

Chatbots as Interface to Ontologies

Agnese Augello, Giovanni Pilato, Giorgio Vassallo and Salvatore Gaglio

Abstract Chatbots are simple conversational agents using “pattern matching rules” to carry out the dialogue with the user and various expedients to improve their credibility. However, the rules on which they are based on are too restrictive and their language understanding capability is very limited. Nevertheless chatbots are widespread in several applications, especially to provide information to users in a new and enjoyable way. In this chapter we describe different chatbot architectures, exploiting the use of ontologies in order to create clever information suppliers overcoming the main limits of chatbots: the knowledge base building and the rigidity of the dialogue mechanism.

1 Introduction

Human computer interaction is a challenging research area having the aim of building more and more usable human-computer interfaces. The use of natural language as an interface allows a fulfilling interaction and a greater accessibility by expert and inexperienced users to the system. For these reasons in last years there has been a growing interest toward the use of conversational agents. Research on this kind of systems involves natural language understanding and the analysis and management

A. Augello (✉) · G. Pilato
Viale delle Scienze, ICAR-CNR, ed. 11, Palermo, Italy
e-mail: augello@pa.icar.cnr.it

G. Pilato
e-mail: pilato@pa.icar.cnr.it

G. Vassallo · S. Gaglio
Viale delle Scienze, DICGIM-UNIPA, ed. 6, Palermo, Italy
e-mail: giorgio.vassallo@unipa.it

S. Gaglio
e-mail: salvatore.gaglio@unipa.it

of conversational practices. However, natural language is characterized by many ambiguities that adult human beings can resolve through their own cultural baggage built with everyday experiences. These difficulties led to the development of simple dialogue systems, called chatbot as an alternative to advanced dialogue systems. The main limits of chatbot technology regard their knowledge representation as well as their information retrieval and dialogue capabilities. Even if the simple technology allows to easily implement a dialogue system, the obtained conversation is limited by pattern matching rules on which chatbots are based. The traditional chatbot dialogue capability is too rigid, it can answer to the user only if there is a pattern which matches the question in its knowledge base. Chatbot are lacking in the intuitive capability of human beings to see meaning, relationships and possibilities beyond the reach of senses. Besides, the chatbot knowledge base is expensive and boring to create and, as highlighted in [1], all possible user questions have to be considered at design time. In our opinion chatbots can be improved by: (a) simplifying the knowledge base design process, extending the chatbot knowledge base with other information repositories, generalizing as much as possible the chatbot pattern-template modules (as an example it could be possible to write generic question answers modules, which the chatbot can adapt and properly complete exploiting other information repositories to answer to a specific user question), using unsupervised methodologies to train automatically the chatbot knowledge base exploiting information available in big textual corpora, (b) enhancing and making more clever their answering mechanism providing chatbots of inferential capabilities about the conversation topics and semantic analysis capabilities in order to handle the different ways of user expression analyzing similarity relations among words.

In last years we have proposed hybrid chatbot architectures combining symbolic and sub-symbolic methodologies for knowledge representation and reasoning, simulating rational and intuitive capabilities of human brain involved in dialogue understanding. In particular the rational component of the chatbot brain is implemented by means of ontologies. In the next sections, after a brief discussion on the state of the art, the main modules of the proposed architectures are described and some related applications will be discussed.

2 State of the Art

Chatbots are conversational agents using “pattern matching rules” to carry out a simple conversation by means of the detection of rules and keywords into the user sentences. During the conversation the chatbot looks for a lexical matching between the user query and a set of question-answer modules stored in its knowledge base. The answers are given using a set of predefined responses. They also use various techniques to improve their credibility, for example they can store user information and preferences, keep trace about the history and the current topic conversation.

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) is a chatbot developed in the Lehigh University (Pennsylvania) and distributed with GNU license.

The knowledge of Alice chatbot is composed of question-answer modules, called categories and described by Aiml (Artificial Intelligence Mark-up Language) language [2]. Aiml is a *mark-up* language in which specific tags are defined to properly interpret the meaningful elements of the sentence written by the user. In particular the tag *aiml* encloses the categories (question-answers modules) belonging to the Aiml file, each category is described by the tag *category*, composed of a tag *pattern* enclosing a sentence that will be compared to the user question, and a *template* tag which encloses the rules to generate the chatbot answer. Other optional tags are also present. The presence of the special symbols “_” and “*”, called “wildcards” in the pattern allows the chat-bot to answer also when in its knowledge base there is not a pattern exactly equal to the user question, the value associated to a wildcard can be read using the *star* tag. The template can contain other Aiml tags; in this case the answer is dynamically composed by properly analyzing them.

