Extending the JAVELIN QA System with Domain Semantics*

Eric Nyberg, Teruko Mitamura, Robert Frederking, Vasco Pedro,
Matthew W. Bilotti, Andrew Schlaikjer and Kerry Hannan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{ehn,teruko,ref,vasco,mbilotti,hazen+,khannan}@cs.cmu.edu

Abstract

This paper presents the current status of work to extend the JAVELIN QA system with domain semantics for question answering in restricted domains. We discuss how the original architecture was extended, and how the system modules must be adjusted to incorporate knowledge from existing ontologies and information provided by third-party annotation tools.

Introduction

This paper describes extensions to the JAVELIN system (Nyberg *et al.* 2003a; 2003b) for question answering in restricted domains. The basic architecture (Figure 1) includes four processing components and two control components:

- Question Analyzer (QA): analyzes the input text to determine question type, answer type, and keywords.
- Retrieval Strategist (RS): uses a successive relaxation of structured queries to retrieve relevant documents.
- Information Extractor (IX): locates passages in the documents which are candidate answers.
- Answer Generator (AG): canonicalizes and aggregates the candidate answers to produce a ranked answer list.
- Execution Manager (EM): creates instances of these component types to answer questions at run time.
- Planner (PL): decides when to invoke each component, and which algorithmic variant to use.

Further detail regarding the basic JAVELIN architecture can be found in (Nyberg *et al.* 2003a; 2003b; Hiyakumoto 2004). To extend the system for use in restricted domains, we first considered the differences between developing an open-domain QA system and developing one for a particular domain. There are three basic considerations: a) there may be a limited amount of data (text) available for development of data-driven techniques; b) in a restricted domain, the text includes both domain-specific terms, and general English terms that carry domain-specific meanings; c) if the domain is narrow enough, it may be possible to model the domain semantics with reasonable human effort.

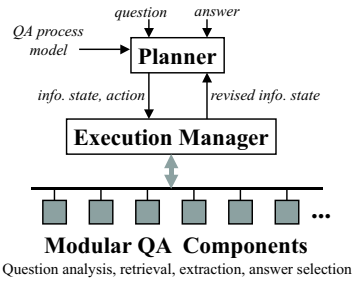

Figure 1: The basic JAVELIN architecture.

These characteristics prompted a discussion of which techniques may or may not be effective a restricted domain. On the one hand, it might be difficult to develop data-driven statistical techniques for text interpretation, given the potential lack of data. On the other hand, symbolic parsing techniques may not fare much better if they do not take into account domain semantics, especially domain-specific meanings of words and phrases. Symbolic NLP might be possible, but only if we can identify the lexical items in the restricted domain and interpret them properly.

Under Phase II of ARDA's AQUAINT program, we are extending JAVELIN for use with the corpus of documents created by the Center for Non-proliferation Studies (CNS). The CNS corpus contains reports on the weapons capabilities of various geo-political entities; for this study we selected a subset of 1130 documents that are potentially relevant to a particular intelligence scenario (development of bioweapons in Egypt). The extended system makes use of the Identifinder (Bikel, Schwartz, & Weischedel 1999) and ASSERT (Pradhan *et al.* 2004) text annotators, WordNet (Fellbaum 1998), and a WMD ontology created by SAIC and Stanford KSL during Phase I of AQUAINT.

We extend the question analysis phase to introduce the notion of *key predicates* - a predicate-argument representation of the kinds of facts that might answer the question. We also add annotators to label important semantic information in the corpus (e.g., named entities, verb-argument structures, semantic roles, etc.). When semantic informa-

*Work supported in part by AQUAINT program award NBCHC040164.

tion is available, the process of question answering becomes one of finding relevant facts and matching them against a key predicate representation of the question.

The remainder of this paper is structured as follows. In Section 2, we present the details of the extended design for closed-domain QA. In Section 3 we present a complete end-to-end example. We conclude in Section 4 with a summary of open issues and ongoing research.

