Acquisition Of New Knowledge In TutorJ

Giuseppe Russo, Arianna Pipitone and Roberto Pirrone

Dept. of Computer Science and Engineering – University of Palermo

Department of Computer Science and Engineering (DINFO) - University of Palermo, Italy
Outline

- TutorJ
- Knowledge base definition and extension
- Proposed methodology
- Application Scenario
- Conclusions and future works
TutorJ is an Intelligent Tutoring System able to interact with users to assess their skills, and to improve their knowledge in a particular domain.

The system’s architecture is inspired to the HIPM cognitive model.
TutorJ
The Knowledge Base of TutorJ makes use of an ontology coded in OWL-DL.

An ontology doesn’t evolve during time.

The previous version of TutorJ
- was unable to reply to questions that were outside the ontology boundaries;
- was not capable to increase its knowledge with respect to the interaction with users, adding new external information sources.
Our goals

We want to add new items on the fly to the knowledge base of TutorJ.

We focus to increase the knowledge base of the system as a consequence of the interaction with users (negative JOL/FOK) and using external information sources (like an online wiki).
Two main problems have to be faced to increase effectively a structured knowledge base using external information sources:

- semi-structured knowledge sources (i.e. a wiki) have to be transformed into a structured form (an ontology-like structure);
- new structured knowledge has to be aligned to the existing knowledge base to use them together.
We use semantic wikis as external information sources.

A Semantic wiki is a wiki with semantic annotations in the hypertextual links of the pages. Many systems have been presented in literature that allow to relate automatically a web page to an ontology. The key idea is to have semantically annotated resources in order to perform a better retrieval process.
Matching of knowledge

The definition of a **semantic similarity measure** to correlate different portions of ontologies is crucial in ontology matching.

**Ontology Matching** can be defined as the set of operations that are applied to define a multiple correspondences between some elements in the first ontology to other elements in the second one referring to a semantic similarity measure.
Formally an alignment between an ontology $O$ and a second ontology $O'$ is the quadruple

$$< e; e'; s; n >$$

where

- $e$ is an element in $O$;
- $e'$ is an element in $O'$;
- $s$ is the relation between elements that defines the similarity;
- $n \in [0..1]$ is the level of confidence (or strength) of the relation.
Acquisition of New Knowledge

Our knowledge acquisition process relies on meta-cognitive strategies.

It is triggered by a negative JOL/FOK:

- when receiving a negative feedback about the presentation of a concept, the system enables a procedure to find new knowledge related to the topic under investigation.
Proposed methodology

We define four macro-steps in our procedure:

1. user requests are inferred from the dialogue performed through the chatbot interface;
2. on-line services are used to retrieve knowledge about a negative JOL/FOK topic;
3. new knowledge is mapped into the domain ontology;
4. results are shown to the user.

The procedure is executed iteratively until the student’s feedback becomes positive so the focus of conversation is moved to another concept.
Implementation of methodology (1)

Inner ontology codes the knowledge the system already owns about the query domain.

Our approach makes use of wikis as a primary source of new knowledge.
STEP 1

- The Chatbot component in TutorJ manages the dialogue with the user
  - deterministic dialogue manager (1st order logic rules)
  - POMDP-based stochastic dialogue manager (under development) to gather uncertainty in the dialogue

- Interesting concepts are extracted from user's most relevant sentences. The selected concepts are candidates to be further investigated.
Implementation of methodology (3)

STEP 2

The system evaluates the existence of some related document in an external wiki.

In our application scenario, we refer to semantic wiki

STEP 3

System calls on line service to export wiki page in an ontology.

We use the function

- [http://semanticweb.org/wiki/Special::ExportRDF](http://semanticweb.org/wiki/Special::ExportRDF)
Fragment of Java code to call online service on a topic:

```java
//topic is the inferred concept from user's sentences
String urlstr = "http://semanticweb.org/wiki/Special:ExportRDF/"+topic;

//Urlstr is the url string of online service
URL url = new URL(urlstr);

//create connection with online service
URLConnection conn = url.openConnection();

//isr is the input stream ontology turned from online service
InputStreamReader isr = new InputStreamReader(conn.getInputStream());

//br is the buffer for isr
BufferedReader br = new BufferedReader(isr);
```
STEP 4

System maps ontology gathered from online service with its inner domain ontology and expands its nodes.

The system calls a mapper module to do this:

```java
/*parFileArray is the parameter file with specification of the ontologies to be mapped*/

Align.main(parFileArray);
```
Scenario (1)

We use the system in art domain.

The fragment of inner domain ontology on arts regarding a property for the concept Artist is:

```xml
<owl:ObjectProperty rdf:ID="has_created">
  <rdfs:domain rdf:resource="#Artist"/>
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="created_by"/>
  </owl:inverseOf>
  <rdfs:range rdf:resource="#Opera"/>
</owl:ObjectProperty>
```
Scenario (2)

The user asks something about Michelangelo Buonarroti.

The system infers by its ChatBot that Michelangelo Buonarroti is an artist, and he could be an instance for the concept Artist in the ontology.
Scenario (3)

The system calls Owl/Rdf feeds from online service about Michelangelo

Here follows an ontology fragment returned by Semantic Wiki:

```xml
<swivt:Subject rdf:about="http://semanticweb.org/id/Michelangelo">
  <rdfs:label>Michelangelo</rdfs:label>
  <swivt:page rdf:resource="http://semanticweb.org/wiki/Michelangelo"/>
  <property:Born_place rdf:resource="http://semanticweb.org/id/ Caprese"/>
</swivt:Subject>
```
Scenario (3)

**Conversation with TutorJ**

**user** said: a other information

**TutorJ** said:

No. I am sorry. I could try to make a simple search in the web for you. Wait a moment.....

**Other data have been gained**

- **Father**: Lodovico di Leonardo di Buonarroti di Simoni, and Judicial administrator
- **Influenced by**: Marsilio Ficino
- **Life place**: Venice, Bologna, and Rome
- **Mother**: Francesca di Neri del Miniato di Siena
- **Studied at**: The Humanist academy

**Dialogue History**

**TutorJ**>

you> a other information

**TutorJ**> No. I am sorry. I could try to make a simple search in the web for you. Wait a moment.....
In our domain we don’t know who is Michelangelo's father. The inner domain ontology doesn’t contain this property. So we can add the property in order to obtain new knowledge.

The output of our procedure in this case is the creation of a new node:

```xml
<owl:ObjectProperty rdf:ID="father">
  <rdfs:domain rdf:resource="#Artist"/>
  <rdfs:range rdf:resource="#Person"/>
</owl:ObjectProperty>
```

and new data tables are created in the instances database.
Conclusions and Future Works

A methodology has been presented to increase TutorJ knowledge that is triggered by negative JOL/FOK about the domain.

Acquisition of new knowledge is achieved through an alignment process with semi-structured wiki data.

Future works are focused to the extraction of new knowledge directly from Wikipedia.
Thanks!

russo@dinfo.unipa.it,

pipitone@dinfo.unipa.it,

pirrone@dinfo.unipa.it