A Temporally-Enhanced PowerAnswer in TREC 2006

Dan Moldovan, Mitchell Bowden and Marta Tatu
Language Computer Corporation
Richardson, Texas 75080
moldovan@languagecomputer.com

Abstract

This paper reports on Language Computer Corporation's participation in the Question Answering track at TREC 2006. An overview of the PowerAnswer 3 question answering system and a description of new features added to meet the challenges of this year's evaluation are provided. Emphasis is given to temporal constraints in questions and how this affected the outcome of the systems in the task. LCC's results in the evaluation are presented at the end of the paper.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software (Question Answering); H.3.7 Digital Libraries; I.2 [**Artificial Intelligence**]: I.2.7 Natural Language Processing

General Terms

Measurement, Performance, Experimentation

Keywords

Open-domain Question Answering, Questions beyond factoids

1 Introduction

Language Computer Corporation participated in the QA track of the 15th annual TREC evaluation. TREC 2006 brought new modifications and challenges from the previous year, which we addressed through new developments in our QA system, PowerAnswer. The format of the main task was largely unchanged when compared to 2005. There were 75 target sets composed of both entity and event targets, each set including FACTOID, LIST and OTHER questions, for a total of 567 questions.

While the format remained the same as last year, there were extra challenges included in this year's question set. There was an increased focus on temporality with time-dependent questions. These are questions for which there is an implicit or explicit specification of a temporal range from which the correct answer must come. This time frame can be expressed in a variety of ways. Present tense questions are seeking the most up-to-date information available in the corpus. Past tense questions declare the time frame explicitly (*What position did she [Janet Reno] have immediately prior to 1993?*) or implicitly through the target (*the Queen Mum's 100th Birthday*).

In light of the new temporal constraints in the question set, there is a new judgement category for responses this year - *correct* has become *locally correct* and *globally correct*. A response is judged *locally correct* if the response is correct in the context of the source document, but the total corpus contains more up-to-date information which is considered *globally correct*. Locally correct answers do not contribute positively to the final system score.

Additionally, LIST and OTHER questions are unchanged from last year, though the final per-series score has been updated to give equal weight to each question type. This gives a higher weight to the more subjective and open OTHER questions than previous years.

This paper describes the PowerAnswer 3 system used in TREC 2006, the improvements made from the previous year and a deep analysis of the system's performance. A discussion on new directions for PowerAnswer inspired by the TREC evaluation will conclude this paper.

2 Overview of PowerAnswer 3

The architecture of the PowerAnswer 3 Question Answering system is illustrated in Figure 1. Automatic question answering requires a system that has a wide range of tools available. There is no one monolithic solution for all question types or even data sources. In realization of this,

