

## Research Article

# A Modular System Oriented to the Design of Versatile Knowledge Bases for Chatbots

**Giovanni Pilato,<sup>1</sup> Agnese Augello,<sup>2</sup> and Salvatore Gaglio<sup>2</sup>**

<sup>1</sup>*Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR), Consiglio Nazionale delle Ricerche (ICAR), Viale delle Scienze, Edificio 11, 90128 Palermo, Italy*

<sup>2</sup>*Dipartimento de Ingegneria Chimica, Gestionale, Informatica e Meccanica, Università di Palermo, Viale delle Scienze, Edificio 6, 90128 Palermo, Italy*

Correspondence should be addressed to Agnese Augello, [augello@info.unipa.it](mailto:augello@info.unipa.it)

Received 9 October 2011; Accepted 22 November 2011

Academic Editor: K. T. Atanassov

Copyright © 2012 Giovanni Pilato et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper illustrates a system that implements a framework, which is oriented to the development of a modular knowledge base for a conversational agent. This solution improves the flexibility of intelligent conversational agents in managing conversations. The modularity of the system grants a concurrent and synergic use of different knowledge representation techniques. According to this choice, it is possible to use the most adequate methodology for managing a conversation for a specific domain, taking into account particular features of the dialogue or the user behavior. We illustrate the implementation of a proof-of-concept prototype: a set of modules exploiting different knowledge representation methodologies and capable of managing different conversation features has been developed. Each module is automatically triggered through a component, named *corpus callosum*, that selects in real time the most adequate chatbot knowledge module to activate.

## 1. Introduction

Research on intelligent systems in last years has been characterized by the growth of the Artificial General Intelligence (AGI) paradigm [1]. This paradigm focuses attention more on learning processes than on the formalization of the domain. According to this theory, an intelligent system should not only solve a specific problem, but it should be designed as a hybrid architecture that integrates different approaches in order to emulate features of human intelligence, like flexibility and generalization capability.

At the same time, particular interest has been specifically addressed in the development of functional and accessible interfaces between intelligent systems and their users, in order to obtain a satisfactory man-machine interaction.

In this context, an ambitious goal is the creation of intelligent systems with conversational skills. The implementation of a conversational agent, however, is a complex task since it involves language understanding and dialogue management [2, 3].

The simplest way to implement a conversational agent is to use chatbots [4], which are dialogue systems based on a pattern matching mechanism between user queries and a set of rules defined in their knowledge base.

In this work we show the evolution of a previously developed model of conversational agent [5, 6]. The cognitive architecture evolved into a modular knowledge representation framework for the realization of smart and versatile conversational agents. A particular module, named “*corpus callosum*,” is dedicated to dynamically switch the different modules. Moreover, it manages their mutual interaction, in order to activate different cognitive skills of the chatbot. This solution provides intelligent conversational agents with a dynamic and flexible behavior that better fits the context of the dialogue [7].

We have modified the ALICE (Artificial Linguistic Internet Computer Entity) [4] core and realized the implementation of a proof-of-concept chatbot prototype, which uses a set of modules exploiting different knowledge representation techniques.

Each module provides specific capabilities: to induce the conversation topic, to analyze the semantics of user requests, and to make semantic associations between dialogue topics.

The dynamic activation of the most adequate modules, capable to manage specific aspects of the conversation with the user, gives to the conversational agent more naturalness of interaction.

The proposed solution tries to overcome the main limits of pattern-matching-based chatbot architectures, which are based on a rigid knowledge base, time consuming to establish and maintain, and a limited dialogue engine which does not take into account the semantic content, the context, and the evolution of the dialogue.

The modularity of the architecture makes it possible to use in a concurrent and synergic way specific methodologies and techniques, choosing and using the most adequate methodology for a specific characteristic of the domain (e.g., an ontology to represent deterministic information, Bayesian Networks to represent uncertainty, and semantic spaces to encode subsymbolic relationships between concepts).

The remaining of the paper is organized as follows: Section 2 gives an overview of related works; Section 3 illustrates the modular architecture; in Section 4 a case of study is described; in Section 5 dialogue examples are reported; finally Section 6 contains the conclusions.

## 2. Related Work

Several systems oriented to the artificial general intelligence approach have been illustrated in the literature. They combine different methodologies of representation, reasoning, and learning.

As an example, ACT-R [8] is a hybrid cognitive system which combines rule-based modules with subsymbolic units represented by parallel processes that control many of the symbolic processes. CLARION [9] uses a dual representation of knowledge, consisting of a symbolic component to manage explicit knowledge and a low-level component to manage tacit knowledge. CLARION consists of several subsystems, each one is based on this dual representation. Subsystems are an action centered system to control actions, a noncentered action system to maintain the general knowledge, a motivational subsystem for perception cognition and action, and a metacognitive subsystem to monitor and manage the operations of other subsystems. The most significant example of hybrid architecture is the OpenCog framework [10]. It is based on a probabilistic reasoning engine and an evolutionary learning engine called Moses. These mechanisms are integrated with a representation of knowledge both in declarative and procedural form. Procedural knowledge is represented by using a functional programming language called Combo. Declarative knowledge is represented in a hypergraph labeled with different types of weights: probabilistic weights representing values of the semantic uncertainty and Hebbian weights acting as attractors of neural networks, allowing the system to make inferences about concepts which are simultaneously activated.

