Available online at www.sciencedirect.com

ScienceDirect

Journal of Applied Logic 5 (2007) 49–69

JOURNAL OF APPLIED LOGIC

www.elsevier.com/locate/jal

COGEX: A semantically and contextually enriched logic prover for question answering

Dan Moldovan *, Christine Clark, Sanda Harabagiu, Daniel Hodges

Language Computer Corporation, Richardson, Texas 75080, USA

Available online 3 February 2006

Abstract

This paper presents the architecture and functionality of a logic prover designed for question answering. The approach transforms questions and answer passages into logic representations based on syntactic, semantic and contextual information. World knowledge supplements the linguistic, ontological, and temporal axioms supplied to the prover which renders a deep understanding of the relationship between the question and answer text. The trace of the proofs provides a basis for generating human comprehensible answer justifications. The results show that the prover boosts the performance of the Question Answering system on TREC 2004 questions by 12%.

Keywords: Question answering; Natural language processing; Temporal reasoning; Automated reasoning; Ontologies

1. Introduction

1.1. Motivation

In spite of significant advances made recently in the question answering technology, there still remain many problems to be solved. Some of these are: bridging the gap between question and answer words, pinpointing exact answers, taking into consideration syntactic and semantic roles of words, better answer ranking, answer justification, and others. Recent TREC[1] results [15] have demonstrated that many performing systems reached a plateau by answering simple factoid questions reasonably well, but failing to extract answers even when simple inferencing is required. It is clear that new ideas based on a deeper language understanding are necessary to push Question Answering (QA) technology further.

In this paper we introduce one such novel idea, the use of automated reasoning in QA, and show that it is a feasible and effective tool for answer validation. We have implemented a logic prover, called COGEX,[2] which uniformly codifies the question and answer text, as well as world knowledge resources, in order to use its inference engine to verify and extract any lexical and semantic relationships between the question and its candidate answers [8].

* Corresponding author.
E-mail address: moldovan@utdallas.edu (D. Moldovan).

[1] The TExt Retrieval Conference (TREC) evaluates open-domain QA systems on a yearly basis. Detailed information is posted at (http://trec.nist.gov).

[2] The name of the prover represents the permutation of the first two syllables of the verb *excogitate*.

1570-8683/$ – see front matter
doi:10.1016/j.jal.2005.12.005

1.2. Technical challenges

The challenges faced when interpreting questions and candidate answers are similar to the challenges encountered when interpreting texts by providing the minimal explanation of why the text would be true. In [6], this form of interpretation was viewed as *abduction*, or inference to the best explanation. In the case of QA, the abductive inference proves the truth of the question given a certain answer. To perform the proof, a QA system must (1) generate logical forms of the question and answer; (2) have access to world knowledge or approximations of it; and (3) resolve several "local pragmatics" problems like linguistic equivalence of semantic patterns, interpretation of complex nominals or interpretation of semantic relations. The abduction is performed as a logical proof which does not always succeed either because of insufficient world knowledge available or because of lack of interpretation rules needed in special contexts. Our solution is to augment the capabilities of COGEX by enriching its initial syntax-based logical forms with (1) semantic relations and (2) predicates available from knowledge sources that enable special forms of reasoning. The enhanced COGEX was integrated into the QA system to augment other previously implemented answer extraction techniques.

1.3. Approach

The logical forms on which COGEX operates are generated by a multi-layered approach. The first layer represents in first order logic syntactic relations derived from the results of a statistical parser similar to [4]. The syntactic relations consist of verb-subject, verb-object, prepositional attachments, adverbial and adjectival adjuncts, complex nominals and coordinations. The second layer adds semantic information of two kinds: (1) semantic entity class provided by named entity recognizers; and (2) semantic relations supplied by a semantic parser. The semantic relations include *is-a*, *part-whole*, *manner*, *purpose* and *cause* relations. The third layer is used to allow the QA system to be adequately responsive to different world views, and thus it is known as the contextual layer. For example, for a time sensitive world view, a *temporal context* generates mappings from language to predicates from the Suggested Upper Merged Ontology (SUMO), which as reported in [12], includes James Allen's temporal primitives [1].

World knowledge, which allows COGEX to generate the abductive proofs, is provided in three forms as well. First, an approximation of world knowledge is available from the glosses defining concepts in WordNet. The WordNet publicly available database, encodes a vast majority of English nouns, verbs, adjectives and adverbs. WordNet is available from www.cogsci.princeton.edu/~wn. It groups words into synonym sets, or *synsets*, which represent a lexico-semantic concept. Each synset is also defined by a gloss. In the eXtended WordNet (XWN) glosses are transformed in the same logical representation on which COGEX operates. The XWN is available from http://xwn.hlt.utdallas.edu. For each word that is encountered in a question or answer, the logical form of the gloss defining it is used as a relevant axiom for the abductive inference.

The second form of world knowledge is provided by *lexical chains* linking any two content words via WordNet synsets and WordNet relations. A method for deriving lexical chains from the WordNet lexico-semantic database was presented in [10]. Lexical chains play an important role since they supply to the abductive inference all axioms representing the defining glosses of the synsets linking any pair of words from the question and the answer. In this way, they fill knowledge gaps corresponding to unstated, but implied information.

The third form of world knowledge corresponds to several pragmatic problems, including paraphrases, idioms and combinations of semantic and contextual relations as a semantic calculus. By combining logic representations of questions and answers with axioms supplying world knowledge COGEX effectively and efficiently re-ranks candidate answers by their ability to justify the questions that asked for them. In this way, the logic prover is a powerful tool in boosting the accuracy of the QA system.

2. A logic prover for textual question answering

A QA system that has the option of justifying the answer to a question has the advantage of ruling out erroneous answers. The justification is (a) provided by the abductive inference when proving the validity of the question for a candidate answer; and (b) it is detailed by the trace of the proof. To integrate a prover in a QA architecture, several interactions with the QA modules need to be made possible.

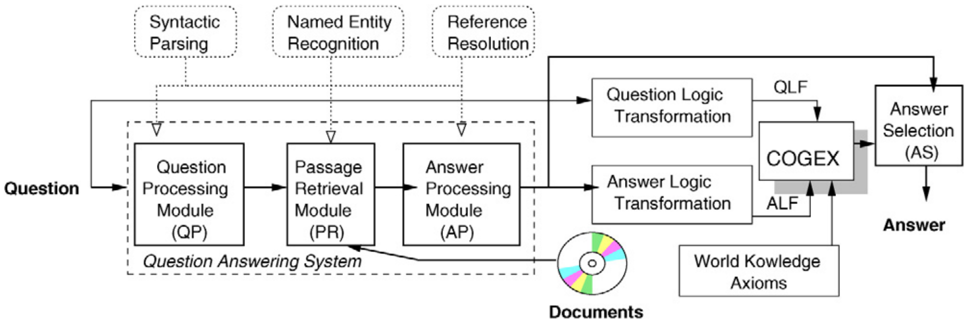

Fig. 1. Architecture of a Question Answering system that includes a logic prover.

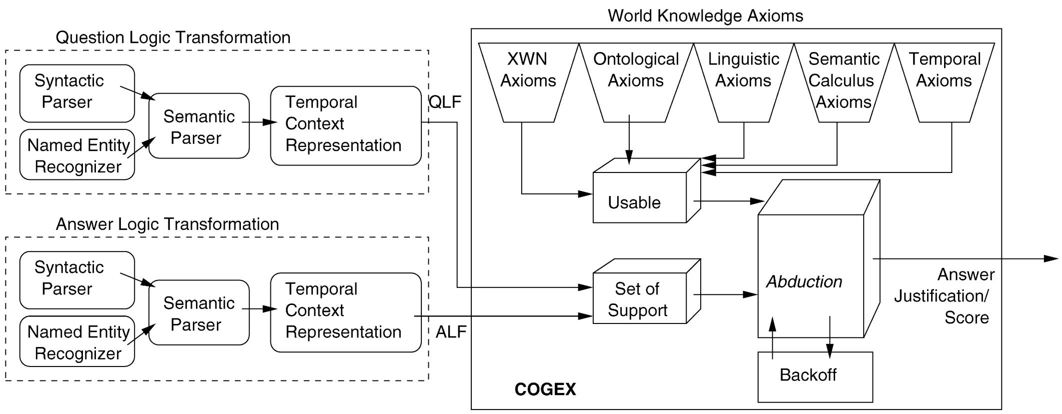

Fig. 2. COGEX architecture.