Special functions are carried out by the *topic*, *that* and *srai* tags. The first tag can be used before a group of categories to set its subject, or can be used inside the template to set or get the conversation topic. The second tag is located between the pattern and template tags and allows the chatbot to remember its last answer and keep trace of the conversation. The *srai* tag allows to recursively call other categories, activating again the pattern matching algorithm which manages the dialogue with the user. Different systems are aimed to the improving of the chatbots technology. In [3] the knowledge base of a community of chatbots, each one expert in a specific topic is coded in a semantic vector space allowing them to estimate their own competence about questions asked by the user. In the CyN project [4] the pattern matching interpreter of Alice, called Program N is linked to OpenCyc [5], the Open source version of Cyc, the largest common sense knowledge base available today. This allows the chatbot to exploit the great quantity of knowledge available in Cyc and to generalize the Aiml categories using OpenCyc information. Reasoning and inferential mechanisms are also included in chatbot systems proposed in [6–8]. In [9] it was proposed the creation of a lexicon extended with thesaurus for simulate a sort of semantic analysis. The implemented chatbot prototype can learn many new lexemes and syntagma structures during the interaction with the user. A deeper attention to conversational features is given into iAiml system [10], where rules to treat intentional information have been added to Aiml KB, exploiting the Conversational Analysis Theory (CAT) as linguistic base for consider intentionality in adjacent pairs in dialogue.

3 Ontology Based Architectures for Conversational Information Suppliers

In this section we discuss some architectures we have introduced in order to the enhance both knowledge representation and dialogue capabilities of chatbots. In particular during our research activities we have integrated symbolic and sub-symbolic knowledge representation approaches to provide chatbots of both rational and

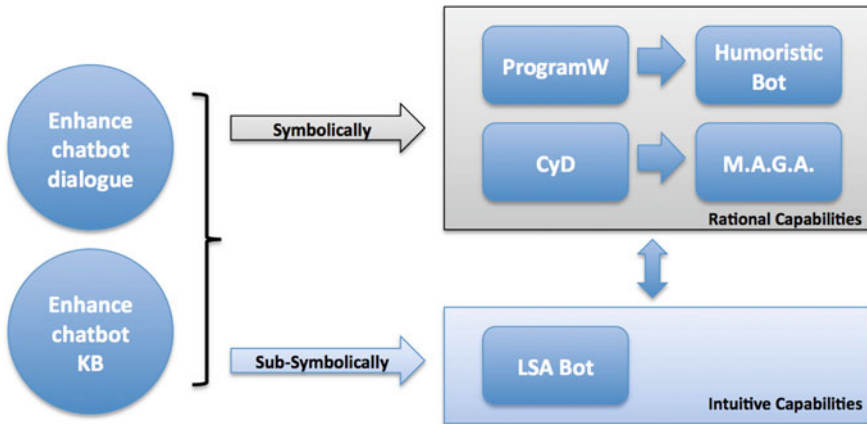


Fig. 1 Systems implemented to enhance chatbot dialogue and KB

intuitive dialogue capabilities. Three main systems are described: the first two implementing the rational and the intuitive components of the chatbot KB respectively, and the last the hybrid rational-intuitive KB architecture.

3.1 Integration of ProgramD and ResearchCyc: Cyd

The idea of extending the chatbot knowledge base with the information stored in a wide common sense repository such as Cyc, has been developed at the Daxtron Laboratories with the implementation of the CyN project [4]. The aim of the project was to provide a natural language interface to the Cyc ontology, making its content available to a wide number of users. In particular CyN is the integration of the Aiml interpreter written in C++ Program N to OpenCyc.

The project is open and easily extensible; therefore we have considered adapting it to the Aiml interpreter written in Java Program D. This choice allows for the exploiting of several advantages deriving from the use of Java language and using the Java API available to access the Cyc knowledge base. At present the technology, called CyD (Cyc + ProgramD) for consistency with CyN, integrates the version of Cyc available for research purpose ResearchCyc with ProgramD. The aim of CyD is to enhance the chatbot KB with the information available in ResearchCyc and provide these conversational systems with common sense reasoning and inferential capabilities. To this end the Aiml language was enriched with new tags to allow the chatbot to query the Cyc ontology directly from its KB rules. The chatbot template can contain these new “ad hoc” AIML tags which transform it into a “meta-answer” that must be processed by the OpenCyc inference engine to produce the chatbot

answer most appropriate to the user question. The main tags for communication with Cyc are illustrated in what follows [4]:

