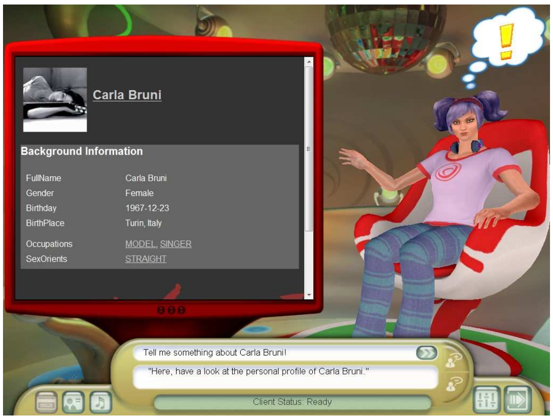
GOSSIP GALORE
A Self-Learning Agent for Exchanging Pop Trivia

Xiwen Cheng, Peter Adolphs, Feiyu Xu, Hans Uszkoreit, Hong Li
DFKI GmbH, Language Technology Lab
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
{xiwen.cheng,peter.adolphs,feiyu,uszkoreit,lihong}@domain.com

Abstract

This paper describes a self-learning software agent who collects and learns knowledge from the web and also exchanges her knowledge via dialogues with the users. The agent is built on top of information extraction, web mining, question answering and dialogue system technologies, and users can freely formulate their questions within the gossip domain and obtain the answers in multiple ways: textual response, graph-based visualization of the related concepts and speech output.



Carla Bruni

Background Information

FullName	Carla Bruni
Gender	Female
Birthday	1967-12-23
BirthPlace	Turin, Italy
Occupations	MODEL, SINGER
SexOrients	STRAIGHT

Tell me something about Carla Bruni!

"Here, have a look at the personal profile of Carla Bruni."

Client Status: Ready

Figure 1: Gossip Galore responding to "Tell me something about Carla Bruni!"

1 Introduction

The system presented here is developed within the project *Responsive Artificial Situated Cognitive Agents Living and Learning on the Internet* (RASCALLI) supported by the European Commission Cognitive Systems Programme (IST-27596-2004). The goal of the project is to develop and implement cognitively enhanced artificial agents, using technologies in natural language processing, question answering, web-based information extraction, semantic web and interaction driven profiling with cognitive modelling (Krenn, 2008).

This paper describes a conversational agent "Gossip Galore", an active self-learning system that can learn, update and interpret information from the web, and can make conversations with users and provide answers to their questions in the domain of celebrity gossip. In more detail, by applying a minimally supervised relation extraction system (Xu et al., 2007; Xu et al., 2008), the agent automatically collects the knowledge from relevant websites, and also communicates with the users using a question-answering engine via a 3D graphic interface.

This paper is organized as follows. Section 2 gives an overview of the system architecture and presents the design and functionalities of the components. Section 3 explains the system setup and discusses implementation details, and finally Section 4 draws conclusions.

2 System Overview

Figure 1 shows a use case of the system. Given a query "Tell me something about Carla Bruni", the application would trigger a series of background actions and respond with: "Here, have a look at the personal profile of Carla Bruni". Meanwhile, the personal profile of Carla Bruni, would be displayed on the screen. The design of the interface reflects the domain of celebrity gossip: the agent is depicted as a young lady in 3D graphics, who communicates with users. As an additional feature, users can access the dialogue memory of the system, which simulates the human memory in dialogues. An example of the dialogue memory is sketched in Figure 2.

As shown in Figure 3, the system consists of a number of components. In principle, first, a user's query is linguistically analyzed, and then inter-

Proceedings of the EACL 2009 Demonstrations Session, pages 13–16,
Athens, Greece, 3 April 2009. ©2009 Association for Computational Linguistics

