Q124.5: *(Rocky Marciano) How many fights did he win?*	
Original answer	"Holmes won his first 48 fights - one short of Rocky Marciano's record."
Web query	"Rocky Marciano won * fights"
Web answer	49
Final answer	"Danish promoter Mogens Palle is touting hapless Brian Nielsen for tying Rocky Marciano's record at 49-0"

PowerAnswer-2 processes the document hits produced by the set of queries, extracts the exact answers, and computes the most probable answer from this set using tiling (Lin, 2002), and the answer confidence assigned by PowerAnswer-2. A boost is then given to answers returned from the TREC collection that best match the answers returned by the web strategy, a larger boost given for higher frequency.

Question Answering systems that utilize syntactic and nominal coreference features are likely to select answers with high question word overlap. By voting with high precision results from the web can prevent the question answering system from extracting syntactically and lexically similar, but incorrect answers, as is the case for question **Q111.3**:

From the above examples, it is clear that the web is a powerful external resource for any open domain question answering system. LCC's TREC 2005 results showed that "web-boosting" provided an added value of 69/331% to the final factoid score.

Q111.3: *(AMWAY) Who is the president of the company?*	
Original ans.	"...Eva Cheng, president of Amway Company China Commodity Ltd."
Web query	"* is the president of AMWAY"
Web ans.	Dick DeVos, Richard DeVos, DeVos
Final ans.	"...said Dick DeVos, president of Amway Corporation"

4 The Role of the Logic Prover

COGEX performs a proof of the question over the candidate passages and scores them according to their syntactic and semantic similarity to the question. In this way, COGEX, operating on world knowledge axioms reranks, extracts, and scores the top N candidates. For example, to process question **106.2**, COGEX utilizes derivational morphology available from WordNet (Fellbaum 1998) to automatically generate an axiom linking the verb *"lose"* with the adjective *"losing"*. This axiom provides the necessary linguistic knowledge for COGEX to accurately verify that *"the losing team"* is entailed by *"Padres"*, and not *"Yankees"*, the statistically extracted answer.

Q106.2 *(1998 Baseball World Series) What is the name of the losing team?*	
Candidate	"...the San Diego Padres team that lost to the Yankees in the 1998 World Series."
Original exact	Yankees
Axiom	derivational morphology rule triggers: $lose\_VB(e1,x1,x2) \rightarrow losing\_JJ(x1)$
Final exact	San Diego Padres

Q91.4 *(Cliffs Notes) What company now owns Cliffs Notes?*	
Candidate	"Cliffs Notes was bought by IDG Books Worldwide"
Axiom	$buy\_VB(e1,x1,x2) \rightarrow own\_VB(e2,x1,x2)$
Final exact	IDG Books Worldwide

Lexical chains (Moldovan and Novischi 2002) continued to be a important resource for COGEX. Lexical chains are derived from the links in WordNet and provide a mechanism for measuring the semantic similarity between keywords in the question and keywords in the answer. The similarity measure associated with each lexical chain is used by COGEX when it scores candidate answers. In the example for **Q91.4**, lexical chains generate the necessary link between *"buy"* and *"own"*. This link is transformed into an axiom and used by COGEX to extract *"IDG Books Worldwide"* as the correct answer to the question. For the TREC 2005 factoid questions, COGEX generated an enhancement of 12.4% to the final factoid score.

5 Temporally Constrained Questions

With the introduction of events as targets in TREC 2005, a large set of questions required the resolution of temporal constraints in candidate answers. Of the 455 questions from the list and factoid track 16% contained temporal references. To meet this anticipated need, LCC's temporal context reasoning system, described in (Moldovan et al. 2005) was incorporated into PowerAnswer-2.