Extending JAVELIN for Restricted Domains

A primary difference between the open-domain JAVELIN system and the version we are extending for restricted-domain QA is the leveraging of an ontology that represents entities and relationships specific to the domain. Where the open-domain system relies on keyword-based retrieval, passage extraction, and reconciliation of answer strings with expected answer types, the extended system makes use of semantic retrieval, fact extraction and reasoning.

Module Extensions

The prototype described in this paper includes changes to the four processing components in the JAVELIN pipeline:

- Question Analyzer: This component, which formerly produced just question keywords, question type and answer type for each question, is extended to provide key predicates as well. Key predicates (described in the following section) are a logical representation of the facts which are sought by the system as candidate answers.
- Retrieval Strategist: To complement existing keyword-based indexing and retrieval, the Retrieval Strategist will be extended to also access a relational database containing instances of predicates (verb-argument frames) and entity mentions. In this new retrieval mode, the system serach for documents containing instances of predicates that match key predicates and contain mentions of the entities referred to in the question. The search is extended through the use of Wordnet and the CNS ontology (details in the following section). No further processing is performed on the retrieved predicate instances at this stage.
- Information Extractor: In previous work, we proposed an approach to question answering based on unification of a logical representation of the question with logical representations for the passages in the corpus (van Durme *et al.* 2003). For the prototype system we decribe here, a new component is created which takes the key predicates as input, as well as a set of documents provided by the Retrieval Strategist which contain predicate instances that match the key predicates, and entity mentions of interest. Predications are instantiated by filling in the entities and binding the logic variables, and then they are checked for consistency against the ontology. Only those that are consistent with our knowledge sources are passed on to the next module as candidate answers.
- Answer Generator: The Answer Generator module for the fact-based QA model extends quite naturally from our previous work in open-domain QA. The Answer Generator is responsible for combining evidence for answers extracted from different documents, and for resolving issues of answer granularity. Developing good metrics for computing confidence scores from combinations of facts is an entire open area of research; our current Answer Generator performs only rudimentary filtering of duplicate facts and presentation of the answer set.

Ontology Resources

Within restricted-domain question answering, an important issue arises when parsing domain specific concepts. Typically these concepts are not represented in open domain knowledge bases and lexical resources such as Wordnet, ThoughtTreasure, etc. In order to efficiently recognize, parse and use these concepts it becomes very important to have an adequate domain specific ontology that describes the relations between the domain concepts, as well as the properties of the objects in those domains. For the prototype system we utilized the CNS ontology, which represents concepts and relations about WMD capabilities of geo-political entities. The CNS ontology was written in KIF (Genesereth & Fikes 1992); we converted the KIF to JGraph format and created a Java API for the resulting ontology.

There are two types of information that we extract from the CNS ontology to support question answering: the Type Hierarchy and the Set of Synonyms (AKAs) for a given concept. These are used to extend the type of a given concept for indexing and semantic-based retrieval.

Type Hierarchy The type hierarchy is derived through the traversal of the graph through the relations of the type *Subclass-Of* for nodes of the type *relation* and *Instance-Of* for nodes of the type *object*[1]. The result is a chain of types from the more specific to the more general:

Anthrax → Biological-Weapon →
Weapon-Of-Mass-Destruction → Weapon →
Physical-Device

Other relation types will be considered in the future, such as *Member-Of*, but the ambiguity ("John is a member of the club?" vs "Portugal is a member of the European Union") in the inference possibilities makes it unclear how useful they will be for type extension in information extraction.

AKA Extraction AKA (or Also Known As) Extraction is derived from the the *Also-Known-As* relation. Most of the nodes reached through this relation are *abstract* nodes, and thus usable only for identification of the defined object, from which more information can be extracted. It is assumed that all synonyms are grounded in the same concept, and usually refer to different ways of saying the same thing:

ICBM → Inter Continental Ballistic Missile →
Inter-Continental-Ballistic-Missile → Intercontinental
Ballistic Missile → Intercontinental-Ballistic-Missile

[1]We note that the CNS ontology is not a tree structure, and therefore there is no guarantee that the traversal of hypernyms will generate a unique path; we perform a greedy traversal, where the first path is chosen, thus producing an ordered list as a hierarchy.