In the field of conversational agents, two different knowledge representation approaches are generally used by intelligent systems to extract and manage semantics in natural language: symbolic and subsymbolic.

Symbolic paradigms provide a rigorous description of the world in which the conversational agent works, exploiting *ad hoc* rules and grammars to make agents able to understand and generate natural language sentences. These paradigms are limited by the difficulty of defining rules and grammars that must consider all the different ways of expression of people. Subsymbolic approaches analyze text documents and chunks of conversations to infer statistic and probabilistic rules that model the language.

In last years we worked on the construction of a system that implements an hybrid cognitive architecture for conversational agents, going to the direction of the AGI paradigm. The cognitive architecture integrates both symbolic and subsymbolic approaches for knowledge representation and reasoning.

The symbolic approach is used to define the agent's background knowledge and to make it capable of reasoning about a specific domain, both in terms of deterministic and uncertain reasoning [11].

The subsymbolic approach, based on the creation of data-driven semantic spaces, makes the conversational agent capable to infer data-driven knowledge through machine learning techniques. This choice improves the agent competences in an unsupervised manner, and allows it to perform associative reasoning about conversation concepts [5].

## 3. A Modular Architecture for Adaptive ChatBots

The illustrated system implements a framework oriented to the design and implementation of conversational agents characterized by a dynamic behavior, that is, capable to adapt their interaction with the user according to the current context of the conversation. With *context* we mean a set of conditions characterizing the interaction with the user, like the topic and the goal of the conversation, the profile of the user, and her speech act.

The proposed work has been realized according to the AGI paradigm: a modular and easily manageable and upgradable architecture, which integrates different knowledge representation and reasoning capabilities.

In the specific case illustrated in this paper, the architecture integrates symbolic and subsymbolic reasoning capabilities. The system architecture is shown in Figure 1, and it is constituted by different components.

- (i) A *dialogue engine* manages the interaction between the user and the chatbot. In particular it is composed of a set of *modules* defining the knowledge and reasoning capabilities of the chatbot. Each module is interconnected with several information repositories of declarative knowledge and defines a set of rules modeling the procedural knowledge. A component, called *collector* links the dialogue interface with the modules of the knowledge base. It sends the request

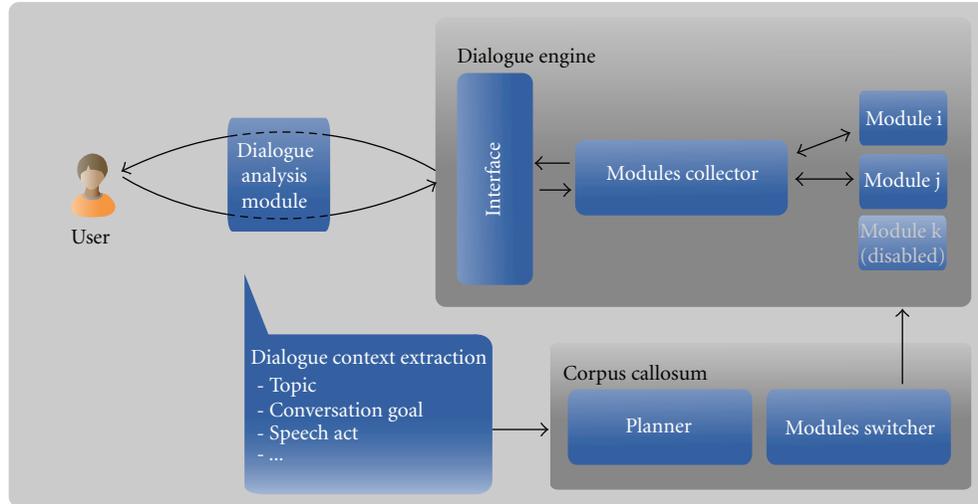


FIGURE 1: System architecture.

of the user to the currently active modules and gives her a proper answer through the interface;

- (ii) A *dialogue analyzer* extracts a set of variables that characterize the context of the conversation;
- (iii) A module named “*corpus callosum*” plans the activations and the deactivations of the dialogue engine modules according to the values of the context variables.

The proposed architecture is quite general and it can be particularized defining specific implementations for each component: for example, it is possible to use different models of knowledge representation, to change the planning module, and to consider different kinds of context variables. Moreover, the proposed architecture is characterized by an adaptive behavior and generic adaptability. In fact, the dynamic activation of the modules makes it possible to obtain a behavior of the chatbot capable to adapt itself to the current context. The behavior depends on the modules specific functionalities and of the corpus callosum planner.

