Model-based Answer Selection

Steven K. Sinha and Srini Narayanan

International Computer Science Institute
University of California, Berkeley
1947 Center Street, Suite 600
Berkeley, CA 94704
SSinha@CS.Berkeley.edu, SNarayan@ICSI.Berkeley.edu

Abstract

Obtaining informative answer passages and ranking them appropriately has previously been error prone for complex, non-factoid questions related to action and event occurrences, causes, and spatiotemporal attributes. A fundamental problem that has hampered the efforts to date has been the inability to extract relations of interest that determine the search for relevant answer passages. We report on a model-based approach to answer selection based on the idea that relations relevant to a question are best captured by an expressive model of events. We outline the essential attributes of the model and report on its application to the AQUAINT focused data corpus for Question Answering.

Introduction

Present-day 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 by a keyword frequency metric to decide which passage contains the exact answer. 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.

Narayanan & Harabagiu 2004, outlined the general architecture of our QA system. In this paper, we focus on an important sub-problem of the overall task, namely the problem of answer selection. State of the art QA systems use a variety of IR and knowledge-based techniques (using lexical chains in WordNet, extended WordNet) to rank the returned answer passages. While such general purpose resources are useful in expanding the query words to related concepts, there is a fundamental limitation to such techniques. Dealing with individual concepts and words cannot distinguish between answers with the same concept (say acquire) and different role bindings (IBM acquired Versatel vs. Versatel acquired IBM). In complex QA scenarios such as the AQUAINT focused data domain (comprising of CNS documents) getting accurate relational information with the right argument bindings is essential to weed out irrelevant answer passages and documents. We consider a new and improved ranking system, utilizing fairly simple, not-so-deep semantic processing, as a first step to improving answer selection.

We setup the problem by describing the state of the state-of-the-art in the second section. Then, we describe our results to date which consists of a post-processing fix to two of the most striking deficiencies of the current baseline, providing a list of issues to tackle when integrating our solution. We also propose modifications to the baseline system itself, to improve the initial identification of answer candidates. We conclude by discussing ongoing extensions designed to increase the scalability of our solution.

A State of the Art QA System Baseline

Ideally, a QA system would read in a natural language question (constrained by nothing more than, say, English grammar), "understand" the context around and the domain of the question being asked, search through expert literature on the subject at hand, extract the relevant information from a variety of sources, balance the reliability of the information, and then generate an answer that combines the information, giving justification for its conclusions, along with links back to the original source documents. That's some way off.

We used the UTD-LCC state-of-the-art QA System (Pasca & Harabagiu 2001) as a baseline for our task. The UTD QA system uses a component based flexible architecture with modules that a) process the question by linking it to an entry in an ontology of answer-types, b) use a variety of IR techniques to retrieve relevant answer passages, and c) extract the answer passage ranked highest amongst the candidates. The system is trained to handle questions related to a closed domain of documents from the Center for Non-Proliferation Studies (CNS). We used questions from AnswerBank, a QA database of question-answer pairs developed by the University of Texas, Dallas as part of the AQUAINT QA project. Each answer passage retrieved comes from a document in the AQUAINT CNS database.

State-of-the-art QA systems such as the UTD system rely on standard IR techniques (like TF-IDF) along with enhancements that expand the query. Such modifications include search patterns and heuristics based on word clusters, synonym sets, and lexical chains, which are a) derived using machine learning techniques, b) extracted from lexical resources such as WordNet or c) a combination of a) and b). Selecting answer passages relies on a quantitative measure that evaluates the degree to which a passage shares the words in the expanded query keyword set.

There are two areas where we believe the current approaches can be improved and which form the foci of the effort described in this paper:

1. The keywords extracted from the question are related to each other and thus extracting relational information such as the frames, predicates and arguments in the question should enable higher precision searches for answer extraction and re-ranking.
2. Processing relations *in the context* of an expressive model is the crucial link between the information sought in the question and the information contained in the answer.

We hypothesized that building ontological schemata of complex entities such as actions and event structure, populating ontologies with instantiations of these schemata, and translating the entries into a form suitable for effective inference will qualitatively improve QA technology. This paper reports on the first of a series of experiments designed to test our hypothesis. We focus on the ability for models of actions to improve the ranking of candidate answers returned by the UTD baseline QA system.

