Managing Conversation Uncertainty in TutorJ

Vincenzo Cannella, and Roberto Pirrone

Dept. of Computer Science and Engineering – University of Palermo
Outline

- TutorJ & WikiArt
- Extensions to the Chatbot Module for dialogue uncertainty management
- Support to gather new data sources to cope with negative JOL/FOK in the conversation
- Examples of dialogue management (video)
From TutorJ to WikiArt
From TutorJ to WikiArt

TutorJ

Intelligent Tutoring System
From TutorJ to WikiArt

TutorJ

Intelligent Tutoring System

Web-based system
From TutorJ to WikiArt

- TutorJ
- Intelligent Tutoring System
- Web-based system

WikiArt: latest version of TutorJ as an ontology-based tutorial system for Arts
From TutorJ to WikiArt

- **TutorJ**
- **Intelligent Tutoring System**
- **Web-based system**

- **WikiArt**: latest version of TutorJ as an ontology-based tutorial system for Arts

- **Treated topics**: artists, their lives and relationships, works of arts, artistic movements.

Department of Computer Science and Engineering (DINFO) - University of Palermo, Italy
inspired to the Human Information Processor Model (HIPM)

Three kinds of modules:
- perceptual modules
- cognitive modules
- motor modules
System Overview

Department of Computer Science and Engineering (DINFO) - University of Palermo, Italy
GraphBot is conversational agent able to merge two interaction modalities:

- **Graphical Interaction**
  - 2D Zooming User Interface (document browsing)
  - GUI widgets that are suitable to the current dialogue move

- **Natural Language**
  - Chat
The new Chatbot Module

Client Side

Chatbot

Netica

Core

OWL API

WN API

Extended AIML Files

Server Side

Bayesian net desc

DB

OWL

WN

Department of Computer Science and Engineering (DINFO) - University of Palermo, Italy
An example of AIML code

```xml
<category>
  <pattern>Yes, I do</pattern>
  <template>
    <that>Do you like music?</that>
    Which artist, in particular?
  </template>
</category>

... System: Do you like music?  
User: Yes, I do  
System: Which artist, in particular?  
...
TutorJ had only short-term conversation planning

- single dialogue move
- context as the sequence of past sentences
New Extensions in the Chatbot module

- Stimulus-response system vs goal-driven system
  - Immediate goal vs final goal

The system is now able to plan the interaction sequence
<category>
<pattern>Yes, I do</pattern>
<goal></goal>
<type></type>
<template>
<that>Do you like music?</that>
Which artist, in particular?
</template>
</category>

Pattern/Goal/Type Matching System

Pragmatics has proposed several possible generic interaction acts goals.

Such goals have been organized into classes and hierarchies too.

With the goal tag, we defined a generic framework allowing to adopt any possible collection of goals, organized into any possible manner.

***

Type tag defined to cover many different chatbot’s behavior (interaction modality, ood, ...)

***

Management of the immediate goal - No long-term planning

Backward compatibility
Two kinds of reasoning investigated

- Deterministic one
  - Prolog language;

- Probabilistic one
  - Belief networks (Netica);
  - Partially Observable Markov Decision Process (POMDP)

Reinforce the understanding of user’s sentences
New Extensions in AIML (Netica)

- Tags to manage a belief networks:
  - `<loadBN>` to load a single bayesian net;
  - `<enterState>` to assert a fact;
  - `<enterStateNot>` to deny a fact;
  - `<retractFindings>` to reset the net;
  - `<delBN>` to delete the net;
  - `<getBelief>` to evaluate the probability of some particular event;
New Extensions in AIML (Netica)

Tags to manage decision networks:

- `<getExpectedUtils>` to estimate rewards for all the possible actions;
- `<getBestDecision>` to return only the best action.
New Extensions in AIML (POMDP)

- Partially Observable Markov Decision Process
  - Extention of Markov chains
  - A way to plan the whole conversation
  - Tags to load the POMDP, to record perception from the environment, and to query for the next action to be performed

- Plans are pre-computed before loading
  - Actions: sentences by the system
  - Perceptions: sentences by the user
  - Goals: a desired knowledge state (KST)
New data sources for the chatbot
- Documents collection
- Database
- WordNet
- Domain Ontology
- Procedural Ontology

Tags to interact with all these different sources

Negative JOL/FOK during conversation
Retrieving contents

Three tiers system
Retrieving contents

- Three tiers system
- Data stored in many ways (relational database, XML database, and documents)
Retrieving contents

- Three tiers system
- Data stored in many ways (relational database, XML database, and documents)
- Formal description of data and knowledge (OWL and Prolog ontologies)
Retrieving contents

- Three tiers system
- Data stored in many ways (relational database, XML database, and documents)
- Formal description of data and knowledge (OWL and Prolog ontologies)
- Querying interface
More than 50,000 documents about artists, and works of art

6,000 artists recorded in the database (name, biographic details, works, ...)

45,000 works of art recorded in the database

Documents analyzed and classified through the Latent Semantic Analysis
Either relational and hierarchical data
24 tables or trees in the database

Many aspects recorded:

- Artist
- Work of art
- Subject
- Technique
- Material
- Artistic period
- Opera collocation
- Role
- Subject type
- ...
OWL Ontology

- Describes the structure of the database and organizes the document collection.
- A document is linked to one or more nodes in the ontology.
- Each table of the database defines a concept of the ontology.

An example of OWL code:

```xml
<owl:Class rdf:ID="Artist"> node of the ontology
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">artista</rdfs:comment> table of the database
</owl:Class>

<owl:Class rdf:ID="Place">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">luogo</rdfs:comment>
</owl:Class>

<owl:Class rdf:ID="Art_Period">
<rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">periodo_artistico</rdfs:comment>
</owl:Class>
```
Describes complex concepts

Structure: Collection of instances of concepts related between them

Focus: element of a structure involved in an interaction

Frame: a focus applied to a constrained structure

Subframes can be derived from frames

Defines the database query management

Examples of Prolog code:

% an instance of artist related to an instance on biography

structure(artist_and_bio,
    instances [a is artist, la is bio],
    hooks [la is_bio_of a]).

frame(artist_and_english_bio,
    focus [a is artist, la is bio],
    structure artist_and_bio,
    constrains ['la.english IS NOT NULL']
).

subframe_of(artist_having_english_bio,
    artist_and_english_bio, [a]).
Four steps:

- Natural Language user query matching
  - Defining the structure
  - Defining the constraint
  - Defining the focus
  - Ontology browsing
- SQL query generation
- Natural language reply generation

Example: “Where and when did the author of Monna Lisa die?”
Where and when did the author of Monna Lisa die?

Structure

- Death
  - happened_in
  - died_in
  - happened_at
  - died_in

- Opera
  - created_by
  - is_subject_of

- Place
- Date
- Artist
- Subject
Where and when did the author of Monna Lisa die?
Where and when did the author of Monna Lisa die?

Focus:
- Place
  - happened_at
- Date
- Death
  - happened_in
  - died_in
- Artist
  - created_by
- Opera
  - is_subject_of

Where and when did the author of Monna Lisa die?
Where and when did the author of Monna Lisa die?

Ontology browsing

Death
- happened_in
- died_in

Place
- happened_at

Date

Artist
- died_in

Opera
- created_by
- is_subject_of

Subject
Where and when did the author of Monna Lisa die?

Ontology browsing

- Death
  - happened_in
  - died_in

- Place
  - happened_at

- Date

- Artist

- Opera
  - created_by
  - is_subject_of

- Subject
Where and when did the author of Monna Lisa die?

Ontology browsing

Death

Place

Date

Opera

Artist

Subject

happened_in

died_in

happened_at

created_by

is_subject_of

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

AAAI FS MCES 2009, Arlington - November 05, 09
Where and when did the author of Monna Lisa die?

Ontology browsing

Death

Place

Date

Artist

Opera

Subject

happened_in

died_in

created_by

is_subject_of

happaned_at
Where and when did the author of Monna Lisa die?
Conclusion and Future works

- Merging of ontology browsing and conversation planning
- Long-term goal-driven system
- Development of a conversational agent integrating deterministic and probabilistic dialogue management
Thanks!

cannella@dinfo.unipa.it
pirrone@dinfo.unipa.it