

Question Answering Using Ontological Semantics

Stephen BEALE, Benoit LAVOIE, Marjorie MCSHANE, Sergei NIRENBURG, Tanya KORELSKY

Institute for Language and Information
Technologies (ILIT-UMBC)
1000 Hilltop Circle
Baltimore, MD, USA 21250
{sbeale,marge,sergei}@umbc.edu

CoGenTex, Inc.
840 Hanshaw Rd, Suite 1
Ithaca, NY, USA, 14850
{benoit,tanya}@cogentext.com

Abstract

This paper describes the initial results of an experiment in integrating knowledge-based text processing with real-world reasoning in a question answering system. Our MOQA “meaning-oriented question answering” system seeks answers to questions not in open text but rather in a structured fact repository whose elements are instances of ontological concepts extracted from the text meaning representations (TMRs) produced by the OntoSem text analyzer. The query interpretation and answer content formulation modules of MOQA use the same knowledge representation substrate and the same static knowledge resources as the ontological semantic (OntoSem) semantic text analyzer. The same analyzer is used for deriving the meaning of questions and of texts from which the fact repository content is extracted. Inference processes in question answering rely on ontological scripts (complex events) that also support reasoning for purely NLP-related purposes, such as ambiguity resolution in its many guises.

1 The Task

People would have no problem answering questions like *Has Tony Hall met with Umid Medhat Mubarak?* – provided they know who these two people are and have witnessed such a meeting or read about it. Even in the absence of overt evidence about such a meeting, people might conclude – based on additional knowledge they might have about the protagonists – that such a meeting could or might have taken place. Some current automatic question-answering (QA) systems might be able to answer such a question if they found a sentence like *Tony Hall met with (saw, talked with) Umid Medhat Mubarak on July 3, 2003 in Baghdad* in some text. But what if the text data was more typical, like, for instance, the following two excerpts:

April 18, 2000 Associated Press. *Representative Tony Hall, a Democrat from Ohio, arrived in Baghdad on a four-day visit to investigate the plight of Iraqi people under sanctions aimed at forcing the government of Iraq to give up its weapons of mass destruction...*

Umid Medhat Mubarak returned to Baghdad on April 17, 2000 after a visit to Jordan and plans to meet with a visiting US politician.

To the best of our knowledge, no current system can input the above texts and return a reasoned response about the likelihood of a meeting between Tony Hall and Mubarak. But in a realistic environment there are even further complications. What if the first text was in English and the second in Arabic? Will the system be able to make even a tentative connection between *Tony Hall* (in the first text) and *US politician* (in the second)? What if the reference to *US politician* was omitted; i.e. if the second text contained only the information that *Umid Medhat Mubarak* was in Baghdad on April 17, 2000? The system would have to infer the possibility of a meeting on the basis of knowledge about (at least) the social and professional background of the protagonists and the times involved.

This paper describes a system that is able to make connections and inferences such as the above. Its most important properties are question answering against structured data stored in a fact repository (FR) and the fact that it uses the same processing machinery and knowledge resources a) to process texts for conversion into facts, b) to understand questions and c) to find answers to questions. We describe the underlying technology that supports such a capability, including the production of text meaning representations (TMRs), reference and date resolution, fact extraction and retrieval, and event scripts that allow us to infer (with some degree of probability) certain events or states not directly stated in any text.

2 The Environment for QA

Our question answering system consists of four main and one auxiliary processing modules (see Figure 1). The **question analysis module** takes as input the text of a user's question and produces its **text meaning representation (TMR)**, see below for an illustration) that contains representations of instances of ontological concepts to which the input refers plus speaker-attitude and communicative information. The TMR is input to the **question interpretation module** that interprets the question in terms of its type and transforms it into a formal query against the **fact repository** or the **ontology** (see below). (Note that the format of the database query is the same as that of the question TMR. In general, all internal representations of knowledge in our system, both elements of knowledge support and results of actual processing, are compatible with the content and format of the ontology and fact repository.)