An Action and Event Ontology

Actions and events are, not surprisingly, the frequent subject of complex queries. *"What will happen if X does Y?"*, *"What does X need before it can do Y?"*, *"If X now has Z, what action Y may have been taken?"*, are but a few. We propose the use of action models to aid in the selection of the best answer to a question asked. Given a set of answer candidate passages returned for an action-related question posed to the QA system, we hypothesize searching for information on to the action's components and related processes will yield documents more relevant than those ranked highly due to query keyword frequency counts.

A general ontology capable of handling scenario and domain questions about events must fulfill some essential requirements. The action model has to be a) *fine-grained* to capture the wide range of possible events and their interactions; b) *context-sensitive* and *evidential* in order to adapt to a dynamic and uncertain environment; c) *cognitively motivated* to allow humans to easily query and make sense of the answers returned; and d) *elaboration-tolerant* so that new domain models can specialize existing representations without changing the basic primitives.

We have developed just such a parameterized model of the structure of events and processes, and it can support tools for programmatic access to enter, modify, advertise and communicate the capabilities, types and details of events for observation, monitoring, inference and control. For further details, comparisons with hybrid system (discrete/continuous) models and an extended treatment of the representation and semantics of events, see (Narayanan 1999; Narayanan & McIlraith 2003).

Figure 1 shows the basic ontology of events. The various components are described below. In each of these cases, we have a precise semantics in terms of the overall structure of the interacting events. 1) The *Basic Structure of an Event*: A basic event is comprised of a set of inputs, outputs, preconditions, effects (direct and indirect), and a set of resource requirements (consuming, producing, sharing and locking). The hasParameter link in Figure 1 depicts the set of parameters in the domain of the basic event type. 2) *Events have a Frame Semantic Structure*: Events are described in language using Frame-like relations. Frames are labeled entities comprised of a collection of roles that include major syntactic and semantic sub-categorization information. The relation *hasFrame* in Figure 1 is a many-to-many link since an individual event may have multiple frames and a single frame could capture multiple events. 3) *Composite Events have a rich temporal structure and evolution trajectories*: The fine-structure of events comprises of a set of key states (such as enabled, ready, ongoing, done, suspended, canceled, stopped, aborted) and a partially ordered directed graph that represents possible evolution trajectories as transitions between key states (such as prepare, start, interrupt, finish, cancel, abort, iterate, resume, restart). Each of these transitions may be atomic, timed, stochastic or hierarchical (with a recursively embedded event-structure). 4) *Process primitives and event construals*: Events may be punctual, durative, (a)telic, (a)periodic, (un)controllable, (ir)reversable, ballistic or continuous. Composite events are composed from a set of process primitives or control constructs (sequence, concurrent, choice, conditionals, etc.) which specify a partial execution ordering over events. The composeBy relation in Figure 1 shows the various process decompositions. 5) *Composite processes support defensible construal operations* of shifting granularity, *elaboration* (zoom-in), *collapse* (zoom-out)) and enable *focus, profiling* and *framing* of specific parts and participants. 6) *Inter-event relations*: A rich theory of inter event relations allows sequential and concurrent enabling, disabling, or modifying relations.