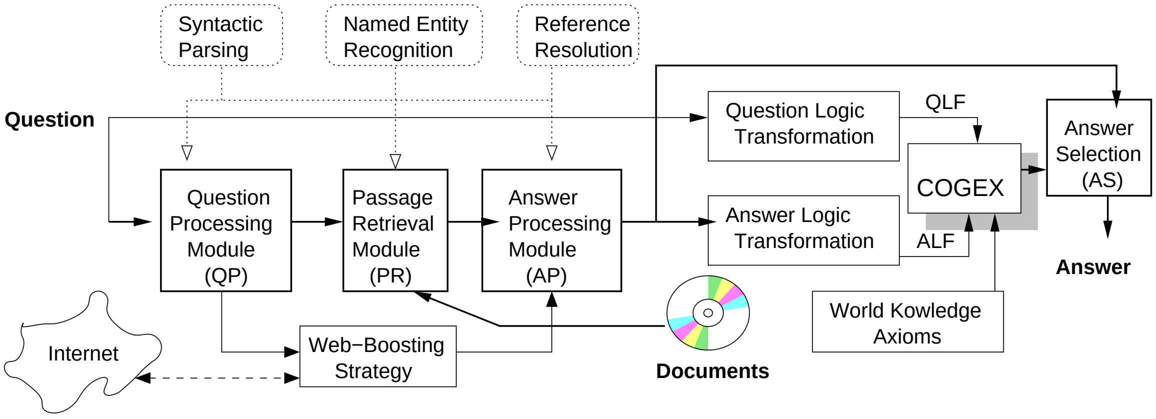

Figure 1: Architecture of PowerAnswer 3

LCC developed PowerAnswer as a fully-modular, distributable, strategy-based question answering system that is highly configurable for large or small tasks. PowerAnswer integrates sophisticated Natural Language Processing components such as embedded ontologies, semantic relation extraction, advanced inference, coreference resolution, temporal contexts, and eXtended WordNet-based lexical chains.

PowerAnswer comprises a set of strategies that are selected based on advanced question processing, and each strategy is developed to solve a specific class of questions either independently or together. A Strategy Selection module automatically analyzes the question and chooses a set of strategies with the algorithms and tools that are tailored to the class of the given question. PowerAnswer can distribute the strategies across workers in the case of multiple strategies being selected, alleviating the increase in the complexity of the question answering process by splitting the workload across machines and processors (Moldovan, Munirathnam et al. 2006).

The major processing steps highlighted in Figure 1 are question processing (QP), passage retrieval (PR) and answer processing (AP). The role of the QP module is to determine (1) temporal constraints, (2) how the target (if given) should relate to the current question, (3) the expected answer type and (4) to select the keywords used in retrieving relevant passages. The PR module ranks passages that are retrieved based on lexical similarity, while the AP determines the extraction of the candidate answers and performs semantic matching and scoring as well as introduces a number of post-processing steps for re-ranking. All modules have access to a syntactic parser, a named entity recognizer and a reference resolution system among many other NLP tools. LCC utilizes a generic set of NLP tools and interfaces that allow us to quickly develop or integrate new tools and processing modules in a pipeline of text processing and understanding.

To improve the methods used for answer selection, we take 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 (Lin, 2002). The web-boosting 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 from LCC's COGEX inference engine.

3 What's new from TREC2005

To meet the challenges of the new complex questions and temporal constraints, LCC developed or included several new innovations into the latest version of PowerAnswer 3. The following subsections will detail these new additions from what was used at TREC 2005 (Harabagiu, Moldovan et al. 2005).

3.1 Question Understanding

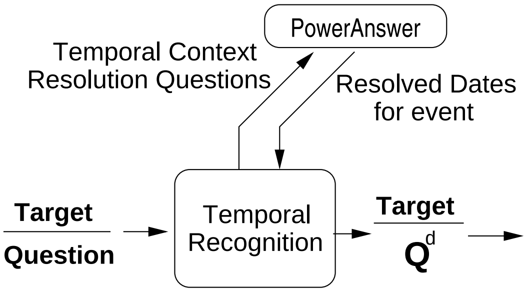

Figure 2: QP: Temporal Resolution

Temporal Resolution

Before strategy selection occurs, some initial question processing takes place. The target and unmodified question are sent to the temporal resolution module, shown in Figure 2 which analyzes the target and question together to resolve any ambiguous temporal context and use this information in the reformulated question to be further processed.

If the target is labelled as an event, any explicit dates are identified, extracted from the target and added to the reformulated question. If there are no explicit temporal contexts in the event target, the module performs a limited question answering procedure to answer "When was TARGET?" The resulting temporal answer is then added to the refomulated question, exemplified in Table 1. If the scores of the answers to this "When" question are similar enough to indicate ambiguity, a date range is created among the top answers and that range is then added to the reformulated question. Our temporal context processing methods will correctly match a date that falls within the desired range. If the question itself contains an explicit temporal context, then none of the above processing is performed and question understanding continues.