The approach taken by LCC for temporal reasoning in QA can be summarized as:

1. Detect absolute dates in the question and prefers passages that match the detected temporal constraints of the question.
2. Discover events related by temporal signals in the question and candidate answers.
3. Perform temporal unification between the question and the candidate answers and boost answers that match the temporal constraints of the question

Additionally, passage retrieval required a temporal index of all the absolute dates detected in the document collection. The temporal index operates on absolute dates, relative dates as well as date ranges. For example, in the question **106.5**, the expression *"1998"* is marked as the required temporal context for the question, and the following query is executed:

Q106.5: *What is the name of the winning manager of 1998 Baseball World Series?*	
Query	winning AND manager AND baseball AND world AND series AND date:[19980101 TO 19981231]

To discover the answer, events anchored by temporal expressions need to be processed. Additionally, we discover temporally related events both in questions and candidate answers. Events linked by temporal constraints are represented as a triple *(S,E1,E2)* which consists of a temporal signal *S*, e.g. *"during"*, *"after"*, and its corresponding event arguments *E1* and *E2*. To produce such triplets, we also had to perform: (1) the disambiguation of signal words and (2) the attachment of events to signal words. When no temporal relations were detected in the candidate passage, the document time-stamp served as the default context based.

The temporal reasoning module integrated into PowerAnswer-2 employs two context unification modules. A full first-order logic reasoning engine was selected when the question contained a temporal event with no absolute date reference, such as "How old was Bing Crosby when he died?", and a a light-weight special purpose reasoner for questions that specified an absolute date, such as, "Who was president of DePauw in 1999?".

The general purpose temporal reasoner is incorporated in COGEX. The events and temporal relations from the question and answer are converted into a Suggested Upper Merged Ontology (SUMO) (Niles and Pease 2001) logic representation and COGEX is used to perform context resolution between the question and answer texts. This approach works for unifying temporal relations based on signal words as well as for ordering questions (first, second, third). If the temporal constraints of the question can not be unified with those in a candidate answer, the answer is discarded from the answer list. For example, question **Q137.4** has a containment relation between *"bombardment"* and *"the 1950's"*. Further *"1950's"* is normalized to the range [19500101, 19501231]. The range contains the time-stamp of expression *"August, 1958"*, which constrains the event *"bombardment"* from the correct candidate answer.

Q137.4: *(Kinmen Island) In the 1950's, who regularly bombarded Kinmen?*	
Original ans.	"(Kinmen Island, Taiwan) - With just two days left before Taiwain chooses a new president, China's military menace ..."
Final ans.	"...Kinmen, also known as Quemoy, was the scene of frequent artillery attacks by Chinese gunners located on the China coast less than 3 kilometers (1.8 miles) away. An intense bombardment beginning August 23, 1958, lasted for 44 days..."

Although 16% of the list and factoid questions in the TREC 2005 test set specified temporal constraints, the temporal reasoner only added a 2% value to the overall system performance. Due to the high degree of keyword overlap in the questions and the candidate answers, PowerAnswer-2 without the temporal reasoner often ranked the correct answer in position one. The temporal reasoner only re-enforced the selected answers.

6 The Processing of 'Other' Questions

The inherent challenge of "other" questions in the TREC QA Track is the filtering and selection of interesting and novel nuggets from a large corpus. The passage recall for information about a target is typically overwhelming, and pruning these passages to pick the best nuggets is time consuming and difficult. This year PowerAnswer-2 experimented with two new techniques for nugget selection that complement the existing "definition" pattern-based method. The three methods are:

[1] *Nuggets discovered by question patterns.*
Returning vital nuggets that are not captured by a definition pattern requires a strategy that seeks characteristics of the target. Since targets can be classified in several target classes, it is natural to generate questions that seek the characteristic of each class. We replied on 33 target classes (e.g. animal, actor, musician, literature) resulting from the analysis previous TREC question sets. To gen-

erate the target classes, we used a Naïve Bayes Classifier that employs features such as WordNet synsets, stemmed surface forms of the tokens, and named entity classes. For example, for the target *Bing Crosby*, the system classified the target as MUSICIAN_PERSON, resulting in the selection of the following set of questions:

Example: **Bing Crosby**

Target Class: **musician_person**

```
What is the name of the band of X?
What record company is X with?
Where was X born?
What kind of singer is X?
When was X born?
Where was X born?
```

The nugget discovered by such questions is:
Tacoma-born Bing Crosby is officially named No. 1 box-office star by Quigley Poll.

