Question Answering Based on Semantic Structures

Srini Narayanan
International Computer Science Institute
1947 Center Street
Berkeley, CA 94704
snarayan@icsi.berkeley.edu

Sanda Harabagiu
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75083
sanda@hlt.utdallas.edu

Abstract

The ability to answer complex questions posed in Natural Language depends on (1) the depth of the available semantic representations and (2) the inferential mechanisms they support. In this paper we describe a QA architecture where questions are analyzed and candidate answers generated by 1) identifying predicate argument structures and semantic frames from the input and 2) performing structured probabilistic inference using the extracted relations in the context of a domain and scenario model. A novel aspect of our system is a scalable and expressive representation of actions and events based on Coordinated Probabilistic Relational Models (CPRM). In this paper we report on the ability of the implemented system to perform several forms of probabilistic and temporal inferences to extract answers to complex questions. The results indicate enhanced accuracy over current state-of-the-art Q/A systems.

1 Introduction

Current Question Answering (QA) systems extract answers from large text collections by (1) classifying the answer type they expect; (2) using question keywords or patterns associated with questions to identify candidate answer passages; and (3) ranking the candidate answers to decide which passage contains the exact answer. Few systems also justify the answer by performing abduction in first-order predicate logic (Moldovan et al., 2003). This paradigm is limited by the assumption that the answer can be found because it uses the question words. Although this may happen sometimes, this assumption does not cover the common case where an informative answer is missed because its identification requires more sophisticated processing than named entity recognition and the identification of an answer type. Therefore we argue that access to rich semantic structures derived from domain models as well as from questions and answers enables the retrieval of more accurate answers as well as inference processes that explain the validity and contextual coverage of answers.

We consider several stages of deeper semantic processing for answering complex questions. A first step in this direction is the incorporation of "semantic parsers" that recognize predicate-argument structures or semantic frames when processing both questions and documents. A second step is the identification of a topic model that contributes to the interpretation of the question and generates a possible index in an off-line battery of ontologies. The third step consists of building a scalable and expressive model of actions and events which allows the sophisticated reasoning imposed by QA within complex scenarios. We embed the three forms of semantic representations and the inference they enable in a novel, flexible QA architecture that allows us to evaluate the impact of each new form of semantic information on the accuracy of answering complex questions.

The remainder of this paper is organized as follows. In Section 2 we present the semantic knowledge that we extract from questions and answers as well as our novel QA architecture. In Section 3 we detail our model of event structure. Section 4 presents the types of inference that are associated with the event structure whereas Section 5 details the results of the evaluations. Section 6 summarizes the conclusions.

2 Semantic Structures for QA

Processing complex questions involves the identification of several forms of complex semantic structures. First we need to recognize the answer type that is expected, which is a rich semantic structure, in the case of a complex question, or a mere concept in the case of a factual question. Second, we need to identify the question class or the question pattern. Third, in the case of a complex question, which is part of a scenario, we need to model the topic of the scenario.

At least three forms of information are needed for detecting the answer type: (1) question classes and named entity classes; (2) syntactic dependency information; and (3) semantic information taking the form of (i) predicate-argument structures or semantic frames and (ii) the representation of the question topic. The following question illustrated the significance of each of the three forms of information:
Q1: "What stimulated India's missile program ?"
The question stem *"what"* is ambiguous, as multiple answer types could be associated with a question pattern *"What stimulated X?"*. To find candidate answers, the recognition of *"India"* and other related named entities, e.g. *"Indian"*, as well as the name of the *"Prithvi missile"* or its related program is important. To better process question *Q1*, the syntac-

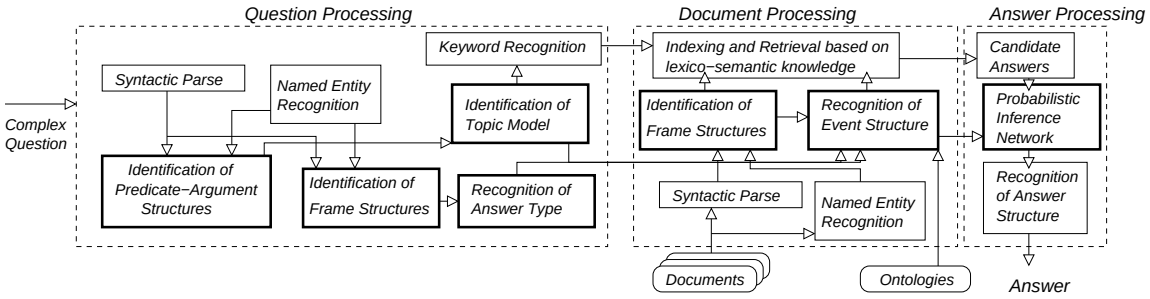

Figure 1: QA architecture based on several forms of semantic structures.

tic dependencies enable the recognition of predicate-argument structures. The predicate-argument structure of *Q1* is:

PREDICATE: Stimulate
ARG0 (role = agent) ANSWER (part 1)
ARG1 (role = thing increasing): India's missile progam
ARG2 (role = instrument) : ANSWER (part 2)

The predicate-argument structure was built based on the definitions of the PropBank project (Kingsbury et al., 2002). The structure indicates that the answer may have the role of *agent* or even the role of *instrument*. When additional information from FrameNet (Baker et al., 1998) is used, we find that the answer may have four other semantic roles, derived as frame elements of two distinct frames:

FRAME: Stimulate
Frame element CIRCUMSTANCES: ANSWER (part 1)
Frame Element EXPERIENCER: India's missile progam
Frame Element STIMULUS : ANSWER (part 2)

FRAME: Subject_stimulus
Frame element CIRCUMSTANCES: ANSWER (part 3)
Frame element COMPARISON SET: ANSWER (part 4)
Frame element EXPERIENCER: India's missile program
Frame element PARAMETER: nuclear proliferation

None of these semantic roles are fully specified. To interpret the semantic information constrained by the thematic roles, we need to also have access to a *topic model* of the scenario in which the question is being asked. For example. for the question: *Q2: "How can a biological weapons program be detected ?"* the topic model consists of (a) a set of typical relations between topic concepts; and (b) a set of possible paths of actions. As it is illustrated in Figure 1, the identification of (a) predicate-argument structures and (b) semantic frames contributes to the recognition of the expected answer as well as to the formation of the topic model.

Question *Q2* is mapped into its *pattern* and its *focus*, which has the role of the topic of the question. The document passages retrieved for the specific topic can be used to extract the most relevant topic relations with the method detailed in Section 2. The event structure, detailed in Section 3 enables the recognition of possible paths of action in the format of chains between the events lexicalized in the topic relations. The set of possible paths of actions generate different interpretations of the questions focus, which facilitate the mapping of the original predicate-argument structure in other predicate structures in which the semantic type of the answer has less ambiguity. Figure 2 illustrates the mapping of the predicate *detect* in the predicates *produce* and *acquire* that can be extracted in parallel. This mapping enabled by the topic model corresponds to the decomposition of the original complex questions into a set of less complex questions.

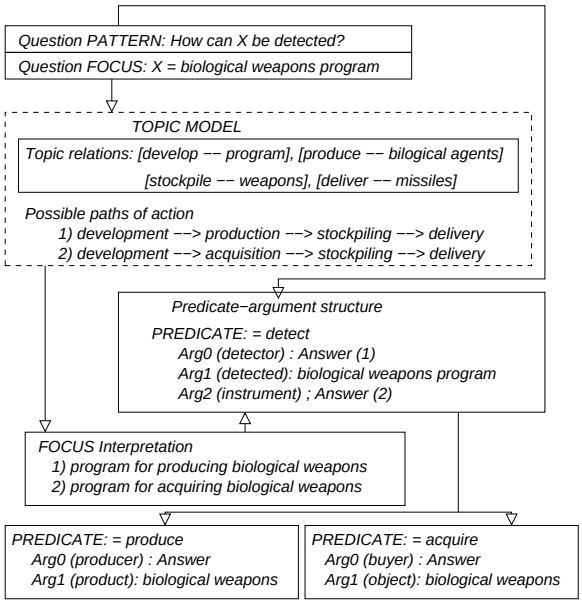

Figure 2: Question processing based on topic models.

Because the model for event structure has the capability of (1) incorporating domain knowledge in OWL-based representations[1]; and (2) performs several forms on inference on this knowledge, it can be used to extract candidate answers from the passages retrieved by the topic relations. The QA architecture that takes advantage of these semantic structures and the inference they enable is illustrated in Figure 1. The syntactic parse is produced by the Collins parser (Collins, 1996), the Named Entity Recognizer (NER) is an implementation of the NER reported in (Bikel et al., 1999) whereas the

[1] OWL is a markup language for the semantic web (http://www.semanticweb.org) which allows for the specification of ontologies and the semantic markup of documents in an xml format on the web

predicate-argument structures and the frame elements are parsed with the techniques described in Section 2.1. All these four operations are performed both in the question processing module and in the document processing module. The topic model, generated at question processing, has three roles: (1) it provides an index for the event structures to find ontological information; (2) it refines the definition of the answer type; and (3) it improves the quality of the retrieved answer passages because it makes topic-relevant relations available. The derivation of the topic model is based on the predicate argument structures derived from the question, whereas the answer type and the event structures rely on the frame semantics available from questions and relevant passages. Because PropBank has higher lexical coverage than FrameNet, whenever the semantic frames cannot be recognized, the QA system falls back on the predicate-argument structure identified in questions and documents. This back-off mechanism enables (1) indexing and retrieving relevant passages from document collections by using lexico-semantic knowledge; and (2) the recognition of the event structure referred by questions and answers. The Probabilistic Inference Networks (PINs) described in Section 5.2 select the answer structures and identify the answers to be returned.

2.1 Predicate and Frame Structures

Proposition Bank or PropBank is a one million word corpus annotated with predicate-argument structures, which were described in (Kingsbury et al., 2002). The corpus consists of the Penn Treebank 2 Wall Street Journal texts (*www.cis.upenn.edu/~treebank*). For every given predicate lexicalized by a verb, a set of arguments sequentially numbered from Arg0 to Arg5 were annotated. The general procedure was to select for each verb the roles that seem to occur most frequently and use these roles as mnemonics for the predicate arguments. Generally, Arg0 would stand for *agent*, Arg1 for *direct object* or *theme* whereas Arg2 represents *indirect object*, *benefactive* or *instrument*, but mnemonics tend to be verb specific. For example, the argument structure for the verb-predicate *steal* has Arg0:*agent*, Arg1:*theme*, Arg2:*source*, and Arg3:*beneficiary*. Additionally, the argument may include functional tags from Treebank, e.g. ArgM-DIR indicates a directional, ArgM-LOC indicates a locative, and ArgM-TMP stands for a temporal.

