

# MOQA Fact Repository Assistant

<sup>1</sup>Benoit Lavoie, <sup>2</sup>Steve Beale and <sup>2</sup>Sergei Nirenburg

<sup>1</sup>CoGenTex, Inc.  
840 Hanshaw Road Suite 1, Ithaca, NY 14850  
benoit@cogentex.com

<sup>2</sup>University of Maryland, Baltimore County  
Institute for Language and  
Information Technology  
1000 Hilltop Circle,  
Baltimore, MD 21250  
{sbeale, sergei}@cs.umbc.edu

## Abstract

This paper describes an implemented prototype, MOQA Fact Repository Assistant, that builds automatically semantic fact repositories (FR) from raw English text, and that offers question-answering functionality by returning facts matching natural language (NL) queries. We focus on the Fact Extraction component used to process raw texts and NL queries, which is implemented using off-the-shelf rule-based and statistical analysis components.

## 1 Introduction

This paper describes an implemented prototype, MOQA Fact Repository Assistant, an application developed in relationship to the ARDA AQUAINT project: MOQA (Meaning-Oriented Question-Answering with Ontological Semantics; Cowie et al., 2004). This application extracts semantic fact repositories (FR) from raw English text, and offers question-answering functionality by returning facts matching NL queries. NL queries are processed by the same components used for the fact extraction from text so that they can be directly matched to the fact repository to facilitate the answer content determination. We focus on the fact extraction component used to process raw texts and NL queries, which is implemented using off-the-shelf rule-based and statistical analysis components.

Section 2 describes the overall system architecture. Section 3 provides a detailed description of the fact extraction. Section 4 describes the question-answer functionality. Section 5 discusses the results, status and related works.

## 2 Overall System Description

The overall system architecture is described in Figure 1. It consists of two main components: (i) a Fact Extraction component (used to process either raw texts or natural language queries), and (ii) a Question-Answering component. Below we provide a description of these components and focus more particularly on the Fact Extraction component.

The Fact Extraction component creates repositories from texts with semantic facts extracted from the texts, and links from those facts to the source texts. To simplify the implementation and facilitate the answer content determination, the system uses the same Fact Extraction component to process raw texts and NL queries; the facts extracted from the NL queries can easily be mapped to the facts extracted from the raw texts by pattern matching for the answer content determination. The Fact Extraction component is described in more detail below.

The Question-Answering uses the Fact Extraction component to extract facts from the NL

query, and matches those facts with those extracted from texts in order to retrieve the facts answering the question. The answer is currently formulated simply by displaying those facts.

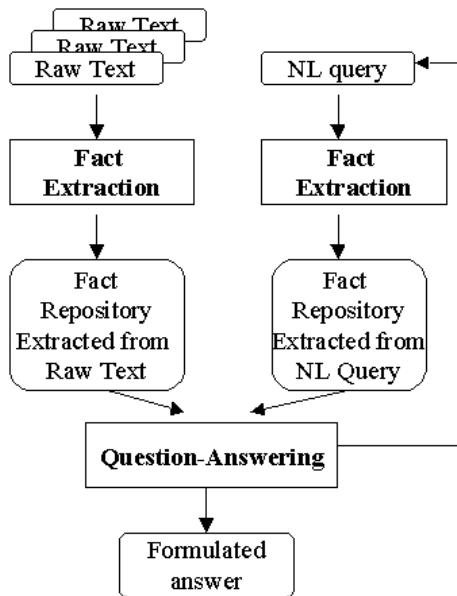


Figure 1. Overall System Architecture

### 3 Fact Extraction

The Fact Extraction component is implemented using three (3) off-the-shelf analysis components:

- Stanford's Lexicalized Parser (Klein and Manning, 2003): a probabilistic parser combining results from phrase structure and lexical dependency models using an A\* algorithm.
- NMSU-CRL's (2003) Text Pre-processor: a context-free lexical tagger assigning words and word sequences with all their possible lexical features (part-of-speech, inflected form, etc...), providing special support for important constructions such as dates and proper nouns.
- UMBC-ILIT (2004) Syntactic and Semantic Analyzer: generates Text Meaning-Representations (TMR) from tagged texts using ontological semantics.