**3.1. Dialogue Engine.** The dialogue engine improves the standard Alice [4] dialogue mechanism. The standard Alice dialogue engine is based on a pattern-matching approach that compares from time to time the sentences written by the user and a set of elements named “*categories*” defined through the AIML (Artificial Intelligence Markup Language) language. This language defines the rules for the lexical management and understanding of sentences [4]. Each category is composed of a *pattern*, which is compared with the user query, and a *template*, which constitutes the answer of the chatbot. The main drawbacks of this approach are (a) the time-expensive designing process of the AIML Knowledge Base (KB), because it is necessary to consider all the possible user requests and (b) the dialogue management mechanism, which is too rigid.

In previous works [5, 6] many approaches have been proposed in order to overcome these disadvantages. The

traditional KB has been extended through the use of *ad hoc* tags capable to query external knowledge repositories, like ontologies or semantic spaces, enhancing as a consequence the inferential capabilities of the chatbots. In fact, incomplete generic patterns can be defined and completed through a search of concepts related to a given topic of conversation in an ontology in order to dynamically build appropriate answers. Furthermore, the KB has been modeled, in an unsupervised manner, in a semantic space, starting from a statistical analysis of documents and dialogue chunks.

The contribute of the new architecture presented in this paper is to enhance the knowledge management capabilities of the chatbot both by declarative and procedural points of view. The goal is reached by splitting the traditional monolithic knowledge base of the chatbot in different components, named *modules*, that are perfectly suited to deal with particular characteristics of dialogue. Besides, a coordination mechanism has been provided in order to select and trigger, from time to time, the most adequate modules to manage the conversation with the user for efficiency reasons.

**3.1.1. Modules.** Each module of the dialogue engine has its own specific features, that make it different from the other modules. For example, the differentiation can be done on functionality, on topics, on mood recognition or emulation, on specific goals to reach, on management of specific user profiles, or on a particular combination of them.

The trivial case is to organize specific modules for determines topics: each module is suited to deal with a particular subject and from time to time the *corpus callosum* evaluates which are the best modules to deal with the current state of the conversation.

Even if AIML has the  $\langle topic \rangle$  tag, the proposed approach has the advantage to separate the KB of the chatbot at module level instead of AIML level and, most important, the recognition of the topic can be realized through a semantic process instead of a lexical, pattern-matching guided, approach.

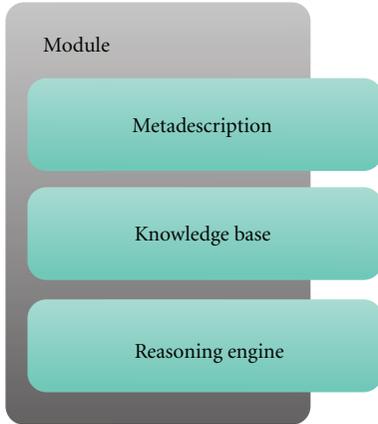


FIGURE 2: Chatbot module.

We have defined a *module* as an ALICE extension, obtained through the insertion of specific plugins in the ALICE architecture, importing the necessary libraries for the module execution. Each module is characterized by the definition of specific AIML tags and processors to query external repositories like ontologies, linguistic dictionaries, semantic spaces, and so on. Each module (see Figure 2) is composed of (a) a metadescription (metadata that semantically characterize the module), (b) a knowledge base, composed of the standard Alice AIML categories, which can be extended with other repositories, like ontologies or semantic spaces, and (c) an inferential engine capable to retrieve information, to select the chatbot answers or to perform actions.

The modular knowledge base is easier to define, design, and maintain. Each module has its own inference engine whose complexity is variable and defined by the module designer. The framework is general purpose: any new module, designed according to the rules of the architecture can be connected or disconnected without affecting the source code, the core of the chatbot or the behavior of any other module. It is possible to create a module at any time in order to manage specific cognitive activities, that mix “memory-oriented elements” (modules specifically suited to manage a specific topic of the conversation) with elements oriented to specific reasoning capabilities (modules oriented to semantic retrieval of information, modules oriented to the lexical analysis of the dialogue, modules oriented at inferring concepts from an ontology-based knowledge representation, modules oriented to evaluation of decisional processes, and so on). The new tags defined in new modules can be used to reach higher complexity levels and add new reasoning capability with the aim of enhancing the interaction characteristics of the conversational agent.

**3.2. Dialogue Analyzer.** This component is capable of capturing particular features related to the context of the dialogue and to manage them as variables. It can analyze the whole dialogue using syntactic, semantic, or pragmatics rules. The dialogue analyzer extracts from the current conversation what we define as “*context variables*.” Possible variables are

the topic of the dialogue, the goal of the conversation, the speech act, the mood of the user, the kind of user, the kind of dialogue (e.g., formal, informal), particular keywords, and so on.

**3.3. Corpus Callosum.** The corpus callosum is equipped with a module selector and a planner. The module selector enables or disables the dialogue engine modules at runtime by selecting the most appropriate modules from time to time. Modules are disabled as soon as they are not useful. The planner uses the context variables in order to define the temporal evolution of the state of each module.