[2] Nuggets discovered by entity classes.
Relevant nuggets of information can be characterized by associations with other named entities in the collection. For this reason we used the semantic classifications generated by our named entity recognizer to discover such relations when looking for relevant passages.

Example: **Akira Kurosawa**

```
_human AND Akira Kurosawa
_date AND Akira Kurosawa
_location AND Akira Kurosawa
_quantity AND Akira Kurosawa
_money AND Akira Kurosawa
_quantity AND Akira Kurosawa
```

The nugget discovered by such questions is:
Akira Kurosawa, renowned Japanese filmmaker, dies at **88**.

[3] Nuggets discovered by patterns.
The traditional method employed by PowerAnswer to extract nuggets is to execute a definition pattern matching module. A list of over 150 positive and negative precomputed patterns is loaded into memory. The target is inserted into these patterns and the resulting query is submitted to an index including stopwords and punctuation. These are high-precision patterns that indicate information of a definitional nature. For example:

Example: **Russian submarine Kursk**

Pattern: 3 1 `ANSWER-target , which`

Answer: The joint British and Norwegian began on Sunday attempt to rescue any survivors on board the sunken **Russian submarine Kursk, which** is lying on the sea bed in the Barents Sea.

7 Answering Relationship Questions

With the accuracy of today's best factoid Q/A systems nearing (and in some cases, exceeding) the 70% F threshold, work in automatic Q/A has begun to focus on the answering of complex questions. Although researchers have not yet agreed upon a standard definition of exactly what constitutes a complex (or "relationship") question, for the purposes of this paper, we claim that a relationship question can be defined as a natural language question whose information need cannot be associated with a single semantic answer type from an idealized ontology of semantic entity or event types.

Unlike factoid questions, which presuppose that a single correct answer can be found that completely satisfies all of the information requirements of the question, relationship questions often seek multiple and different types of information and do not presuppose that one single answer could meet all of its information needs simultaneously. For example, with a factoid question like *Who are the members of the Rat Pack?*, we assume that a user is looking for a list of names (specifically, person names) who were a part of the Rat Pack. In this case, users do not expect systems to return additional related information (such as which member of the Rat Pack was considered its leader), as the answer itself is sufficient to meet the information need of the question. In fact, returning more than the requested information is undesirable, as the system would be more informative than necessary and violate the Gricean Maxim of Quantity. In contrast, with a relationship question like *What impact did the Rat Pack have on the rise of the Las Vegas tourism industry?*, the wider focus of this question indicates that users may not have a clearly defined (or pragmatically restrictive) information need, and therefore would be amenable to receiving additional supporting information that was relevant to their overall goal.

In this section, we describe the approach we used to answer complex questions for the TREC 2005 Relationship Task. Since we believe that answering relationship questions depends on sophisticated representation of the information need of these complex questions, we have implemented an approach which employs three question representation strategies to find answers: (1) an approach based on keyword selection, (2) an approach based on topic representation, and (3) an approach based on automatic lexicon generation. We show that by combining results from each of these components, we can achieve high levels of performance with almost no manual processing.

8 The PALANTIR Complex Question-Answering System

We chose to extend LCC's PALANTIR Q/A system for the TREC 2005 Relationship Task. Developed for interactive question-answering applications like LCC's FERRET Dialog System, PALANTIR incorporates techniques for keyword selection, passage retrieval, and answer ranking that were developed to address the types of complex questions that are typically asked by users in an interactive Q/A dialog. Furthermore, since complex questions lack a single identifiable semantic answer type, we felt that PALANTIR's multiple answer finding strategies could be leveraged to find a wider range of answers than the question processing and answer justification modules implemented in POWERANSWER-2.