UMBC-ILIT's analyzer already provided support for the output produced by the NMSU-CRL's Text Pre-processor. We integrated with Stanford's

Lexicalized Parser in order to help filter some incorrect tagged results produced by the Text Pre-processor that sometimes caused problems with UMBC-ILIT's analyzer.

Figure 2 illustrates the design of the Fact Extraction main component. The processing is described below.

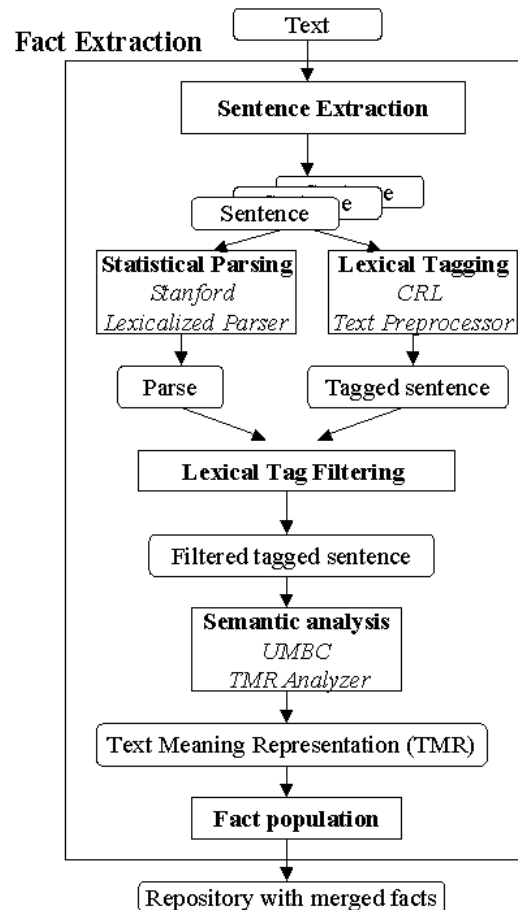


Figure 2. Fact Extraction Component

The Sentence Extraction component first extracts individual sentences using heuristics based on punctuation marks. Text can be in plain format or be annotated with HTML/XML tags (in which case XML parsing is done to extract the sentence content).

The extracted sentences are then processed independently using NMSU-CRL's Text Pre-processor and Stanford's Lexicalized Parser. Figure 2 illustrates the tagged sentence obtained for the sentence *Tony Hall arrived in Baghdad on June 2, 2000*. Figure 3 illustrates the parse obtained for the same sentence.

```

((0,3;5,8), root='tony hall
tony',post=pn,type=definite,dict=err,text_form='To
ny Hall Tony')
((0,3;5,8), root='tony
hall',post=pn,type=name,case=capitalized,dict=names,te
x
t_form='Tony Hall')
((5,8),
root='hall',post=pn,case=capitalized,dict=names,text_for
m='Hall')
...
((10,16), root='arrive',post=v,form=past-
participle,case=lower,dict=pos,text_form='arrived')
((18,19),
root='in',post=adv,case=lower,dict=pos,text_form='in')
((18,19),
root='in',post=prep,case=lower,dict=pos,text_form
= 'in')
((21,27),
root='baghdad',post=pn,type=city,case=capitalized,
dict=places,text_form='Baghdad')
((29,30),
root='on',post=adv,case=lower,dict=pos,text_form='on')
((29,30),
root='on',post=prep,case=lower,dict=pos,text_form
= 'on')
((32,35;37,38;40,40;42,45), root='(year 2000) (month
06) (day 12)',post=date,dict=err,text_form='June
12 , 2000')
((32,35),
root='june',post=pn,case=capitalized,dict=names,text_for
m='June')
((37,38), root='12',post=num,dict=err,text_form='12')
((40,40), root=',',post=punct,dict=err,text_form=',')
((42,45), root='(year 2000) (month 00) (day
00)',post=date,dict=err,text_form='2000')
((42,45), root='2000',post=num,dict=err,text_form='2000')
((47,47), root='.',post=punct,dict=err,text_form='.')