Typically, QA systems have three different modules: the question processing (QP) module, the passage retrieval (PR) module and 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 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 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 original QA system. Fig. 1 illustrates the architecture of a QA system that includes the COGEX logic prover.

As illustrated in Fig. 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. Fig. 2 details the operation of COGEX on its input axioms. The first order logical translations of the question and its associated candidate answers are loaded into the Set of Support (SOS) [14] of COGEX, since they are intended to guide the search of the proof. The QLF and the ALF are produced in a three-layered first order logic representation which is detailed in Section 3. The first layer relies on the syntactic parse and the named entity recognition, available also to the QA modules, as was illustrated in Fig. 1. The second layer relies on the recognition of semantic relations processed by the semantic parser reported in [2]. 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 [7] and it is detailed in Section 3.3. Fig. 2 also illustrates the translation of questions and candidate answers into QLFs and ALFs before being supplied to the SOS.

The usable axioms illustrated in Fig. 2 provide world knowledge to COGEX. There are five sources of usable axioms. First, we employ axioms derived from the eXtended WordNet (XWN). To select the axioms from WordNet we use two principles: (a) we chose all logical transformations of glosses defining question or candidate answer words, regardless of their semantic sense; (b) we chose all logical transformations of glosses defining synsets from the lexical chains linking any word from a question with an answer word. Second, we use ontological axioms generated by the JAGUAR knowledge acquisition tool described in [2]. JAGUAR encodes subsumption relations derived from an ontological hierarchy. Third, linguistic axioms are generated to account for the interpretation of several linguistic phenomena, e.g. appositions, nominal coreference or possessives. Section 4.2 details the linguistic axioms that are provided in COGEX. The fourth source is a semantic calculus operating on a subset of the semantic relations recognized in questions and answers. It is detailed in Section 4.3. The fifth source of axioms is based on temporal reasoning axioms available from the SUMO knowledge base. The usable list [14] is a set of axioms used by COGEX to supplement the knowledge in the set of support.

Once the axioms are loaded into COGEX, the abductive proof begins. If a proof fails, the backoff module is invoked. The purpose of this module is twofold: (1) to compensate for errors in the text parsing and logical form transformation phase, such as prepositional attachments and recognition of subject-verb or object-verb relations; (2) to enable abductions when the world knowledge is insufficient for providing all the necessary inferences. During the backoff, arguments of predicates from the question are incrementally unbound, the proof score is reduced, and the abduction is re-attempted. The loop between the abduction and the backoff module continues until the proof succeeds, or the proof score is below a threshold based on the connectivity of the proof. After all the candidate answers are processed, they are re-ranked based on their proof scores. COGEX passes to the answer selection module the re-ranked list of answers, their scores and their proofs as possible answer justifications.

3. A three layered first order logic representation of text

The knowledge representation used by COGEX has three layers that contribute to its ability to soundly reason about the similarity between a question and its candidate answers: (1) syntax-based representation, (2) semantic relation representation, (3) contextual representation. Each will be detailed below.

3.1. Syntactic representation

The first layer localizes the syntactic ambiguities in questions and answers by having access to full syntactic parses and semantic classes provided by named entity recognizers. The logical representation acknowledges syntax-based relationships such as: (1) syntactic subjects, (2) syntactic objects, (3) prepositional attachments, (4) complex nominals, and (5) adjectival/adverbial adjuncts. Additionally, as reported in [11], there is a one to one mapping of the words of the text into the predicates in the logical form. A predicate is generated for every noun (NN), verb (VB), adjective (JJ), adverb (RB), preposition or conjunction encountered in the question of candidate answer. The name of the predicate is a concatenation of the lexeme's base form and the part of speech of the word. Nouns, adjectives and adverbs have predicates with a single argument. Nouns and adjectives have the same argument when the adjective modifies the noun.

In the spirit of the Davidsonian treatment of the action predicates [5], all verb predicates (as well as their nominalizations representing actions, events or states) have three arguments: `action/state/event-predicate`(e, x_1, x_2), where:

- e represents the *eventuality* of the action, state or event stated by the verb to take place;
- x_1 represents the *syntactic subject* of the action, event or state; and
- x_2 represents the *syntactic direct object* of the action, event or state.

When the verb is bi-transitive, the predicate has an additional argument x_3, corresponding to the indirect object. The arguments of the verb predicates are always in the same order: subject, direct object, indirect object. In the case when one of these syntactic roles is missing, its respective argument is not filled. Moreover, adverbs modifying verbs share the same eventuality argument with the verbs.

Compound noun phrases (NNC), or complex nominals are represented by a grouping predicate that has the participant nouns of the phrase as its arguments. Similarly, preposition (IN) predicates have a fixed argument allocation: the first argument corresponds to the predicate of the phrase head to which the prepositional phrase is attached, whereas the second argument corresponds to the prepositional object. Conjunctions are transformed into predicates as well, which enables the aggregation of several predicates under the same syntactic role (e.g. subject, object or prepositional object). Predicates corresponding to conjunctions coordinate also several different events, actions or states. By convention, conjunction-predicates have a variable number of arguments, since they cover a variable number of predicates. The first argument represents the "result" of the logical operation induced by the conjunction (e.g. a *logical and* in the case of the `and` conjunction, or *logical or* in the case of the `or` conjunction). The rest of the arguments indicate the predicates covered by the conjunction, as they are arguments of those predicates as well.

For text that is in the form of a factoid seeking question, a special predicate is introduced to capture the target type, or answer type of the question. The answer type predicate (AT), is the focal point for the factoid driven proof search, and the marker used for answer extraction inside of COGEX. An example of a natural language sentence and its logical form representation is provided below.

Example 1. Bin Laden reportedly purchased anthrax a half decade ago from a supplier in North Korea.

```
Bin_NN(x1) & Laden_NN(x2) & nn_NNC(x3,x1,x2) & _human_ NE(x3) &
& reportedly_RB(x4,e1) & purchase_VB(e1,x3,x5) & anthrax_ NN(x5) &
& half_JJ(x6,x7) & decade_NN(x7) & ago_ JJ(x8,x7) & from_ IN(e1,x9)
& supplier_NN(x9) & in_IN(x9,x12) & North_ NN(x10) & Korea_ NN(x11)
& nn_NNC(x12,x10,x11) & _location_NE(x12)
```

A notable feature of the logical form representation used in COGEX is the fixed-slot allocation mechanism of the verb predicates following the Davidsonian notation introduced in[6]. This allows for the logic prover to distinguish the roles of the subjects and objects in a sentence and based on this syntactic information to select an answer that is expected to fill a specific role. The syntactic information adds to the semantic class information indicated by the expected answer type recognized by the QP module of the QA system. In Example 2, the sentence mentions two human entities and one only one of them is the correct answer to the question. The verb *give* has two objects: a direct object (*tickets*) and an indirect object (*Bill*). For this reason, the predicate `give_VB` has four arguments instead of the typical three arguments of verb predicates. The additional argument corresponds to the indirect object. The additional argument indicates the *recipient* role of the predicate, which is the entity asked about in the question. It is the role of the linguistic axioms detailed in Section 4.2 to establish the identity between the subject of receiving and the recipient of giving.

Example 2. John gave Bill tickets to the game.

```
John_NN(x1) & _human_NE(x1) & give_ VB(e1,x1,x2,x4) & Bill_ NN(x2)&
& _human_NE(x2) & ticket_NN(x4) & to_TO(x4,x5) & game_NN(x5)
```

Who received tickets to the game?

```
_human_AT(x1) & receive_VB(e1,x1,x2) & ticket_ NN(x2) & to_ TO(x2,x3)
& game_NN(x3)
```

Answer: *Bill_NN*$(x2)$

3.2. Semantic relation representation

The second layer of the logical representation of the text is obtained by using a semantic parser, namely POLARIS, which was described in [2]. The semantic relations discovered by POLARIS are abstractions of underlying relations between concepts that exist within a word, between words, between phrases, and between sentences. Semantic relations together with the syntactic relationships captured in the text serve as building blocks for marking contextual boundaries in the answer candidate text. Table 1 lists the relations that Polaris extracts from text.