Let  $C_t$  be the set of  $m$  context variables  $C_{jt}$  ( $j = 1, 2, \dots, m$ ) at time  $t$ ; the planner maps the context representation on the module states. The state  $s_{it}$  of the  $i$ th module at time  $t$  can be a binary value (e.g., active, not active) or a real value in  $[0, 1]$  representing its probability of activation.  $C_t$  contains all the past values for each one of the variables:

$$C_t = \begin{bmatrix} C_{1,t} & C_{2,t} & \dots & C_{m,t} \\ C_{1,(t-1)} & C_{2,(t-1)} & \dots & C_{m,(t-1)} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1,0} & C_{2,0} & \dots & C_{m,0} \end{bmatrix}. \quad (1)$$

The planner is characterized by the mapping function:

$$S_t = f(C_t) \quad (2)$$

with

$$S_t = [s_{1t}, s_{2t}, \dots, s_{mt}]. \quad (3)$$

Given the metadescription of the modules, the corpus callosum must determine the function  $f$ . The corpus callosum reconfigures the mapping function when new modules are enabled/connected or disabled/disconnected. It modifies the mapping using a learning algorithm that enhances the chatbot behavior in terms of activation/deactivation of the most fitting modules. It is possible to define a training set and an evaluation feedback mechanism of the chatbot’s answers.

The module selector activates or deactivates the chatbot modules, checking the value of the  $s_{it}$  state of each module  $i$  at time  $t$ : for binary values the activation is straightforward; for continuous values it is necessary to use a thresholding mechanism that can be the same for all modules or a specific threshold for each module, or dynamic, computed from time to time according to specific constraints.

## 4. A Case Study

As a proof of concept of the proposed system, we have implemented a conversational agent oriented at assisting people, typically students, in the Computer Science Engineering Department of the University of Palermo, Italy.

The conversational agent plays the role of a virtual doorman, or secretary, who is also capable of showing a different behavior according to the current dialogue context. Possible users are students, professors, researchers, or other people. Requests can vary from generic information to particular questions regarding specific people.

In the following subsections we will describe the key implemented components. For the dialogue analyzer we will describe the extracted contextual variables, and the modality of extraction. Particular emphasis will be given to the extraction of variables like speech acts, which are a fundamental characteristic that conduct conversations.

**4.1. Dialogue Analyzer Component.** In the particular implementation we have chosen to extract the following *context variables*: the speech act, the kind of user, the goal of the user, and the topic of the conversation. Speech acts derive from John Austin studies [12] and characterize the evolution of a conversation. According to Austin each utterance can be considered as an action of the human being in the world. Specific sequences of interaction are common in spoken language (e.g., question-answer), and they can be recognized and evaluated in order to better understand the meaning of sentences and generating the most appropriate answer to a user question. A conversation is affected also by the kind of interlocutor. As an example a simple language would be adopted to speak with a child, while a refined style is more adequate to speak to a well-educated person. Also the kind of terms between people is important during a dialogue: a conversation can be more or less formal. An agent can also have a thorough knowledge or not of a specific topic of conversation.

In consideration of this, we have identified four main variables that characterize the dialogue:

- (i) *Topic*: it is the topic of the conversation; it can be *artificial intelligence, image processing, computer languages, administration questions, and generic*.
- (ii) *Speech act*: the kind of speech act characterizing the dialogue at a given time; it can assume the values *assertive, directive, commissive, declarative, expressive with positive, neutral, or negative connotation*.
- (iii) *User*: the kind of user that is interacting with the chatbot. According to the definition of the scenario users are *student, professor, and other*.
- (iv) *Goal*: the goal of the user; it can be “*looking for people*” or “*looking for information*”.

**4.1.1. Topic Extraction.** In order to detect the topic of the conversation, we semantically compare the requests of the users with a set of documents, which have been previously classified according to the possible topics of conversation.

The comparison is based on the induction of a semantic space. A semantic space is a vector model based on the principle that it is possible to know the meaning of a word analyzing its context. The building of a semantic space is usually an unsupervised process that consists of the analysis of the distribution of words in the documents corpus.

The result of the process is the coding of words and documents as numerical vectors, whose respective distance reflects their semantic similarity.

In particular, a large text corpus composed of microdocuments classified according to the possible topics of conversation has been analyzed. Each document used to create

the space has been then associated with a very specific topic. A semantic space has been therefore built according to an approach based on latent semantic analysis (LSA) [13], reported in [5].

Given  $N$  documents of a text corpus, let  $M$  be the number of unique words occurring in the documents set. Let  $\mathbf{A} = \{a\}_{im}$  be an  $M \times N$  matrix whose  $(i, j)$ th entry is the square root of the sample probability of finding the  $i$ th word in the vocabulary in the  $j$ th document. According to the Truncated Singular Value Decomposition technique, a matrix  $\mathbf{A}$  can be approximated, given a truncation integer  $K < \min\{M, N\}$  as a matrix  $\mathbf{A}_k$  given by the product  $\mathbf{A}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$ , where  $\mathbf{U}_k$  is a column-orthonormal  $M \times K$  matrix,  $\mathbf{V}_k$  is a column-orthonormal  $M \times K$  matrix, and  $\mathbf{\Sigma}_k$  is a  $K \times K$  diagonal matrix, whose elements are called singular values of  $\mathbf{A}_k$ .