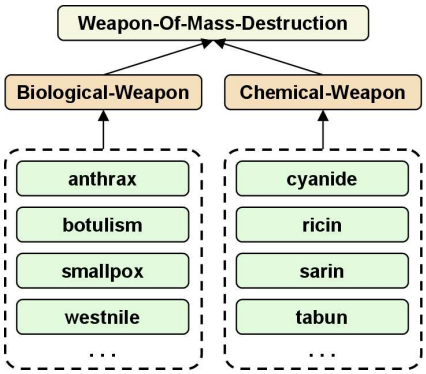

Figure 2: Example from the CNS ontology in JGraphT

Figure 2 shows an example from the CNS ontology in JGraphT format. The figure illustrates the *Chemical-Weapon* and *Biological-Weapon* objects and local class hierarchy via *Subclass-Of* and *Instance-Of* relations.

A Detailed End-to-End Example

For this work, we have chosen to show a detailed example that highlights the differences between general, open-domain question answering and narrow-domain question answering. We chose an example question extracted from a hypothetical dialog between JAVELIN and an intelligence analyst reporting on Egypt's chemical weapons programs:

"What chemical weapons does Egypt have?"

The example question is a list question that poses several specific challenges. The first of these challenges is the "paraphrase problem" as identified in (Woods 1997), namely that the phrase "chemical weapons" could be realized in the text in any number of equivalent ways, including "chem-weapons", "chemical arms" and "chemical agents", etc., or even by terms that subsume it like "CBW" or "WMD."

Querying on "chemical weapons" using traditional, keyword-based IR will fail to retrieve documents that use one of the paraphrases listed above; this is a known limitation of keyword-based IR. Although automatic query expansion techniques exist that claim to address this problem, they rely on cooccurrence between query terms and related words in top-ranking documents. Given this, it seems unlikely that "chem-weapons" would be generated as an alternation for "chemical agents." As Woods also notes, thesaurus-based methods are equally prone to error because the words in a traditional thesaurus entry are not guaranteed to be pairwise synonymous; some other relation might hold between them, such as hypernymy, hyponymy or meronymy, but this is underspecified. A traditional thesaurus might identify "chemical weapons" and "WMD" as synonyms, but suggest no relationship between "chemical arms" and "chem-weapons."

Another challenge posed by list questions is that the name of the category queried (e.g. "chemical weapons") does not usually appear in relevant documents. Often, a particular article will discuss one or more specific chemical weapons, mentioning them by name, but never using any expression for the name of the object type they belong to, such as "chemical agents." If we add a keyword to the query to represent the object type, we will incur a recall penalty because not all relevant documents contain that keyword.

To address these two challenges, we are leveraging an ontology built for the CNS data that is capable of detecting a variety of equivalent expressions that refer to the type Chemical-Weapon. The ontology can also tell us that the phrase "WMD" refers to a superset, not an equivalent set, of objects that contains not only chemical weapons, but also biological and nuclear weapons. Furthermore, the ontology understands several instances of the type Chemical-Weapon, and we have augmented the ontology to include an even wider variety of Chemical-Weapon instances for the purpose of this example. See Figure 2 for a visualization of the Chemical-Weapon type in the CNS ontology.

A third challenge underscored by this particular example is that the verb meaning is difficult to decipher. The question could be asking which chemical weapons Egypt has currently in its arsenal in some quantity, or which agents Egypt has the capability to produce. It could also be asking which potentially lethal substances Egypt may be storing in, say, for example, a research laboratory, but has not yet been able to weaponize. A related difficulty is how to generate alternations for the verb so that documents containing its synonyms can be retrieved; the previously cited comments from Woods regarding the use of thesauri also apply here.