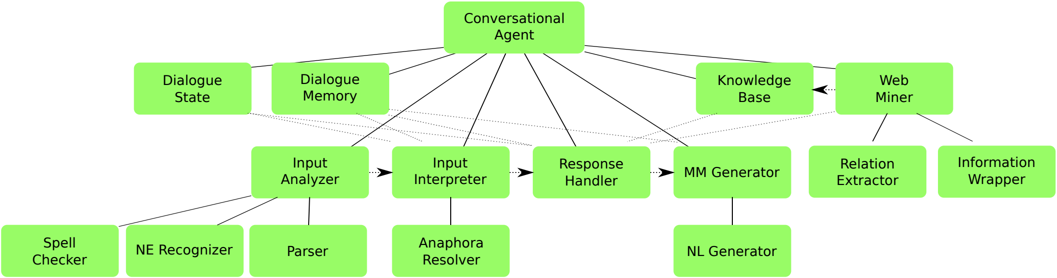


Figure 3: Agent architecture and interaction of components

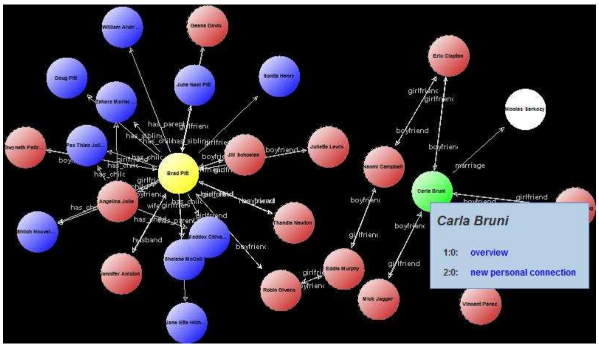


Figure 2: Representation of Social Network in Dialogue Memory

preted with respect to the context of the dialogue. A Response Handler will then consult the knowledge base pre-constructed by extracting relevant information from the Web, and pass the answer, in an abstract representation, to a Multimodal Generator, which realizes and presents the answer to the user in multiple ways. The main components are described in the following sections.

2.1 Knowledge Base

The knowledge base is automatically built by the Web Miner. It contains knowledge regarding properties of persons or groups and their social relationships. The persons and groups that we concern are celebrities in the entertainment industry (e.g., singers, bands, or movie stars) and their relatives (e.g., partners) and friends. Typical properties of a person include name, gender, birthday, etc., and profiles of celebrities contain additional properties such as sexual orientation, home pages, stage names, genres of their work, albums, and prizes. Social relationships between the persons/groups such as parent-child, partner, sibling, influencing/influenced and group-member, are also stored.

2.2 Web Miner

The Web Miner fetches relevant concepts and their relations by means of two technologies: a) information wrapping for exaction of personal profiles from structured and semi-structured web content, and b) a minimally supervised machine learning method provided by DARE (Xu et al., 2007; Xu et al., 2008) to acquire relations from free texts. DARE learns linguistic patterns indicating the target semantic relations by taking some relation instances as initial seed. For example, assume that the following seed for a parent-child relationship is given to the DARE system:

(1) **Seed**: ⟨Angelina Jolie, Shiloh Nouvel Jolie-Pitt, daughter⟩

One sentence that matches the entities mentioned in the seed above could be (2), and from which the DARE system can derive a linguistic pattern as shown in 3.

(2) **Matched sentence**: *Angelina Jolie* and Brad Pitt welcome their new *daughter Shiloh Nouvel Jolie-Pitt*.

(3) **Extracted pattern**: ⟨subject: celebrity⟩ welcome ⟨mod: “new daughter”⟩ ⟨object: person⟩

Given the learned pattern, new instances of the “parent-child” relationship can be automatically discovered, e.g.:

(4) **New acquired instances**: ⟨Adam Sandler, Sunny Madeline⟩ ⟨Cynthia Rodriguez, Ella Alexander⟩

Given the discovered relations among the celebrities and other people, the system constructs a social network, which is the basis for providing answers to users’ questions regarding celebrities’ relationships. The network also serves as a resource for the active dialogue memory of the agent as shown in Figure 2.

2.3 Input Analyzer and Input Interpreter

The Input Analyzer is designed as both domain and dialogue context independent. It relies on several linguistic analysis tools: 1) a spell checker, 2) a named entity recognizer SProUT (Drozdzynski et al., 2004), and 3) a syntactic parsing component for which we currently employ a fuzzy paraphrase matcher to approximate the output of a deep syntactic/semantic parser.

In contrast to the Input Analyzer, the Input Interpreter analyzes the input with respect to the context of the dialogue. It contains two major components: 1) anaphoric resolution, which refers pronouns to previously mentioned entities with the help of the dialogue memory, and 2) domain classification, which determines whether the entities contained in a user query can be found in the gossip knowledge base (cf. "Carla Bruni" vs. "Nicolas Sarkozy") and whether the answer focus belongs to the domain (cf. "stage name" vs "body guard"). For example, a simple factoid query such as "Who is Madonna", an embedded questions like "I wonder who Madonna is", and expressions of requests and wishes such as "I'm interested in Madonna", would share the same answer focus, i.e., the "personal profile" of "Madonna". In addition to the simple answer types such as "person name", "location" and "date/time", our system can also deal with complex answer focus types such as "personal profile", "social network" and "relation path", as well as domain-relevant concepts such as "party affiliation" or "sexual orientation".

Finally, the analysis of each query is associated with a meaning representation, an answer focus and an expected answer type.

2.4 Response Handler

This component executes the planned action based on the properties of the answer focus and the entities in a query. In cases where the answer focus or the entities cannot be found in the knowledge base, the system would still attempt to provide a constructive answer. For instance, if a question contains a domain-specific answer focus but entities unknown to the knowledge base, the agent will automatically look for alternative knowledge resources, e.g., Wikipedia. For example, given the question "Tell me something about Nicolas Sarkozy!", the agent would attempt a Web search and return the corresponding page on Wikipedia about "Nicolas Sarkozy", even if the knowledge base does not contain his information since he is a politician rather than an entertainer.

In addition, specific strategies have been developed to deal with negative answers. For instance, the agent would answer the question: *When did Madonna die?*, with "As far as I know, Madonna is still alive.", as it cannot find any information regarding Madonna's death.