Table 1
List of semantic relations recognized by POLARIS

Semantic relations			
Possession	Source-From	Possibility	Kinship
Topic	Certainty	Property-Attribute Holder	Manner
Agent	Means	Result	Theme-Patient
Temporal	Accompaniment-Companion	Stimulus	Depiction
Part-Whole	Experiencer	Extent	Recipient
Hyponymy	Frequency	Belief	Predicate
Entail	Influence	Goal	Cause
Associated-with/Other	Meaning	Make-Produce	Measure
Instrument	Synonymy-Name	Explanation	Justification
Location-Space	Antonymy	Purpose	Plausibility-of

Semantic relations provide the semantic background of the text, allowing for a denser connectivity between the words and concepts expressed in the text. The following example shows the semantic relations detected by POLARIS for the same sentence as in Example 1:

Example 3. Bin Laden reportedly purchased anthrax a half decade ago from a supplier in North Korea.

AGENT(Bin Laden, purchased)

TOPIC(purchased, reportedly)

THEME(anthrax, purchased)

RECIPIENT(a supplier in North Korea, purchased)

TEMPORAL(a half decade ago, purchased)

MEASURE(a half, decade)

LOCATION(in North Korea, a supplier)

When semantic relations are discovered, they are mapped into first order logic representations. The name of the predicates encoding semantic relations has two components: (a) the type of relation it encodes; and (b) the suffix SR indicating that the predicates stands for a semantic relation. The arguments of these predicates are events and entities participating in the relation. Because POLARIS uses the same syntactic information as the first layer of the logic representations, it is natural to layer the semantic information on top of the syntactic information:

```
& AGENT_SR(x3,e1) & TOPIC_SR(e1,x4) & THEME_SR(x5,e1)
& RECIPIENT_SR(x9,e1) & TEMPORAL_SR(x7,e1) & MEASURE_ SR(x6,x7)
& LOCATION_SR(x12,x9)
```

3.3. Temporal context representation

As a final layer to the knowledge representation we present contextual reasoning predicates. These are derived from both the semantic relations detected in the text, as well as from a module which utilizes a machine learning algorithm for detecting temporally ordered events [7]. The temporal detection module marks detected events as triples, (*S,E1,E2*), where S corresponds to the temporal signal that links the two events, E1 and E2, together. E1 and E2 can be verbal events (or nominalizations) such as *purchase*, or time events such as *three hours*. From this triple, temporally related SUMO predicates are generated based on hand-coded interpretation rules for the signal classes. The purpose of the interpretation rules is to define an algorithm for assigning a signal word to a SUMO predicate and define the manner in which the slots for the predicate are filled. Table 2 enumerates the signal classes, and the SUMO predicate corresponding to the interpretation rule.

Table 2
Signal class to SUMO mapping

Signal class	SUMO logic	Interpretation
⟨S sequence, E1, E2⟩	earlier(E1,E2)	E1 happened in full before E2
⟨S contain, E1, E2⟩	during(E1,E2)	E1 is contained by E2
⟨S overlap, E1, E2⟩	overlapsTemporally(E1,E2)	An interval exists that is contained by E1 and E2
⟨S open_right, E1, E2⟩	meetsTemporally(E1,E2)	E1 is the right boundary of E2, left is undefined
⟨S open_left, E1, E2⟩	meetsTemporally(E2,E1)	E2 is the right boundary of E1, left is undefined
⟨S closed, E1, E2⟩	duration(E1,E2)	E1 lasts for the duration of E2

The SUMO predicates *meets* and *overlaps* are qualified with the word "Temporally" to distinguish between other senses of overlap and meeting. This is not a concern with the other SUMO predicates, such as earlier and during, because they only apply in a temporal domain.

The appropriate SUMO predicate is chosen and the identifiers for the events associated with the signal expression serve as arguments to the SUMO predicate. Further, a date resolution module processes all underspecified and relative dates to accurately anchor these temporal references in a normalized calendar year based on the document timestamp. For instance, relative times, such as "last week", are resolved by adding the offset specified by the relative time text to the date of the document. The normalized times found within the text are expressed in this layer of the logical form in the following manner:

```
time_TMP(BeginFn(event), year, month, date, hour, minute, second)
& time_TMP(EndFn(event), year, month, date, hour, minute, second)
```

Since all temporal SUMO predicates operate on time intervals, absolute times must be represented as an interval as well. The *BeginFn* and *EndFn* functions tie the interval to a specific event and define the begin and end of the interval. The length of the interval is determined by the level of specification that is provided by the time in the text. The following example illustrates how temporal context would be represented for the sentence that was used in Examples 1 and 3:

Example 4. Bin Laden reportedly purchased anthrax a half decade ago from a supplier in North Korea. (Document Date: 8/24/2003)

```
& during_TMP(e1,t1) & time_TMP(BeginFn(t1),1998,1,1,0,0,0)
& time_TMP(EndFn(t1),1998,12,31,23,59,59)
```

The above temporal context representation replaces the syntactic representation for half a decade ago, and captures the desired information that the purchase was sometime in the year 1998. The final representation after the candidate text has been processed by the third layer is shown below:

```
Bin_NN(x1) & Laden_NN(x2) & nn_NNC(x3,x1,x2) & _human_ NE(x3)
& reportedly_RB(x4,e1) & purchase_VB(e1,x3,x5) & anthrax_ NN(x5)
&from_IN(e1,x9) & supplier_NN(x9) & in_ IN(x9,x12) & North_ NN(x10)
& Korea_NN(x11) & nn_NNC(x12,x10,x11) & _location_NE(x12)
& AGENT_SR(x3,e1) & TOPIC_SR(e1,x4) & THEME_SR(x5,e1)
& RECIPIENT_SR(x9,e1) & TEMPORAL_SR(x7,e1) & MEASURE_ SR(x6,x7)
& LOCATION_SR(x12,x9)
& during_TMP(e1,t1) & time_TMP(BeginFn(t1),1998,1,1,0,0,0)
& time_TMP(EndFn(t1),1998,12,31,23,59,59)
```

Work in progress addresses contextual dimensions such as conditionality, subjectivity, planning, spatial, and defaults as described in [3].

4. Natural language axioms

A major problem in QA is that often an answer is expressed in words different from those in the question. World knowledge is necessary to conceptually link questions to candidate answers and draws on several resources as shown in Fig. 2: (1) To increase lexical connectivity XWN is employed for the derivation of lexical chains as well as gloss axioms for the WordNet concepts. (2) Linguistic rules drive the generation of paraphrasing axioms. (3) A Semantic Relation Calculus facilitates inference over the detected semantic relations so as to derive unstated semantic relations. (4) Ontology axioms allow domain specific topics to be connected via hypernymy relations and perform a similar function to that of lexical chains. (5) Temporal reasoning axioms based on the Allen primitives are necessary to perform temporal context unification. Each of these knowledge resources will be detailed along with examples in the following subsections.

4.1. WordNet axioms

WordNet glosses contain an abundant source of world knowledge. To be useful in automated reasoning, the glosses need to be transformed into logical forms. The eXtended WordNet [16] provides such transformations of the glosses. Example 5 illustrates the logic representation of the gloss definition of concept `sport_NN#1`, which is *an active diversion requiring physical exertion and competition*.

Example 5.

```
active_JJ(x1) & diversion_NN(x1) & require_VB(e1,x1,x2) &
and_CC(x2,x3,x4) & physical_JJ(x3) & exertion_NN(x3) &
competition_NN(x4).
```

A much improved source of world knowledge is obtained when the gloss words from WordNet are semantically disambiguated, as was reported in [10]. When gloss words are semantically disambiguated, the connectivity between synsets is dramatically increased. Lexical chains can be established between synsets in different hierarchies. Lexical chains between two words are sequences of synsets related through WordNet relations. To derive such lexical chains automatically, we developed software that finds paths between any two WordNet synsets S_i and S_j up to a certain distance [10]. A lexical chain may contain (1) any of the WordNet relations; (2) a *GLOSS* relation between a synset and any of the synsets from its own gloss; and (3) a *SIM_DERIV* relations between a WordNet synset S_a containing a word W_a and a synset S_b containing a word W_b which is a morphological derivation of W_a. A *SIM_DERIV* relation exists between any word from S_a and any word from S_b.

Because there often exists more than one path between any two concepts, a path ranking algorithm is necessary in order to control the quality of lexical chain axioms input to the logic prover. When ranking the paths the following heuristics are used:

(1) Shorter paths are generally better than longer paths.
(2) Relations are not equal.
(3) Order of relations in a path is important.
(4) The type of nodes along paths is important.