The FrameNet project annotates roles defined for each semantic frame. A frame is a schematic representation of situations involving various participants, props and other conceptual roles, all called Frame Elements (FEs). For example the frame THEFT describes situations in which a PERPETRATOR takes GOODS that belong to the VICTIM . The MEANS by which this is accomplished may be also expressed. The British National Corpus is used for annotations.

(Gildea and Jurafsky, 2002) and (Gildea and Palmer, 2002) report on the same statistical method that labels argument roles from PropBank or FEs from FrameNet on any English sentence that is syntactically parsed. Their method consists of two classification tasks: (1) identifying the parse tree constituents corresponding to the predicate arguments or the FEs; and (2) recognizing the role of the argument or FE. They have introduced seven features that (a) were used for training both classifiers; and (b) worked both for PropBank and FrameNet. In (Surdeanu *et al.*, 2003) seven additional features were proposed, that enhanced the performance of the classifiers. By using both sets of features in our implementation using the SVM-light software available from *http://svmlight.joachims.org*, we automatically transformed the Question *Q3* into the predicate-argument structure *PAS(Q3)* and the Frame Structure *FS(Q3)*:

Q3: What kind of nuclear materials were stolen from the Russian navy ?

PAS(Q3): What [Arg1: kind of nuclear materials] were [Predicate: stolen] [Arg2: from the Russian navy]?

FS(Q3): What [GOODS: kind of nuclear materials] were [target−Predicate: stolen] [VICTIM: from the Russian navy]?

The expected answer, as predicted by *PAS(Q3)* is the Arg1 of the predicate *'steal'*, when the Arg2 has the head *'Russian navy'*. Additionally, the answer needs to be in the same semantic class as *'nuclear materials'*. The FEs from *FS(Q3)* show that we should search for an FE with the role GOODS whenever we find a target word of the frame STEAL. The paragraphs containing candidate answers are parsed similarly. For example, the correct answer *A(Q3)* is transformed into the predicate-argument structure *PAS((A(Q3))* and the Frame Structure *FS(A(Q3))*:

A(Q3): Russia's Pacific Fleet has also fallen prey to nuclear theft; in 1/96, approximately 7 kg of HEU was reportedly stolen from a naval base in Sovetskaya Gavan .

PAS(A(Q3)): [Arg1(P1) Russia's Pacific Fleet] has [ArgM−DIS(P1) also] [Predicate(P1): fallen] [Arg1(P1): prey to nuclear theft]; [ArgM−TMP(P2): in 1/96], [Arg1(P2): approximately 7 kg of HEU] was [ArgM−ADV(P2) reportedly] [Predicate(P2): stolen] [Arg2(P2): from a naval base] [Arg3(P2): in Sovetskaya Gavan]

FS(A(Q3)): [VICTIM: Russia's Pacific Fleet] has also fallen prey to [GOODS: nuclear] [target−Predicate(P1): theft]; in 1/96, [GOODS(P2): approximately 7 kg of HEU] was reportedly [target−Predicate(P2): stolen] [VICTIM(P2): from a naval base] [SOURCE(P2): in Sovetskaya Gavan]

In *PAS(A(Q3))* we identify two predicates, indexed P1 and P2. P2 is lexicalized with the same word-lemma as the predicate from *Q3*, thus its Arg1(P2): *'approximately 7 kg of HEU'* provides the exact answer. It is to be noted that its Arg2(P2) is *'a naval base'* which has a meronym relation with the previously mentioned NP *'Russia's Pacific Fleet'*, a meronym of *'Russian navy'*. The same meronymy needs to be resolved between the FE VICTIM of *stolen'* and the FE of VICTIM of *theft* in the *FS(A(Q3))*. In the second case the meronymy is identified since the second frame identifies an event which is an example of the event identified by the first frame.

2.2 Topic Models

In question processing two objects need to be identified: (1) the *expected answer type* and (2) the *focus* of the question. For example, in question *Q2: How can a biological weapons program be detected ?"*, the expected answer type is MANNER(of detection) and the focus is *'biological weapons program'*. When processing complex question the role of the focus becomes more important, since it guides the recognition of the topic model associated with the question, which in turn enables the identification of partial answers and the relations between them. To identify the expected answer type, we can rely on the question stem (e.g. *"How"*) and its associated semantic classes or we can determine the answer type by using a combination of features associated with the question stem and one or more of the question words. For example, the question *"How long does it take to produce weapons of mass destruction ?"* has the answer type TIME_SPAN determined by the combination of the stem *'how'* and the adverb *'long'*. This information is much more relevant for identifying the expected answer type than the fact that the predicate *'take'* has ArgM=*'how long'* and Arg2=*'produce weapons of mass destruction'*, which represents the focus of the question.