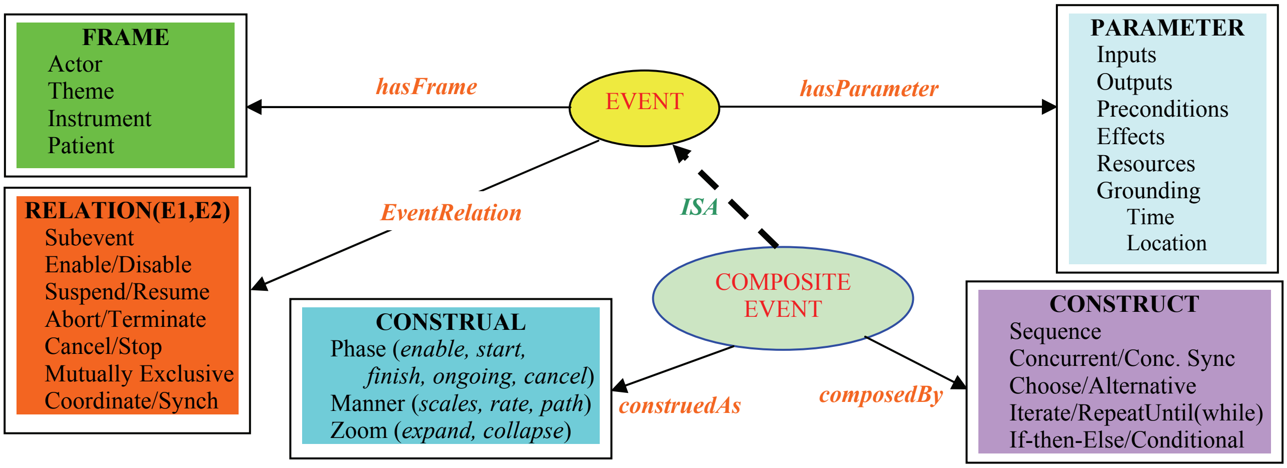

Figure 1

Examples include interrupting, starting, resuming, canceling, aborting or terminating relations.

Any specific action can be described in structured form, as an instance of the event ontology. For example, the process of manufacturing a table entails a carpenter, a need for a certain amount of wood, as well as the existence of a workshop, all of which will enable production of a table. The manufacturing process itself may be broken down into sub-actions of sawing, sanding, construction, applying lacquer, and polishing, each with its own set of actors, preconditions and effects, and sub-actions. This event description is specified by a set of relations that fill the multi-valued role slots defined in the ontology. Here, Precondition : [possess(carpenter, wood, 5 feet), possess(carpenter, shop)], Effect : [possess(carpenter, table)], etc.

From any event description, we are able to build an event model. The chief added feature of a model is its ability to hold state. A model can be instantiated with quantities of a resource available. From this, simulations can be run. For example, using our table example, if it takes five feet of pine board to build a table, and in a given table building scenario, a carpenter has 12 feet, you could infer two tables can be built by simulating the construction cycle until you lacked sufficient resources to build another. The exact mechanisms of simulation are discussed in (Narayanan 1999; Narayanan & McIlraith 2003) and are outside the scope of this paper.

Model-Based Answer Selection

Given an action model for a query, we have the key component to rank answer candidates.
To do this, we

1. Use a Prop-Bank trained parser to parse the question and answer candidates
2. Use the extracted predicate/arguments from the question to index into our model database.
3. Take all the relations from the model expanded into predicate/argument form and match them to the predicate/arguments in each of the parsed answer candidates.
4. Rank answer candidates by relation match count.

Here's an example from the less-innocent domain of WMD production, discussed frequently in the CNS database. *Question: Does Pakistan possess the technological infrastructure to produce biological weapons?* The question, when parsed into predicate/argument form, contains two relations:

- possess (Pakistan, technological infrastructure)
- produce (Pakistan, biological weapons)

The latter is most salient. A Named Entity Recognizer determines Pakistan is a country, and biological weapons is a weapon type.

'produce (?country, ?weapon)' keys into our model database and triggers the WMD Acquisition event model. (Production is one potential sub-action of acquisition.) All the relations in the WMD Possession event description are returned.