Additionally, the number of paths between two concepts can be a clue of how related two concepts are; the more paths, the stronger the relation between the concepts. Examples 6 and 7 illustrate two lexical chains that link pairs of words selected from the questions and candidate answers evaluated in TREC 2004. The selected words are underlined in the examples.

Example 6.
Q28.2: When was Abercrombie & Fitch established?
Answer: Abercrombie & Fitch began life in 1892 as a high-end camping, fishing and hunting gear store in New York City.

Lexical chain 1:
establish:v#2 → SIM_DERIV → constitution:n#2 → SIM_DERIV → organize:v#5 → HYPERNYM → initiate:v#2 → SIM_DERIV → initiation:n#2 → SIM_DERIV → begin:v#1
Lexical chain 2:
establish:v#2 → HYPERNYM → open:v#2 ← GLOSS ← start:v#1 ← SYNONYM ← begin:v#2 ← GLOSS ← life:n#2

Example 7.
Q4: Which was the first movie that James Dean was in?
Answer: Dean, who began as an actor on early television dramas and the New York stage, didn't make his screen debut until 1951's "Fixed Bayonet."
Lexical chain 1:
movie:n#1 → HYPERNYM → show:n#3 → HYPERNYM → communication:n#2 → SIM_DERIV → communicate:v#1 → HYPERNYM → convey:v#3 → HYPERNYM → transfer:v#3 → HYPERNYM → move:v#2 → HYPONYM → drag:v#7 → SIM_DERIV → screen:n#1
Lexical chain 2:
first:a#4 → SIMILAR[3] → opening:a#1 → SIM_DERIV → begin:v#1 → SIM_DERIV → beginning:n#5 → HYPONYM → debut:n#1

Example 6 illustrates the semantic relationship between the verb v_1 = `"establish"` and the idiomatic expression x_1 = `"begin life"` accounted by two different lexical chains. Lexical chain 1 connects v_1 with v_2 = `"begin"` whereas lexical chain 2 connects v_1 with the noun n_2 = `"life"`. The first lexical chain corresponds to the connection of the aspectual information of v_1 and v_2 whereas lexical chain 2 corresponds to the state of existence referred by v_1 and n_2.

Example 7 illustrates the role of lexical chains in establishing a connection between the phrases P_1 = `"the first movie"` and P_2 = `"screen debut"`. The first lexical chain relates the nouns n_1 = `"movie"`, the head of P_1, and n_2 = `"screen"`, the modifier from P_2. The second lexical chain relates the adjective `"first"`, the modifier from P_1, with the noun `"debut"`, the head of P_2.

Lexical chains improve the performance of question answering systems in two ways: (1) they increase the document retrieval recall and (2) they improve the answer extraction by providing the much needed world knowledge axioms that link question keywords with answers concepts.

4.2. Linguistic axioms

Some world knowledge axioms are used by COGEX to encode several forms of linguistic pragmatics. These axioms are necessary to account for syntactic, semantic and morphological variations between the question and the answer words and also for several forms of coreference. These axioms are instantiated based on patterns found in the parse trees for the questions and candidate answers. Since most of the axioms rely on world knowledge that originates in linguistic information, we refer to them as linguistic axioms and present some examples below together with sample TREC questions that invoke them.

4.2.1. Coreference axioms for complex nominals

Several forms of coreference need to be resolved between concepts used in questions and in candidate answers. A special kind of coreference is the case of name alias, in which an entity is referred to by its full proper name, whereas the same entity may be referred to in another place by an acronym, a partial name or by an alias. This is demonstrated for the question, *"Which company created the Internet browser Mosaic?"*. The correct candidate answer refers to the Internet browser Mosaic as Mosaic. Using an abductive assumption an axiom is built such that the head noun of the complex nominal in the question implies the remaining nouns in the complex nominal:

```
all x1(mosaic_nn(x1) → internet_nn(x1) & browser_nn(x1))
```

[3] SIMILAR is a WordNet-encoded relation.

Similarly, a question may use an abbreviated form of a name, while the answer uses the full name of the entity. For example in the correct candidate answer for the question, *"When was Microsoft established?"*, Microsoft is referred to as Microsoft Corp. in the candidate answer. For this case an axiom is instantiated that assigns each component noun of the complex nominal the head argument of the this nominal.

```
all x1 x2 x3 ((microsoft_nn(x1) & corp_nn(x2) & nn_ nnc(x3,x1,x2)
→ microsoft_nn(x3) & corp_nn(x3))
```

These are considered weak axioms and any proof that uses them will be penalized with a lower proof score than those that do not.

4.2.2. Derivational morphology axioms

A candidate answer for a question will often contain concepts that are expressed by a different lexical form than is expressed in the question (e.g. nominalizations referring to the event expressed by verbs, or adjectives referring to the same property as a noun). This lexical variation poses a serious problem for the logical representation employed by COGEX. In order to verify the argument roles for a given candidate, axioms are constructed for a pair of words from the question and the answer that have a derivational link of length one in WordNet. Based on automatic analysis of morphological variation, axioms are built to test their syntactic similarity. For the question, *"Who is the sponsor of the International Criminal Court?"*, the correct candidate answer, *"... the adoption last week of a UN-sponsored agreement on the creation of an international criminal court... "*, the pair (sponsor_n, sponsor_vb) is detected and the following axiom is generated where the right-hand side is derived from the answer parse tree and the left-hand side is derived from the parse of the question:

```
all x1 x2 (_organization_ne(x1) & sponsor_vb(x3,x1,x2) →
sponsor_nn(x1) & of_in(x1,x2)
```

4.2.3. Apposition axioms

Many times, the answer to a question can be found in the appositive of one of the concepts mentioned also in the question. For instance, in the answer an apposition might describe an attribute that is sought by the question. For example, the question, *"What is Carlos the Jackal's real name?"*, has the candidate answer: *"... agents could be indicted in France for collaborating with "Carlos the Jackal", whose real name is Ilich Ramirez Sanchez"*. In this case an apposition modifies Carlos the Jackal to describe his real name. An apposition axiom is generated to link the head nouns for the two phrasal constituents by using the resolution of the relative pronoun *whose*. The apposition axiom is illustrated in Example 8.

Example 8.

```
all x1 x2 x3 x4 x5 x6 X7 X8 X9 (Carlos_nn(x1) & the_nn(x2) &
& Jackal_nn(x3) & real_JJ(x5) & name_nn(x5) & be_ v(e1,x5,x9)
& Ilich_nn(x6) & Ramirez_nn(x7) & Sanchez_nn(x8) &
& nn_nnc(x9,x6,x7,x8) → nn_ nnc(x4,x1,x2,x3) & whose_prp(x4,x9))
```

Another case when apposition axioms help was noticed for question, *"What kind of animal is an agouti?"*, which is answered by the appositive construction in *"... skinks* (*a type of lizard*)*, agoutis* (*rabbit-sized nocturnal rodents*)*... "* which instantiates the axiom from Example 9.

Example 9.

```
all x1 x4 (agouti_nn(x1) & rabbit_ sized_JJ(x4) & nocturnal_JJ(x4)
& rodent_nn(x4) → agouti_nn(x4) & rodent_(x1))
```

4.2.4. Part-whole relations for location questions

A location seeking question can have a candidate answer that identifies the sought location by referring to part of the location. For example, in the question, *"Where is Devil's Tower?"*, the answer contains the text, *"American Indians won another court battle... at Devils Tower National Monument in the northeast corner of Wyoming"*, that describes Devils Tower by specifying the part of Wyoming in which it lies. The axiom illustrated in Example 10 is built to connect Wyoming to its part.