Complex questions rely on topic models for finding the answer since it is unlikely that in a text collection the exact answer to a complex questions can be found, but it is more likely that partial answers can be detected, and then they may be combined for generating the most informative answer. We used an incremental topic representation that was introduced in (Harabagiu, 2004). Information about a topic is modeled through two incremental enhancements of the topic signatures introduced in (Lin and Hovy, 2000). The first enhancement determines a set of seed relations. The methodology considers:
(1) *filtering out outliers* of the terms identified as relevant with the statistical method based on likelihood ratio reported in (Lin and Hovy, 2000)
(2) *morphological expansion* of the nouns and verbs from the topic signature;
(3) *semantic normalization* through the NER and an off-line ontology of 22,000 words; and
(4) *selection of the topic seeds* with the same likelihood ratio method applied for acquiring the topic concepts. The seeds are the most relevant [*Verb-Noun*] pairs which have a predicate-argument relationship.
For question *Q3* words like *'say'*, *'have'* or *'identify'* were filtered out, living words like *'weapons'*, *'sarin'* and *'produce'* as the most relevant topic concepts. The morphological expansion added words like *'production'* whereas the semantic normalization unified *'Russian'* and *'Iraqi'* into NATIONALITY and *' bomb'* or *'building'* into ARTIFACT.

The seed relations that was selected for question *Q3* is [*develop - program*]. The relation is further used to produce a corpus of paragraphs related to the corpus, from which new topic relations can be extracted. Two types of relations are targeted: (1) syntax-based relations (e.g. *Verb - Subject*, *Verb - Object* and *Verb - Prepositional Attachment*) and (2) salience-based relations, which model long-dependency relations to a seed concept. The relations are ranked based on a methodology introduced in (Riloff, 1996) each relation is ranked based on its *Relevance-Rate* and its *Frequency*. The *Frequency* of an extracted relation counts the number of times the relation is identified in the relevant paragraphs. The *Relevance-Rate* = *Frequency* / *Count*, where *Count* measures the number of times an extracted relation is recognized in any paragraph considered.

This ranking allows us to select a new topic relation, and to resume the topic modeling procedure, this time on a new corpus generated by the most recently discovered relation. We stop the discovery process when we have identified 20 topic relations. Some of the topic relations discovered for question *Q2* are illustrated in Figure 2.

The second enhancement of topic representations reported in (Harabagiu, 2004) considers the notion of *topic theme* that associates clusters of topic relation with text segments. The segmentation is produced by the TEXTTILING algorithm (Hearst, 1997). The nominalization of the verb corresponding to the most relevant topic relation in a segment is considered to be linked to the nominalization from the following topic-relevant segment. Such segments are called *themes* and the chains of nominalizations represent possible paths of actions. Two such paths are represented in Figure 2

3 From Semantic Extraction to Inference for QA

Semantic extraction allows us to identify predications in the input text. For processing complex questions we further identify the question class or the question pattern as well as relevant parts of the scenario which we refer to as the topic model. A significant gap remains between a) the unstructured and intuitively chosen tag sets used in FrameNet or PropBank and the relation names and clusters in the topic model and b) a formal characterization of the interrelated events, actions, states and relations holding among them. The explicit representation of such frame semantic and event structure information is needed for for the potential use of such resources for question answering.

In previous work (Chang *et al.*, 2002), we bridged the gap by defining a formalism that unpacks the shorthand of frames into structured event representations. This allows annotated FrameNet data to parameterize event simulations (Narayanan, 1999) that produce fine-grained, context-sensitive inferences. We have extended this work to further incorporate the topic model and theme described earlier. Currently, the list of extracted *predicate-argument*

structures, *the topic model* and *the answer type* predicate are used to index into a set of parameterized event representations instantiated to specific values based on the extracted predicate-argument bindings (see Figure 3). The *answer type* predicate translates to a specific inference procedure.

Figure 3 (middle) shows the representation of extracted predicate-argument bindings in our parameterized event formalism, Embodied Construction Grammar (ECG)(Bergen and Chang, in press), that maps annotations to event simulations. ECG is a constraint-based formalism similar in many respects to other unification based linguistic formalisms such as HPSG or LFG (features, roles, constraints, simple and complex slots, subcasing, and a *self* reference). ECG differs from other linguistically motivated proposals in 1) the use of an *evokes* relation that models the *priming* of a background schema (role inheritance is lazy and explicitly specified) and 2) the complex network of conceptual schemas in ECG are designed to map utterances to mental *simulations* in context to produce a rich set of inferences. It is thus ideally suited for our current goal of translating frames to conceptual representations. Figure 3 (middle left) shows the THEFT schema instantiated to the bindings extracted from the answer passage. Figure 3 (middle right) shows the schema instance enhanced with inferentially derived additional bindings.