After the application of LSA the corpus documents have been coded as vectors in the semantic space.

Let  $N$  be the number of documents used to build the semantic space,  $\mathbf{d}_i$  and  $\mathbf{s}$  the numerical vectors associated respectively, to the  $i$ th document of the corpus and to the current sentence of conversation, and let  $T(\mathbf{d}_i)$  be the topic associated to  $\mathbf{d}_i$ . During the conversation, to evaluate the topic associated to the current sentence, we encode it as a vector in space, by means of the folding its technique [14], obtaining the vector  $\mathbf{s}$ , as reported below.

Let  $\mathbf{v}$  be a vector representing the current sentence. The  $i$ th entry of  $\mathbf{v}$  is the square root of the sample probability of finding the  $i$ th word in the vocabulary in the sentence, then:

$$\begin{aligned} \mathbf{x} &= \mathbf{v}^T \mathbf{U}_k \mathbf{\Sigma}_k^{-1}, \\ \mathbf{s} &= \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{x}. \end{aligned} \quad (4)$$

Then we compare the obtained vector with all the documents encoded in the space by using an appropriate geometric similarity measure  $\text{sim}$  defined in [5]. The topic of the sentence  $T(\mathbf{s})$  is equal to the topic of the closer document  $T(\mathbf{d}_k)$ :

$$T(\mathbf{s}) = T(\mathbf{d}_k) \quad (5)$$

according to the following similarity measure [5]:

$$\text{sim}(\mathbf{s}, \mathbf{d}_k) = \begin{cases} \cos^2(\mathbf{s}, \mathbf{d}_k) & \text{if } \cos(\mathbf{s}, \mathbf{d}_k) \geq 0, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where  $\cos(\mathbf{s}, \mathbf{d}_k)$  is the cosine between the vectors  $\mathbf{s}$  and  $\mathbf{d}_k$ .

**4.1.2. Speech Acts.** Exploiting the support of a linguist, we have adopted two elements of the speech act theory: the *illocutionary point* and the *illocutionary force*. An illocutionary point is the basic purpose of a speaker in making an utterance. It is a component of illocutionary force. The illocutionary force of an utterance is the speaker's intention in producing that utterance. An illocutionary act is an instance of a culturally defined speech act type, characterized by a particular illocutionary force, for example,

*promising, advising, warning*, and so forth. Recalling the Searle definition [15], we have five kinds of illocutionary points:

- (i) *assertive*: to assert something;
- (ii) *directive*: to commit to doing something;
- (iii) *commissive*: to attempt to get someone to do something;
- (iv) *declarative*: to bring about a state of affairs by the utterance;
- (v) *expressive*: to express an attitude or emotion.

We have simplified the concept of illocutionary force introducing three kinds of *act connotation*:

- (i) *positive* connotation,
- (ii) *neutral* connotation,
- (iii) *negative* connotation.

In our approach a directive act has a negative connotation when it is conducted with some sort of coercion. An explanation request conducted in a polite manner has a positive connotation; in other cases it will be labeled as a “neutral” connotation. For assertive acts we have a negative connotation when the assertion induces an adverse mood of the interlocutor, while it will be labeled as “positive” on the contrary. Commissive acts will be basically “neutral,” apart from promises (positive connotation), and threats (negative connotation). Expressive acts, like wishes, have a positive connotation; greetings have a neutral connotation, and complaints have a negative connotation. Declarative acts have not been used so far, since in dialogues schema that we have considered they are substantially absent (characteristic that is reported also in the literature [16, 17]).

This classification is not obvious and clear; therefore we have used some heuristics that associate a positive connotation to acts that cause in the agent a more favorable behavior (e.g., empathy and understanding are kinds of favorable behaviors). Illocutionary point and illocutionary force are variables of a speech act in an absolute sense, since they do not have any reference to a previous act.

**4.1.3. Kind of User and Goal of the Dialogue.** At present these kind of variables are extracted through ad hoc AIML categories suited to capture the information.

**4.2. Created Modules.** We have realized five kinds of modules.

- (i) Module 1: it is aimed at characterizing a friendly behavior of the chatbot.
- (ii) Module 2: it is oriented to characterize the chatbot with though behavior as a consequence of a negative evolution of the dialogue.
- (iii) Module 3: it makes the chatbot empathetic with the user.

(iv) Module 4: this module is designed to induce a situation of submissiveness of the chatbot with respect to the interlocutor.

(v) Module 5: it is built to make the chatbot capable to manage informative tasks.

Each module is activated by the corpus callosum, according to specific thresholds.

**4.3. Corpus Callosum Implementation.** In this section we illustrate the corpus callosum component and its relative mapping function realized with a Bayesian network. The network is shown in Figure 3.

The network makes inference on the variables extracted from the context in order to trigger the activation/deactivation of a module.

The status of a module is directly influenced by variables such as the conversation goal, the topic and the kind of user, while it is indirectly influenced by speech acts sequences (see Figure 3).