Example 10.

```
all x1 x2 (corner_nn(x1) & of _in(x1,x2) & _ location_ne(x2)
→ _location_ne(x1))
```

4.2.5. Possessive/location axioms

Possession is a frequent shorthand for expressing location relations. To accommodate for this commonality, an axiom is constructed and loaded into the knowledge base for the question answering system when the target of a question is a location. Such a relation is detected for the question, *"Where is the Berkman Center for Internet and Society* (*located*)*?"*, which has as a candidate, *"... Jonathon Zittrain, executive director of Harvard University's Berkman Center for Internet Society"*, and results in the axiom from Example 11 being invoked and added to the axiom base:

Example 11.

```
all x1 x2 x3 e1 (_location_ne(x1) & pos(x2,x1)... → in_in(x1,x2)
|locate_vb(e1,x3,x1) & in_in(e1,x2))
```

4.3. Semantic relation calculus

TREC 2004 saw a rise in the number of questions that strayed from the traditional factoid genre, such as those seeking judgment relations like: causality, manner, purpose, and goal. Examples of these include: *"Why is "The Tale of Genji" famous?"*, *"What is Public Citizen's purpose?"*, *"What is the Muslim Brotherhood's goal?"*, *"How did James Dean die?"*, and more. Approximately 5% of all factoid questions could be classed as *judgment* seeking questions and require a semantically enhanced logic prover to accurately answer them. Semantics detected in the text include relations such as *purpose, part-whole, manner, means, cause, synonymy*. In order to verify the semantic connectivity between a question and its candidate answer, a set of rule pairing axioms for the semantic relations is required. These enable inference of unstated semantics from those detected in the candidate text. Examples of such axioms include:

```
all x1 x2 x3 (purpose_sr(x3,x2) & partwhole_sr(x3,x1) → purpose_sr(x2,x1))
all x1 x2 x3 (synonymy_sr(x1,x2) & agent_ sr(x2,x3) → agent_sr(x1,x3))
all x1 x2 (synonymy_sr(x1,x2) → synonymy_ sr(x2,x1))
all x1 e1 e2 (agent_sr(x1,e1) & purpose(e2,e1) → purpose(e1,x1))
all x1 x2 x3 (cause_sr(x1,x2) & cause_sr(x2,x3) → cause_sr(x1,x3))
all x1 x2 x3 (purpose_sr(x1,x2) & topic_sr(x2,x3) → purpose_sr(x1,x3))
all x1 x2 (accompaniment_sr(x1,x2) → accompaniment_sr(x2,x1))
all x1 x2 x3 (instrument_sr(x1,x2) & makeproduce_sr(x2,x3)
→ makeproduce_sr(x1,x3))
```

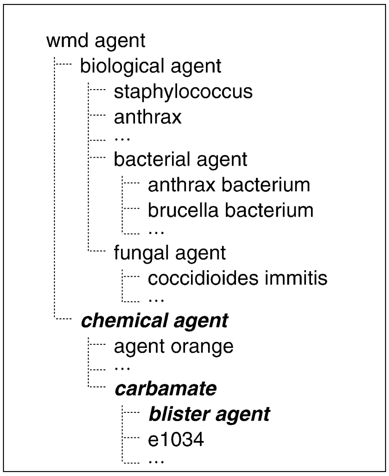

Fig. 3. Sample ontological hierarchy acquired for the CNS collection.

4.4. Ontological axioms

In a document collection, one or several topics may be covered. For each topic, an ontology can express the most relevant concepts and their typical relationships. Such ontologies supply critical domain knowledge for the QA system. Each ontology can be acquired offline from a topic-relevant collection of texts and serve as a resource for on-demand axiom construction. For a document collection developed by the Monterey Center for Nonproliferation Studies (CNS), that contains **20,000** articles about terrorist activities and weapons of mass destruction, an ontology was constructed that contains more that **611** new concepts connected through a series of relations. To build the ontology sentences that contain manually selected seed concepts are retrieved from the document collection, and are syntactically and semantically parsed. The compound nominals from the resulting parse are stored in the ontology as discovered concepts. In a final stage of ontology construction a classification is performed in which the stored noun phrases are organized into a hypernymy tree using WordNet as an upper ontology to draw connections between these discovered concepts. Examples of such concepts include weapons of mass destruction, biological and chemical agents, terrorist groups, and more. Fig. 3 shows a sample of the ontological concepts detected in the collection that are used for domain specific axioms at the time of answer extraction.

Ontological axioms are generated starting from the concepts that are encountered in the candidate answers. For the question, *"How many tons of chemical agents might have been destroyed at the Blue Grass Army Depot?"*, the correct candidate, *"The US Army is asking area residents to give their opinions on what sort of disposal plant should be built to destroy* 523 *tons of obsolete nerve and blister_agent stored at the Blue Grass Army Depot."* has a reference to a *blister agent* which triggers the invocation of the following axioms from the ontology:

```
all x1 (blister_agent(x1) → carbamate(x1))
all x1 (carbamate(x1) → chemical_agent(x1))
```

As an example of an alternate domain that is useful for corporations, a Human Resources ontology can also be constructed offline. This includes concepts for procedures, forms, employee types, policies, and more. This ontology was built by relying on a text collection of more than **15,000** documents related to company products, policies, contracts, and more. An ontology targeted at the Human Resources needs generated **1574** new concepts. A sample of the concepts extracted from the documents is shown in Fig. 4.

Such an ontology allows the question, *"What are the full-time employee rates for our medical plan?"* to be answered as *"...Semi-Monthly" Health Care Premium Rates for Premium Choice Plus-Plan* 008: $32.55 *for the employee coverage,* $61.18 *for employee-spouse coverage..."* by using the following axioms to connect the concept medical plan in the question to the concept premium choice plan in the answer:

```
all x1 (premium_choice_plus(x1) → premium_ plan(x1))
all x1 (premium_plan(x1) → medical_plan(x1))
```

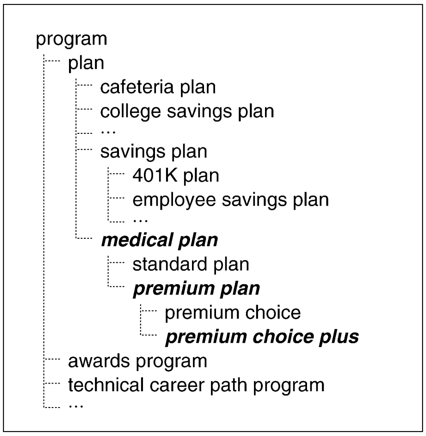

Fig. 4. Sample ontological hierarchy acquired for the domain of Human Resources.

4.5. Temporal reasoning axioms

Temporal reasoning about intervals, time points, and their relationship to events they constrain requires a temporally enhanced knowledge representation as detailed in Section 3. Additionally, a knowledge base of temporal reasoning axioms are required. These are taken from the SUMO Knowledge Base [12] that consists of axioms for a representation of time points and time intervals, Allen [1] primitives, and temporal functions. A sample of the 127 temporal reasoning axioms follows:

- Example Axiom 1: during is a transitive Allen primitive:

```
during(TIME1, TIME2) & during(TIME2, TIME3)) → during(TIME1, TIME3)
```

- Example Axiom 2: TimePoint *POINT1* is before TimePoint *POINT2*

```
Time(POINT1,Y1,M1,D1) & Time(POINT2,Y2,M2,D2) & ($LT(Y1,Y2)
|($EQ(Y1,Y2) & $LT(M1,M2))|($ EQ(Y1,Y2) & $ EQ(M1,M2) & $LT(D1,D2))
→ before(POINT1,POINT2)
```

- Example Axiom 3: Every TimeMeasure is either a TimeDuration or a TimePosition

```
ISA(TIME, TimeMeasure) → (ISA(TIME, TimeDuration)
|ISA(TIME, TimePosition)
```

- Example Axiom 4: *INTERVAL1* starts after and ends before *INTERVAL2*

```
ISA(TIME1, TimeInterval) & ISA(TIME2, TimeInterval) &
& during(TIME1,TIME2) → temporalPart(TIME1, TIME2)
```

- Example Axiom 5: The definition of equality assignment for TimePoints

```
ISA(TIME1,TimePoint) & ISA(TIME2,TimePoint) &
beforeOrEqual(TIME1,TIME2) & beforeOrEqual(TIME2,TIME1)
→ (TIME1 = TIME2)
```

5. Control strategy

5.1. Axiom partitioning mechanism

To perform abductive inference, the knowledge must be divided into (a) a set of axioms known as the Set of Support (SOS); (b) a set of Usable axioms; and (c) a set of equations known as *demodulators*, as reported in [13]. COGEX works by continually resolving an element from the SOS against one of the usable axioms. The search strategy is a form of best-first search, in which the *"weight"* of each axiom is measured and lighter axioms are preferred. Unit axioms are treated as light, thus the search can be viewed as a generalization of the unit preference strategy. Additionally, the strategy restricts the search such that a new clause is inferred if and only if one of its parent clauses come from the Set of Support (SOS).

For each candidate answer returned by the question answering engine, COGEX attempts to prove that the question is entailed by the answer. The existentially quantified logical form representation for the candidate answer is placed into the Set of Support to ensure that the prover will focus its inference generation and search on the candidate answer. In addition, the negated question logical form is universally quantified and added to the Set of Support so as to invoke the proof by contradiction.