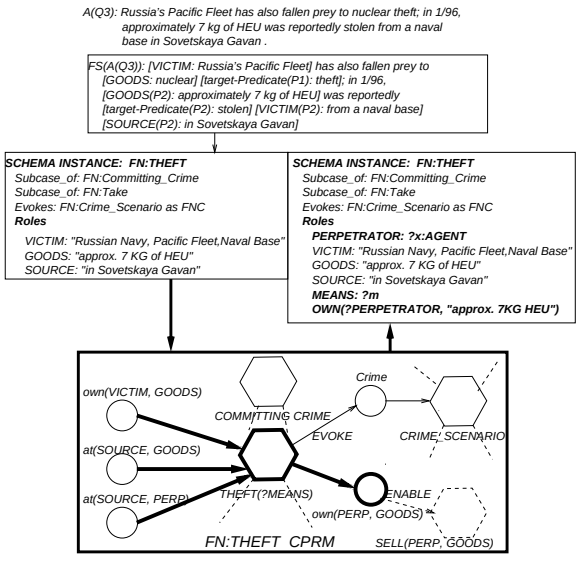

Figure 3: From Semantic Extraction to Inference

Figure 3 (bottom) shows a fragment of the event simulation for the THEFT frame (all the information in this simulation is generated from information in the FrameNet database). Preconditions and world states that obtain before the event include a) VICTIM owns the GOODS, b) the PERPETRATOR is at the SOURCE and c) the GOODS are at the SOURCE. The THEFT event can be a simple transition or can zoom-in to a complex event with phases (such as *start*, *ongoing*, *finish*, *interrupt*, *cancel*, *resume*, *stop*). Complex events can include monitoring and detection conditions as well as resource production, consumption and locking. The *completion* of THEFT results in a) the PERPETRATOR owning the GOODS and b) the evocation of the CRIME SCENARIO schema, which gets simulated if other conditions obtain (such as AUTHORITIES notice the CRIME). The effect of one action may probabilistically enable, disable, interrupt, or terminate other possible events (such as OWN provides evidence for the future SELL event). The result of running the inference process for this example results in 1) identification of relevant unbound roles (PERPETRATOR and MEANS) and 2) highly probable new assertions and bindings (the perpetrator owns the goods after the theft). 1) suggests new scenario-based query expansion strategies and is a result of updating the state variables after the new evidence (extracted predicate-arguments) is asserted as this process is called **filtering**. 2) is the resultant state after executing the action and is computed by a) executing the action and identifying **reachable states** and b) updating the state after the action to find the **Maximum A Posteriori (MAP)** probabilities. These procedures are amongst the important inference methods for structured stochastic processes and are directly supported by our implementation.

Technically, the event structure implementation uses a factorized model of *states* based on Temporally Extended (aka Dynamic) Probabilistic Relational Models (Murphy, 2002; Pfeffer, 2000; Getoor *et al.*, 2001) that enable a variety of inferences that update and revise the state variables (forward and backward in time). Central to the representation of actions and events is an event model called **executing schemas** (or **x-schemas**), motivated by research in both sensorimotor control and cognitive semantics (Narayanan, 1997). X-schemas are active structures based on Stochastic Petri Nets (Ciardo *et al.*, 1994) that cleanly capture sequentiality, concurrency and event-based asynchronous control[2]. Our implementation integrates the PRM based state model with the x-schema based action model and is called Coordinated Probabilistic Relational Models or CPRM. Our CPRM implementation, KarmaSIM, is linked to existing linguistic resources (FrameNet and WordNet) and to ontologies on the semantic web. To address the vexing issue of domain specific Knowledge Acquisition (KA), in past work we have constructed automatic translators from OWL-based event and process ontologies (such as OWL-S) to the CPRM modeling framework, KarmaSIM (Narayanan and McIlraith, 2003). WordNet, OpenCYC, and SUMO are also available in OWL. For the experiments reported here, we used the OWL-

[2]X-schemas have been shown to provide a cognitively motivated basis for modeling diverse event-structure related linguistic phenomena, including aspectual inference (Chang *et al.*, 2002), metaphoric inference (Narayanan, 1997) and event-based reasoning in narrative understanding (Narayanan, 1999).

based Teknowledge WMD ontology[3] to instantiate the general frames obtained from FrameNet[4]. The CPRM model populated with domain knowledge to functions as a QA system component for answer extraction (see Figure 2).

We have developed a protocol that allows us to take predicates and frames extracted from the input text and perform a variety of causal and event structure related inferences for QA. Currently, the main API between the semantic extraction and inference components makes use of 1) extracted *predicate-argument* structures, 2) extracted *topic models* and 3) a set of extracted *answertype predicates*. The topic models provide an index into the CPRM model database (compiled from existing FrameNet and Semantic Web (OWL-based) databases). CPRM Models matching the topic model are retrieved and instantiated by the predicate argument bindings specified by the semantic parse output. The *answertype* predicates are mapped to specific structured probabilistic inference procedures afforded by the CPRM models. The next section outlines the currently implemented CPRM inference algorithms and their use for question and answer processing.

4 Inference With CPRMs for QA

Inference in structured probabilistic models of dynamic systems (as in the CPRM model) consists of the following kinds of computations. Here X_t is a state variable at time t (lowercase x_t is a value assignment), and y_t is an observation value at time t.
Filtering:Compute $P(X_t|y_{i...t})$. State update based on the observation sequence.
Prediction: Compute $P(X_{t+h}|y_{1...t})$. Predict the state at some future time $t+h$ based on the observation sequence up to time t.
Smoothing: Compute $P(X_{t-m}|y_{1...t})$. Recompute previously estimated states in the present of current evidence.
MAP: Compute $argmax_{x_{1...t}}(P(x_{1...t}|y_{1...t})$. Compute the best assignment of state values given the observation sequence.
Reachability:Given a CPRM $\mathcal{S}$ with an initial state X_t and a final state X_f, is $X_f \in \mathcal{R}(\mathcal{S}, X_t)$?

We compiled a list of *complex, semantically rich, high frequency* answer types for questions in the AQUAINT CNS data.[5] The top four categories were to 1) *Support/Justification* for a proposition, 2) *the ability* of an agent to perform a specific act, 3) *temporal projection* or predictions from a state, and 4) *hypothetical situations* (including counterfactuals). In our model, these map straightforwardly into the running of various inference procedures (including their sequential application) described in Section 3. For counterfactuals, we use the idea of model intervention (proposed by (Pearl, 2000)). The exact details of the algorithm for counterfactuals is outside the scope of this paper. Table 4 summarizes the various query types and the corresponding inference algorithms. We don't know of any previously implemented QA system (going from text to inference) capable of handling these kinds of questions.