Before relationship questions for this task are submitted to PALANTIR, questions are first manually processed to resolve pronouns and other referring expressions and to remove instances of ellipsis. Questions are then sent to an automatic question decomposition module which uses a set of heuristics to break complex questions into a set of syntactically simpler questions. Keywords are then extracted by a keyword selection module which detects collocations and ranks keywords with an approximation of their importance. Documents are then retrieved and submitted to a trio of answer finding strategies. Candidate answers from each strategy are then merged and ranked; the 7 top-ranked passages are returned as answers. A block diagram for our system is presented in Figure 3.

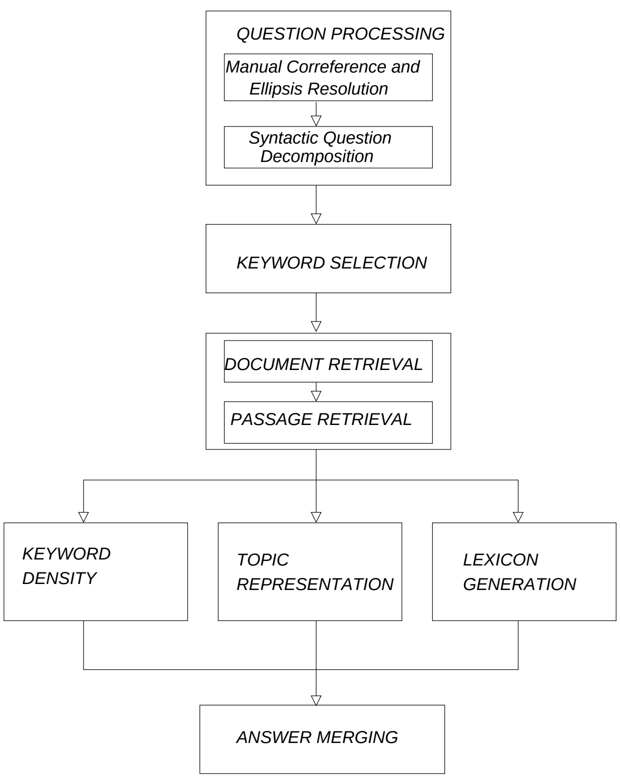

Figure 3: Architecture of LCC's PALANTIR Complex Q/A System

Question Processing
As in our submission to the 2004 AQUAINT Relationship Pilot, coreference and ellipsis resolution were done manually for each question. Relationship questions were automatically syntactically decomposed using heuristics that split conjoined NPs and lists into separate questions and extracted embedded questions. For example, a complex question context like *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.* was automatically split into the three questions in Figure 4. Semantic question decomposition of the type described in (Hickl et al., 2004) was not performed for any questions in this year's task.

Complex Question: 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.

Syntactic Decomposition

Q_1: What possible relationship is there between the Cuban and Congolese governments? Q_2: What attempts by the *Cuban and Congolese governments* to form trade alliances? Q_3: What attempts by the *Cuban and Congolese governments* to form military alliances?

Figure 4: Question Decomposition

Keyword Selection

The following techniques were employed for keyword selection:
<u>Collocation Detection.</u> When doing keyword selection for complex questions, precise collocation detection is necessary. For example, in order to retrieve documents concerning the *Organization of African States*, systems must retrieve documents containing the collocation as a whole, and not just its individual tokens.
<u>Keyword Ranking.</u> PALANTIR assigns a weight to each keyword extracted from a complex question as part of its document retrieval strategy. Based on an approach first outlined in (Moldovan et al., 2004), weights are assigned heuristically based on a rough approximation of the keyword's overall importance to a query. For example, in the current version of PALANTIR, the highest weights were assigned to proper names (NNPs), followed by comparative and superlative adjectives, ordinal numbers, and quoted text.
<u>Keyword Expansion.</u> Synonyms and alternate forms for each keyword were added from a database of similar terms developed for past TREC Q/A evaluations.