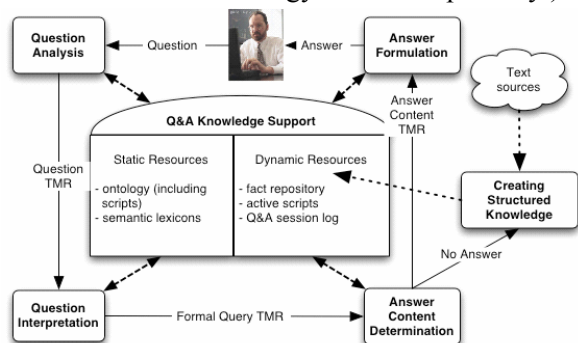


Figure 1. The top-level architecture of the system.

Thus, text meaning representation in our approach “doubles” as the basis for reasoning processes. The query serves as input to **answer content determination**. This latter module uses the knowledge resources of the system to infer the most preferred answer, once again, formulated in the TMR metalanguage. If an answer cannot be found, the system has the option to call the auxiliary module for **creating structured knowledge, CSK**. The CSK module works also in the background mode, using text sources to continuously update the fact repository (in the work reported here there has been some human involvement in the process of TMR production for CSK; we plan to study the degradation of the system when used in a fully automatic mode). When called by the answer content determination module, the CSK module analyzes texts and generates entries in the fact repository that help to answer the original question. The text analysis process in this module is the same as that used in question analysis. The final module in the system is **answer formulation**. The current

implementation simply returns fragments of facts (and fact reasoning chains) that answer the initial question. In the future, natural language generation will be employed to produce textual responses.

In order to answer complex questions in context, a system must extract, manipulate and generate the meaning of natural language texts. Question answering against a structured knowledge base, especially when the latter contains interpretable knowledge elements (e.g., instances of events and objects defined in an ontology, not uninterpreted text strings), can attain better results than QA that works by manipulating templates filled with snippets of actual texts – at the least because of the added benefit of disambiguation and reference resolution. The prerequisite for such a system is the existence of a structured knowledge base used as a source of answers to questions. In a real application, the knowledge must be ample and dynamic, so that the knowledge resources must be constantly and promptly augmented. This is not practical if knowledge is acquired entirely by people. Automating structured knowledge acquisition from open text is, therefore, a necessary condition for the success of an advanced QA application. The CSK module of our system is a step toward this functionality, albeit not yet in a fully automatic way. At this point, we rely on TMRs that are obtained automatically but improved through human interaction (see Nirenburg et al. 2004 for details). Note that fully automatic methods for creating structured knowledge of a quality even remotely approaching that needed to support realistic QA do not at this point exist. Few of the numerous current and recent machine learning and statistical processing experiments in NLP deal with the analysis of meaning at all; and those that do address partial tasks (e.g., determining case role fillers in terms of undisambiguated text elements in Gildea and Jurafsky 2002) in a rather “knowledge-lean” manner. The results are very far away indeed from either good quality or good coverage, either in terms of phenomena and text. We believe that our approach, using as it does statistical as well as recorded-knowledge evidence for extracting, representing and manipulating meaning is the most practical and holds the most promise for the future. Indeed, it is not even as expensive as many people believe.

3 The Knowledge Support Infrastructure

The process of deriving TMRs from text is implemented in our Ontosem text analyzer. Semantic analysis in OntoSem is described in some detail in Nirenburg and Raskin 2004; Nirenburg et al., 2004; Beale et al. 1995, 1996,

2003; Mahesh et al. 1997. Our description here will be necessarily brief. Also note that the analysis process is described here as if it were a strict pipeline architecture; in reality, semantic analysis is used to inform and disambiguate syntactic analysis, for example, in cases of prepositional phrase attachment.

Text analysis in OntoSem relies on the results of a battery of pre-semantic text processing modules. The **preprocessor** module deals with mark-up in the input text, finds boundaries of sentences and words, recognizes dates, numbers, named entities and acronyms and performs morphological analysis. Once the morphological analyzer has generated the citation forms for word forms in a text, the system can activate the relevant lexical entries in its lexicons, including the onomasticon (a lexicon of proper names). Figure 2 presents a sample of preprocessor output.

```

- <phrase>
  <word root="*person*" pos="n" features="((type pn) (first tony) (last hall))" repeat="nil"/>
  <word root="arrive" pos="v" features="((tense past) (form past-participle))" repeat="nil"/>
  <word root="in" pos="prep" features="nil" repeat="nil"/>
  <word root="*place*" pos="n" features="((type pn) (city baghdad))" repeat="nil"/>
</phrase>

```

Figure 2: Sample preprocessor output

The task of **syntactic analysis** (see Figure 3) in ontological semantics is, essentially, to determine clause-level dependency structures for an input text and assign grammatical categories to clause constituents (that is, establish subjects, direct objects, oblique objects and adjuncts).