Answer Type	Inference Type
Just(Proposition)	**MAP**
Ability(Agt,Act)	**F;S**
Prediction(State)	**P;R;MAP**
Hypothetical(I,State)	**F;R_I**

Table 1: The type of answer required and the inference algorithm used in the CPRM model. Here **MAP** stands for Maximum A Posteriori estimation, **F** for filtering, **S** for smoothing, **R** for reachability, and **P** for predictive inference. , indicates sequential application. The symbol I represents a specific intervention into the CPRM network (Pearl 2000) as specified by the hypothetical condition. Computing reachability after the intervention is given by R_I.

5 Evaluating Semantically based QA

The previous sections described techniques to incorporate semantic components at increasing levels of depth and complexity. We now report on experiments conducted to evaluate the utility of these differing We report on results pertaining to the impact of (1) the identification of semantic structures and (2) inference through CPRMs on a baseline state-of-the-art Q/A system that emerged after five years of TREC evaluations.

5.1 Evaluating semantic information

To evaluate our novel QA architecture we have used a set of 400 questions pertaining to four different topics: (T1) *UN inspections*; (T2) *Thefts in Russia's nuclear navy*, (T3) *Status of India's Prithvi ballistic missile project* and (T4) *China's participation in non-proliferation regimes*. For each topic we have created a gold standard consisting of (1) 100 questions; (2) one or several text spans considered correct answers by two independent judges; (3) the syntactic parse produced by the Collins parser (Collins, 1996) which was manually corrected; (4) the predicate argument structures of the questions and its corresponding answer, produced automatically and then corrected manually; (5) the semantic frames whenever they could be identified. The answers were extracted from the AQUAINT CNS corpus. The gold standard was used for evaluating the precision (P(Arg)) and recall (R(Arg)) of identifying the correct boundaries of predicate arguments. We have also computed and F_1-score as $F_1(\text{Arg})=\frac{2P(Arg)\times R(Arg)}{P(Arg)+R(Arg)}$. Table 2 lists

[3]http://www.reliant.teknowledge.com/DAML/WMD.owl

[4]The compilation process is not completely automated, since none of the owl ontologies were rich enough to cover our event structure model. For the experiment, we restricted any information added to the OWL-based ontologies to the class documentation strings provided in the ontology. We are currently trying to use semantic extraction to automatically generate this information from the documentation.

[5]AQUAINT is an ARDA sponsored QA program. The Center for Non-Proliferation (CNS) data is a data source released to the AQUAINT project.

Corpus	P(Arg)	R(Arg)	F_1(Arg)
PropBank	85.4	85.6	85.5
AnswerBank	89.4	89.5	89.4
Corpus	P(Role)	R(Role)	F_1(Role)
PropBank	88.5	92.7	90.5
AnswerBank	86.8	95	90.7

Table 2: Identification of predicate-argument structures.

Corpus	P(FE)	R(FE)	F_1(FE)
FrameNet	75.2	77	76.08
AnswerBank	73.5	74	73.74
Corpus	P(Role)	R(Role)	F_1(Role)
FrameNet	91.57	89.13	90.33
AnswerBank	90.2	88.5	89.34

Table 3: Identification of frame structures.

the results. The Table also lists the precision of classifying the arguments (P(Role)), the recall for argument classification (R(Role)) and the corresponding F_1-score. The results are presented for two corpora: the PropBank section 23; and AnswerBank, which represents our gold standard. Table 3 presents similar results for recognizing the boundaries of frame elements (FEs) from FrameNet and for classifying their semantic roles.

5.2 Evaluating the CPRM model for QA

We experimented with the QA system on the AQUAINT CNS data. Since there are no implemented QA systems that perform the kinds of complex inferences described above, our evaluation with respect to the current state-of-the-art baseline relates to the enhanced set of questions and answer types our system can handle. We wanted to calibrate to extent and type of inferences needed for different questions in the CNS scenario data as well as the extent to which such inferences require manual domain model building. To this end, we created a set of 400 hand-annotated question answer passages for the gold standard. We measured the performance of our system with along the following dimensions. 1) How well did the automatically constructed CPRM domain models (from the OWL ontologies) fare when compared to the manually constructed (from gold-standard CNS data) CPRM model? 2) How capable was our CPRM event model in performing a set of complex event-structure based inferences required for QA?

To test (1), we manually compiled CPRM domain models based on our core theory of events and on the gold standard annotations (we used a 60-20-20 build-validate-test dataset). We compared this to the semi-automatically generated from the OWL databases of WMD processes. For our first experiment, we looked at how many of the complex, semantically rich inference types could be made by our system for the two models. Figure 4 shows the