The Cyc responses are embedded in a natural language sentence according to the rules of the template. As an example the following category that allows the chatbot to verify if a concept belongs to a specific collection.

```
<category>
  <pattern>IS * A *</pattern>
  <template>
    <cycsystem>
      (cyc-query '
        (#$genls
          <cycterm><star index="1"/></cycterm>
          <cycterm><star index="2"/></cycterm>
        )#$EverythingPSC)
    </cycsystem>!!
  </template>
</category>
```

A possible dialogue could be:

User: *Is Palermo a City?*

Chatbot: *Yes!!*

The integration with Cyc has several advantages among which being the benefit of making categories writing easier; it is possible to think of different ways to express general topics creating generic, default categories. The string matching the wildcard symbol belonging to the category pattern will be searched into the Cyc KB; if a corresponding Cyc constant is found, the chatbot can analyze all the information associated to it exploiting the different Cyc predicates.

For example, it can extract a definition of the constant by means of the *#\$comment* predicate, verify if the constant is a collection or a collection instance and analyze the other related concepts by means of the *#\$isa* or *#\$genls* predicates. Or in specific contexts it can analyze “ad hoc” predicates.

Is possible to arrange all these information together, creating dynamically exhaustive answers, not present in the traditional AIML knowledge base. The possibility of exploiting a wide, common-sense ontology, like Cyc, makes the dialogue more fluent and reduces the number of default answers needed in traditional chatbots to fill up their “cultural” gaps.

Additionally, the possibility of interacting with the Cyc knowledge base by means of natural language allows inexpert users to create or extend ontologies with new concepts, facts and relations, without having to learn CycL statements.

3.1.1 A Cyd Application to User Mobile Conversational Assistants: MAGA

The CyD technology has been used into a more complex system to implement multimodal natural language user assistants. This project advanced with an analysis of

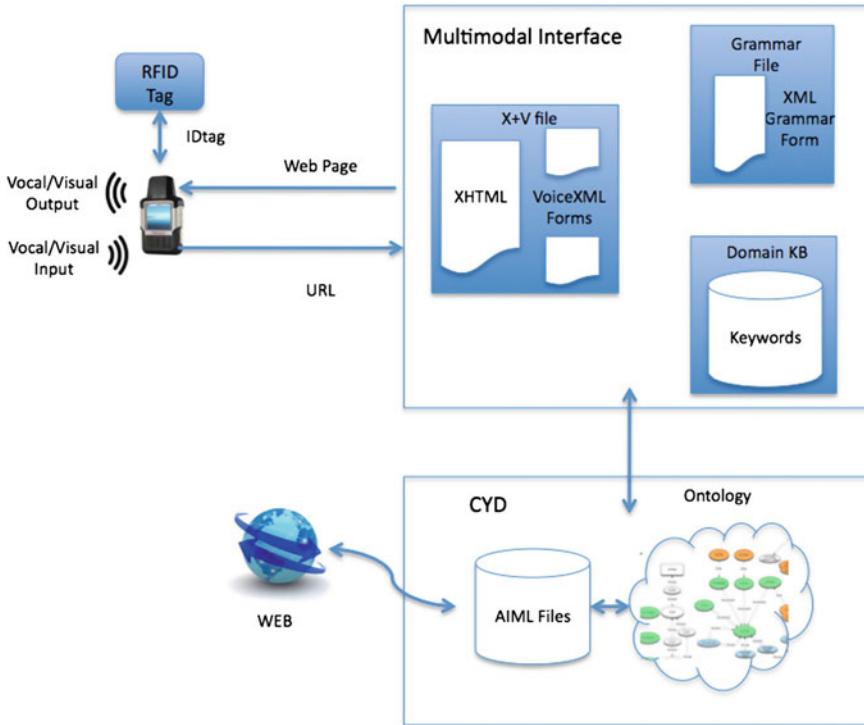


Fig. 2 An architecture for mobile, conversational assistants

the potentialities offered by ubiquitous computing technology and by personal mobile devices, equipped with third-generation wireless communication technologies, which can be exploited with the aim of providing people with useful information in relation to the environment. The aim is to offer an assistance service to users for information retrieval, accessible from mobile devices like PDAs and smartphones, in order to satisfy mobility needs of the user. We applied this architecture to the cultural heritage field by proposing a multimodal guide created to provide information during cultural heritage sites tours, and named it M.A.G.A. (Mobile Archaeological Guide at Agrigento) [11].

An overview of the M.A.G.A. architecture, based on a client-server paradigm, is shown in Fig. 2.

The main feature of the system is the integration of different technologies. In fact, the application is accessible using a PDA equipped with an RFID-based, auto-localization module, while the information retrieval service is provided by means of a spoken, natural language interaction with a conversational agent, integrated with reasoning capabilities based on CyD technology.