Here is a partial list of the relations:

```
; Preconditions for the Develop Expertise stage
Possess( ?country, Expert( ?weapon ) )
Possess( ?country, ResearchInstitution( ?weapon ) )

; Process of Developing Expertise
Acquire( ?country, DevelopmentKnowledge( ?weapon ) )
Research( ?country, DevelopmentKnowledge( ?weapon ) )

; Effect of the Develop Expertise stage
Possess( ?country, DevelopmentKnowledge( ?weapon ) )

; Process of obtaining raw and intermediate materials
; to build weaponized agent
Acquire( ?country, Resources( ?weapon ) )
Buy( ?country, Resources( ?weapon ) )
Steal( ?country, Resources( ?weapon ) )
```

```
Find( ?country, Resources( ?weapon ) )

; Effect of obtaining resources
Possess( ?country, Resources( ?weapon ) )

; Process of obtaining manufacturing facility
; (alternative ways of building manfacturing plant)
Build( ?country, ManufacturingPlant( ?weapon ) )

; Effect of building manufacturing plant
; Precondition for manufacturing
Possess( ?country, ManufacturingPlant( ?weapon ) )

; Process of obtaining weaponized agent
; (actions are alternatives)
Manufacture( ?country, WeaponizedAgent( ?weapon ) )
Acquire( ?country, WeaponizedAgent( ?weapon ) )
Buy( ?country, WeaponizedAgent( ?weapon ) )
Steal( ?country, WeaponizedAgent( ?weapon ) )

; Effect of obtaining weaponized agent
; Precondition (#1) for storage/stockpile
Possess( ?country, WeaponizedAgent( ?weapon ) )
```

The returned relations' variables are then bound to those extracted from the question (?country = Pakistan, ?weapon = biological weapons).

One answer returned by the baseline system is:

> While Pakistan is not known to possess biological weapons (BW), it has talented biomedical and biochemical scientists and well-equipped laboratories, which would allow it to quickly establish a sophisticated BW program, should the government so desire. (Pakistan Country Profile, CNS 2004)

When parsed, this answer contains the following relations, among others:

- possess (Pakistan, biological weapons, not known)
- has (Pakistan, biomedical scientists, talented)
- has (Pakistan, biochemical scientists, talented)
- has (Pakistan, laboratories, well-equipped)

all of which match relations in the event model (dropping the third argument). Our matching algorithm counts preconditions, resource relations, and effects of an action as the relevant set for a direct query about the action. The relations are all weighted equally, so the match score is just a count of relational matches (relations and bindings). Answer re-ranking is directly based on the model match score.

Results

Of the complex questions (i.e. non-definitional/factoid) in the ~2700 AnswerBank question/answer pairs, at least half were about either WMD acquisition, use, hiding, elimination, or control through treaties. Thus, we estimated that just five, fairly simple, parameterized models based on the action and event ontology outlined earlier could cover well over 1000 questions in the AnswerBank database. Preliminary results confirmed our estimates.

We also found that the use of relations did successfully eliminate irrelevant answer candidates which happened to contain the same keywords, but didn't relate them to one another in the same manner as the question posed.

Recall that our expectation was that expanding the query to cover relevant predicates in the model would result in responsive answer candidates for the end user. To evaluate this hypothesis we selected questions randomly from AnswerBank dealing with the WMD Acquisition scenario. We then used the state of the art baseline system and took the top seven answer candidates produced for each question, adding in the gold standard answer as a candidate where it wasn't automatically retrieved. We then re-ranked the answer candidates for each question using our model-based approach. If the model-based approach was superior to the baseline, we expected it would produce the gold-standard as the best answer more often than the baseline and also more often than chance (1 in 7 or 1 in 8). Somewhat to our surprise, running this experiment with the model-based relations revealed a top-ranked gold-standard answer with a 100% success rate. It became clear that the ability to use action-model-based relations could cleanly separate the gold-standard answer from other answers returned by the system, thus validating our approach.

System Design Issues

While these initial results are promising, there are a number of system design issues that we are currently tackling to make the model-based approach a scalable component of an open domain QA system.

Availability of models. For our technique to scale up, we need to be able to easily construct models of actions and events in different domains. Our current technique is to automatically cull as much of this information as possible from the SemanticWeb. To this end, we have built an automatic translator using the SemanticWeb markup language, OWL, to be able to compile our models automatically. However, the specifics of the various action parameters for specific domains will not likely be represented completely on the Web. These must be hand designed or converted from other manually created descriptions. We have been investigating options for direct knowledge entry by Subject Matter Experts (SMEs) using public domain ontology editors (like Protégé' from Stanford). We have had encouraging preliminary results from this approach and efforts are underway to formally evaluate the technique and the utility of the resulting model base.