Document Retrieval
As with the TREC 2004 version of PALANTIR (Moldovan et al., 2004), we used an preprocessed and indexed version of the TREC Q/A corpus that had been previously annotated with part-of-speech information, syntactic parse information, and named entity information taken from LCC's CICEROLITE named entity recognition soft-

ware. After keyword selection is complete, PALANTIR's document retrieval system uses a state machine-based approach that iteratively drops keywords of lesser importance in order to find the most relevant documents. Once a set of documents has been retrieved, documents are segmented into sentences and text passages are extracted that contain clusters of keywords. Passage length is determined dynamically based on the number of keywords found within a set of sentences; the average passage length is three sentences.

Once a set of passages have been retrieved, PALANTIR employs three different strategies to find answers. The first strategy, known as *Keyword Density*, ranks candidate answers using a score based on the number, weight, and relative position of the question words (and alternations) found in the passages. (A version of this approach was the only strategy used in our submission in the 2004 AQUAINT Relationship Pilot.) The second strategy uses sophisticated *Topic Representations* to select candidate answers that contain specific topic-relevant words and relations derived from the set of documents retrieved during document retrieval. Finally, PALANTIR uses an approach based on *Lexicon Generation* to automatically expand keywords that may denote a set of terms (e.g. *South American countries*, *Latin America*, *high-tech weaponry*) to their full membership.

Keyword Density Strategy

PALANTIR's Keyword Density (Moldovan et al., 2004) heuristic assigns a score to each passage returned during passage retrieval. Top passages are run through a series of redundancy filters to remove duplicate (or overlapping) answers. Two sets of features are then used to rank the remaining passages:

Surface Ranking. 50 of the more than 80 surface features used in PALANTIR for factoid Q/A were chosen to rank answers to relationship questions. Features were selected based on their compatibility with passage-length answers as well as for the overall speed and performance of the system.

Relation Ranking. In addition, candidate answer passages were ranked based on a series of relational features. This relation-based ranking considers a set of features based on a dependency parse of both the question and each of the sentences in the candidate answer passage.

Topic Representation Strategy

In this approach, we used two different topic representation strategies to rank candidate answer passages. Similar to the approach employed by LCC's LITE-GISTEXTER question-directed summarization system (Lacatusu et al., 2004) for the DUC 2005 summarization evaluations, this approach assumes that answers to complex relationship questions can be identified by selecting those passages that contain a preponderance of topic terms and topic relations.

PALANTIR's *Topic Representation* strategy uses two different topic representations to identify the set of relevant sentences that should be included in a summary: topic signatures (TS_1) and enhanced topic signatures (TS_2). Originally developed for single-document summarization (Lin and Hovy, 2000), the topic signature algorithm computes a weight for each term in a document cluster based on its relative frequency in a relevant set of documents. (Complete details of our topic signature implementation are provided in (Lacatusu et al., 2004).) In DUC 2005, we also experimented with the enhanced topic signatures first described by (Harabagiu, 2004). Unlike Lin and Hovy's topic signatures (which are limited to sets of individual terms), Harabagiu's enhanced topic signatures can be used to discover a set of relevant relations that exist between topic signature terms and to provide each relation with a weight depending on its overall significance to the topic being modeled. With enhanced topic signatures, topics are represented as the set of relevant relations that exist between topic signature terms: $TS_2 = (topic, < (r_1, w_1)...(r_m, w_m) >)$, where r_i is a binary relation between two topic concepts. Two different forms of topic relations are considered by this approach: (1) syntax-based relations that exist between the verbs and their arguments; and (2) context-based relations (C-relations) that exist between entities. We calculate enhanced topic signatures in the manner described in (Harabagiu, 2004). Examples of TS_1 and TS_2 for Question 9 (*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.*) are presented in Table 1.