In particular, the RFID module allows the system to sense changes in user's environment, and to automatically adapt itself. The chatbot is provided with inferential

capabilities thanks to the use of CyD technology. It is possible to define an “ad hoc” microtheory for the specific context, properly defining the collections of concepts and facts regarding the analyzed domain. It is possible to exploit the information already present in Cyc, by hooking up the created microtheory to some of the existing ones.

The dialogue between the chatbot and the user is verbally carried on, by using a spoken, natural language in addition to traditional visual and keyboard (or stylus)-based commands. To this aim we used XHTML + Voice [12], X + V for short, while the multimodal interface has been developed with the IBM Multimodal Tools 4.1.2.2 [13] for WebSphere Studio V5.1.2. The speech recognition process is carried out through an “ad-hoc” built-in speech grammar, including a set of rules which specifies utterances that a user may say. The system is accessible through a web page in a multimodal browser from the handheld device. The interaction occurs through the loading of X + V pages, which can be triggered by user vocal and visual command or RFID detection; in every page the user can have a vocal dialogue with the chatbot. The chatbot searches for the best match rule in its knowledge base. The AIML rule can directly produce an answer, or it might be necessary to query Cyc in order to construct a more suitable answer to the user’s request. The chatbot answer could also be the result of a query to standard search engines which will search for local or remote documents related to the user query. The interaction between the application running on the PDA and the system is also started by the detection of a RFID tag, which is used to estimate the PDA position within the environment. According to this feature, people can go on asking questions about the current object to the chatbot with vocal queries, or they can discard the information and continue their tour.

The system is easily adaptable to application domain changes: the grammar is easy-fitting and can be improved with minimal effort, the choice of using Cyc allows the system developers to exploit the large amount of data already organized and described in this ontology. This makes the system more adaptable to domain changes, as it is not necessary to write the entire set of knowledge every time, but only the most specific. For this reason the system was applied for other purposes: to provide services in a university campus [14], to assist users in shopping activities [15].

3.2 Integration of a ChatBot and WordNet: ProgramW

The chatbot knowledge base has been also extended with the information available in the WordNet lexical database. The idea has been made concrete with the setting up of an open source project, called ProgramW, available at the sourceforge web site [16]. Program W is an AIML interpreter written in Java, which extends Program D technology in order to enable the interaction between the chatbot and WordNet. In particular we created new Aiml tags which allow chatbots to query WordNet directly from the rules belonging to its knowledge base. Chatbots can exploit information about lexical terms, for example their lemmas or the corresponding glosses (*wordnetlemma* and *wordnetgloss* tags) and evaluate existing relations between words defined in

Wordnet such as synonymy (*wordnetsynset* tag), antonymy (*wordnetantonym* tag) hyponymy-hypernymy (*wordnethyponym* and *wordnethypernym* tags), meronymy-holonymy (*wordnetmeronym* and *wordnetholonym* tags). It is also possible to find the relationship joining two words by means of *wordnetrelation* tag.

This integration can be used to improve the language analysis capability of the chatbot agent.

3.2.1 An Application of ProgramW: EHeBby, the Humoristic Bot

In recent years there was an interest in enhancing the realness of interaction with the conversational agents, providing these systems with the capability of change their behaviour according to the conversation content like in conversation between human beings. An important feature in social human interactions is represented by the capability of generating and understanding humour, therefore it is auspicious to reproduce this ability also in conversational agents [17]. Therefore a humoristic chatbot was proposed; it is capable of generating humorous expressions, proposing riddles to the user, telling jokes and ironically answering the user. Besides this, the chatbot is capable of detecting, during the conversation with the user, the presence of humorous expressions, listening and judging jokes, and reacting by changing the visual expression of the avatar, according to the perceived level of humour. The generation of humorous text is well suited for conversational agents. As a matter of fact, it is possible to define inside the chatbot knowledge base, composed of question-answer modules, the funniest answers most fitting to the user query. We focused our attention to the recognition of humour rather than to the generation.

As a consequence we have analysed the literature techniques in computational humour aimed to the recognition of humour in very short sentences (“one-liners”) for the design of a humorous conversational agent. A humour recognition methodology has been implemented in a chatbot, through the research, inside the sentences introduced by the user, of the main features that characterize the text as humoristic, in particular the features which can be computationally detected, such as alliteration, antinomy and adult slang, according to what was suggested by the authors of [18]. The core of the system consists of the chatbot knowledge base, composed of three kinds of AIML categories:

- the set of standard Alice categories, which allow the chatbot to hold a general conversation with the user;
- a set of categories aimed at the humorous sentences generation. These categories allow the chatbot to answer the user in a humoristic way, by means of jokes.
- a set of categories which allow the chatbot to recognize an humoristic intent in the user sentences. This feature is obtained by connecting the chatbot knowledge base to external resources, such as the lexical dictionary WordNet and the CMU pronouncing dictionary [19], in order to detect the presence of humorous linguistic features in the sentence.