An open-domain QA system might try to generate the most common synonyms using Wordnet (Fellbaum 1998):

have: hold, feature, experience,
receive, get, undergo, own, possess

Without making a judgment as to whether or not expansion of synonyms is good retrieval practice on average, it is clear that the expansion given for "have" above is not taking advantage of assumptions that we can make about our closed domain. We would rather have a verb expansion such as the following, which is conditioned on the domain and pinpoints several potential meanings for "have" that are exceptionally relevant for the CNS domain:

have: possess, develop, maintain,
research, deploy, stockpile, sell, trade

As this domain-specific expansion capability is not yet available in the ontology we are working with, the expansion of key predicates into surface forms was performed by hand for the purposes of this example.

As a part of the retrieval process, JAVELIN uses corpus annotation to mark mentions of different kinds of chemical weapons in the text and link them into the ontology. In addition, JAVELIN relies on the ASSERT system (Pradhan *et al.* 2004) to generate predicate-argument structures for verbs in the text. These annotations are stored in a separate database to facilitate search and retrieval.

When our example question enters the JAVELIN pipeline, the Question Analyzer is the first module to process it. It returns an expected answer type of Chemical-Weapon, which corresponds to a type node in our CNS ontology. It also recognizes the question as a list question, which means that the result should be a list of different instances of the type

Chemical-Weapon. The final and most important job of the Question Analyzer is to break the question down into key predicates that can be used by the RS to retrieve documents. Since extensions are ongoing, these tasks were carried out manually for this example. In this case, the question is broken down into the following conjunction of predicates. Note that the Question Analyzer interprets the verb "have" correctly for this context:

$$\text{POSSESS}(\text{ EGYPT, } ?x \text{ }) \wedge \text{IS}(\text{ } ?x\text{, Chemical-Weapon })$$

This formulation of the user's information need goes on to the Retrieval Strategist, which is responsible for generating a set of queries and retrieving documents relevant to that information need. Making use of the ontology, the closed-domain RS can improve on the quality of keyword-based search by indexing and retrieving on predicate instances that match the information need. The first step is to expand the set of key predicates with domain-appropriate expansion, e.g.:

$$\begin{gathered}(\text{ POSSESS}(\text{ EGYPT, } ?x \text{ }) \vee \text{DEVELOP}(\text{ EGYPT, } ?x \text{ }) \vee \\ \text{MAINTAIN}(\text{ EGYPT, } ?x \text{ }) \vee \text{RESEARCH}(\text{ EGYPT, } ?x \text{ }) \vee \\ \text{DEPLOY}(\text{ EGYPT, } ?x \text{ }) \vee \text{STOCKPILE}(\text{ EGYPT, } ?x \text{ }) \vee \\ \text{SELL}(\text{ EGYPT, } ?x \text{ }) \vee \text{TRADE}(\text{ EGYPT, } ?x \text{ }) \text{ }) \\ \wedge \text{IS}(\text{ } ?x\text{, Chemical-Weapon })\end{gathered}$$

With this expanded pattern, the RS can then use a combination of database and IR techniques to retrieve a set of documents from the collection to pass along the pipeline. The subcorpus used in this example is approximately 1130 documents extracted from a cleaned, TREC-formatted version of the CNS data. The subcorpus is centered on the Egypt chemical weapons program scenario described above.

The RS generates a list of documents that contain mentions of the geo-political entity Egypt, as well as instances of one of the predicates, such as POSSESS or DEVELOP, and passes that list of documents along to the IX module. The majority of the documents passed to the IX contain instances of predicates that do not satisfy the constraints in the information need; for example, they may discuss items that Egypt possesses, stockpiles or trades that are not chemical weapons. It is the job of the IX to separate the truly useful facts from the rest using unification and constraint checking techniques grounded in the ontology.