Axioms placed in the Usable list are those described in Section 4: (1) WordNet axioms, (2) Linguistic Rule (or NLP) Axioms, (3) Semantic Relation Calculus, (4) Ontology Axioms, and (5) Temporal Reasoning Axioms. Since the SOS and Usable lists are populated with the candidate answer axiom, question axiom, and axioms derived from world knowledge, the working knowledge base only consists of approximately **20,000** axioms which virtually guarantees termination.

5.2. Inference rules

The inference rule sets are based on *hyperresolution* and *paramodulation*. Hyperresolution is an inference rule that performs multiple binary resolution steps at once, where binary resolution is an inference mechanism that looks for a positive literal in one clause and negated form of that same literal in another clause such that the two literals can be canceled, resulting in a newly inferred clause. Paramodulation introduces the notion of equality substitution so that axioms representing equality in the proof do not need to be explicitly included in the axiom lists. Additionally, similar to hyperresolution, paramodulation combines multiple substitution steps into one.

All modern theorem provers use hyperresolution and paramodulation inference rules since they allow for a more compact and efficient proof by condensing multiple steps into one, which becomes increasingly more important with the new integration of semantic and contextual reasoning requirements. Specifically, paramodulation is applied to the demodulators in the SUMO temporal axiom set and hyperresolution applies to all resolution steps. An example of a demodulator from the SUMO temporal axiom set was illustrated in Section 4.5 by the Example Axiom 5.

5.3. Backoff strategy

COGEX will continue searching for a proof until the Set of Support becomes empty, a refutation is found, or the proof score drops below a failure threshold based on the connectivity of the proof. COGEX initially assigns a perfect score to the proof it is attempting to create, and as the system is forced to employ its backoff strategy, it deducts points from the score. The backoff strategy consists of unbinding arguments in the predicates of the question, and/or removing predicates that are not critical to the meaning of the question. The backoff strategy establishes a much more robust inference engine by providing a way to handle minor logical form representation defects, resulting from parse errors, and missing world knowledge in the knowledge base.

During the search process, COGEX selects axioms from the Set of Support, generates inferences, and places any newly inferred clauses back into the SOS. The order in which COGEX chooses axioms from the SOS is dependent upon the weight assigned to each axiom. The question axiom is assigned the highest weight in the SOS so that when resolutions are attempted using this axiom, it is guaranteed that all other inferences have been made within the knowledge base. With the question as the last axiom to be processed, we are able to control the flow of the search.

If an SOS empty condition is reported, arguments to the predicate of the question that the proof failed on are incrementally relaxed. For verbs, the subject is unbound, then the object, and if the predicate still fails to unify with any other predicates in the Usable list, the predicate is flagged as dropped and no longer participates in the proof. For complex nominals, each component of the complex nominal is unbound. The arguments of prepositions, possessives, personal possessives, verbs, and coordinated conjunctions are unbound in a similar manner. For standalone predicates such as nouns, the predicate is dropped as soon as it causes a proof failure.

To force the predicates in the question to be dropped in a specific order, the predicates from the logical form of the question axiom are ordered by their semantic and syntactic importance based on their part of speech class. Answer type, semantic relation, temporal context, and named entity predicates are first, complex nominals and coordinate conjunctions are second, nouns and verbs are third, adjectives and adverbs are fourth, and prepositions, possessives, and personal pronouns are last in the logical form ordering.

The failure threshold is established such that the prover will exit in failure once all that remains of the question axiom is a series of unbound keywords. Also, if an answer type or temporal context predicate is dropped, the prover immediately exits in failure. This eliminates answers that do not fulfill the baseline requirements that the candidate contains the target type (such as human, organization, location) and that the candidate answer has the same temporal context as the question. When the prover exits in failure, a score of zero is returned for the proof and no answer is extracted.

If the prover successfully creates a proof and extracts an answer, the answer is analyzed for validity. If there are no more connecting predicates that link the expected answer type predicate to other predicates in the question axiom, then the answer is considered invalid. Connecting predicates are defined as predicates that are used to link other predicates, such as prepositions, verbs, complex nominals, coordinated conjunctions, and modifiers. If the answer extraction is flagged as invalid, the proof scores are still used to re-rank the previously extracted answers.

The justification of a proof provides a means for tracking the axioms used to entail the question by the current candidate answer. The certainty of an axiom is measured by its weight. Heavier weights indicate that an axiom is less certain, so the penalty for its use will be higher. The strength or certainty of a lexical chain, for example, is a function of the relation types in the chain and its length. Further, the class of NLP axioms that are generated to handle coreferencing are weighted heavily due to the abductive nature of their inferences. The proof score can then be computed based on the axiom weights supplied by the justification trace as well as the number of predicates that are dropped, and arguments in predicates that are unbound. The calculated proof score is assigned to the candidate answer and used in the final answer re-ranking process.

In Example 12, the expected answer is the agent of the sponsoring event whereas the object of the event is *the International Criminal Court*. In the candidate answer, the sponsoring event modifies another event, *the agreement*, whose THEME is the third event, *"the creation"* with the object *"the International Criminal Court"*. To perform the abduction, the backoff algorithm in COGEX decides to unbind the argument $x8$ of the lexicalized predicate *sponsor_VB*. In this way $x8$ becomes a free variable and allows the proof to succeed after assessing a small penalty.

Example 12. Who is the sponsor of the International Criminal Court?

```
_organization_AT(x1) & sponsor_NN(x1) & of_IN(x1,x5) &
international_NN(x2) & criminal_NN(x3) &
court_NN(x4) & nn_ NNC(x5,x2,x3,x4)
```

South Africa welcomed the adoption last week of a UN-sponsored agreement on the creation of an international criminal court.

```
south_NN(x1) & africa_NN(x2) & nn_NNC(x3,x1,x2)
& welcome_VB(e1,x3,x4) & adoption_NN(x4) & last_JJ(x5,x6)
& week_NN(x6) & of_IN(x4,x8) & sponsor_VB(e2,x7,x8)
& UN_NN(x7) & agreement_NN(x8)
& on_IN(x8,x9) & creation_NN(x9) & of_ IN(x9,x13)
& international_NN(x10) & criminal_NN(x11) & court_NN(x12)
& nn_NNC(x13,x10,x11,x12)
```

The control strategy of COGEX provides the mechanisms for assigning confidence scores to the candidate answers based on their adherence to the syntactic, semantic, and contextual properties of the question submitted to the system. These scores are then used to re-rank the candidate answers. Further, answer extraction is achieved by focusing the inference on the answer type predicates (*AT* or *SR*). If the resulting proof is well connected to the answer predicate, then the predicate is mapped back to the text from which it is was derived for the purposes of answer extraction. Finally, contextual constraints are enforced, so that incorrect answers are eliminated from the candidate answer list if the temporal context does not unify with that in the question. Examples of the system performing these functions follow in Section 6.

6. More examples

This section shows how semantic and temporal context reasoning extensions to COGEX interact with the original components of the logic prover to answer more complicated questions. Example A demonstrates the semantic relation extensions and the use of the semantic relation calculus, while Example B presents the temporal context reasoning capabilities of the system.

6.1. Example A

The following example relies heavily on the logical form representation's ability to express the deeper semantic information within the text. Rather than being concerned with information available from the Named Entity Recognizers, e.g. names of organizations, locations or persons, this question is seeking an answer whose semantic type cannot be captured by semantic entity classes. To be able to answer such kinds of questions, semantic relations between entities have also to be considered. Semantic relations allow COGEX to detect the answer type being expressed by the question and reason over semantic information found within the text.

Question: *What is the Muslim Brotherhood's goal*?
Question Semantic Relations:
PURPOSE(x, Muslim Brotherhood)
Question Axiom:
-(exists x1 x2 x3 x4 (Muslim_NN(x1) & Brotherhood_NN(x2) & nn_ NNC(x3,x1,x2) & PURPOSE_SR(x4,x3))).