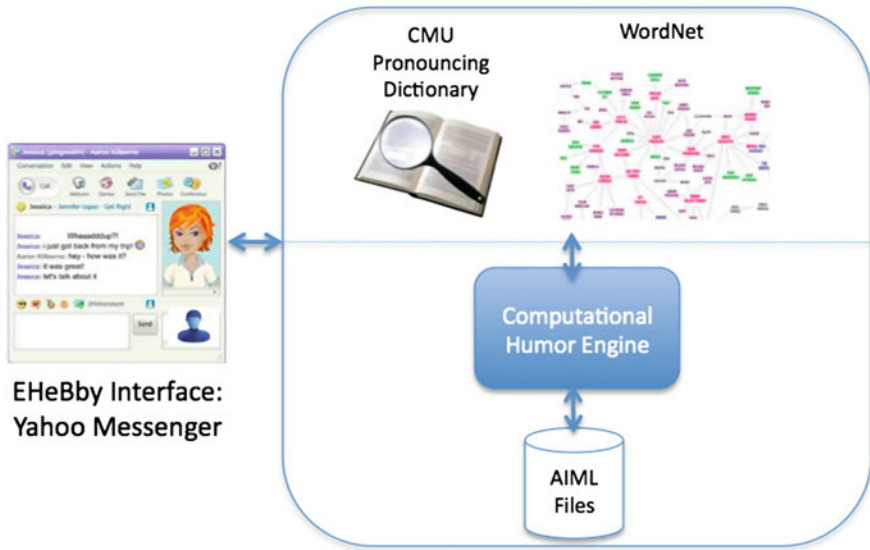


Fig. 3 Humoristic Bot architecture

The humour recognition has been implemented by means of the recognition, into the user sentences, of the peculiar humoristic texts features selected by Mihalcea and Strapparava [18].

3.3 Hybrid Architectures for Chatbots

3.3.1 Introduction of an “Intuitive” Component in the Chatbot’s Brain: LSAbot

The aim of this system is to add some sort of intuitive reasoning ability to chatbots, attempting to overcome the rigid pattern matching rules, proposing a sort of “semantic” matching”. We believe that this intuitive-associative capability can be obtained using the LSA (Latent Semantic Analysis) methodology [20]. The representation of information in a LSA based semantic, “conceptual” space and a consequent sub-symbolic geometric representation of the chatbot knowledge space, can help to better design a human-like conversational interface provided with intuitive, associative capabilities.

The proposed approach involves the following steps:

- Chatbot creation and training;
- Generation of a semantic space where to code the chatbot competences;
- Choice of the dialogue management technique necessary to obtain the selection of the best answer to deal with the current subject.

The first step consists in the creation of the traditional AIML KB and in the choice of a natural language micro-documents corpus appropriate to extend the chatbot competences. The knowledge base of the conversational agent is then sub-symbolically coded in the conceptual space by means of the Latent Semantic Analysis methodology [20]. The association capability is then obtained mapping the user question in the same conceptual space of the chatbot and comparing the coded query with the sub-symbolically coded knowledge elements of the chatbot using a suitable similarity measure between the user query vector and each sentence vector, representative of the answers present in the chatbot knowledge base. In particular each query of the user is coded in the same conceptual space by means of the “folding-in” technique. Let q be the user query and \mathbf{q} its associated vector, let s be the one of the knowledge base sentences and \mathbf{s} its corresponding vector; the similarity between the query and the sentence can be evaluated according to vector similarities measures, like, for example, the cosine between the two vectors \mathbf{s} and \mathbf{q} .

The conversation between the user and the chatbot can take place after the choice of the dialogue management technique, necessary to obtain the selection of the best answer to deal with the current subject.