```

Figure 2. Tagged Sentence Sample

The tagged sentence illustrated in Figure 2 lists all possible word sequence analyses, but those appearing in bold are the best ones to provide as input to UMBC-ILIT's analyzer. The ordering of the analysis results indicates that except for the prepositions *in* and *on*, all word sequences are properly analyzed, including the proper noun and the date analyzed in terms of month, day and year. The parse illustrated in Figure 3 shows the best overall analysis but it does not provide an analysis of the date needed by the semantic analyzer.

The next step consists of filtering the tagged sentence using the parse and simple heuristics: e.g. if more than one alternative tagged word/expression analysis can be matched with a given parse node based on the word form and word positioning, then only the tagged analyses matching this parse node best based on the part-of-speech are kept. This heuristic can be used to eliminate the analyses of *in* and *on* as adverbs (*post=adv*) in the specification of Figure 2.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <node pos="S">
- <node pos="NP">
- <node pos="NNP">
  <node label="Tony" />
</node>
- <node pos="NNP">
  <node label="Hall" />
</node>
</node>
- <node pos="VP">
- <node pos="VBD">
  <node label="arrived" />
</node>
- <node pos="PP">
- <node pos="IN">
  <node label="in" />
</node>
- <node pos="NP">
- <node pos="NP">
- <node pos="NNP">
  <node label="Baghdad" />
</node>
</node>
</node>
- <node pos="PP">
- <node pos="IN">
  <node label="on" />
</node>
- <node pos="NP">
- <node pos="NNP">
  <node label="June" />
</node>
- <node pos="CD">
  <node label="2" />
</node>
- <node pos=",">
  <node label="," />
</node>
- <node pos="CD">
  <node label="2000" />
</node>
</node>
</node>
</node>
</node>
</node>
- <node pos=",">
  <node label="." />
</node>
</node>

```

Figure 3. Parse Sample

The filtered tagged sentences are then passed to UMBC-ILIT's analyzer to produce the corresponding TMR. Figure 4 illustrates an XML extract of the TMR for the sentence analyzed above (due to space constraints, the Figure only displays nodes that are relevant for our discussion). In this TMR extract, HUMAN-166 is an instance of the concept HUMAN appearing in the text with the root word TONY HALL and known as the concept labeled (HAS-NAME) TONY HALL. This instance is the agent of COME 165, an instance of the concept COME whose destination is CITY-16 (Baghdad, not defined in Figure 4). Those instances are also assigned with word position in the text not illustrated in the Figure. More details about TMRs can be found in UMBC-ILIT (2004).

```

- <node label="HUMAN-161">
- <node label="AGENT-OF">
- <node label="VALUE">
  <node label="COME-165" />
</node>
</node>
- <node label="HAS-NAME">
- <node label="VALUE">
  <node label="TONY-HALL" />
</node>
</node>
- <node label="ROOT-WORDS">
- <node label="VALUE">
  + <node label="TONY-HALL">
</node>
</node>
+ <node label="WORD-NUM">
- <node label="INSTANCE-OF">
- <node label="VALUE">
  <node label="HUMAN" />
</node>
</node>
</node>
- <node label="COME-165">
- <node label="DESTINATION">
- <node label="VALUE">
  <node label="CITY-163" />
</node>
</node>
- <node label="AGENT">
- <node label="VALUE">
  <node label="HUMAN-161" />
</node>
</node>
+ <node label="TIME">
- <node label="ROOT-WORDS">
- <node label="VALUE">
  <node label="ARRIVE" />
</node>
</node>
+ <node label="WORD-NUM">
- <node label="INSTANCE-OF">
- <node label="VALUE">
  <node label="COME" />
</node>
</node>
</node>

```

Figure 4. TMR Extract for *Tony Hall arrived in Baghdad*