Answer: *The Muslim Brotherhood, Egypt's biggest fundamentalist group established in* 1928, *advocates turning Egypt into a strict Muslim state by political means, setting itself apart from militant groups that took up arms in* 1992.
Answer Semantic Relations:
SYNONYMY(Muslim Brotherhood, Egypt's biggest fundamentalist group)
TEMPORAL(1928, establish)
AGENT(Muslim Brotherhood, advocate)
PURPOSE(turning Egypt into a strict Muslim state, advocate)
PROPERTY-ATTRIBUTE HOLDER(strict, Muslim state)
MEANS(political means, turning Egypt into a strict Muslim state)
TEMPORAL(1992, took up arms)
Answer Axiom:
exists e1 e2... x1 x2... (Muslim_NN(x1) & Brotherhood_NN(x2) & nn_ NNC(x3,x1,x2) & Egypt_NN(x4) & _s_POS(x7,x4) & biggest_JJ(x5,x7) & fundamentalist_JJ(x6,x7) & group_NN(x7) & establish_ VB(e1,x8,x7) & in_IN(e1,x9) & 1928_CD(x9) & advocate_ VB(e2,x3,x10) & turn_VB(e3,x3,x11) & Egypt_NN(x11) & into_ IN(e3,x12) & strict_JJ(x13,x14) & Muslim_NN(x14) & state_NN(x15) & nn_NNC(x16,x14,x15) & by_IN(e3,x18) & political_JJ(x17,x18) & means_NN(x18) & ··· & AGENT_SR(x3,e2) & PURPOSE_SR(e3,e2) & ···

SR Calculus Axioms:
The purpose of any action executed by an agent is also the purpose of that agent.
all x1 e1 e2 (AGENT_SR(x1,e1) & PURPOSE_SR(e2,e1) → PURPOSE_SR(e1,x1)).

Proof:
The prover begins by selecting predicates from the candidate answer that will be used to infer new clauses.
1 [] Muslim_NN($x1).
2 [] Brotherhood_NN($x2).
3 [] nn_NNC($x3,$x1,$x2).
16 [] AGENT_SR($x3,$e2).
17 [] PURPOSE_SR($e3,$e2).
Axioms from the Usable list are invoked to infer new clauses.
45 [] -AGENT_SR(x1,e1)|-PURPOSE_SR(e2,e1)|
PURPOSE_SR(e1,x1).
49 [hyper,16,17,45] PURPOSE_SR($e2,$x3).
The question is inserted last to ensure that all possible inferences have been made from the candidate answer.
58 [] -PURPOSE_SR(x4,x3)|-nn_NNC(x3,x1,x2)|-Muslim_NN(x1)
|-Brotherhood_NN(x2).
The presence of positive forms of each of the negated predicates in the question causes a refutation and terminates the proof.
59 [hyper,1,2,3,49,58] $F.

Steps 1–17 of the proof consist of predicates extracted from the candidate answer that are used later in the proof to infer new clauses and finally produce a refutation. In step 45 the prover is using the semantic relation calculus to infer that the purpose of advocating is also the purpose of the Muslim Brotherhood. In step 58 the question axiom is inserted and produces a refutation with the clauses from the candidate answer and inferred clauses. The determination of what answer to return is made by examining the trace of the proof for what clause unified with $PURPOSE\_SR(x4,x3)$.

The following is a candidate answer initially returned by the question answering engine with a higher confidence than the candidate that has been presented in the proof:

The fundamentalist Muslim Brotherhood, a major force in Jordan's political arena holding 22 *seats in the outgoing lower house, and nine other left-leaning opposition parties boycotted the elections in protest of the government's tightened grip on press and publication.*

However, the semantic parser does not find a purpose relation within the text of this candidate answer which forces the prover to drop the answer type predicate in attempting to build a proof for this candidate answer. The drop of the answer type predicate signals that this candidate does not contain a valid answer to the question and the candidate is discarded.

Selected Answer: *turning Egypt into a strict Muslim state by political means*

6.2. Example B

In this example, the question is looking for a factoid answer, with a temporal context constraint. Reasoning over the temporal context of the candidate answers is necessary to ensure that the returned answer matches the given temporal constraints of the question.

Question: *Of what country was Syria a part in* 1930?
Question Axiom:
-(exists x1 x2 x3 x4 (_country_AT(x1) & Syria_NN(x2) & _ country_ NE(x2)
& part_NN(x3) & of_IN(x3,x1) & overlapsTemporally_TMP(x3,x4)
& time_TMP(BeginFn(x4),1930,1,1,0,0,0)
& time_TMP(EndFn(x4),1930,12,31,23,59,59))).

Answer: *But France has a different relationship with Syria, partly because it was once a French-mandated territory, from* 1920–1946.

Answer Axiom:
exists e1 e2 x1 x2 x3 x4 x9 x10 (France_NN(x1) & _country_NE(x1)
& have_VB(e1,x1,x3) & different_JJ(x2,x3) & relationship_NN(x3)
& with_IN(x3,x4) & Syria_NN(x4) & _country_NE(x4) & ···
& French_JJ(x5,x4) & mandated_JJ(x6,x4) & territory_NN(x4)
& during_TMP(x4,x6) & time_TMP(BeginFn(x6),1920,1,1,0,0,0)
& time_TMP(EndFn(x6),1946,12,31,23,59,59) & time_TMP(BeginFn(x7),1930,1,1,0,0,0)
& time_TMP(EndFn(x7),1930,12,31,23,59,59).

Lexical Chains Axioms:
all x1 part_NN(x1) → territory_NN(x1)

Temporal Axioms:
Two time intervals overlapsTemporally if and only if each one is a temporalPart of the other.
all INTERVAL1 INTERVAL2 (ISA(INTERVAL1, TimeInterval)
& ISA(INTERVAL2, TimeInterval)
& overlapsTemporally_TMP(INTERVAL1, INTERVAL2) ↔
& temporalPart_TMP(INTERVAL2, INTERVAL1)
& temporalPart_TMP(INTERVAL1, INTERVAL2)

Interval1 is temporalPart of Interval2 if Interval1 is during Interval2.
all TIME1 TIME2 (ISA(TIME1, TimeInterval) & ISA(TIME2, TimeInterval)
& during_TMP(TIME1, TIME2) → temporalPart_ TMP(TIME1, TIME2)).

One TimePoint is before another if it occurs before the other one in the calendar. Checks all possible cases in which one time could be earlier than another.
all POINT1 YEAR1... POINT2 YEAR2...
(time_TMP(POINT1,YEAR1,MONTH1,DAY1,HOUR1,MINUTE1,SECOND1)
& time_TMP(POINT2,YEAR2,MONTH2,DAY2,HOUR2,MINUTE2,SECOND2)
& ($LT(YEAR1,YEAR2) | ($EQ(YEAR1,YEAR2) & $LT(MONTH1,MONTH2))
|... | ($EQ(YEAR1,YEAR2) & $EQ(MONTH1,MONTH2) & $ EQ(DAY1,DAY2)
& $EQ(HOUR1,HOUR2) & $EQ(MINUTE1,MINUTE2) & $ LT(SECOND1,SECOND2)))
→ before_TMP(POINT1,POINT2)).

Interval1 occurs during Interval2 if Interval1 starts after Interval2 and Interval1 ends before Interval2.
all INTERVAL1 INTERVAL2 (before_TMP(EndFn(INTERVAL1), EndFn(INTERVAL2))
& before_TMP(BeginFn(INTERVAL2), BeginFn(INTERVAL1))
→ during_TMP(INTERVAL1,INTERVAL2)).

Proof:
The prover begins by selecting predicates from the candidate answer that will be used to infer new clauses.
8 [] Syria_NN($x4).
9 [] _country_NE($x4).
17 [] territory_NN($x4).
18 [] during_TMP($x4,$x6).
19 [] time_TMP(BeginFn($x6),1920,1,1,0,0,0).
20 [] time_TMP(EndFn($x6),1946,12,31,23,59,59).
21 [] time_TMP(BeginFn($x7),1930,1,1,0,0,0).
22 [] time_TMP(EndFn($x7),1930,12,31,23,59,59).
The prover uses SUMO temporal axioms from the Usable list to infer new knowledge about the times provided in the candidate answer. Inference path is not included for brevity.
989 [hyper,981,822,765,979] overlapsTemporally_TMP($f4($ x6,$x7),$ x7).

The question is inserted last to determine if there is a refutation after all inference generation is complete.
1008 [] -_country_AT(x1)|-overlapsTemporally_TMP(x3,x4)
|-time_TMP(BeginFn(x4),1930,1,1,0,0,0)
|-time_TMP(EndFn(x4),1930,12,31,23,59,59)|-_country_ NE(x2)
|-Syria_NN(x2)|-part_NN(x3)|-of_IN(x3,x1)
The detection of a refutation signals the prover to stop and return the proof.
1009 [hyper,8,9,17,989,21,22,1008] $F.