4 Integration of CyD and LSAbot: A Rational/Intuitive Chatbot Architecture

In this architecture common sense and intuitive reasoning capabilities have been integrated in a chatbot. The main idea is to allow the chatbot to exploit information stored in both a common sense ontology and in a semantic space, but the peculiar feature relates the interconnection of these two different knowledge bases. In fact we propose the induction of a sub-symbolic layer in an ontology by means of a mapping of the concepts into a semantic space. Given a specific Cyc microtheory its mapping is performed building a LSA semantic space where the corpus training is composed of the comments associated to the ontology concepts and other micro-documents, each associated to a single concept. To each concept are associated one or more documents, which are represented by vectors into the space. Therefore each concept is projected in the space, and a layer of sub-symbolic, semantic relationships between concepts, given by their mutual geometric distance in the space, is automatically created. In this way the ontology concepts can be related to each other by means of their sub-symbolic relationships as well as by means of the relationships defined into the ontology. The conversational agents can exploit this sub-symbolic layer to retrieve semantic relations between ontological concepts already stored in the KB which are not easily reachable by means of the traditional ontology rules but that are more easily reachable through associative sub-symbolic paths.

Therefore the chatbot brain is comprised of two different but interconnected areas as shown in Fig. 4.

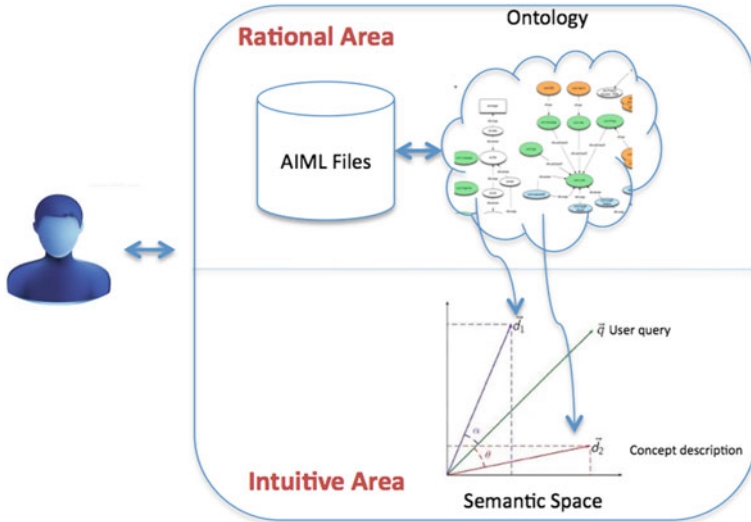


Fig. 4 Rational/Intuitive chatbot architecture

The former is a rational area, made of the Cyc ontology and the standard AIML knowledge base of the chatbot. The latter is an intuitive area, made of a semantic space in which both Cyc concepts, AIML categories and user queries are mapped.

Each ontology concept is encoded as a point in the multi-dimensional semantic space using its Cyc definition and its related documents and is, as a result, identified by a set of vectors. The geometric similarity measure, establishes a correlation between the vectors and therefore a semantic sub-symbolic link among the concepts.

In particular given a concept c_k , and its associated vector \mathbf{c}_k , the set CR of concepts sub-symbolically conceptually related to the concept c_k can be obtained determining the n concepts associated to the set of vectors whose similarity measure with c_k is higher than an experimentally fixed threshold T :

$$CR = \{ \mathbf{c}_i, \text{ with } i = 1, \dots, n \mid sim(\mathbf{c}_k, \mathbf{c}_i) \geq T \} \tag{1}$$

where $sim(\mathbf{c}_k, \mathbf{c}_i)$ is a properly defined geometric distance, related to the cosine between the vectors \mathbf{c}_k and \mathbf{c}_i .

The chatbot can exploit the semantic layer through new specific Aimpl tags introduced for this interaction. In particular, the *relatedConcept* tag allows the chatbot to retrieve the concepts conceptually related to a specific ontology concept, while *sentenceConcept* tag allows the chatbot to retrieve the concepts related to a sentence. In summary, the chatbot can dialogue with the user, exploiting its standard knowledge base or properly query the ontology by means of “ad-hoc” defined Aimpl tags, but it can also exploit the intuitive, sub-symbolic layer through other Aimpl tags, created for trigger an intuitive behaviour of the chatbot. Furthermore, the chatbot can improve

its knowledge base adding unknown concepts introduced by the user in the conversation, by means of an “ontology targeting” mechanism. Every time an unknown concept is introduced by the user in the conversation, the chatbot invites the user to give him a definition of the concept. The process is similar to what happens in real life when someone introduces a new term or concept, and we ask them for further explanation. The description of the new concept provided by the user is then mapped in the semantic space by means of the “folding-in” technique. The resulting vector is then compared through the vector similarity measure according to formula 1 with the concepts already stored into the microtheory. This procedure is obtained by means of the aforementioned *sentenceConcept* tag. The new concept is then added by the chatbot into the microtheory through the *addConcept* tag and then is linked to the sub-symbolically conceptually related concepts through the *cyc-assert* tag by means of a *#\$conceptuallyRelated* predicate.