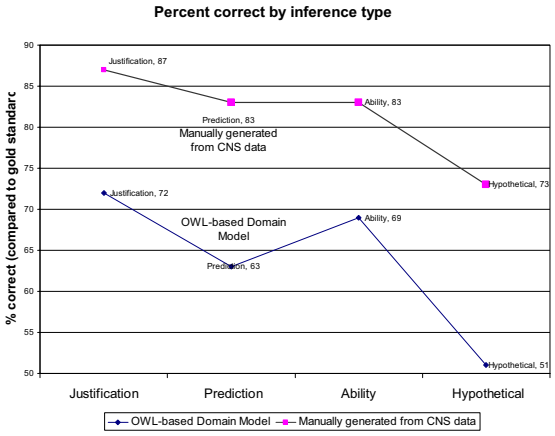

Figure 4: Performance of the CNS-based (gold standard) and OWL-derived CPRM models based on inference type

performance of the two systems on the CNS gold-standard annotations (the results are for the test data of 80 questions). Note that both the manually built and the OWL-based models perform reasonably well for the different inference types we looked at. This is somewhat encouraging given that this is the first inference based QA system (that we are aware of) that goes from textual input to inference. The main shortcoming of the OWL-derived models was that they lacked detailed specifications of the processes, their resource requirements, and a detailed list of agent abilities, preconditions, effects and maintenance conditions. We are seeking to overcome this deficiency through a variety of automatic techniques, semantic web resources, and Subject Matter Expert (SME) input using the CPRM GUI to bootstrap and enhance the acquisition of domain specific knowledge. However, results from these efforts remains future work.

To test (2), we looked at the percentage of inferences by different types of event-structure inferences that had to be made to generate the answer for the questions in the 400 gold standard annotations. The categories we looked at were *aspectual inferences* (Phases of events, viewpoints (zoom-in, zoom-out)), *action and process-feature inferences* (Preconditions, Effect, Resources (produced, consumed, locked)), *metaphoric inferences* (we only looked at Event Structure Metaphors (Lakoff 1999). We counted the number of inferences made by the human and by the model (the CNS-based manually built model) for each category in the annotated data. We looked at the precision (number of correct inferences) and recall (number of total made) .[6]

Table 4 shows our initial results. Note that all

[6]We computed an f-score based on ($\frac{2PR}{P+R}$) for both the CNS gold-standard based CPRM model and for the OWL derived model.

Component	Number	M_{1f}	M_{2f}
Aspectual	375	.74	.65
Action-feature	459	.62	.45
Metaphor	149	.70	.62

Table 4: Inferences broken by Event Structure component. M_{1f} refers to the f-score of the manually constructed CNS gold-standard model, M_{2f} to the model derived from OWL.

AH	PAS	FS	TM
49 (12%)	130 (32%)	78 (19%)	42(10%)
PAS+TM	FS+TM	ES+TM	ES+Inf
141(35%)	94(23.5%)	203(50%)	294(73.5%)

Table 5: Number of correct answer types identified by semantic information originating in: the Answer Hierarchy (AH), the predicate-argument structure (PAS); the topic model (TM); the event structure (ES) and the CPRM inference (Inf) for a set of 400 complex questions.

three of the categories of inferences are fairly common in the data, and our initial results are quite encouraging. The more domain general inference types regarding the aspectual and metaphoric inferences about events seem to fair reasonably well (recall that all these inferences are impossible in the state-of-the-art baseline QA system). The lower score the action-feature inference seems to tied to the lack of domain knowledge in our model regarding domain specific process details (such as the specific resources for the production (or dispersal) of WMD). We expect this number to increase considerably with more domain specific knowledge using the techniques described earlier. We are also conducting a detailed study of other important categories of event related causal inferences.

5.3 Evaluating the Answers

The focus of our experiments was to measure the impact of (1) the identification of semantic structures and (2) inference through CPRMs on state-of-the-art Q/A techniques that emerged after five years of TREC evaluations. As reported in (Moldovan et al., 2002), most of the errors of Q/A systems are determined by (a) the incorrect identification of the expected answer type and (b) the inability to expand question keywords with the ideal words that enhance the retrieval of the candidate answers.

Table 5 lists the results obtained for the identification of correct answer types. The answer hierarchy (AH) comprising more than 8000 WordNet concepts and mapping into 15 name classes was the source of only 12% of the correctly recognized answer types, in contrast with the more than 70% that is correctly identified for factoid questions when processing TREC-like data. To evaluate the contribution of predicate-argument structures (PAS), we considered that the answer type can be defined not only as a semantic class, but also as an argument of a specific predicate. Whenever the answer would be recognized as the same argument of the same predicate or of a directly related predicate[7] we considered that the answer type is recognized correctly. Similarly, when the frame structures could be identified in the question and the answer, the answer type can be indicated by the frame element (FE), and its correct identification accounts for our resolution of a correctly predicted answer type. The topic models (TMs) contribute to the recognition of the answer type if any of the relations they induce pertains to the expected answer, which may be either the relation itself, a more complicated structure that includes any of the topic relations or any concept that takes part in any topic relation but was not accessible directly from the question words. The event structure (ES) was considered a valid source for finding the answer type if any of the schemas that were instantiated contained at least a semantic class or relation that corresponds even partially to the answer structure, whereas the combination between ES and the inference procedures (Inf) determines the answer type either by considering only the semantic information available from the ES or by adding to it the answer types determined by inference. The results listed in Table 5 show that the schema instantiations, through their very general semantic coverage account for most of the answer types which are recognized, whereas the addition of answer types determined by inference accounts for almost 73.5% of the correct answer types of the evaluated complex questions. When processing the test questions only with the AH, 8% of the answers were correct. In contrast, when all the other semantic structures were available and probabilistic inference could be performed, 52% of the extracted answers were correct. In future work we plan to investigate ways in which the semantic structures presented in this paper could improve the quality of paragraph retrieval and keyword selection.