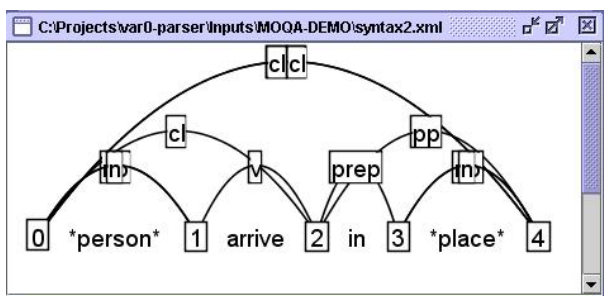


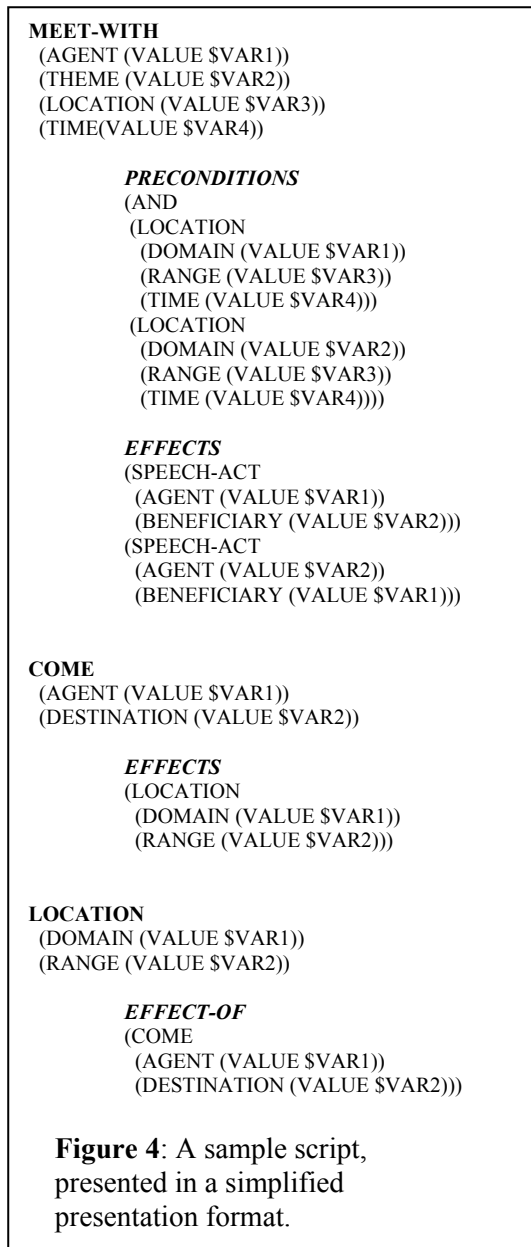
Figure 3: Sample parser output. in graphical

Semantic analysis proper uses the information (mutual constraints) in the active lexicon entries, the ontology and the results of earlier processing to carry out, at the first step, word sense disambiguation and establish basic semantic dependencies in the text. The results are recorded in *basic* TMRs (see below). At the next step, the analyzer determines the values of the various modalities, aspect, time, speech acts, speaker attitudes and other knowledge elements that are

produced in OntoSem using a variety of “microtheories,” to produce *extended* TMRs. At both steps, the analyzer has to deal with ambiguity, incongruity between the input and the expectations recorded in the static knowledge sources (SKSs), unknown words, and non-literal language. In a recent evaluation, the basic analyzer was shown to carry out word sense disambiguation at over 90% and semantic dependency determination at 87% on the basis of correct syntactic analysis and on sentences of an average length of over 25 words with 1.33 unknown words on average per input sentence (see Nirenburg et al., 2004). While not perfect, these results show promise as training data for machine learning work.