At the end of the dialogue, the conversation files can be analyzed by the ontology expert which analyzes the concepts added by the chatbot together with their associated values of closeness with the Cyc concepts to which they have been linked. He then can establish whether to keep the new concepts in the ontology or to delete it. The proposed architecture was evaluated in [21] mapping the concepts of two different microtheories of Cyc in a semantic space, a smaller one in order to validate the proposed technique on the entire set of microtheory concepts and a greater one in order to carry out a test on one practical scene. The chosen application domains are the american academic structure, described by *AcademicOrganizationMt* microtheory, and the *BiologyMt*. The following dialogue is a small example of a possible interaction with the chatbot.

User: *I need information about departments related to the study of plants.*
 Chatbot: *Do you want information about the BiologyDepartment?*
 User: *Can you indicate me other related Departments?*
 Chatbot: *I have found a relation with the Agriculture Department.*
 User: *Can you give me information about it?*
 Chatbot: *It is a specialization of AcademicDepartment, instances of which conduct teaching and research in Agriculture.*
 User: *Can you give me information about some Private University?*
 Chatbot: *Sorry, I don't have any information about it. Can you help me in finding a definition of Private University?*
 User: *The definition is a university that is run without the control of any government entity.*
 Chatbot: *Ok, I can add this information In my KB. It is right to relate this concept to the University concept?*
 User: *Yes, it is.*

The following category allows the chatbot to understand the user information requests about a generic argument matched in the pattern by the wildcard *. Its value is recovered in the template through the *star* tag ,stored in the variable *userSentence* and hidden to the user by means of the *think* tag.

```
<category>
  <pattern> I NEED INFORMATION ABOUT * </pattern>
  <template>
    <think>
      <set name='userSentence'></star>
    </set>
    </think>
    <srai>RELATED CONCEPT</srai>
  </template>
</category>
```

The following category is recursively called by the previous one, by means of the *srai* tag. In its template the Cyc concept most related to the user request is searched through the *sentenceConcept* tag. If a concept is detected, it is stored in the *cycConcept* variable, and the chatbot asks to the user if the concept detected is that for which he was querying.

```
<category>
  <pattern> RELATED CONCEPT </pattern>
  <template> Do you want information about
    <set name='cycConcept'>
      </sentenceConcept >
    </set>?
  </template>
</category>
```

If the user asks for an information related to an unknown concept, the chatbot asks him for a definition.

```
<category>
  <pattern>
    CAN YOU GIVE ME INFORMATION ABOUT *
  </pattern>
  <template>
    <think><set name='cycConcept'>
      <cycterm></star></cycterm>
    </set></think>
    <condition>
      <li name=' cycConcept' value='NULL'>
        Sorry, I don't have any information about it.
        Can you help me in finding a definition of </star>?
      </li>
      <li>... </li>
    </condition>
  </template>
</category>
```

The user gives to the chatbot the concept definition, and the chatbot searches for conceptually related concepts to which the new one can be linked.

```
<category>
  <pattern> THE DEFINITION IS * </pattern>
  <template>
```

```

<think> <set name='userSentence'></star></set>
</think>
Ok, I can add this information in my KB.
It is right to relate this concept to the
<set    name='conceptuallyRelated'>
  </ sentenceConcept >
</set> concept?
</template>
</category>

```

5 Conclusion

In this chapter we summarize several works proposed to design human like, conversational agents overcoming the limitations of traditional rule-based chatbots like Alice. The first step was the formalization of a rational component in the chatbot KB in order to provide chatbots of reasoning and natural language processing capabilities. Afterwards, an intuitive component has been added to the chatbot KB, exploiting the associative properties deriving from natural language elements representation in a semantic space model. Finally the two components have been integrated together with the aim of simulating the main areas of the human brain. The proposed approaches have been implemented and tested in several applications on specific domains. Future works will regard the enhancement of the proposed framework introducing a dynamic analysis of conversation in the semantic, “conceptual” space.

Acknowledgments This work has been partially supported by the PO FESR 2007/2013 grant G63F12000240004 funding the OnSicily project.