Since specific sequences of speech acts can imply a mood change in the chatbot, we have defined a “chatbot-induced behavior” variable, representing the chatbot behavior, with the aim of realizing a more realistic and plausible conversational agent. Speech acts are detected through a simple rule-based speech act classifier, whose description goes beyond the scope of this paper. Once recognized, speech acts relating both to current and past time slices are used to code the current behavior of the chatbot. The corpus callosum selects therefore the most appropriate module to activate by analyzing the chatbot behavior induced by the speech acts sequence and the other context variables.

The module with the highest probability is then selected, according to the causal inference schema coded in a dynamic bayesian network.

In particular, a variable “Modules” is defined in the network, whose states represent the possible modules to activate. The probability value associated with each state will determine whether that particular module must be turned on or off. The mapping function is then obtained by evaluating the conditional probability of the variable “Modules,” given its parents (see Figure 3)

$$S_t = f(C_t) \Rightarrow \text{Modules} = P(\text{Modules} \mid \text{Parents}(\text{Modules})) = P(\text{Modules} \mid \text{Goal, Topic, User, ChatbotInducedBehavior}). \quad (7)$$

The relationship between the variables follows the Bayes’ rule:

$$p(y|x) = \frac{p(x|y) \cdot p(y)}{p(x)}. \quad (8)$$

The labeled arcs are temporal arcs, for example, the arc labeled as “1” represents the influence of the variable at the former instant of time. The behavior of the chatbot is influenced by a triplet of speech acts (see Figure 3):

- (i) the act of the user at time  $t$ ;

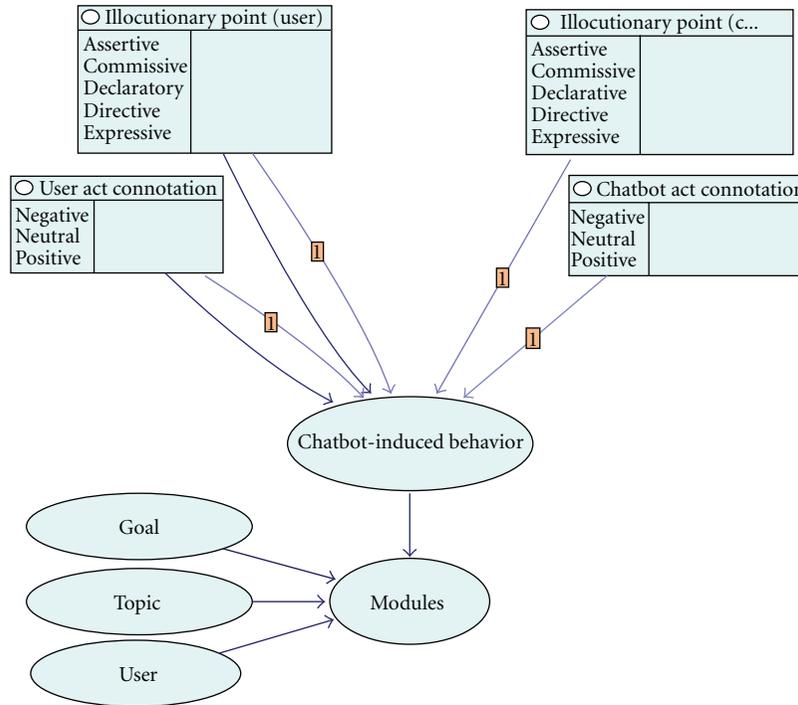


FIGURE 3: Bayesian planner of the corpus callosum.

- (ii) the act of the chatbot at time  $t - 1$ ;
- (iii) the act of the user at time  $t - 1$ .

The acts sequencing is therefore given by the last three speech acts in the dialogue. Of course, it is possible to extend the temporal window to more than three steps of speech acts. We have restricted the behaviors of the chatbot to 4:

- (i) friendly,
- (ii) determined,
- (iii) submissive,
- (iv) reassuring.

In Figure 4 we show an example of a possible sequence that induces a positive behavior in the chatbot, that can be mapped as a friendly behavior in our domain.

### 5. Dialogue Examples

In this section we show some samples of dialogues obtained during the interaction of users with a proof of concept of our prototype that we have realized at Department of Ingegneria Informatica of the University of Palermo, Italy.

The first example illustrates how the behavior of the chatbot changes according to different sequences of speech acts.

User: Buongiorno! (*Good morning!*)  
 Chatbot: Buongiorno, cosa desidera? (*Good morning, may I help you?*)

User: Sto cercando il professore Gaglio. (*I'm looking for professor Gaglio*)  
 Chatbot: Chi lo desidera? (*Who are you?*)  
 User: Sono un suo studente (*I'm one of his students.*)  
 Chatbot: Aspetti che verifico (*Please wait, I'll check.*)  
 User: Grazie (*Thank you.*)  
 Chatbot: Mi dispiace, il professore non c'è (*I apologize, the professor is not here at the moment.*)  
 User: Grazie lo stesso. (*Thanks anyway.*)  
 Chatbot: Mi dispiace, arrivederci. (*I am sorry, goodbye.*)

Table 1 shows the triplets of speech acts for the previous dialogue. Time starts from “-1” since the first triplet is characterized only by the speech act of the user. As a consequence the starting of the dialogue is in the last column at time  $t - 1$ , where an expressive act of the user is detected. With this kind of dialogue we expect the activation of Module 1, which has been designed to manage friendly communications with students and have the goal of looking for people. At step 3 it is detected the goal of the user, and at step 5 the kind of user.