```

- <node label="HUMAN-110">
- <node label="AGENT-OF">
- <node label="VALUE">
  <node label="MEET-WITH-111" />
</node>
</node>
- <node label="HAS-NAME">
- <node label="VALUE">
  <node label="TONY-HALL" />
</node>
</node>
+ <node label="ROOT-WORDS">
+ <node label="WORD-NUM">
- <node label="INSTANCE-OF">
- <node label="VALUE">
  <node label="HUMAN" />
</node>
</node>
</node>
- <node label="MEET-WITH-111">
- <node label="TIME">
- <node label="VALUE">
  - <node label="THEME">
    - <node label="VALUE">
      <node label="HUMAN-112" />
    </node>
  </node>
  - <node label="AGENT">
    - <node label="VALUE">
      <node label="HUMAN-110" />
    </node>
  + <node label="<">
  </node>
</node>
</node>
- <node label="ROOT-WORDS">
- <node label="VALUE">
  <node label="MEET" />
</node>
</node>
+ <node label="WORD-NUM">
- <node label="INSTANCE-OF">
- <node label="VALUE">
  <node label="MEET-WITH" />
</node>
</node>
</node>

```

Figure 5. TMR Extract for *Tony Hall met with ...*

Figure 5 illustrates another TMR extract for the sentence *Tony Hall met with Umid Medhat Mubarak*. In this case, Tony Hall is represented by the instance HUMAN-110 and the meet event is represented by the instance MEET-WITH-111.

The Fact Population component processes such TMRs in order to extract facts that can be added or merged to the current facts. This processing is done following these steps:

- First, a copy is made of each concept instance (HUMAN-161, HUMAN-110, COME-165, MEET-WITH-111, CITY-163, etc) with all their conceptual dependencies that are not sentence specific (such as ROOT-WORDS and WORD-NUM illustrated in Figure 4);
- Each copied instance represents is a potential fact entry. It is compared with the existing entries, in order to be added or merged with those entries. The merging is currently done only for concepts that are not events such as HUMAN, CITY, etc. and require concept types (HUMAN) and concept names (TONY HALL) to match. Although this is not currently implemented, the assumption is that concepts instances referring to different objects will be assigned with different names.
- References to the original TMR nodes from which facts content we extracted is Each new Content added or merged to the facts also contain references to the concept instances in the TMR where the fact content was derived.

Figure 6 illustrates an extract of the facts obtained from the specification illustrated in Figure 4 and Figure 5. This specification contains an entry for a HUMAN instance, HUMAN-1007, with name TONY HALL and which is derived from the TMR concept instances HUMAN-110 and HUMAN-161. HUMAN-1007 is agent of both MEET-WITH-1008 and COME-1010 (illustrated on Figure 6) where COME-1010 is derived from the TMR concept instance COME-165.