References

1. L'Abbate, M., Thiel, U., Kamps, T.: Can proactive behavior turn chatterbots into conversational agents?. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT '05), pp. 173–179. IEEE Computer Society, Washington, DC, USA (2005). doi:[10.1109/IAT.2005.49](https://doi.org/10.1109/IAT.2005.49).<http://dx.doi.org/10.1109/IAT.2005.49>
2. Wallace, R.S.: The anatomy of A.L.I.C.E. In: Epstein, R., Roberts, G., Beber, G. (eds.) Parsing the Turing Test, pp. 181210. Springer Science+Business Media. London. ISBN 978-1-4020-6710-5 (2009)
3. Pilato, G., Vassallo, G., Augello, A., Vasile, M., Gaglio, S.: Expert chat-bots for cultural heritage. *Intell. Artif.* **2**(2), 25–31 (2005)
4. Kino, H.: Coursey living in CYN: mating AIML and CYC together with program N (2004)
5. Resource. <http://www.cyc.com>
6. Stock, O.: Language-based interfaces and their application for cultural tourism. *AI Mag.* **22**(1), 85–98 (2001)
7. Mori, K., Jatowt, A., Ishizuka, M.: Enhancing conversational flexibility in multimodal interactions with embodied lifelike agent. In: Proceedings of the 8th international conference on Intelligent user interfaces (IUI '03), pp. 270–272. ACM, New York (2003). doi:[10.1145/604045.604096](https://doi.org/10.1145/604045.604096).<http://doi.acm.org/10.1145/604045.604096>

8. Koutamanis, A.A.: An intelligent verbal interface for the retrieval of pictorial architectural information. In: Agger, K., Christiansson, P., Howard, R (eds.) *Distributing Knowledge in Building Proceedings*, vol. 2, pp. 6–13. Aarhus School of Architecture, Aarhus (2002)
9. Zdravkova, K.: Conceptual framework for an intelligent chatterbot. In: *Proceedings of the 22nd International Conference on Information Technology Interfaces*, 2000. ITI 2000, p. 189, 194, pp. 16–16 (2000)
10. Neves, A.M.M., Barros, F.A., Hodges, C., iAIML: a mechanism to treat intentionality in AIML chatterbots. Tools with artificial intelligence, 2006. In: *18th IEEE International Conference on ICTAI '06*, p. 225, 231 (2006). doi:[10.1109/ICTAI.2006.64](https://doi.org/10.1109/ICTAI.2006.64)
11. Augello, A., Santangelo, A., Sorce, S., Pilato, G., Gentile, A., Genco, A., Gaglio, S.: MAGA: a mobile archaeological guide at agrigento, workshop giornata nazionale su guide mobili virtuali 2006. ACM-SIGCHI, Torino, 18 Oct 2006 Online Proceedings <http://hclab.uniud.it/sigchi/doc/Virtuality06/index.html>
12. Axelsson, J., Cross, C., Ferrans, J., McCobb, G., Raman, T., Wilson, L.: XHTML+Voice profile 1.2, W3C Note. (2004)
13. IBM Multimodal Tools. <http://www-306.ibm.com/software/pervasive/multimodal/>
14. Sorce, S., Augello, A., Santangelo, A., Pilato, G., Gentile, A., Genco, A., Gaglio, S.: A multimodal guide for the augmented campus. In: *Proceedings of the 35th annual ACM SIGUCCS fall conference (SIGUCCS '07)*, pp. 325–331. ACM, New York (2007). doi:[10.1145/1294046.1294123](https://doi.org/10.1145/1294046.1294123)<http://doi.acm.org/10.1145/1294046.1294123>
15. Antonella, S., Agnese, A., Salvatore, S., Giovanni, P., Antonio, G., Alessandro, G., Salvatore, G.: A virtual shopper customer assistant in pervasive environments. In: Meersman, R., Tari, Z., Herrero, P(eds.) *Proceedings of the 2007 OTM Confederated International Conference on the Move to Meaningful Internet Systems—Volume Part I (OTM'07)*, pp. 447–456. Springer, Berlin (2007)
16. Caronia, A., Pilato, G., Augello, A., Gaglio, S.: Getting started with programW. Resorces <http://programw.sourceforge.net>
17. Heylen, D.: Talking head says “Cheese!” humour as an impetus for embodied conversational agent research CHI-2003 workShop: humour modeling in the interface
18. Mihalcea, R., Strapparava, C.: Bootstrapping for fun: web-based construction of large data sets for humour recognition. In: *Proceedings of the Workshop on Negotiation, Behaviour and Language (FINEXIN 2005)*, pp. 25–30. Ottawa (2005)
19. CMU Dictionary. <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>
20. Landauer, T.K., Foltz, P.W., Laham, D.: An introduction to latent semantic analysis. *Discourse Process.* **25**, 259–284 (1998)
21. Pilato, G., Augello, A., Vassallo, G., Gaglio, S.: Sub-symbolic semantic layer in Cyc for intuitive chat-Bots. ICSC, pp. 121–128. IEEE Computer Society, Los Alamitos (2007)