Candidate answer passages are assigned a composite score equal to the sum of the weights of all of the topic terms and/or relations they contained; answers are ranked according to this score.

Lexicon Generation Strategy

Although relationship questions do not feature semantic answer types, they do often include non-specific entities which denote a set of individuals. For example, in question topic 23, *The analyst is interested in knowing which South American countries are involved in nuclear proliferation.*, the NP *South American countries* denotes the set of countries found on the continent of South America. Although named entity recognition systems (NER) such as LCC's CICEROLITE have been used successfully identify sets of candidate answers for more than 150 different kinds of answer types, no current NER system includes enough semantic types to identify the extension of every possible set-denoting entity. With relationship questions, knowing the full extension for a particular entity – like *South American countries* – can pro-

Topic Signature

congo(pn)	1961	congolese(pn)	1374
cuba(pn)	1322	kabila(pn)	1278
cuban(a)	895	rebel(n)	846
rwandan(pn)	819	rwanda(pn)	795
castro(pn)	440	kinshasa(pn)	421
uganda(pn)	371	ugandan(pn)	338
troop(n)	202	embargo(n)	200
zimbabwe(pn)	175	laurent_kabila(pn)	171
eastern_congo(pn)	156	angola(pn)	154
president_laurent_kabila(pn)	153	ally(n)	153
cuban(n)	125	cuban(pn)	116
first(a)	113	tutsi(pn)	112
havana(pn)	109	rebellion(n)	106
namibia(pn)	100	war(n)	99
hutu(pn)	90	island(n)	88
mobutu_sese_seko(pn)	87	cease(v)	85
fidel_castro(pn)	75	dictator(n)	73
exile(n)	64	rebel(a)	59
soldier(n)	59	border(n)	57
accuse(v)	56	fight(v)	54
zambia(pn)	53	back(v)	51

Enhanced Topic Signature

Congolese - rebel	103	NE:LOCATION - policy	83
Congolese - NE:PERSON	65	Congolese - NE:OTHER	55
dictator - Mobutu Sese Seko	55	Rwandan - troop	55
NE:LOCATION - back	52	dictator - NE:OTHER	47
Congolese - government	45	NE:OTHER - rebel	43
rebel - group	39	back - rebel	38
ally - NE:LOCATION	38	food - medicine	35
dia - NE:PERSON	35	Hutu - militia	35
Cuban - NE:PERSON	34	Rwandan - soldier	33
food - sale	33	rebel - try	33
NE:PERSON - government	33	cash - transfer	31
PPE:DATE_TIME - genocide	31	NE:LOCATION - war	30
charter - flight	30	back - NE:PERSON	28
Rwandan - government	28	NE:LOCATION - most	28
Lead - move	28	send - troop	28
Cold - War	28	Hutu - rebel	28
come - power	28	PPE:NUMBER - troop	25

Table 1: Signatures for question Q_9

vide an invaluable source of keywords that can be used to find additional relevant information. In order to answer relationship questions like the above, we have implemented a system based on a weakly-supervised learning approach described in (Thelen and Riloff, 2002) that can identify a set of entities semantically related to a set of automatically-generated seed tokens. Crucial to this approach is the generation of a large database of syntactic frames which are used to approximate the different types of semantic relationships that can exist between entities. We populated this database with a variety of extraction patterns first written for LCC's CICERO information extraction software.

In this strategy, decomposed questions are sent to an *Answer Type Term Detection* module used in PALANTIR's factoid Q/A system. NPs that are detected as potential answer type terms are sent to a Seed Generation module that searches in a frame database for potential matches. If more than 5 matches are found in the database, the NP is sent to a Lexicon Generation module which uses the (Thelen and Riloff, 2002) method to generate potential expansions. For example, given the NP *South American countries*, this approach returns two additional South American countries: *Brazil* and *Argentina*. Once lexicon generation is complete, each original question (not including the generated terms) is sent to PALANTIR's Keyword Selection module and candidate answers are generated as with the Keyword Density strategy. Terms identified by Lexicon Generation are used to filter candidate answers: only answers containing the generated terms are returned as final answers to relationship questions.