2.5 Multimodal Generator

The agent (i.e., the young lady in Figure 1) is equipped with multimodal capabilities to interact with users. It can show the results in textual and speech forms, using body gestures, facial expressions, and finally via multimedia output to an embedded screen. We currently employ template-based generators for producing both the natural language utterances and the instructions to the agent that controls the multimodal communication with the user.

2.6 Dialogue State

The responsibility of this component is to keep track of the current state of the dialogue between a user and the agent. It models the system's expectation of the user's next action and the system's reactions. For example, if a user misspelled a name as in the question "Who is Roby Williams?", the system would answer with a clarification question: "Did you mean Robbie Williams?" The user is then expected to react to the question with either "yes" or "no", which would not be interpretable in other dialogue contexts where the user is expected to ask a question. The fact that the system asks a clarification question and expects a yes/no answer as well as the repaired question are stored in the Dialogue State component.

2.7 Dialogue Memory

This component aims to simulate the cognitive capacity of the memory of a human being: construction of a short-time memory and activation of long-time memory (our Knowledge Base). It records the sequence of all entities mentioned during the conversation and their respective target foci. Simultaneously, it retrieves all the related information from the Knowledge Base. In figure 2, the dialogue memory for the three questions "Tell me something about Carla Bruni.", "Can you tell me some news about her?", "How many kids does Brad Pitt have?" is shown. Green and yellow bubbles are entities mentioned in the dialogue context, where the yellow one is the last mentioned entity. White bubbles indicate the newest records which are acquired in the last process of online QA.

3 Implementation

The system uses a client-server architecture. The server is responsible for accepting new connections, managing accounts, processing conversations and passing responses to the clients. All the server-side functions are implemented in Java 1.6. We use Jetty as a web server to deliver multimedia representations of an answer and to provide selected functionalities of the system as web services to our partners. The knowledge base is stored in a MySQL database whose size is 11MB, and contains information of 38,758 persons including 16,532 artists and 1,407 music groups. As for the social connection data, there are 14,909 parent-child, 16,886 partner, 4,214 sibling, 308 influence/influenced and 9,657 group-member relational pairs. The social network is visualized in JGraph, and speech output is generated by the open-source speech synthesis system *OpenMary* (Schröder and Hunecke, 2007).

There are two interfaces realizing the client-side of the system: a 3D software application and a web interface. The software application uses a 3D computer game engine, and communicates with the server by messages in an XML format based on BML and SSML. In addition, we provide a web interface[1], implemented using HTML and Javascript on the browser side, and Java Servlets on the server side, offering the same core functionality as the 3D client.

Both the server and the web client are platform independent. The 3D client runs on Windows with a dedicated 3D graphics card. The recommended memory for the server is 1GB.

4 Conclusions

This paper describes a fully implemented software application, which discovers and learns information and knowledge from the Web, and communicates with users and exchanges gossip trivia with them. The system uses many novel technologies in order to achieve the goal of vividly chatting and interacting with the users in a fun way. The technologies include information extraction, question answering, dialogue modeling, response planning and multimodal presentation generation. Please refer to (Xu et al., 2009) for additional details about the "Gossip Galore" system.

The planned future extensions include the integration of deeper language processing methods to discover more precise linguistic patterns. A prime candidate for this extension is our own deep syntactic/semantic parser. Another plan concerns the required temporal aspects of relations together with credibility checking. Finally, we plan to exploit the dialogue memory for moving more of the dialogue initiative to the agent. In cases of missing or negative answers or in cases of pauses on the user side, the agent can use the active parts of the dialogue memory to propose additional relevant information or to guide the user to fruitful requests within the range of user's interests.

[1] `http://rascalli.dfki.de/live/dialogue.page`

References

Witold Drozdzynski, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. 2004. Shallow processing with unification and typed feature structures – foundations and applications. *Künstliche Intelligenz*, 1:17–23.

Brigitte Krenn. 2008. Responsive artificial situated cognitive agents living and learning on the internet, April. Poster presented at CogSys 2008.

Marc Schröder and Anna Hunecke. 2007. Mary tts participation in the Blizzard Challenge 2007. In *Proceedings of the Blizzard Challenge 2007*, Bonn, Germany.

Feiyu Xu, Hans Uszkoreit, and Hong Li. 2007. A seed-driven bottom-up machine learning framework for extracting relations of various complexity. *Proceedings of ACL-2007*, pages 584–591.

Feiyu Xu, Hans Uszkoreit, and Hong Li. 2008. Task driven coreference resolution for relation extraction. In *Proceedings of ECAI 2008*, Patras, Greece.

Feiyu Xu, Peter Adolphs, Hans Uszkoreit, Xiwen Cheng, and Hong Li. 2009. Gossip galore: A conversational web agent for collecting and sharing pop trivia. In *Joaquim Filipe, Ana Fred, and Bernadette Sharp (eds). Proceedings of ICAART 2009*, Porto, Portugal.