The **OntoSem ontology** provides a metalanguage for describing the meaning of the lexical units in a language as well as for the specification of meaning encoded in TMRs. The ontology contains specifications of concepts corresponding to classes of things and events in the world. It is a collection of frames, or named sets of property-value pairs, organized into a hierarchy with multiple inheritance. The expressive power of the ontology and the TMR is enhanced by multivalued fillers for properties, implemented using the “facets” DEFAULT, SEM, VALUE, and RELAXABLE-TO, among others. At the time of this writing, the ontology contains about 6,000 concepts (events, objects and properties), with, on average, 16 properties each. Temporally and causally related events are encoded as values of a complex event’s HAS-EVENT-AS-PART property. These are essentially scripts that provide information that is very useful in general reasoning as well as reasoning for NLP (e.g., Schank and Abelson 1977, Lin and Hovy 2000, Clark and Porter 2000). We use scripts in the answer content determination module of the question answering system. Figure 4 illustrates a rather simple script that supports reasoning for our example question answering session.

The **OntoSem lexicon** contains not only semantic information, it also supports morphological and syntactic analysis. Semantically, it specifies what concept, concepts, property or properties of concepts defined in the ontology must be instantiated in the TMR to account for the meaning of a given lexical unit of input. At the time of writing, the latest version of the English semantic lexicon includes over 12,000 handcrafted entries. These entries cover some of the most complex lexical material in the language – “closed-class” grammatical lexemes such as conjunctions, prepositions, pronouns, auxiliary and modal verbs, etc. as well as about 3,000 of the



most frequent main verbs. We illustrate the structure of the lexicon entry on the example of the first verbal sense of *alert*:

```

alert-v1
cat v
morph regular
ex "He alerted us to the danger"
syn-struct
subject $var1
root "alert"
indirectobject $var2
pp (opt +)
root "to"
object $var3
sem-struct
WARN
agent ^$var1
beneficiary ^$var2
theme ^$var3

```

For lack of space, we will not be able to discuss all the representational and descriptive devices used in the lexicon or the variety of ways in which semantic information in the lexicon and the ontology can interact. See Nirenburg and Raskin (2004, Chapters 7 and 8) for a discussion.

The English **onomasticon** (lexicon of proper names) currently contains over 350,000 entries semantically linked to ontological concepts; it is increasing in size daily by means of semi-automated knowledge-extraction methods.

The **TMR** (automatically generated but shown here in a simplified presentation format) for a short sentence (*He asked the UN to authorize the war*) from a recently processed text about Colin Powell is presented below. The numbers associated with the ontological concepts indicate instances of those concepts: e.g., REQUEST-ACTION-69 means the 69th time that the concept REQUEST-ACTION has been instantiated in the world model used for, and extended during, the processing of this text or corpus.

REQUEST-ACTION-69

AGENT	HUMAN-72
THEME	ACCEPT-70
BENEFICIARY	ORGANIZATION-71
SOURCE-ROOT-WORD	ask
TIME	(< (FIND-ANCHOR-TIME))

ACCEPT-70

THEME	WAR-73
THEME-OF	REQUEST-ACTION-69
SOURCE-ROOT-WORD	authorize

ORGANIZATION-71

HAS-NAME	UNITED-NATIONS
BENEFICIARY-OF	REQUEST-ACTION-69
SOURCE-ROOT-WORD	UN

HUMAN-72

HAS-NAME	COLIN-POWELL
AGENT-OF	REQUEST-ACTION-69
SOURCE-ROOT-WORD	he ; <i>ref. resolution done</i>

WAR-73

THEME-OF	ACCEPT-70
SOURCE-ROOT-WORD	war

The above says that there is a REQUEST-ACTION event whose agent is HUMAN-72 (Colin Powell), whose beneficiary is ORGANIZATION-71 (United Nations) and whose THEME is an ACCEPT event. That ACCEPT event, in turn, has the THEME WAR-73. Note that the concept ACCEPT is *not* the same as the English word *accept*: its human-oriented definition in the ontology as “To agree to carry out an action, fulfill a request, etc”, which fits well here.