Mapping. The mapping and matching of predicates and arguments to model relations is difficult, but recent work has made several advances in resolving this issue. As we saw in the WMD example, to work, we required a match between "manufacture" (in the model) and "produce" (in the question), synonyms in this domain. These problems are obviously compounded by nominal descriptions of events and compound nouns. Our current experiments attempt to exploit lexical resources like WordNet and FrameNet for this purpose. Initial results identifying frame matches for relations extracted from the input questions and answers in AnswerBank suggest a high hit rate (frames found for relations extracted from the QA database). We are currently cooperating with University of Texas, Dallas to incorporate their frame parser into the extraction system. Once this is accomplished, we plan to systematically evaluate and report on the impact of frame extraction on the model indexing and answer selection process.

External KB lookup. To provide wide coverage over possible questions from domains like WMD with only a few models, access to external KBs is required. For example, in the WMD production model, the types of experts needed for 'Developing Expertise' for nuclear weapons is entirely different from those needed for biological weapons. The same model can be used for both types of weapons development, through use of parameterization. Documents about biological weapons production in Pakistan might well have instances of 'possess (Pakistan, Expert(biological weapons)), when the function, Expert (biological weapons), is expanded into 'biochemical scientist' and 'biomedical scientist', among other terms. We are currently experimenting with a named entity tagger built by our collaborators at the University of Texas, Dallas which has been trained on an extended ontology of entities in the various WMD domains. Our initial results show that this technique in combination with the model-directed search reported in this paper is extremely promising. We are formally evaluating the impact of combining the UTD extended named entity tagger and should have results to report by the time of the Workshop.

Relevance of relations. In the overly-simplistic-yet-effective metric we propose, all relation matches count equally. Certainly, though, a document which directly addresses the question asked may be more valuable to the user than one that discusses, say, the preconditions of a process which provides the resources needed by another process whose effect is being asked about. We assume a user wants the most direct answer as possible to a query posed. As such, we propose an outward expansion of searching, from the direct preconditions and effects of a query relation, to more indirect preconditions and effects and tangential processes, if the previous searches return no good answer candidates.

Query Expansion

One major difficulty for the full QA system is producing relevant answer candidates in the first place. Increasing the number of relevant answer candidates requires intervention into the baseline system itself. As mentioned, the baseline system has neither the domain knowledge, nor a sense of how keywords are related. Therefore a good number of the returned passages are entirely irrelevant. (Query about the US's own stockpiles of weapons, and the documents returned have the US as the source of intelligence about someone else's stockpiles.) To rectify this, our model-based approach has to be moved up into the initial process of selecting answer candidates. Since the domain of our source documents is closed (the CNS database), the documents can be preprocessed – parsed into predicate/argument structures and indexed accordingly. Then, as opposed to searching for high keyword frequency, the system can search for model-directed relation frequency. By using synonym and action-model relation expansion, the number of query terms will increase, as will the quality of answer candidates, we believe.

Future Work

We are focusing on connecting more of the subsystems together for automated processing of text, from query to production of model ranked answers, including making the necessary modifications for query expansion.

Work on model simulation and inference is an ongoing effort that feeds into the model based answer selection process. Model simulation and inference has the potential to not only calculate exact answers given the necessary evidence, but also compute values for relation arguments, providing the necessary information for additional searches of as yet out of reach relevant answer candidates.

Acknowledgements

This work was funded by an ARDA AQUAINT grant. Special thanks go to Jerry Feldman and the NTL group at ICSI and UC Berkeley. We are also grateful to the Sanda Harabagiu and the AQUAINT QA group at UTD for providing access to their QA system and for helpful discussions on this work.

References

Pasca, M. A., and Harabagiu, S. 2001. High Performance Question/Answering. In *Proceedings of the 24th Annual International ACL SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2001)*, 336-374. New York, NY: ACM Press.

Narayanan, S. 1999. Reasoning About Actions in Narrative Understanding. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI '99)*, 350-357. San Francisco, CA: Morgan Kaufmann.

Narayanan, S., and Harabagiu, S. 2004. Question Answering based on Semantic Structures. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*. San Francisco, CA: Morgan Kaufmann.

Narayanan, S., and McIlraith, S. 2003. Analysis and Simulation of Web Services. *Computer Networks* 42: 675-693.