```

- <node id="1007" inst="HUMAN" label="HUMAN-1007" name="TONY-HALL"
ref="HUMAN-110,HUMAN-161" />
- <node label="AGENT-OF" ref="HUMAN-110" />
- <node label="VALUE" />
  <node id="1008" inst="MEET-WITH" label="MEET-WITH-1008" />
</node>
- <node label="AGENT-OF" ref="HUMAN-161" />
- <node label="VALUE" />
  <node id="1010" inst="COME" label="COME-1010" />
</node>
- <node label="HAS-NAME" ref="HUMAN-110,HUMAN-161" />
- <node label="VALUE" />
  <node label="TONY-HALL" />
</node>
- <node label="INSTANCE-OF" ref="HUMAN-110,HUMAN-161" />
- <node label="VALUE" />
  <node label="HUMAN" />
</node>
- <node id="1010" inst="COME" label="COME-1010" ref="COME-165" />
- <node label="DESTINATION" />
- <node label="VALUE" />
  <node id="1011" inst="CITY" label="CITY-1011" />
</node>
- <node label="AGENT" />
- <node label="VALUE" />
  <node id="1007" inst="HUMAN" label="HUMAN-1007" />
</node>
- <node label="TIME" />
- <node label="VALUE" />
  <node label="<" />
  <node label="FIND-ANCHOR-TIME" />
</node>
- <node label="INSTANCE-OF" />
- <node label="VALUE" />
  <node label="COME" />
</node>
</node>

```

Figure 6. Facts Extract Sample

The facts are stored in a XML repository with the texts and TMRs from which those facts were derived. The system provides a simple interface that lets the user or developer create and visualize fact-based repositories from text files or user-defined texts. This interface is illustrated in Figure 7.

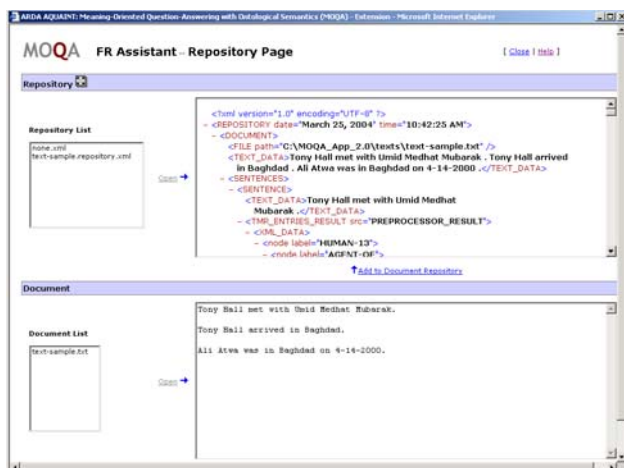


Figure 7. Repository Interface

## 4 Question-Answering Functionality

The system currently has a basic question-answering functionality integrated with the automatically extracted fact-based repository. This functionality is supported by the interface illustrated in Figure 8.



Figure 8. Question-Answer Interface to the Repository

The interface can display:

- The current user-defined NL query
- The representation of the analyzed query (described below);
- A simple paraphrase of the analyzed query (to let the user know if the system has understood his/her query);
- A simple display of the facts answering the query.

The NL query is first analyzed using the same Fact Extraction component used to extract facts from raw texts. This approach was used for two practical reasons: to save time by reusing the same components, and to facilitate the answer determination by comparing similar representations (facts extracted from the NL query and facts extracted from raw texts) – see below.

Then, the facts extracted from the NL query are compared with the facts extracted from the raw texts in order to unify the references. As a result, in the analyzed query, Tony Hall will be assigned with the same reference it has in the facts extracted from raw texts, if found. Otherwise, a feature (*known="false"*) will indicate that nothing is known about this instance in the extracted texts.

Figure 8 illustrates the facts extracted from the query *Who is Tony Hall?* The fact instance re-labeled “HUMAN-1007” (from HUMAN-1001) indicates that this instance is known in the facts extracted from raw text. The fact instance labeled REQUEST-INFO-1000 indicates the nature of the query (an IS-A question about the theme).

```

- <node id="1007" inst="HUMAN" label="HUMAN-1007"
  OLD-ref="HUMAN-1001" name="TONY-HALL"
  ref="HUMAN-124" root-word="*PERSON*">
- <node label="HAS-NAME">
- <node label="VALUE">
  <node label="TONY-HALL" />
  </node>
</node>
- <node label="THEME-OF">
- <node label="VALUE">
  <node id="1000" inst="REQUEST-INFO"
  label="REQUEST-INFO-1000" />
  </node>
</node>
- <node label="INSTANCE-OF">
- <node label="VALUE">
  <node label="HUMAN" />
  </node>
</node>
</node>
- <node id="1000" inst="REQUEST-INFO" known="false"
  label="REQUEST-INFO-1000" ref="REQUEST-INFO-
  123" root-word="WHO">
- <node label="THEME">
- <node label="VALUE">
  - <node label="DOT">
    <node label="IS-A" />
  </node>
  </node>
</node>
</node>
- <node label="INSTANCE-OF">
- <node label="VALUE">
  <node label="REQUEST-INFO" />
  </node>
</node>
</node>

```

Figure 9.

Facts Extracted from the Query: *Who is Tony Hall?*

The paraphrasing of the analyzed query is done using a simple template-based approach. Figure 10 illustrates a simplified version of a template that can be matched with the facts in order to generate “*Search for details about the person named “TONY HALL” where <value-of type=“\$X”/> will give person if \$X = HUMAN, and \$X as default.*