Q175.4: *(repatriation of Elian Gonzales) Who was the U.S. Attorney General at the time?*	
Prelim. Q	When was repatriation of Elian Gonzales?
Prelim. A	2000
New Q	Who was the U.S. Attorney General at the time *in 2000*?
Answer	"Earlier Thursday, U.S. Attorney General Janet Reno said ..." (from a 2000 document)

Table 1: Temporal Constraint Example

Target Understanding

The next step in question processing is to understand how the target relates to the question. There are times when the target simply provides context and other times when the question directly references the whole target, or only a part. Furthermore, at TREC, the references to answers from previous questions of the same target is constantly increasing.

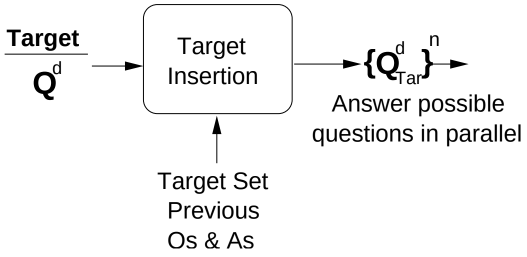

Figure 3: QP: Target Understanding

The first step of the module described in Figure 3 is to examine the possibility of references to previous answers, exemplified in Table 2. Steps, such as type, gender and number matching, are taken to examine the current question, the previous questions within the target set, and their answers and to determine if a previous answer should be inserted. If this succeeds, the new question is saved to a list of reformulations. Next, the module attempts to insert information from the target or use it as context. If no point of insertion is found in the question, the target is added as the question's context which restricts the document retrieval. Several steps are taken to resolve where and in what capacity the whole target or a portion of it should be used in the question. For all the above steps that succeed, a new question is saved in the refomulation list.

Target 158: *Tufts University*	
Q158.1	**Who** became Tufts University President in 1992?
A158.1	*John DiBiaggio*
Q158.2	Over which other university did **he** *(John DiBiaggio)* preside?
Q158.3	What was Tufts' endowment in 1992 when **he** *(John DiBiaggio)* became president?

Table 2: Previous Answer Insertion

Next, all the reformulated questions are sent through the remainning processing pipeline and, at the end, voting is performed to determine which of the ambiguous target understanding reformulations have higher confidence. The top answer of the most unambiguous reformulated question is selected and returned as the final result. The information about target insertions is stored in a cache which shall be used for subsequent questions in the same target set.

Answer type detection

We extended PowerAnswer's answer type detection module by moving it to a hybrid system which takes advantage of precise heuristics as well as machine learning algorithms for ambiguous questions. A maximum entropy model was trained to detect both answer type terms and answer types. The learner's features for answer type terms include part-of-speech, lemma, head information, parse path to WH-word, and named entity information. Answer type detection uses a variety of attributes such as additional answer type term features and set-to-set lexical chains derived from eXtended WordNet[1] which links the set of question keywords to the set of potential answer type nodes.