Figure 5 shows the evolution of the Bayesian network for each step of the dialogue. In this picture the probability of activation of each module is shown during the temporal evolution of the dialogue, which consists of 6 time slots. On the  $x$ -axis time slots are reported, while  $y$ -axis shows the probability of activation for each module. The Module 1 is the prevalent one.

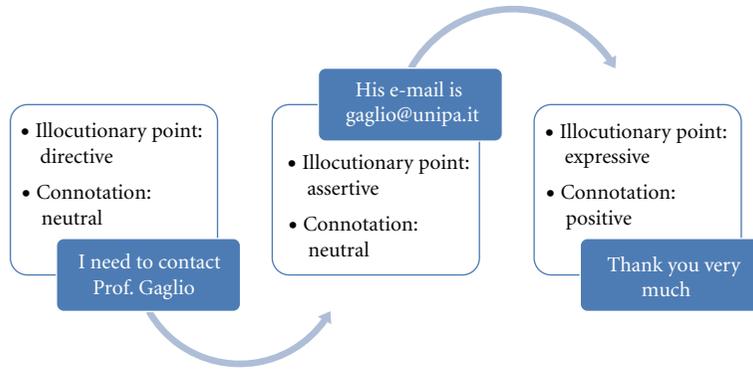


FIGURE 4: An example of speech acts sequence inducing a positive behavior in the chatbot.

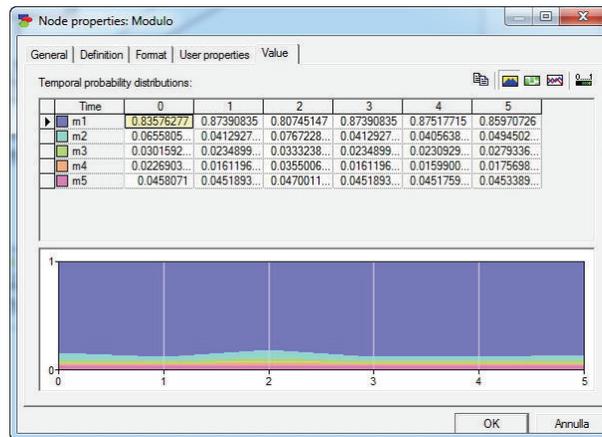


FIGURE 5: Bayesian network behavior evolution for dialogue 1.

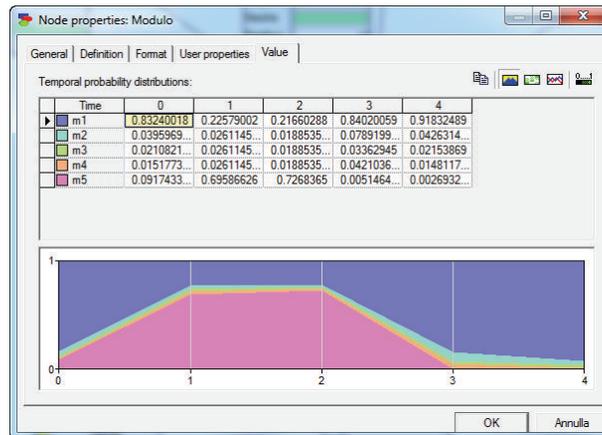


FIGURE 6: Bayesian network behavior evolution for dialogue 2.

TABLE 1: Speech act triplets for dialogue 1.

Tempo $t$	User speech act ( $t$ )	Chatbot speech act ( $t + 1$ )	User speech act ( $t + 1$ )
-1			Expressive-Neutral
0	Expressive-Neutral	Expressive-Neutral	Directive-Neutral
1	Directive-Neutral	Directive-Neutral	Assertive-Neutral
2	Assertive-Neutral	Commissive-Neutral	Expressive-Neutral
3	Expressive-Neutral	Assertive-Neutral	Expressive-Neutral
4	Expressive-Neutral	Expressive-Neutral	

In the following dialogue we show how the system detects a change of goal of the user: from a search for information, managed by Module 5, to the search for a professor, managed by Module 1.

1. . .
- 2 User: Sto cercando informazioni sulle tesi. (*I am looking for information about degree theses.*)
- 3 Chatbot: Mi dica pure. (*Yes, tell me please.*)
- 4 User: Quale professore si occupa di I.A.? (*Which professor deals with A.I.?*)
- 5 Chatbot: Il professore Gaglio. (*Professor Gaglio.*)
- 6 User: Il Professore Gaglio si trova in dipartimento? (*Is professor Gaglio at the Department right now?*)
- 7 Chatbot: Chi lo desidera? (*Who is looking for him?*)
8. . .

The shape of the probabilities of activation of the modules is shown in Figure 6: at time 1 the user is looking for information, and module 5 is activated; at time 3 the goal of the user becomes the search of a professor (step 6) and Module 1 becomes the most probable.