In future work, we plan to experiment with different applications of this technique in order to best determine how to use terms identified by the Lexicon Generation module.

Answer Merging

Answers from each answer-finding strategy were combined using an Answer Merging module. The Answer Merging module first combined the top answers from each strategy for a single decomposed question. This merging is done by heuristically normalizing the scores that each strategy assigns to every answer. Duplicate and overlapping answers are filtered again using the same filters employed by the Keyword Density strategy. Once the answers for every decomposed question are ranked, at least the top three answers from each decomposed question are presented as the answers to the complex question. non-redundant, non-overlapping candidate answers as answers to each relationship question. strategy's answers were separately top answers from each strategy are merged answers to every decomposed question are then to the complex question.

9 Results

The following table illustrates the final results of Language Computer's efforts in the 2005 TREC main QA track obtained by PowerAnswer-2.

	PowerAnswer-2
Factoid	0.713
List	0.468
Other	0.228
Overall	0.534

Table 2: Results in the main task.

This year, our two submissions for the Relationship Task differed in terms of the total number of keywords considered in document retrieval. In Run 1, only a limited (high-confidence) set of alternations were used; Run 2 used a much larger set of alternations for each question. Table 9 compares performance of those two runs.

For both runs, the F-Measure was above the median score for all groups: Run 1 scored 0.204, while Run 2 received a 0.179 score. Although we kept our manual processing of question to a minimum – using only automatic methods for keyword expansion and syntactic question decomposition – we did perform manual resolution of coreference and ellipsis for 14 of the 25 scenarios. (Both of our runs were considered to be in the set of "manually processed" systems by the NIST assessors.) When only "manually processed" scores were considered, only Run 1 was above the median; Run 2 has the median score among the 9 groups. Since we believed that including a greater number of keyword alternations would enable us to find a wider range of answers, we were surprised by the fact that Run 2 received a somewhat lower score than Run 1. However, we doubt if this result is truly significant: Run 2 featured only 1.1 more keywords than Run 1 on average – a difference which resulted in the loss of only 5 total nuggets overall.

In addition to the overall F-measure, we calculated two additional recall measures to help interpret our results: Document Retrieval Recall and Passage Retrieval Recall. We define Document Retrieval Recall as the percentage of vital nuggets returned in the documents under consideration; similarly, Passage Retrieval Recall is defined as the percentage of vital nuggets returned in the passages under consideration. (When relationship questions were decomposed into sets of subquestions, we computed a single recall (Document or Passage) measure which combines results from each of the subquestions.)

Run	Run 1	Run 2
Document Retrieval Recall	0.533	0.504
Passage Retrieval Recall	0.504	0.452
Answer Recall	0.306	0.244
Answer Precision	0.079	0.079
Answer F-measure	**0.204**	**0.179**

Table 3: The two submitted results of PALENTIR.

Document Retrieval Recall for both runs was simliar, just above 50%. This is substantially lower than typical factoid Document Retrieval Recall, which for the factoid version of PALANTIR on TREC 2004 was above 90%. We believe this is for two reasons. First, relationship questions contain many more keywords on average than factoid questions. In order to be successful in answering relationship questions, systems must identify exactly which keywords should be submitted to a document retrieval query; inclusion of less relevant keywords could result in the retrieval of spurious documents. Second, without an overt semantic answer type, relationship Q/A systems cannot make use of the valuable semantic information provided by available named entity recognition systems. Passage Retrieval Recall for both runs was slightly lower than their respective Document Retrieval Recall numbers. This is similar to factoid Passage Retrieval Recall, which is usually around 10% lower than Document Retrieval Recall. We believe our substantially lower Answer Recall is due to our answer ranking algorithm. In future work, we will experiment with novel answer ranking techniques that will enable us to keep more of the answers we retrieve.