[1] `http://xwn.hlt.utdallas.edu`

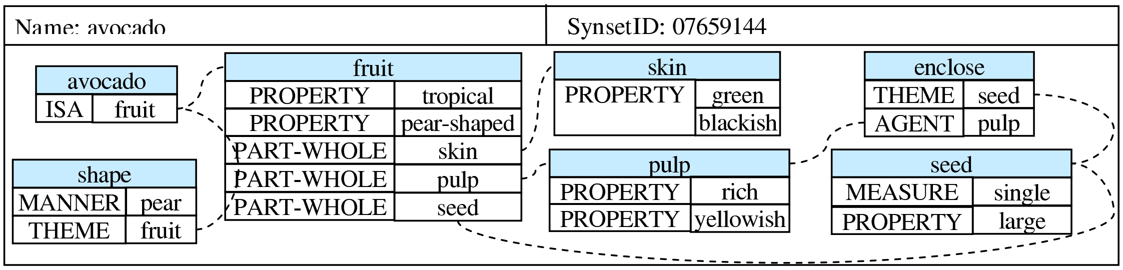

Figure 4: Semantic representation of *avocado* in XWN-KB

3.2 Temporal context/temporal ranking of answers and documents

For TREC 2006, we extended the temporal context mechanisms which originally relied on signal words such as "in", "of", "prior", and "during", and the detection of SUMO (Niles and Pease 2001) events to flag if a temporal context should be marked within a text snippet (Moldovan et al. 2005). Although this approach is precise, the coverage for attaching a temporal context to an event was lower than required for the TREC 2006 task. For this reason, LCC introduced a more robust default temporal context mechanism that operates at several levels of granularity: phrase, sentence, passage, and document. If a date that unifies with the temporal requirements of the input question is detected in the discourse surrounding a candidate answer, a temporal boost is applied, which is scaled according to the syntactic proximity of the detected date to the candidate answer.

To handle the new distinction between "locally correct" and "globally correct", we also included temporal post-processing of the answer list. Dates for documents and the temporal context of the answer are maintained through question answering and after initial ranking, answers are given a boosting factor on top of their current relevance score that is intended to give greater priority to strong answers that are more recent than other strong answers. Answers that appear further down the response list and have lower relevance scores will not be affected by this boosting. One disadvantage to this is that the scoring of a response as "locally correct" was not exclusive to out-of-date information. There can be misreported information in the document, which might be much more recent than a correct answer found in an older document. There were several cases of later documents contradicting correct answers with incorrect ones.

For example, Q195.2 *What percentage of the vote was for (East Timor) independence?* has the exact correct answer of "78.5 percent". Participant responses of "78 percent" to "79 percent" were returned and marked as correct from documents ranging from 1999/09/04-07. One document (NYT19991001.0212) dated 1999/10/01 states "80 percent of the voters chose independence", a more recent response with misreported data, and marked as "locally correct". LCC had a total of 11 locally correct factoids.

Because temporal answers can have a range of granularity, when pre-processing the data collection, the named entities stored in the IR index are extracted in a greedy fashion, so both "March 14, 1592" and "2000" will be tagged as *_date* to give PowerAnswer the best flexibility for entity selection. During answer processing, if the question is seeking just a month, or a year, then the excess information from the *_date* entity selected is removed after a more fine-grained NE recognition is performed on the answer nugget. Questions 182.4 *In what month is the Edinburgh Fringe held?* and 182.6 *In what year was the Edinburgh Fringe begun?* demonstrated the utility of this method. Otherwise, if a simple "When was ..." question is asked, the entity with the most detailed temporal information would be the final answer. This method operated on 38 questions seeking *year, month,* or *day* in this year's factoid set.

3.3 eXtended WordNet Knowledge Base

eXtended WordNet Knowledge Base (XWN-KB) is the result of LCC's ongoing research which captures and stores the rich world knowledge encoded in WordNet's glosses into a knowledge base.

To achieve this goal, LCC used the parsed and sense disambiguated glosses of WordNet (eXtended WordNet) and its semantic parser, Polaris (Bixler et al. 2005), to derive a highly semantic representation of WordNet's definitions. For example, the first sense of the word "avocado" defined as *a pear-shaped tropical fruit with green or blackish skin and rich yellowish pulp enclosing a single large seed* is represented in XWN-KB as shown in Figure 4.

Given XWN-KB's semantic clusters, the accuracy of our lexical chains module increased greatly. The ambiguous `GLOSS` relation which used to link a concept $w1$ with any concept $w2$ present in $w1$'s gloss was replaced by chains which describe the exact association between $w1$ and $w2$. For instance:

(n-avocado#1, alligator_pear#1, avocado_pear#1, aguacate#1) GLOSS **(n-seed#2)**

became:

(n-avocado#1, alligator_pear#1, avocado_pear#1, aguacate#1) ISA (n-fruit#1) PART-WHOLE (n-seed#2)

whose meaning and strength is very different than:

(n-avocado#1, alligator_pear#1, avocado_pear#1, aguacate#1) GLOSS **(a-rich#2)**

which is explicited as:

(n-avocado#1, alligator_pear#1, avocado_pear#1, aguacate#1) ISA (n-fruit#1) PART-WHOLE (n-pulp#2, flesh#3) PROPERTY (a-rich#2).