The given candidate answer contains a date range rather than the specific date requested by the question. Through the use of the SUMO temporal axioms, we are able to infer that the year 1930 occurs during the range 1920–1946, and thus all information in the context of 1920–1946 also exists within the context of 1930 and COGEX is able to extract the contextually constrained answer.

Proofs for answers which do not have the correct context will fail, resulting in the candidate answer being pruned from the answer list. This in turn promotes all candidates below. As an example, the context resolution module will fail to find a proof for the first candidate answer returned by the answer processing module (without temporal reasoning), whose document date is 1998:

The United States did not punish Israel when it occupied territories of some Arab countries such as Palestine and Syria, and refused to comply with relevant UN resolutions on the Middle East issues.

Since the document date of the candidate does not meet the temporal constraints of the question, the incorrect answer is filtered from the list and the correct answer, candidate 5, is rightly promoted to position 1.

Selected answer: *France*

7. Results

COGEX was implemented and integrated into a state-of-the-art question answering system that participated in TREC 2004 evaluations. Attempts of abductive proofs were done for all 230 factoid questions, but when the proof failed, the QA system resorted to other answer extraction methods that were part of the system before the prover. These methods include statistical approaches based on linguistic features of the text for which a confidence score is computed. Thus, some questions are answered by the QA system without the prover, some only by the prover, and some by both the non-prover system and the prover.

Failures of the abductive proofs are due mainly to the insufficient world knowledge, which was described in Section 4. Additionally, missing NLP axioms and incorrect lexical chains prevented the prover from completing an inference. So as to make COGEX robust the backoff strategy was employed. Abductive proofs were considered failed when we had to use the backoff strategy degrading to keyword matching as described in Section 5. The derivation of the knowledge was not always correct. For the three layers of the knowledge representation: (1) Syntactic, (2) Semantic, and (3) Temporal Context, Table 3 summarizes the accuracy of the logical representations based on each form of knowledge.

To compute the accuracy of the syntactic parser and of the first layer of the logic transformations, we have manually corrected the parse tree and the logical forms for all 230 questions from TREC 2004 and their first 20 candidate answers. The accuracy of the semantic relations was also tested against manual annotations on the whole set of questions and each of the first 20 candidate answers. Finally, for the detection of the temporal context, we have

Table 3
Accuracy of knowledge representations

Layer	Component	Accuracy (%)
1	Syntactic parser and logical form transformations	86
2	POLARIS semantic relations	40
3	Temporal context detection	51

Table 4
Contribution of COGEX and supporting components

Module	Rate increase (%)
Enhanced logic prover	12
Textual semantics	4
Lexical chains	19

created a gold standard on the same data. On the temporal context gold standard, we have measured the accuracy of recognizing temporal signals correctly (by disambiguating them when they could have other linguistic functions); the accuracy of attaching temporal expressions correctly to events and the accuracy of detecting the temporal context. Temporal signals were detected correctly in 71.4% of the cases, temporal intervals were attached correctly to events in 49.1% of the cases and the temporal context was recognized correctly in 51% of the cases. To create the gold standard, three linguists have annotated the data, with an inter-annotator agreement with a KAPPA score of 82%.

A separate set of 200 time-dependent questions was used as a benchmark for measuring the contribution of temporal context to the question answering system. The questions were derived from a 300 MB sub-collection of the TREC corpus that included a sample of XIN, APW, and NYT articles. When comparing the answer accuracy when the temporal context is enabled with the answers obtained without this form of knowledge available, and enhancement of 22% was obtained.

Table 4 shows the contribution of COGEX and its supporting technologies to the success of the question answering system at the TREC 2004 Question Answering Track test set reported at [9].

When COGEX was using semantic relations and lexical chains, it enhanced the accuracy of answers with 12% over the case when such resources were not available. Textual semantics alone, expressed by the recognition of semantic relations provided by POLARIS contributed 4% to the overall accuracy. Lexical chains provide (1) means for adding XWN axioms; and (2) lexico-semantic information for generating keyword expansions. Thus the 19% contribution of lexical chains includes their support of knowledge for the logic prover, as well as in the selection of alternations at the Information Retrieval level (this is why it exceeds the 12% contribution made by the logic prover as a whole).

The overall results of the question answering system at TREC 2004 including the enhanced logic prover with textual semantics and lexical chains was reported with an f-measure of 77% for the 230 factoid questions.

8. Conclusions

The desire to answer more complex questions is tightly coupled with the need to introduce innovative automated reasoning methods in question answering. Recent TREC QA results have clearly demonstrated that deeper text understanding methods and inferences are necessary to push QA technology further.

This paper has introduced a complex logic prover especially designed for textual QA. A main innovation is the multi-layer logic representation of text. In addition to syntactically based logic forms, we have enhanced the logic representation with a rich set of semantic relations as well as temporal contexts. This enriched representation allows us to capture essential relations between events.

Another innovation is the multitude of logical axioms provided to the prover, some derived at run time, some generated apriori. World knowledge is provided via lexical chains formed on eXtended WordNet. Links are established between concepts from questions and concepts from answer texts as needed. A large set of different types of linguistic axioms are introduced. These are derived from the parse of the candidate answers and the questions and generated automatically before the proof. Generated manually were the semantic calculus relations that link basic semantic relations. When reasoning on fixed domains, specially designed ontologies can significantly boost the accuracy of QA systems. Domain ontologies represent an important source of axioms. Finally, there are temporal axioms to assist with temporal reasoning aspects.

The logic prover has been integrated into a QA system and tested on the last several TREC QA experiments. Results on TREC QA 2004 show that the logic prover contributes to an overall increase of accuracy by 12%. This percentage may become more significant as the complexity of questions increases.

Improved inference mechanisms, semantic relation detection, as well as improved world knowledge extraction schemes are required to handle higher levels of question complexity.

References

[1] J. Allen, Time and time again: the many ways to represent time, Internat. J. Intelligent Syst. 4 (6) (1991) 341–355.
[2] D. Bixler, D. Moldovan, A. Fowler, Using knowledge extraction and maintenance techniques to enhance analytical performance, in: Proceedings of the 2005 International Conference on Intelligence Analysis, Washington, DC, 2005.
[3] C. Clark, D. Hodges, J. Stephan, D. Moldovan, Moving QA towards reading comprehension using context and default reasoning, in: Papers from the AAAI 2005 Workshop on Inference for Textual Question Answering, AAAI Press, 2005, pp. 6–12.
[4] M. Collins, Head-driven statistical models for natural language parsing, Computational Linguistics 29 (4) (2003) 589–637.
[5] D. Davidson, The logical form of action sentences, in: N. Rescher (Ed.), The Logic of Decision and Action, University of Pittsburgh Press, 1967, pp. 81–95.
[6] J. Hobbs, M. Stickel, P. Martin, Interpretation as abduction, Artificial Intelligence 63 (1993) 69–142.
[7] D. Moldovan, C. Clark, S. Harabagiu. Temporal context representation and reasoning, in: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-2005), Edinburgh, Scotland, 2005, pp. 1099–1104.
[8] D. Moldovan, C. Clark, S. Harabagiu, S. Maiorano. 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), Edmonton, Alberta, Canada, 2003, pp. 87–93.
[9] D. Moldovan, S. Harabagiu, C. Clark, M. Bowden, J. Lehman, J. Williams, Experiments and analysis of LCC's two QA systems over TREC 2004, in: Proceedings of TREC 2004 Conference, http://trec.nist.gov, 2004.
[10] D. Moldovan, A. Novischi, Lexical chains for question answering, in: Proceedings of 19th International Conference on Computational Linguistics (COLING-2002), Taipei, Taiwan, 2002, pp. 674–680.
[11] D. Moldovan, V. Rus, Logic form transformation of WordNet and its applicability to question answering, in: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001), Toulouse, France, 2001, pp. 394–401.
[12] I. Niles, A. Pease, Towards a standard upper ontology, in: Proceedings of the 2nd International Conference on Formal Ontology in Information Systems, Ogunquit, Maine, 2001.
[13] S. Russell, P. Norvig, Artificial Intelligence—A Modern Approach, Prentice-Hall, 2003.
[14] L. Wos, Automated Reasoning, 33 Basic Research Problems, Prentice-Hall, 1998.
[15] E. Voorhees, Overview of the TREC 2004 question answering track, in: Proceedings of TREC 2004, http://trec.nist.gov, 2004.
[16] eXtended WordNet, http://xwn.hlt.utdallas.edu.