The **Fact Repository** contains a list of remembered instances of ontological concepts. For

example, whereas the ontology contains the concept CITY, the fact repository contains entries for London, Paris and Rome; and whereas the ontology contains the concept WAR, the fact repository contains the entry WWII. The main difference between an ontological concept and its instance is the nature of the fillers of properties for each. In the former, the fillers of properties are, typically, overridable constraints; in the latter, they are actual values (when known), or they are left unfilled when not known. A simple fact repository entry is illustrated below:

HUMAN-33599	
NAME	George W. Bush
ALIAS	George Bush, President Bush, George W, the president of the United States, the US president
SOCIAL-ROLE	PRESIDENT
GENDER	male
NATIONALITY	NATION-213 ;(USA)
DATE-OF-BIRTH	July 6, 1946
spouse	human-33966 ;Laura Bush

4 The Question Answering Modules

Referring back to Figure 1, we now will briefly describe the three central question answering modules of Question Analysis, Question Interpretation and Answer Content Determination. Figure 5 shows a question answering session that exemplifies these three stages. The user enters a question in the “Natural Language Query” text box and clicks on the “Submit” button. The OntoSem analyzer is then invoked to analyze the question.

The screenshot shows the MOQA FR Assistant interface. At the top, there is a navigation bar with 'MOQA FR Assistant -- Search Page' and links for 'Refresh', 'Repository Page', 'Back to Welcome Page', and 'Help'. Below this is a 'Natural Language Query' section with a text input field containing 'Who is Tony Hall?' and a 'Submit' button. The 'Query Analysis Details' section shows the following XML-like structure:


```

  <SENTENCE>
  <TEXT_DATA>Who is Tony Hall ?</TEXT_DATA>
  - <TMR_ENTRIES_RESULT src="PREPROCESSOR_RESULT">
  - <XML_DATA>
  - <node label="REQUEST-INFO-116">
  
```

 The 'Query Paraphrase' section displays: 'Search for details about the person named "TONY-HALL".'. The 'Answer Details' section shows: 'The search found 1 match: 1. HUMAN-1000 "TONY-HALL" o COME-1003 (AGENT (...("TONY-HALL")) DESTINATION (...("BAGHDAD"))) o MEET-WITH-1001 (AGENT (...("TONY-HALL")) THEME (...("UMID-MEDHAT-MUBARAK")))'.

Figure 5: Querying about a known person.

The TMR of the question, the result of the Question Analysis module, is displayed in the Query Analysis Details box. Obviously, not many of the details can be seen in these figures, but in the interface, the user can scroll through the TMR output. We will, in fact, be integrating our existing TMR and Fact graphical browsers into this interface in the near future. The results of the next module, Question Interpretation, are then displayed in the Query Paraphrase box. From there, the fact repository is queried, an answer is returned (perhaps utilizing the inference techniques to be described), and the supporting fact (or fact reasoning chain) is displayed in the Answer Details box. Below we present three example sessions, the third of which will be the basis for the more detailed description of the three modules.

For this discussion, we concentrate on the following three sentences that have been processed by the CSK module and the facts that were derived from them and stored in the fact repository (the facts are represented using instances of ontological concepts, of course):

1. Tony Hall Met with Umid Medhat Mubarak.
2. Tony Hall arrived in Baghdad (on Thursday).
3. Ali Atwa was in Baghdad (on April 18 and later).

In Figure 5, the question is “Who is Tony Hall?” In its current state, the system simply finds and responds with the stored fact about Tony Hall, which includes the information about his arrival (the COME-1003 event derived from sentence 2) and the MEET-WITH-1001 event (derived from sentence 1). Figure 6 shows the results of the query, “Did Tony Hall meet with Umid Medhat Mubarak?” A matching fact was found in the FR (derived from sentence 1) and is displayed in the Answer Details. Figure 7 presents a more complex case. The query is “Did Tony Hall meet with Ali Atwa?” The FR contains no fact that can directly answer this question. The system uses facts from sentences 2 and 3 to return the possible inference MEET-WITH-??? (the tentative nature of the inference is marked by the -??? appended as the instance number; and in the

Natural Language Query

Type a question in English:

Query Analysis Details

```
<TEXT_DATA>Did Tony Hall meet with Umid Medhat Mubarak ?</TEXT_DATA>
- <TMR_ENTRIES_RESULT src="PREPROCESSOR_RESULT">
- <XML_DATA>
- <node label="HUMAN-135">
  <node label="AGENT-06">
```

Query Paraphrase

Search for details about meetings involving "TONY-HALL" and "UMID-MEDHAT-MUBARAK".