3.4 COGEX - Natural Language Logic Prover

LCC's natural language logic prover, COGEX, is used by PowerAnswer to provide the final re-ranking of the candidate answers based on the degree of semantic entailment between the candidate answer passage (CAP) and the question (Q).

3.4.1 Knowledge Representation Model

COGEX uses a three-layered logical representation which captures the syntactic, semantic and temporal propositions encoded in a text fragment (either question or answer passage). The details of the logic transformation methodology are presented in (Moldovan et al., 2003; Moldovan et al. 2005). Several extra modules were added to this process.

Negation detection and representation

In cases similar to the best ranked candidate answer passage for Q141.1,

If the Seattle Seahawks do not increase their one-year, $1.8 million offer, ...[2],

the system removes `not_RB(x3,e1)` and negates the verb's predicate `(-increase_VB(e1,x12,x17))` unless the verb is modified by an adverb. For example, one of the top passages retrieved for Q143.3,

Ms. Kirkpatrick's spokeswoman at the institute (American Enterprise Institute) did not immediately return a call for comment

is represented as:

```
ms__NN(x1) & kirkpatrick_NN(x2) &
nn_NNC(x3,x1,x2) & _human_NE(x3) &
_s_POS(x3,x4) & spokeswoman_NN(x4)
& at_IN(x4,x5) & institute_NN(x5)
& -immediately_RB(x8,e1) &
return_VB(e1,x4,x7) & call_NN(x7)
& for_IN(x7,x6) & comment_NN(x6) &
PART-WHOLE_SR(x4,x5) & AGENT_SR(x4,e1)
& THEME_SR(x7,e1) & PURPOSE_SR(x6,x7).
```

Similarly, for nouns whose determiner is *no*, for example, for *No NFL starter has more experience than Moon, ...* (candidate answer passage for Q141.4), the verb's predicate is negated:

```
NFL_NN(x1) & _organization_NE(x1) &
starter_NN(x2) & -have_VB(e1,x2,x3) &
more_JJ(x7,x3) & experience_NN(x3)
```

Quantification detection

The quantification of entities plays a role when a proof's final score is computed. If we assume that both the candidate answer passage and question are existential, then $CAP \vdash Q$ if and only if CAP's entities are a subset of Q's entities, such as:

Some smart people read $\vdash$ *Do people read ?* (yes)

and penalizing a pair whose Q contains predicates that cannot be inferred is a correct way to ensure entailment:

Some people read $\nvdash$ *Do smart people read ?* (yes).

But, if both CAP and Q are universally quantified, then the groups mentioned in Q must be a subset of the ones from CAP

All people read $\vdash$ *Do all smart people read?* (yes)

and:

All smart people read $\nvdash$ *Do all people read?* (yes).

Thus, COGEX's proof-scoring module adds back the points for the modifiers dropped from Q and subtracts points for CAP's modifiers not present in Q.

Conditional detection and representation

To ensure the correctness of the logic form which impacts the prover's reasoning process, we adapted the logic form representation of a text snipet to reflect the conditional dependencies between two or more clauses.

For example, the sentence from one of the top passages returned for Q164.6:

If you have a farm, bet it

is represented as

```
(you_PRP(x3) & have_VB(e1,x3,x1)
& farm_NN(x1) & AGENT_SR(x3,e1) &
THEME_SR(x1,e1)) -> (bet_VB(e2,x3,x1)
& AGENT_SR(x3,e2) & THEME_SR(x1,e2)).
```

3.4.2 Natural Language Axioms

Our NLP axioms are linguistic rewriting rules that help break down complex logic structures and express syntactic equivalence. After analyzing the logic form and the parse trees of each text fragment, the system, automatically, generates NLP axioms to break down complex nominals and coordinating conjunctions into their constituents so that other axioms can be applied, individually, to the components. These axioms are made available only to the (CAP,Q) pair that generated them. For example, the noun compound *Ben Cohen* (related to Q172.3) is broken down into *Ben* and *Cohen*.