## 6. Conclusion

We have presented a system which implements a framework capable to dynamically manage knowledge in conversational agents. As a proof of concept we have illustrated a prototype, which is characterized by a Bayesian network planner. The planner is aimed at selecting, from time to time, the most adequate knowledge modules for managing specific features of the conversation with the user. As a result, the architecture is capable to generate complex behaviors of the agent. The architecture is analogous to the structure of the human brain: two hemispheres cooperate for the management and understanding of a dialogue, through a connection element like the corpus callosum. The left hemisphere is specialized in logic, reasoning, linguistic, rule-oriented, and syntactic processing of language; the right hemisphere is more oriented to intuition and emotions and processes information holistically. The corpus callosum is the bridge that connects the two hemispheres. It makes possible their mutual interaction through the migration of information between them. The dialogue analyzer extracts context information (the topic, the goal, the linguistic act, etc.) and the planner exploits these information to select and activate only the most appropriate modules, that incorporate the most adequate rules to process the specific sentences typed by the user. The conversational agent is also capable to perform analogical reasoning in order to understand the context and the structures in a general manner; the corpus callosum analyzes this information and selects the most appropriate module that is capable to properly process the specific sentence. However, it is worthwhile to point out that it is possible to realize other modules that fulfill other kind of specific functions.

## Acknowledgments

This work has been partially supported by Italian MIUR (Ministero dell'Istruzione, dell'Universita' e della Ricerca) within the SINTESYS "Security and INTElligence SYStem" project. The authors would thank Mr. Giuseppe Miceli for his partial contribution to the experimental trials.

## References

- [1] B. Goertzel and C. Pennachin, *Artificial General Intelligence (Cognitive Technologies)*, Springer, Secaucus, NJ, USA, 2007.
- [2] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, 2000, ISBN0-13-095069-6.
- [3] S. Larsson, "Dialogue systems: simulations or interfaces?" in *Proceedings of the 9th workshop on the semantics and pragmatics of dialogue*, In Gardent and Gaiffe (eds.), 2005.
- [4] A.l.i.c.e.chatbot, <http://www.alicebot.org/>.
- [5] F. Agostaro, A. Augello, G. Pilato, G. Vassallo, and S. Gaglio, "A conversational agent based on a conceptual interpretation of a data driven semantic space," in *Proceedings of the Advances in Artificial Intelligence: 9th Congress of the Italian Association for Artificial Intelligence (AIIA '05)*, S. Bandini and S. Manzoni, Eds., vol. 3673 of *Lecture Notes in Artificial Intelligence*, pp. 381–392, Milan, Italy, September 2005.
- [6] A. Augello, G. Pilato, G. Vassallo, and S. Gaglio, "A semantic layer on semi-structured data sources for intuitive chatbots," in *Proceedings of the 2nd International Workshop On Intelligent Interfaces For Human-Computer Interaction (IIHCI '09)*, pp. 760–765, 2009.
- [7] G. Pilato, A. Augello, and S. Gaglio, "A modular architecture for adaptive chatbots," in *Proceedings of the 5th IEEE International Conference on Semantic Computing*, Stanford University, Palo Alto, Calif, USA, September 2011.
- [8] J. R. Anderson, "Using brain imaging to guide the development of a cognitive architecture," in *Integrated Models of Cognitive Systems*, W. D. Gray, Ed., pp. 49–62, Oxford University Press, New York, NY, USA, 2007.
- [9] S. Hélie and R. Sun, "Incubation, insight, and creative problem solving: a unified theory and a connectionist model," *Psychological Review*, vol. 117, no. 3, pp. 994–1024, 2010.
- [10] B. Goertzel, "Opencogbot: achieving generally intelligent virtual agent control and humanoid robotics via cognitive synergy," in *Proceedings of the (ICAI '10)*, Beijing, China, 2010.
- [11] G. Pilato, A. Augello, G. Vassallo, and S. Gaglio, "Sub-symbolic semantic layer in cyc for intuitive chat-bots," in *Proceedings of the 1st IEEE International Conference on Semantic Computing*, pp. 121–128, September 2007.
- [12] J. L. Austin, *How to do things with words*, Harvard University Press, Cambridge, Mass, USA, 1975.
- [13] T. K. Landauer, P. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, pp. 259–284, 1998.
- [14] M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM Review*, vol. 37, no. 4, pp. 573–595, 1995.
- [15] J. R. Searle, "A taxonomy of illocutionary acts," in *Language, Mind, and Knowledge*, K. Gnderson, Ed., vol. 7, 1975.
- [16] D. Twitchell and J. F. Nunamaker, "Speech act profiling: a probabilistic method for analyzing persistent conversations

and their participants,” in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, 2004.

- [17] D. P. Twitchell, M. Adkins, J. F. Nunamaker, and J. K. Burgoon, “Using speech act theory to model conversations for automated classification and retrieval,” in *Proceedings of the 9th International Working Conference on the Language-Action Perspective on Communication Modeling*, 2004.




**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