Answer Details

The search found 1 match:

1. MEET-WITH-1001 (AGENT (...("TONY-HALL")) THEME (...("UMID-MEDHAT-MUBARAK")))

Figure 6: Querying about a known event.

future, we will also generate a numerical value reflecting the confidence level of the inferences and sub-inferences). The supporting reasoning chain is also displayed. We will now use this example to discuss the three main question answering modules.

The **Question Analysis** module takes the input text question and produces the TMR using the resources described in section 3 above. Figure 8 illustrates the resulting TMR in a graphical browser. (One can inspect the content of the various concept instances – e.g., OBJECT-56 – by clicking on graphical objects representing them). The main thing to point out is that the TMR instantiates a MEET-WITH event which is the basis for querying the FR, itself comprised of facts represented by ontological concept instances.

The **Question Interpretation** module then derives the canonical form of an input question TMR (determining question type and reformulating the content of an actual question in a standard format), and applies reference resolution. The answer displayed in Figure 7 involves reformulating the query *Did \$X meet with \$Y?* to *Find meetings involving \$X and \$Y* or more particularly, *Find meetings where a person named \$X is AGENT and a person named \$Y is BENEFICIARY, and meetings where a person named \$Y is AGENT and a person named \$X is BENEFICIARY.* Such a query can be specified as a standard DBMS query for the actual search. The knowledge that we are dealing with people and knowledge of their names is used to help resolve the references

between the instances *Tony Hall* and *Ali Atwa* appearing in the query and the references to *Tony Hall* and *Ali Atwa* that may be stored in the fact repository. We will report on the actual methods of question interpretation and reference resolution we use separately.

The **Answer Content Determination** module is invoked next. The possible queries constructed in the previous module are processed. First, direct queries are attempted.

If an answer is found, it is returned directly. In this example, no fact directly states that Tony Hall met Ali Atwa. Scripts are then activated which allow us to reason about the question. In the script of Figure 4, the preconditions of a MEET-WITH event include both participants having to be in the same place at the same time. This will invoke a series of queries that will determine if Tony Hall and Ali Atwa were indeed in the same location at the same time. In general, if the preconditions of an event are satisfied, we infer that the event itself possibly took place. In this case, the fact that Ali Atwa was in Baghdad is present in the FR by virtue of sentence 3 above. Using this knowledge, the system seeks to prove that Tony Hall was also in Baghdad at the same time. Once again, there is no direct fact that states this. However, the facts about Tony Hall include information that he arrived in Baghdad at a certain time (at the present time, we do not attempt to match the times of the facts, although this will be a focus of our ongoing work).

MOQA FR Assistant -- Search Page [Refresh | Repository Page | Back to Welcome Page | Help]

Natural Language Query

Type a question in English:

Query Analysis Details

```
<?xml version="1.0" encoding="UTF-8" ?>
- <QUERY_DOCUMENT date="March 31, 2004" time="12:55:34 PM">
- <DOCUMENT>
- <FILE />
```

Query Paraphrase

Search for details about meetings involving "TONY-HALL" and "ALI-ATWA".

Answer Details

The search found 1 possible match for:

```
MEET-WITH-??? (AGENT (...("TONY-HALL")) THEME (...("ALI-ATWA")))
```

1. because COME-1003 (AGENT (...("TONY-HALL")) DESTINATION (...("BAGHDAD"))) and LOCATION-1006 (DOMAIN (...("ALI-ATWA")) RANGE (...("BAGHDAD")))

Figure 7: Querying about an unknown event.