The axioms

```
nn_NNC(x3,x1,x2) & _human_(x3)
& ben_NN(x1) & cohen_NN(x2) ->
ben_NN(x3)
```

and

```
nn_NNC(x3,x1,x2) & _human_(x3)
& ben_NN(x1) & cohen_NN(x2) ->
cohen_NN(x3)
```

help the system infer the answer for Q172.3 *What is Ben's last name of Ben & Jerry's ?*

In this year's TREC competition, we introduced a new type of NLP axioms which capture the equivalence of names. In the passage

The jury of 11 whites and one African-American, whom they elected foreman, had convicted Bill King of capital murder ...,

`Bill King` refers to the same entity as `John William King` from Q145.1 *How many non-white members of the jury were there of John William King convicted of murder ?*

Using the keyword alternation information provided by PowerAnswer's Question Processing (QP) module, COGEX automatically created and used during inference the axiom:

```
bill_NN(x7) & king_NN(x8) &
nn_NNC(x9,x7,x8) -> john_NN(x2)
& william_NN(x3) & king_NN(x4) &
nn_NNC(x5,x2,x3,x4)
```

3.4.3 Lexical Chain Axioms

For the task of identifying the semantic entailment between a question and its candidate answer, the ability to recognize two semantically-related words is an important requirement. Therefore, we, automatically, construct lexical chains of WordNet relations annotated between the synsets, from the words in CAP to Q's constituents (Moldovan and Novischi 2002). For each relation in the best chain[3], the system generates, on demand, an axiom with the predicates of the synsets in the WordNet relation. For example, given the XWN lexical chain

(v-call#4,send_for#1) -DERIVATION-> (n-caller#5) -HYPERNYM-> (n-announcer#1) -DERIVATION-> (v-announce#3)

between *call/VB* and *announce/VB*, the system constructs three axioms:

`call_VB(e1,x1,x2) -> caller_NN(x1)`,
`caller_NN(x1) -> announcer_NN(x1)` and
`announcer_NN(x1) -> announce_VB (e1,x1,x2)`.

We adopted this new one-axiom-per-chain-relation approach (compared with one-axiom-per-chain) because XWN-KB contains a much larger set of semantic relations and it is very difficult to reduce and entire lexical chain to only one implication which captures its entire meaning. By assigning a set of axiom templates to each semantic relation (examples in Table 3), a lexical chain is broken down into several axioms whose relations are combined by the logic prover as it sees fit. Moreover, the weight of each axiom depends on the relation that defines it and the weight of the chain where it originated from.

Other improvements added to the lexical chain builder module include lexical chain axioms which append the entity name of the target concept, whenever it exists, to the created axiom. For example, COGEX uses the axiom

```
south_african_JJ(x1,x2) ->
south_africa_(x1) & _country_NE(x1)
```

when it tries to infer *president of South Africa* (Q183.1) from *South African president*.

We ensured the relevance of the lexical chains by limiting the path length to three relations and the set of WordNet relations used to create the chains by discarding the paths that contain certain relations in a particular order. For example, the automatic axiom generation module did not consider chains with an IS-A relation followed by a HYPONYMY link. Without removing these types of chains, our system inferred, for instance, *John lives in Detroit* from *John lives in Chicago* ($Chicago \xrightarrow{is-a} city \xrightarrow{hyponymy} Detroit$). Similarly, the system rejected chains with more than one HYPONYMY relations. Although this relation links semantically related concepts, the questions are more general than the candidate answer passages and too many HYPONYMY relations can lead to a too specific concept in Q.