6 Issues and Discussion

The last few years have witnessed a good deal of activity on predicate extraction (aka semantic parsing (Gildea and Jurafsky, 2002; Kingsbury et al., 2002)). Until now it has been unclear if and how predicate extraction might help in the performance of an actual NLP task. Often the intuitive justification offered was that predicate extraction was an intermediate step toward semantic inference (Gildea and Jurafsky, 2002). As far as we know the results reported in this paper constitute the first demonstration that sophisticated textual analysis including predicate-argument extraction can be combined with deep semantic representation and inference models to enhance a state-of-the-art QA system to answer new question types that pertain to causal and tempo-

[7] Directly related predicates are those that (a) belong to the same verb hierarchy in WordNet or (b) are arguments of the target predicate (either because they are infinitives or because they belong to a relative clause).

ral aspects of complex events. Importantly, we believe our work demonstrates a flexible architecture and methodology that harnesses the increasingly widespread availability of semantically motivated resources (such as WordNet, FrameNet, and the Semantic Web). Our current efforts are directed at more effective knowledge acquisition and at expanding the coverage of system both in terms of the domain models and question and answer types supported. We believe that our flexible architecture and CPRM based computational model for combining predicate and frame parsing with deep inference could point the way for building the next generation of semantically rich QA systems.

Acknowledgements

This work was funded by an ARDA AQUAINT grant. We would like to thank Steve Maiorano for the many discussions of ideas he shared with us in this project. Special thanks go to Jerry Feldman and the NTL group at ICSI and UC Berkeley. We are also grateful to the researchers and students working on the AQUAINT project in the Human Language Technology Research Institute at UT Dallas.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of COLING/ACL '98*:86-90,. Montreal, Canada.

Benjamin K. Bergen and Nancy Chang. in press. Simulation-based language understanding in Embodied Construction Grammar. In *Construction Grammar(s): Cognitive and Cross-language dimensions*. John Benjamins.

Daniel M. Bikel, Richard Schwartz and Ralph M. Weischedel. 1999. An Algorithm that Learns What?s in a Name. *Machine Learning Journal*.

Nancy Chang, Srini Narayanan and Miriam R.L. Petruck. 2002. Putting frames in perspective. In *Proc. Nineteenth International Conference on Computational Linguistics (COLING 2002)*.

Ciardo, Gianfranco, Reinhard German, and Christoph Lindemann. 1994. A characterization of the stochastic process underlying a stochastic petri net. *Software Engineering* 20.506–515.

Michael Collins. A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, ACL-96*, pages 184–191, 1996.

Getoor, Lise, Nir Friedman, Daphne Koller, and Avi Pfeffer. 2001. Learning probabilistic relational models. In *Relational Data Mining*, ed. by DzeroskiLavrac, 307–335. SV.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245-288.

Daniel Gildea and Martha Palmer. 2002. The Necessity of Parsing for Predicate Argument Recognition. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL 2002)*:239-246, Philadelphia, PA.

Sanda Harabagiu. 2004. Incremental Topic Representations. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*.

M. Hearst. 1997. TextTiling: Segmenting text into multi-paragaph subtopic passages. *Computational Linguistics*, 23:33-64.

Chin-Yew Lin and Eduard Hovy. 2000. The Automated Acquisition of Topic Signatures for Text Summarization. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*:495-501.

Paul Kingsbury, Martha Palmer, and Mitch Marcus. 2002. Adding Semantic Annotation to the Penn TreeBank. In *Proceedings of the Human Language Technology Conference (HLT 2002)*:252-256, San Diego, California.

George Lakoff, and Mark Johnson. 1999. Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought . Basic Books, New York 1999.

Dan Moldovan, Marius Pasca, Sanda Harabagiu and Mihai Surdeanu. 2002. Performance Issues and Error Analysis in an Open-Domain Question Answering System. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL 2002)*:33-40, Philadelphia, PA.

Dan I. Moldovan, Christine Clark, Sanda M. Harabagiu, Steven J. Maiorano: COGEX: A Logic Prover for Question Answering. HLT-NAACL 2003

Dan Moldovan, Christine Clark, Sanda Harabagiu and Steven Maiorano. 2003. A Logic Prover for Question Answering. In *Proceedings of the HLT-NAACL 2003*. Edmonton, Canada.

Murphy, Kevin, 2002. *Dynamic Bayesian Networks: Representation, Inference, and Learning*. University of California, Berkeley dissertation.

Narayanan, Srini, 1997. *Knowledge-based Action Representations for Metaphor and Aspect (KARMA)*. Computer Science Division, University of California at Berkeley dissertation.

Narayanan, Srini. 1999. Reasoning about actions in narrative understanding. In *Proc. Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*. Morgan Kaufmann Press.

Narayanan, Srini and Sheila McIlraith. 2003. Analysis and Simulation of Web Services In *Computer Networks*, 42 (2003), 675-693, Elsevier, NH.

Pearl, Judea, 2001. *Causality : Models, Reasoning, and Inference*. Cambridge University Press, 2000.

Pfeffer, Avi, 2000. *Probabilistic Reasoning for Complex Systems*. Stanford University dissertation.

Ellen Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*:1044-1049.

Mihai Surdeanu, Sanda Harabagiu, Paul Aarseth and John Williams. 2003. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL 2003)*:8-15, Sapporo, Japan.