From 1130 documents in the subcorpus, 14 sentences were judged by the IX to contain relevant predications; the predications from three of those sentences are shown below:

$$\begin{gathered}\text{PRODUCE}(\text{ EGYPT, mustard gas }) \\ \text{PRODUCE}(\text{ EGYPT, nerve gas }) \\ \text{PRODUCE}(\text{ EGYPT, cyanide gas }) \\ \text{POSSESS}(\text{ EGYPT, VX gas }) \\ \text{IS}(\text{ mustard gas, Chemical-Weapon }) \\ \text{IS}(\text{ nerve gas, Chemical-Weapon }) \\ \text{IS}(\text{ cyanide gas, Chemical-Weapon }) \\ \text{IS}(\text{ VX gas, Chemical-Weapon })\end{gathered}$$

The actual sentences that gave rise to these facts are shown below, with the mentioned chemical weapons in italics, and the filename of the original CNS corpus document given in parentheses for each sentence:

> Cordesman states that Egypt appears to have several production facilities for *mustard and nerve gas*, and that the Sakr-80 missile could be modified and armed with chemical warheads. (n9716893.htm)

> Egypt has the infrastructure to rapidly produce *cyanide gas.* (n9716893.htm)

> Egypt has hinted that it is willing to destroy its stock of *VX gas* if Israel agrees to open its nuclear facilities to international inspection and to sign an agreement on the non-proliferation of weapons of mass destruction. (n9716894.htm)

The IX also found facts about Egypt's delivery of chemical weapons to third parties, and Egypt's battlefield use of such weapons, both of which might satisfy the information need with the help of some inference. Other retrieved information includes facts about Egypt's development of missiles that could be used to deliver chemical weapons, and mentions of Egyptian chemical weapons experts.

As a baseline for comparison with our method of search by extraction of predicate instances, we ran a Boolean keyword search engine over the document collection using the following query, in which the last term matches any term from a document that begins with the prefix "chem:"

```
egypt AND weapon AND chem*
```

The query returned 235 of the 1130 documents in the Egypt scenario subcorpus, and they were manually judged for relevance under the criterion that they link Egypt or its government to having possession of or any other connection to any sort of chemical agent that can be used as a weapon, regardless of the level of inference necessary. It turned out that only 17 of the 235 (0.0723 Precision) were relevant to the scenario in that they discussed chemical weapons at all in connection with Egypt. The vast majority of the documents concerned nuclear activity on Egypt's part, disputes between Egypt and Israel, development and use of non-chemical weapons by Egypt, development of Scud and other types of missile technology and trade of these technologies between Egypt and other countries.

Status and Future Work

The example presented in this paper demonstrates how we are extending the JAVELIN QA system to work with restricted domain semantics. We are evolving toward a model in which facts are extracted from text, reasoned over, and used to pinpoint answers supported by the root text in the corpus documents. Our goal is to extend the coverage of JAVELIN to the entire CNS corpus, in support of unrestricted question-answering dialogs with the user. In the remainder of this section we mention a few of the related research topics we are currently pursuing.

Graphical Models for Semantics and Search

Advances in fact extraction and inference may be achievable via graphical models of semantics and graph-theoretic

approaches to search. Using an existing set of NLP tools including named-entity taggers, shallow semantic parsers, and within-document coreference tools, a series of interconnected semantic annotations has been constructed across available corpora. Methods of refining the resulting network of typed entities and predicates to produce a basic level of global entity coreference-resolution are under investigation. In addition, entities, along with predicate target verbs, will be tied into lexical and ontological knowledge bases to further ground the types of these objects, and aid further modes of inference across this data. Following previous research by (Bhalotia *et al.* 2002), question results may be modeled as trees within this graph which connect nodes matching query elements. In the case of our semantic network, many refinements to heuristics influencing the search for interconnecting trees are possible, as the relational model is much more complex than those previously investigated.