3.4.4 Semantic Calculus

The Semantic Calculus axioms combine two semantic relations. Their purpose is three fold: (1) increase the semantic connectivity of a text fragment by combing relations identified in text, (2) which can make explicit unstated relationships and longer dependencies within a text fragment, and (3) pinpoint the semantic association between concepts linked by lexical chains. Once the system breaks down the generic lexical chain *(w1) -SR1-> (w2) -SR2-> (w3)* into `w1(x1) -> w2(x2) & SR1(x1,x2)` and `w2(x2) -> w3(x3) & SR2(x2,x3)`, the semantic calculus axiom `SR1(x1,x2) & SR2(x2,x3) -> SR3(x1,x3)` defines the semantic connection `S3` = (`SR1` ∘ `SR2`) between `w1` and `w3` across the lexical chain.

3.4.5 Temporal Axioms

One of the types of temporal axioms that we load in our logic prover links specific dates to more general time intervals. For example,

October 2000 entails the year *2000*

```
time_TMP(BeginFn(e1), 2000, 10,
1, 0, 0, 0) & time_TMP(EndFn(e1),
2000, 10, 31, 23, 59, 59)          →
time_TMP(BeginFn(e1), 2000, 1, 1,
0, 0, 0) & time_TMP(EndFn(e1), 2000,
12, 31, 23, 59, 59).
```

[3] Shorter chains are better than longer ones. The relations are not equally important and their order in the chain influences its strength.

Semantic Relation	**Axiom Templates**

ISA	n1(x1) -> n2(x1)
	v1(e1,x1,x2) -> v2(e1,x1,x2)

DERIVATION	n(x1) -> v(e1,x1,x2) & AGENT_SR(x1,e1)
	n(e1) -> v(e1,x1,x2)
	v(e1,x1,x2) -> n(x1)
	v(e1,x1,x2) -> n(e1)

CAUSE	v1(e1,x1,x2) -> v2(e2,x2,x3) & CAUSE_SR(e1,e2)
AGENT	n1(x1) -> n2(x2) & AGENT_SR(x1,x2)
PERTAIN	a(x1,x2) -> n(x1)

Table 3: Semantic Relation - Axiom Template mapping

These axioms are automatically generated before the search for a proof starts. Additionally, the prover uses a SUMO knowledge base of temporal reasoning axioms that consists of axioms for a representation of time points and time intervals, Allen primitives, and temporal functions. For example,
during is a transitive Allen primitive, giving:

```
 during_TMP(e1,e2) & during_TMP(e2,e3)
 -> during_TMP(e1,e3).
```

4 Challenges at TREC 2006

The largest challenge introduced for this year's evaluation is the new emphasis on time-dependent questions. These are FACTOID questions for which the target set or question explicitly or implicitly describes a temporal range in which the answer must lie. The result of this is the new type of response judgement, "locally correct" where the response is correct for the local document but not "globally" for the whole data collection. Table 4 lists several examples of these types of questions. In TREC 2005, 16% of the FACTOID and LIST questions set temporal constraints. For this year, that number rose to 20.1% and included more complexity. Errors in temporal context understanding and processing comprised 14% of the errors this year.

Q149.3: *(The Daily Show) Who is host of The Daily Show?*	
Locally Correct:	NYT19980810.0417 - Kilborn
Globally Correct:	NYT19991220.0172 - John Stewart (who replaced Craig Kilborn)

Table 4: Questions with "locally correct" answers

Another complexity added in this year's evaluation were questions in the same target set that are syntactically very similar with only one or two words changed that completely alter the goal of the question. Examples can be seen in Table 5. If the QA system was not able to fully understand the difference or if the important keyword was incorrectly relaxed from the IR query because of insufficient documents retrieved, the semantic difference is lost and the results of the two questions is the same.

Q187.1:	In what country is the **origin** of the Amazon River?
Q187.2:	In what country is the **mouth** of the Amazon River?
Q166.1:	How many humans **were infected** with avian flu in Hong Kong in 1997?
Q166.2:	How many humans **died** of avian flu in Hong Kong in 1997?