```

<IF>
  <node inst=" $X" name=" $Y">
    <node label="THEME-OF">
      <node label="VALUE">
        <node inst="REQUEST-INFO">
          </node>
        </node>
      </node>
    </node>
  </IF>
  <THEN>
    Search for details about the <value-of
    type=" $X"/> named "<value-of
    name=" $Y"/>".
  </THEN>

```

Figure 10. Paraphrase Template Sample

The answer content determination also relies on a simple template based approach. There are two main types of query pattern currently supported:

- Who/What is \$X?  
e.g. *Who is Tony Hall? What is Baghdad?*
- Did/Do/Will/... \$Y?  
e.g. *Did Tony Hall met with Umid Medhat Mubarak?*

For query like *Who/What is \$X?*, the answer content determination will return the fact extracted from the raw text that has the same reference as the fact \$X in the analyzed query.

For query like *Did/Do/Will/... \$Y?*, the answer content determination will return the facts extracted from the raw text that has the same type as \$Y (e.g. concept MEET-WITH), and where conceptual actants in \$Y, can be matched with those instances.

Figures 10 and 11 illustrate two formulated answers obtained for the above mentioned queries.

**The search found 1 match:**

1. HUMAN-1000 "TONY-HALL"
  - COME-1010 (AGENT (...("TONY-HALL"))  
DESTINATION (...("BAGHDAD")))
  - MEET-WITH-1008 (AGENT (...("TONY-HALL"))  
THEME (...("UMID-MEDHAT-MUBARAK")))

Figure 10. Answer Formulation for Facts Returned to Query: *Who is Tony Hall?*

**The search found 1 match:**

1. MEET-WITH-1008 (AGENT (...("TONY-HALL"))  
THEME (...("UMID-MEDHAT-MUBARAK")))

Figure 11. Answer Formulation for Facts Returned to Query: *Did Tony Hall met with Umid Medhat Mubarak?*

## 5 Results, Status and Related Work

The current results obtained with the system currently implemented are still rudimentary, since the end-to-end integration was just completed and further refinement is still needed to improve the integration between the components; the filtering of the tagged sentences with the parses can still be improved greatly, the patterns used for the question-answering functionality can also be significantly refined, etc.

The automatic fact extraction can currently be applied on any raw text in plain format or annotated with XML/HTML tags, but performance can be slow for long texts, and sentences that are too complex are not currently (fully) analyzed. The extraction also works best for sentences about meetings, people or locations, and involving no referential expression. NL queries are currently limited to patterns like *Who/What is \$X?* and *Did/Do/Will/... \$Y?*

**A detailed evaluation will be included in the final version of this paper.**

In a related work (Cowie et al., 2004), an implementation of the Fact Extraction was done using only NMSU-CRL's Text Pre-processor and UMBC-ILIT's analyzer. However, the process was not fully automated, in part because of the

incomplete support of the analyzer to handle the incorrect tagging produced by the context-free Text Pre-processor. An additional reason was that the representation of the extracted facts had a formalism very different from those of the TMRs and the mapping between those formalisms was incomplete. This work extends that previous work by fully automating the extraction process.

## Acknowledgements

This work was partially funded by the ARDA AQUAINT program under contract number 2002\*H167200\*000. We are grateful to Richard Kittredge and Ted Caldwell for helpful comments on the current draft.

## References

- Jim Cowie, Sergei Nirenburg, Tanya Korelsky, Stephen Helmreich, Benoit Lavoie, Valeriy Sibirtsev, Steve Beale and Ted Caldwell. To appear. MOQA: Meaning-Oriented Question-Answering. *Proceedings of RIAO 2004*.
- Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.
- NMSU-CRL. 2003. *English Pre-processor Overview: Version 3.4a*. Technical report of New Mexico State University, Computing Research Laboratory.
- UMBC-ILIT. 2004. *Introduction to the UMBC-ILIT Syntactic and Semantic Analyzer*. Technical report of University of Maryland, Baltimore County, Institute for Language and Information Technology.