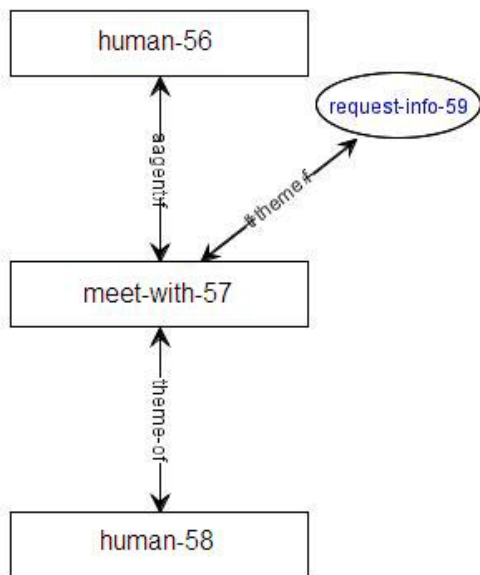


Figure 8. Sample output viewed through the TMR browser

Matching times of This information is represented by the COME-1003 fact. We can look up COME in the script of Figure 4 and see that one effect of a COME event is that the agent's location becomes the destination of the COME event. In general, we can use known facts to infer additional facts about their effects. In this case, we can infer that Tony Hall was, in fact, in Baghdad, which, in turn, allows us to make the top level inference that he might have met with Ali Atwa, who we previously determined was also in Baghdad. We are aware that the conjecture about the possible meeting should involve additional knowledge of the background and histories of the participants (e.g., if a cobbler and a head of state are in the same place at the same time, that does not imply a potential meeting between them). We are working on enhancing the knowledge (centered on the ontological MEET-WITH script) to improve such reckoning.

In a separate article in preparation, we will go into much more detail about the reasoning process. There are obviously many additional issues, including:

- events. Our time resolution meaning procedures enable this;
- Assigning probabilities to inferences. For example, if two people were in the same room, the possibility of their meeting is much higher than if they were in the same country;
- Controlling the inference process.

With regard to this last issue, the OntoSem environment provides a useful mechanism. In particular, the scripts that we are developing encode expectations and are meant to constrain and

direct the inference process – even we are fully aware of the abductive, defeasible nature of this knowledge.

The inference steps described above were directed by the information in the MEET-WITH and COME scripts. Also, known facts about one of the participants, Tony Hall, were used to direct queries to support a possible inference. Obviously, much work remains to be done. We must populate a large fact repository that should include a large collection of facts about individuals as well as places, organizations and event instances. At this time, we are starting to use our TMR production environment for extracting facts. We hope to be able to report on the progress of this work at the workshop.

5 Conclusion

We have presented the first experiment with a knowledge-based QA system in which text processing is integrated with reasoning on the basis of shared knowledge and processing infrastructures. Indeed, the same processing and knowledge resources in the system carry out reasoning for the purposes of QA and reasoning that is necessary to create a high-quality unambiguous text meaning representation itself. While this is just a small experiment, we have specific and, we believe, realistic plans for scaling this system up – through automatic population of the fact repository, semi-automatic enlargement of the lexicons and the ontology and expansion of the inventory of scripts.

We believe that integrating a comprehensive throughput system for an advanced application, even one in which some of the modules are still on a relatively small scale, is a very important kind of work in our field. It tackles real problems head on, without resorting to a rather defeatist – though quite common in today's NLP – claim that certain goals are infeasible.

References

- S. Beale, S. Nirenburg and K. Mahesh. 1995. *Semantic analysis in the Mikrokosmos machine translation project*. In Proceedings of the 2nd Symposium on Natural Language Processing, Kaset Sart University, Bangkok, Thailand.
- S. Beale, S. Nirenburg and K. Mahesh. 1996. *Hunter-Gatherer: Three search techniques integrated for natural language semantics*. In Proceedings of the 13th National Conference on Artificial Intelligence. Portland, OR.
- S. Beale, S. Nirenburg and M. McShane. 2003. *Just-in-time grammar*. In Proceedings HLT-NAACL-2003, Edmonton, Canada.

- P. Clark and B. Porter. 2000. *\$RESTAURANT Reⁿ-visited: A KM implementation of a compositional approach*. Technical Report, AI Lab, University of Texas at Austin.
- D. Gildea and D. Jurafsky. 2002. *Automatic labeling of semantic roles*. Computational Linguistics 28(3). 245-288.
- C. Lin and E. H. Hovy. 2000. *The automated acquisition of topic signatures for text summarization*. In Proceedings of the COLING Workshop on Text Summarization. Strasbourg, France.
- K. Mahesh, S. Nirenburg and S. Beale. 1997. *If you have it, flaunt it: Using full ontological knowledge for word sense disambiguation*. In Proceedings of Theoretical and Methodological Issues in Machine Translation (TMI-97). Santa Fe, NM.
- S. Nirenburg and V. Raskin. 2004. *Ontological Semantics*. MIT Press.
- S. Nirenburg, M. McShane and S. Beale. 2004. *Evaluating the performance of OntoSem*. In Proceedings ACL Workshop on Text Meaning and Interpretation, Barcelona.
- R. Schank and R. Abelson. 1977. *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Erlbaum.