Table 5: Syntactically-similar questions

Target understanding remained a challenge this year as seen in the 18% error rate in Table 6. Complex targets such as *Pakistani government overthrown in 1999* and *cloning of mammals (from adult cells)* require better precision from PowerAnswer's question processing module to determine the relation between the target and the information being sought in the question. Questions that refer to answers from previous questions complicate the processing even more, as in the target series: **Q197.1** *What animal was the first mammal successfully cloned from adult cells?*, the answer to which is used in **Q197.2** *What year was this animal born?* and answers from both these is further required as a context to **Q197.3** *At what institute was this procedure done?*

Error	Dist
Answer Ranking	20%
Target Understanding	18%
Temporal Context	14%
Semantic Selection	12%
Answer Type Detection	11%
Keyword Expansion	9%
Keyword Selection	8%
NIL Answer	4%
Strategy Selection	3%
Passage Selection	1%

Table 6: Error Distribution (Factoid)

Type	Num
(Globally) Correct	233
Inexact	32
Locally Correct	11
Unsupported	12
Wrong	115

Table 7: Answer Distribution (Factoid)

Evaluating question answering results continues to be a challenge as well, as each judge holds differing opinions on what makes an insufficient or oversufficient response. This subjectivity in evaluation in general tends to affect all systems to the same extent but does lead to discussion about correctness of responses. Examples this year involving "inexact": for one question, a response of only "Washington" (meaning District of Columbia) was judged correct, whereas a reponse of "Salzberg" (Austrian city) was judged inexact and only "Salzburg, Austria" marked as correct. For Q180.6 *How many years are in a term of the Lebanese Parliament?*, responses of "four years" were judged inexact and only an answer of "four" was correct. This leads to a discussion of what information is needed to be sufficient, especially in cases of location answers and does clarifying information such as "times" or "years" make a response over-specified. LCC had a total of 32 inexact responses, as seen in Table 7.

5 Evaluation Results

Table 8 illustrates the final results of Language Computer's efforts in the 2006 TREC main QA track obtained by PowerAnswer 3. As seen in Table 8, PowerAnswer 3 was the top performer in 2 of 3 question types and best overall system.

	PowerAnswer 3	Best	Median
Factoid Acc.	0.578	0.578	0.186
List F-score	0.433	0.433	0.087
Other F-score	0.167	0.250	0.125
Overall	0.394	0.394	0.134

Table 8: Results for the main QA task

6 Conclusion

Questions of increasing complexity will continue to demand greater use of contextual information to decrease ambiguity and increase precision for similar questions and targets. More advanced reasoning is required to operate on these contexts to find the right answer through the noise of the collection. Future directions for PowerAnswer include better detection and usage of contexts in text, more complex hierarchical representation of knowledge, and problem solving using question answering. This year's TREC yet again helped to push the envelope of what kinds of questions current systems are able to handle. The question sets with close syntactic similarity test the semantic understanding of the participating systems and the emphasis on temporal correctness demands better reasoning.

Acknowledgements

We would like to thank the researchers and engineers from Language Computer Corporation for their invaluable contributions to this work. This work, and also our broader research in QA, is supported by the Disruptive Technology Office's (DTO) 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. In *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, D. Moldovan, C. Clark, M. Bowden, A. Hickl and P. Wang. Employing Two Question Answering Systems in TREC-2005. In *Proceedings of TREC 2005*.

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

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, S. Harabagiu, C. Clark and M. Bowden. PowerAnswer-2: Experiments and Analysis over TREC 2004. In *Proceedings of TREC 2004*.

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

D. Moldovan, S. Munirathnam, A. Fowler, A. Mohammed and E. Jean. Synergist: Tools for Intelligence Analysis. *NIMD Conference*, Arlington, VA, 2006.

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

M. Tatu and D. Moldovan. 2006. A Logic-based Semantic Approach to Recognizing Textual Entailment. In *Proceedings of the COLING/ACL 2006 Conference*.

M. Tatu, B. Iles, J. Slavick, A. Novischi and D. Moldovan. 2006. COGEX at the Second Recognizing Textual Entailment Challenge. In *Proceedings of the PASCAL Second Recognising Textual Entailment Challenge*.

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*.