Strategies	Total	Vital	Okay	Rec	Prec	F
Density	154	23	21	0.259	0.065	0.167
Lexicon	3	1	1	0.020	0.005	0.015
Topic	29	3	2	0.105	0.021	0.072
Combined	**186**	**27**	**24**	**0.306**	**0.079**	**0.204**

Table 4: Strategy comparison (Run 1).

Table 9 presents a comparison of the three answer-finding strategies in PALANTIR. We found that the traditional density strategy – first introduced in the AQUAINT 2004 Relationship Pilot – remains dominant, providing 83% of our total answers, 85% of vital nuggets, and 87% of okay nuggets. Although the Topic Representation- and Lexicon Generation-based strategies were used much less frequently, they did contribute approximately 14% of the total nuggets (vital and okay) that system returned. We are encouraged by the fact that our overall F-measure – combining results from all three strategies – is higher than the performance of any individual strategy. Although the sample size is rather small, these results suggest that using a hybrid approach that combines results from multiple answer-finding strategies may be effective for these types of relationship questions.

Acknowledgements

We would like to thank the researchers and engineers from Language Computer Corporation for their valuable contributions to this work. This work, as our broader research in QA is supported by the Advanced Research Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program.

References

D. Bixler, D. Moldovan, A. Fowler. 2005. Using Knowledge Extraction and Maintenance Techniques to Enhance Analytical Performance. *Proceedings of the 2005 International Conference on Intelligence Analysis*, Washington D.C.

E. Breck, M. Light, G. Mann, E. Riloff, B. Brown, P. Anand, M. Rooth and M. Thelen. 2001. Looking under the hood: Tools for diagnosing your question answering engine. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001) Workshop on Open-Domain Question Answering.*

C. Fellbaum. 1998. WordNet: An Electronic Lexical Database. *MIT Press.*

A. Fowler, B. Hauser, D. Hodges, I. Niles, A. Novischi and J. Stephan. 2005. Applying COGEX to Recognize Textual Entailment In *Prooceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, 2005.*

S. Harabagiu. 2004. Incremental topic representations. In *Proceedings of the 20th COLING Conference*.

A. Hickl, J. Lehmann, J. Williams, and S. Harabagiu. 2004. Experiments with interactive question-answering in complex scenarios. In *Proceedings of the Workshop on the Pragmatics of Question Answering at HLT-NAACL 2004*.

F. Lacatusu, A. Hickl, S. Harabagiu, and L. Nezda. 2004. Lite-gistexter at DUC 2004. In *Proceedings of DUC 2004*.

C.Y. Lin and E. Hovy. 2000. The automated acquisition of topoic signatures for text summarization. In *Proceedings of the 18th COLING Conference*.

J. Lin. 2002. The Web as a Resource for Question Answering: Perspectives and Challenges In *Proceedings of the third International Conference on Language Resources and Evaluation (LREC 2002)*

D. Moldovan, C. Clark, S. Harabagiu and S. Maiorano. 2003. Cogex: A Logic Prover for Question Answering. In *Proceedings of the Human Language Technology and North American Chapter of the Association for Computational Linguistics Conference (HLT-2003)*, 87-93.

D. Moldovan, C. Clark, S. Harabagiu. 2005. Temporal Context Representation and Reasoning To appear in the *Proceedings of the Nineteenth Internation Joint Conference on Aritificial Intelligence*

D. Moldovan, S. Harabagiu, C. Clark and M. Bowden.

PowerAnswer-2: Experiments and Analysis over TREC 2004 2004.

Moldovan, D.; Novischi, A. 2002 Lexical Chains for Question Answering. In *Proceedings of COLING 2002*, pp.674-680.

I. Niles and A. Pease. 2001. *Towards a Standard Upper Ontology*. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems. Ogunquit, Maine, October 2001

M. Thelen and E. Riloff. 2002. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*.