Moving toward Semantic Retrieval

To move to a question-answering model that is based on fact extraction and semantic retrieval, we are investigating different strategies of indexing semantic content, such as predicate instances, arguments and entity mentions. Our Information Extractor module will need to be tightly integrated with our ontologies and sources of world knowledge so that it can evaluate the consistency and check the constraints of candidate predications discovered among the documents in the collection by the Retrieval Strategist. The challenge in JAVELIN will be to seamlessly merge the fast, shallow search capabilities of IR with slower, deeper search through the predicate argument structures and ontologies on the Information Extraction side.

Probabilistic Inference Rules via NLP

One of the advantages of working in a restricted domain is that it is usually possible to find a domain expert who has spent many hours (sometimes, years) becoming familiar with the domain. With experience comes the capacity to perform intuitive associations between concepts, events and persons that are very difficult to capture with automatic data analysis. In the future, we would like our system to support the creation of new rules of inference, not in a complicated logical representation, but in natural language. We are working towards a system design which will allow an analyst to specify rules of inference over the semantic types and relations in an ontology (e.g., "If an individual belonging to an organization purchases weapons, the organization may now possess the weapons"). To reason with such probabilistic expressions will require a rule engine that can associate likelihoods with facts and the rules that produce them; we are actively engaged in research in this direction.

Conclusion

In this paper, we described a set of extensions to the JAVELIN QA system which support the use of domain semantics (ontologies and semantic annotations) for question answering that is based on key predicates and fact matching, rather than keywords and string matching. We have successfully integrated a significant portion of the knowledge in the CNS ontology into JAVELIN. We have also integrated the Identifinder and ASSERT modules for named-entity annotation and verb-argument (semantic role) annotation. We presented a detailed example of how the system works on a sample input; due to the ongoing status of the implementation, some of the data used in the example was manually generated. While the current system is still at the proof-of-concept stage, we expect to complete the process of integrating the CNS data with JAVELIN for an evaluation with real users during the coming year.

References

Bhalotia, G.; Hulgeri, A.; Nakhe, C.; Chakrabarti, S.; and Sudarshan, S. 2002. Keyword searching and browsing in databases using banks. In *In Proceedings of the 18th International Conference on Data Engineering (ICDE-2002)*, 431–440.

Bikel, D. M.; Schwartz, R. L.; and Weischedel, R. M. 1999. An algorithm that learns what's in a name. *Machine Learning* 34(1-3).

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

Genesereth, M., and Fikes, R. 1992. Knowledge interchange format, version 3.0, reference manual. Technical Report Logic-92-1, Stanford University.

Hiyakumoto, L. S. 2004. Planning in the javelin qa system. Technical Report CMU-CS-04-132, Carnegie Mellon University School of Computer Science.

Nyberg, E.; Mitamura, T.; Callan, J.; Carbonell, J.; Frederking, R.; Collins-Thompson, K.; Hiyakumoto, L.; Huang, Y.; Huttenhower, C.; Judy, S.; Ko, J.; Kupść, A.; Lita, L. V.; Pedro, V.; Svoboda, D.; ; and van Durme, B. 2003a. The javelin question-answering system at trec 2003: A multi-strategy approach with dynamic planning. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*.

Nyberg, E.; Mitamura, T.; Carbonell, J.; Callan, J.; Collins-Thompson, K.; Czuba, K.; Duggan, M.; Hiyakumoto, L.; Hu, N.; Huang, Y.; Ko, J.; Lita, L.; Murtagh, S.; Pedro, V.; and Svoboda, D. 2003b. The javelin question-answering system at trec 2002. In *Proceedings of the Eleventh Text REtrieval Conference (TREC 11)*.

Pradhan, S.; Ward, W.; Hacioglu, K.; Martin, J.; and Jurafsky, D. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL-2004)*.

van Durme, B.; Huang, Y.; Kupść, A.; and Nyberg, E. 2003. Towards light semantic processing for question answering. In *Proceedings of HLT/NAACL 2003 Workshop on Text Meaning*, 54–61.

Woods, W. A. 1997. Conceptual indexing: A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories.