

EHeBby: An evocative humorist chat-bot

Giovanni Pilato^{a,*}, Agnese Augello^b, Giorgio Vassallo^b and Salvatore Gaglio^{a,b}

^a*ICAR-CNR Istituto di Calcolo e Reti ad alte prestazioni, Italian National Research Council, Viale delle Scienze, Ed. 11, 90128 Palermo, Italy*

^b*DINFO – Dipartimento di Ingegneria Informatica, University of Palermo, Viale delle Scienze, Ed. 6, 90128 Palermo, Italy*

Abstract. A conversational agent, capable to have a “sense of humor” is presented. The agent can both generate humorous sentences and recognize humoristic expressions introduced by the user during the dialogue. EHeBby is an entertainment oriented conversational agent implemented using the ALICE framework embedded into an Yahoo! Messenger client. It is characterized by two areas: a rational, rule-based area and an evocative area. The first one is based on well founded techniques of computational humor and a standard AIML KB. The second one is based on a conceptual space, automatically induced by a corpus of funny documents, where KB items and user sentences are mapped. This area emulates an associative/evocative behavior of the conversational agent, making her more attractive. EHeBby includes also an avatar that changes the face expression according to humoristic content of the dialogue.

1. Introduction

The interest about enhancing the interface usability of applications and entertainment platforms has increased in last years. Intelligent user interfaces can help people during the interaction with a system in a natural manner, trying to understand and anticipate user needs [22].

Pattern matching, finite-state-machines and frame-based models are commonly used as methodologies for designing chat-bots. This kind of techniques suffice for simple tasks, since they are based on a static process that assigns in advance all possible types to match [24]. One of the most widespread conversational agent technologies is ALICE [5], whose knowledge base is composed of question answer modules, called categories and described by the AIML language. This kind of interfaces can be improved through the integration of more sophisticated techniques [5,27]. In particular, there has been a great deal of research in order to integrate symbolic and sub-symbolic approaches [19].

As an example, in [23] ALICE-based chat-bots have been provided with advanced reasoning capabilities through the linking of the AIML interpreter with the OpenCyc commonsense ontology [21]. The benefits of ontological resources for a spoken dialog system have also been reported in [25]. In [18] the Latent Semantic Analysis (LSA) technique [20] has been exploited in order to achieve an associative matching between user questions and chat-bot answers, using the pattern-matching mechanism just as a “default” behavior.

One of the most sought goals is to enhance the realness of interaction of conversational agents, and one of the main features in conversations between human beings is the capability to generate and

*Corresponding author. E-mail: g.pilato@icar.cnr.it.

understand humor. As reported in [1] “Humor is part of everyday social interaction between humans”, and computational humor [3] deals with the automatic generation and recognition of humor.

Verbally expressed humor has been surveyed in literature [2] in last years. The analysis has concerned in particular very short expressions (jokes), like one-liners. A one-liner is a short sentence with comic effects, simple syntax, intentional use of rhetoric characteristics (e.g., alliteration, rhyme), and frequent use of creative language constructions [6,17].

Since during a conversation the user says short sentences, one-liners, jokes or gags can be good candidates for the generation of humorous sentences. As a consequence, computational humor techniques regarding one-liners can be customized for the design of a humoristic conversational agent.

In this paper we present a humoristic conversational agent capable to generate humoristic expressions, proposing to the user riddles, telling jokes, and ironically answering to the user. The work is an evolution of the architecture illustrated in [16]. In particular, we propose an enhanced chat-bot architecture, named EHeBby, with both rational and associative capabilities. In particular we have applied LSA [20] technique to a traditional, rule-based knowledge representation, in order to add an evocative module that can be embedded in an entertainment-oriented conversational agent. For this reason we have implemented two different but interconnected areas in the chat-bot “brain”.

The first one is a rational area, based on the standard AIML KB, which exploits both WordNet [14] and the CMU pronouncing dictionary [15] in order to understand the humorist level of user sentences.

The second one is an associative area obtained building an LSA-inspired semantic space in which the sentences stored in the standard AIML KB and a corpus of funny sentences are mapped. User questions and chat-bot answers are therefore connected each other by geometric similarity relationships in this space. As a matter of fact, the reciprocal geometric distance between concepts implicitly defines a “sub-symbolic” relationship net, which can be seen as a new “sub-symbolic semantic layer” automatically added to the chat-bot Knowledge Base. This sub-symbolic layer, which has the same psychological basis declared by LSA [20] can be exploited by the conversational agent during the dialogue with the user through ad hoc AIML tags, specifically created for trigger an associative behavior of the chat-bot. As a result, the chat-bot can dialogue with the user exploiting its standard pattern matching mechanism, but it can also make use of the evocative area. This second feature allows to exploit semantic relations between the items already stored in the EHeBby KB that are not easily reachable by means of the traditional pattern-matching rules but that are strictly related to the sentence introduced by the user.

EHeBby can detect, during the conversation with the user, the presence of humoristic expressions, listening and judging jokes and react changing the visual expression of the avatar, according to the perceived level of humor. The chat-bot shows a smiling face if the user sentence is perceived as “funny”, while it will show an “anger” face if the sentence contains adult slang. If none of the features characterizing a humoristic phrase is recognized in the sentence through the use of the rational area, the user question is mapped in the semantic space. If the chat-bot is not capable of recognizing the humorous level of the user sentence, the evocative module computes the semantic similarity between what is said by the user and the sentences stored in the chat-bot KB and tries to answer to the user with a funny expression which is conceptually closer to the user input.

Furthermore, EHeBby can improve its KB adding new jokes or riddles introduced by the user in the conversation. Specific categories in the AIML KB allow storing and mapping them in the evocative area of the chat-bot.

A prototype of the system has been implemented as a Yahoo instant messaging contact.

The subsequent sections report the background about chat-bots, WordNet and CMU dictionary, and computational humor; in section 3 the EHeBby architecture description is given, in section 4 some

```

<category>
<pattern> HELLO</pattern>
<template>
Hi, user!!!!
</template>
</category>

```

Fig. 1. An example of AIML category.

example of categories are illustrated, in section 5 a set of examples obtained on a set of both humoristic and non humoristic sentences is reported, together with a set of sentences obtained from the evocative area of the chat-bot. Conclusions are finally given at the end of the paper.

2. Background

2.1. Chat-bots

Chat-bot systems are dialogue software agents able to handle a conversation with the user by means of simple matching rules. An example is the open source chat-bot architecture ALICE [5]. In this kind of systems, the dialogue is based on a pattern matching algorithm looking for a match between the user's sentences and the information stored in the chat-bot knowledge base. The Alice knowledge base is described by a set of question-answer modules, named *categories*. Categories are structured with an XML-like language called AIML (Artificial Intelligence Mark-up Language). The main elements of the chat-bot knowledge base are described by means of specific AIML tags. Figure 1 shows an example of AIML category. The tag *pattern* encloses the user question, while the tag *template* encloses the chat-bot answer. The presence in the pattern of special symbols named *wildcards* allows a chat-bot to achieve a partial matching between the user question and the pattern. The template can contain other AIML tags, which enhance the dialogue chat-bot capabilities. As an example *set* and *get* tags allow chat-bot to save and get the values of variables, the *system* tag enables the execution of other programs, the *srai* tag recursively calls the pattern matching on another category.

The ALICE KB can be constantly increased by the botmaster by means of a "targeting" mechanism. The targeting procedure consists in the analysis of the conversation files in order to detect those user questions, which had an incomplete matching with the AIML pattern. As an example, a user question that matches a pattern with a wildcard is an opportunity to create a new, more specific pattern. As a consequence the botmaster can write new categories for that kind of questions.

2.2. WordNet and the CMU Pronouncing Dictionary

WordNet [14] can be described as a set of lexical terms organized in a semantic net. Each node represents a *synset*, a set of terms with similar meaning. A synset is characterized by a *gloss*, that may contain a short definition and, in some cases, also one or more example sentences. Each arc is a semantic or lexical relation between synsets. The relations are characterized by a hierarchical structure. In fact a hypernymy relation connects each synset with his higher synset. A synset s_1 is hypernym of the synset s_2 if s_1 conceptually includes s_2 . Each synset has at least one common hypernym with all the others. The root of this hierarchy is the concept *Entity*.

In order to explore the structure of WordNet we have introduced new, ad-hoc AIML tags which allow the chat-bot to extract for a specific synset its gloss (tag *gloss*), its hypernyms (tag *instanceOf*), its meronyms (tag *taghasPart*), its olonyms (tag *memberOf*), and so on.

The Carnegie Mellon University (CMU) Pronouncing Dictionary [15] is a machine-readable pronunciation dictionary for North American English that includes more than 125000 words with their associated pronunciation transcriptions in the given phoneme set. Currently, this set consists of 39 phonemes.

2.3. Computational humour

Humor is one of the most fascinating and complicated sides of the human behavior. Dealing with the cognitive aspect of humor, it contributes in getting and keeping people's attention, helping also retaining concepts, anecdotes, and ideas in memory [6].

Computational humor deals with the analysis of the humor with the main aim of computationally managing verbal humor. Two main issues of this field are the automatic generation of humorous text and the automatic recognition of humorous sentences [6,17].

The first issue has been coped with the HAHACRONYM project whose aim was the achievement of an acronym ironic re-analyzer and generator in a given context. The humorous effect has been obtained by automatically changing some words present in an acronym preserving the rhyme and rhythm [6]. In [2], a punning riddles generation system has been proposed equipped with an interactive user interface. Other systems are illustrated in [7–9].

The second issue, i.e. the automatic humor recognition, is more difficult to manage since it is based on both syntax and semantics of sentences entered by the user. An attempt has been proposed in [10] where the humorous intent inside short dialogues is recognized. In another work a humor classification system over a corpus of Italian quotations manually extracted and tagged from the Wikiquote project has been implemented [11].

One of the most known attempts has been presented in [4] where the problem of recognition of verbal humor has been managed both as a classification problem and as a task of style-features recognition. In particular the attention has been focused on the type of humor present in very short sentences, called one-liners. In particular the authors have detected the main texts humoristic features analyzing the humor literature [3,12,13] and selecting the features which can be computationally detected, such as alliteration, antinomy and adult slang.

3. EHeBby: The evocative humorist bot

EHeBby is a female chat-bot provided with sense of humor, she is capable of telling humorous anecdotes to the user and she is also capable of listening jokes, trying to understand their humorous level. We have chosen to design a female character since it is easier to justify “angry” expressions in case of rude riddles given by the user. This gives us also the possibility to change the face expression ranging from happy to angry. EHeBby reacts accordingly to the user jokes, showing she smiling if she considers the sentence “funny”. She will be angry if she considers the joke in poor taste. If she cannot understand the humorous level of sentences given by the user, she uses her evocative area looking for humorous replies which are conceptually close to the user input. If neither the evocative area of her brain is capable to detect any humor in the joke, she shows herself indifferent and answers with a neutral default category.

The system framework is illustrated in Fig. 2. The chat-bot can interact with two main areas. The first one is a “rational area” which consists of structured knowledge bases the chat-bot can use: the AIML

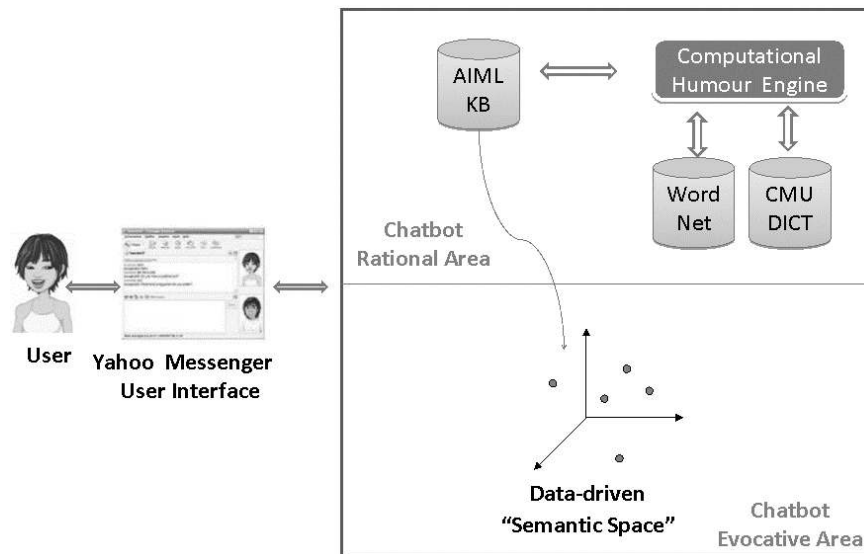


Fig. 2. The evocative chat-bot architecture.

chat-bot Knowledge Base [5] made of categories from which the chat-bot can also exploit by means of a computational humour engine, the well-founded WordNet lexical database [14] and the pronunciation dictionary CMU [15]. The second one is an “evocative area”; it is made of a semantic space in which AIML categories, user queries and amusing, humoristic sentences such as quotations, aphorisms, jokes and one liners are mapped.

Besides, the chat-bot has also been implemented as a Yahoo Messenger instant messaging client, in order to make the system easily accessible to the final user.

3.1. Rational area

The rational area consists of the ontology given by the WordNet [14] lexical database, the CMU [15] pronunciation dictionary and the chat-bot Knowledge Base. A computational humor engine and the chat-bot knowledge base give the core of the rational area. Four kinds of AIML categories are present:

1. the set of standard Alice categories, which allow the chat-bot to hold a general conversation with the user;
2. a set of categories aimed at the humorist sentences generation. These categories allow the chat-bot to answer the user in a humoristic way, by means of jokes. The generation of humor is obtained writing specific funny sentences in the template of the category;
3. a set of categories aimed at humorist or funny sentences evocation by means of the comparison between the user input with the sentences mapped in the semantic space belonging to the evocative area. The chat-bot will answer with the sentence semantically closer to the user input. These categories link the rational area with the evocative area of the chat-bot brain. The evocation is obtained by means of the presence of a specific AIML tag in the category template;
4. a set of categories which allow the chat-bot to recognize an humoristic intent in the user sentences. This feature is obtained connecting the chat-bot knowledge base to other resources, such as the lexical dictionary WordNet [14] and the CMU pronunciation dictionary [15], in order to detect the presence of humoristic linguistic features in the sentence.

We have paid more attention to the humour recognition techniques, because they have more impact to the final user even if they are more difficult to handle. The humour recognition has been implemented in the rational area by means of the identification, into the user sentences, of particular humoristic texts features. Among the features described in literature, we have considered the ones suggested by Mihalcea and Strapparava [4], which are *alliteration*, *antinomy* and *adult slang*.

3.1.1. Alliteration

The alliteration, the rhetoric figure consisting in the repetition of a letter, a syllable or a phonetic sound in consecutive words is a feature present in the most of humoristic sentences. The phonetic effect induced by these repetitions captures the attention of people listening it, and in the most of cases, produces a funny effect [4].

A module has been implemented in order to recognize this characteristic in a sentence. A pre-processing removes punctuation marks and stopwords from the sentence, then its phonetic transcription is obtained using the CMU dictionary [15]. The module then analyzes the phonetic transcription of the sentence in order to discover possible repetitions of the beginning phonemes in subsequent words.

The module finds an alliteration if:

- at least three words have in common the first phoneme;
- at least three words have in common the first two phonemes;
- at least three words have in common the first three phonemes.

Two examples of one-liners including one or more alliteration chains with their respective phonetic transcription are reported:

“C program run. C program crash. C programmer quit”

S IY1 . P R OW1 G R AE2 M . R AH1 N . S IY1 . P R OW1 G R AE2 M . K R AE1 SH . S IY1 . P R
OW1 G R AE2 M ER0 . K W IH1 T .

“If you can’t convince them, confuse them”

IH1 F . Y UW1 . K AE1 N T . K AH0 N V IH1 N S . DH EH1 M . K AH0 N F Y UW1 Z . DH EH1 M .

3.1.2. Antinomy

The presence of antinomies in a sentence is another feature producing a humoristic effect. A specific module for the detection of antinomies (of nouns, verbs, adverbs and adjectives) in a sentence has been developed; the module exploits the semantic relation defined into the lexical dictionary WordNet. In particular three different procedures have been implemented in order to detect into a sentence:

- a direct antinomy relation among nouns, verbs, adverbs and adjectives;
- an extended antinomy relation, which is an antinomy relation between a word and a synonym of its antonym. The relation is restricted to the adjectives;
- an indirect antinomy relation, which is an antinomy relation between a word and an antonym of its synonym. The relation is restricted to the adjectives.

These humoristic sentences contain antinomy relation:

“A *day* without sunshine is like, *night*”

“Artificial *intelligence* usually beats real *stupidity*”

“*Good* girls are *bad* girls that never get caught”

3.1.3. Adult slang

In many cases, the comic effect of jokes is due to the presence of words pertaining to the sexual domain, mainly used in the adult slang.

The chat-bot analyzes the presence of this kind of words using a list of terms classified as adult slang. As an example the following sentences are reported:

“100,000 sperm and you were the fastest?”
“Sex is like air; it’s not important unless you aren’t getting any”

3.1.4. AIML categories for generation and recognition of humor

The generation of humoristic sentences is obtained through the writing of suitable AIML categories, exploiting the pattern matching mechanism of Alice chat-bot.

The recognition of humoristic sentences is obtained by means of special AIML categories which allow the chat-bot to execute the modules developed in order to detect the humoristic features previously analyzed. The modules are executed by means of the system tag inserted into the template.

3.2. Evocative area

The evocative area consists in a sub-symbolic semantic representation added to the chat-bot Knowledge Base. This model is obtained encoding sentences stored into the AIML KB as vectors in a semantic space. The space is created through the statistical analysis of words co-occurrences into a corpus of texts by means of LSA technique. The semantic space will be used to store and retrieve also funny sentences that are not directly coded as AIML categories, but that can be more easily retrieved by exploiting an associative behaviour [18] according to a case-based reasoning-like model.

After the creation of the semantic space, each item of the KB is encoded as a point in the multi-dimensional semantic space using the folding-in technique [26]. The chat-bot exploits the associative area, attempting to “guess” semantic relations between the sentences written by the user and the items of its own KB. This feature is obtained mapping the user question in the same conceptual space of the chat-bot and comparing the coded query with the sub-symbolically coded knowledge elements of EHeBby, and applying a suitable similarity measure between the user query vector and each sentence vector, representative of the answers present in the chat-bot knowledge base.

EHeBby can also enhance its own AIML KB mapping in the evocative area new items like jokes, riddles and so on introduced by the user during the dialogue. The botmaster can also code these sentences in the rational area transforming them into AIML categories.

3.2.1. Building of an LSA-based semantic space

Given N documents of a text corpus let M be the number of unique words occurring in the documents set. Let $\mathbf{A}=\{a_{ij}\}$ be a $M\times N$ 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 paragraph. According to the Singular Value Decomposition theorem, \mathbf{A} can be decomposed in the product $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where \mathbf{U} is a column-orthonormal $M\times N$ matrix, \mathbf{V} is a column-orthonormal $N\times N$ matrix and $\mathbf{\Sigma}$ is a $N\times N$ diagonal matrix, whose elements are called singular values of \mathbf{A} .

The matrices \mathbf{U}_R and \mathbf{V}_R obtained after decomposition process reflect a breakdown of the original relationships into linearly independent vectors [2]. These independent R dimensions of the \mathfrak{R}^R space can be tagged in order to interpret this space as a “conceptual” space. Since these vectors are orthogonal,

they can be regarded as principal axes representing the “fundamental” concepts residing in the data driven space S generated by the LSA [18].

To evaluate the distance between two vectors \mathbf{x}_i and \mathbf{x}_j belonging to this space that is coherent with this probabilistic interpretation, a similarity measure is defined as follows:

$$sim(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \cos^2(\mathbf{x}_i, \mathbf{x}_j) & \text{if } \cos(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

We have chosen this kind of similarity measure since the elements of the starting matrix \mathbf{A} are the square roots of sample probabilities. If we distribute the contribution of Σ on both \mathbf{U} and \mathbf{V} , the dot product between two vectors of the space gives the element the square root of joint co-occurrence probability. In order to obtain a probability we have to calculate the square of the dot product. For the same reason we consider zero the products lower than zero.

The use of the square of the cosine does not affect the relative computation of the distance between two vectors, since the cos function is monotone. It is useful only for the statistical “interpretation” of SVD and not for “improving” the similarity recognition capability.

3.2.2. Evocation of answers during the dialogue

After the creation of the semantic space S , AIML categories, sentences riddles and quotations, are encoded as points in S , using the folding-in technique [26].

The geometric similarity measure between two items as defined in Eq. (1), establishes a semantic, weighted, sub-symbolic link between them. The net of these semantic connections can be seen as a semantic “layer” superposed to the existing AIML categories. In particular given a vector \mathbf{u} , associated to a sentence \mathbf{s} , the set of vectors sub-symbolically conceptually related to the sentence \mathbf{s} is given by:

$$CR_{(s)} = \{\mathbf{u}_i \mid sim(\mathbf{u}, \mathbf{u}_i) \geq T\} \quad (2)$$

where $CR_{(s)}$ is the set of vectors \mathbf{u}_i , associated to the sentence \mathbf{s} whose similarity measure is higher than an experimentally fixed threshold T ($T \in \mathbb{R}; 0 \leq T \leq 1$).

The chat-bot exploits the semantic layer through new specific AIML tags introduced for this interaction: the *relatedSentence* and the *randomRelatedSentence* tags. The former allows the chat-bot to retrieve the closer riddle to the user sentence according to the computation of the similarity measure given by Eq. (1). The latter allows the chat-bot to randomly select one of the most conceptually related riddles to a sentence introduced by the user among the sentences included in $CR_{(s)}$, where s is the sub-symbolic coding of the sentence written by the user.

3.2.3. AIML Targeting

The AIML Targeting is a mechanism which allows increasing the chat-bot KB during the conversation with the user. Every time the user wants to tell the chat-bot a new joke or a riddle, the chat-bot memorizes it into its evocative area for further recall mapping it as a vector into the semantic space. Its similarity with the riddles, jokes and generally speaking, items of the KB already mapped into the space is then computed according to Eq. (2).

The new riddle is added by the chat-bot into the evocative part of its KB through the *addRiddle* tag. At the end of the dialogue, the botmaster can furthermore analyze the conversation files. He can evaluate the new sentences inserted by the chat-bot and decide if the new items should be retained or discarded.



Fig. 3. Expressions for the EHeBby avatar.

3.3. EHeBby Yahoo! Messenger avatar

EHeBby has been connected also to instant messaging software in order to make it more interesting and accessible to end-users. EHeBby can communicate in real time with other users and it can be incorporated into a list of contacts as a real-user.

Among the possible instant messaging programs, it has been decided to connect the conversational agent with Yahoo Messenger, since it is one of the most widely used in the world and it owns a Java classes (JYMSG) library, which allows us to interact with it. Therefore a Yahoo client for our agent has been created. Through the appropriate configuration of the AIML interpreter (Program D) it has been possible to realize the connection between the agent and instant messaging software.

The use of Instant Messaging program has also allowed the use of available avatars and their different expressions associated according to the different moods, as shown in Fig. 3.

4. Examples of chat-bot knowledge base

4.1. AIML categories for the generation of humoristic sentence

The following categories are two examples of AIML question-answer modules needed to hold this dialogue. In these categories the potentiality of the AIML tags are exploited in order to allow the chat-bot keeping trace of the conversation (*that* tag) and to set and get dialogue information (*set* and *bot* tags in the example).

User: What is your name?

EHeBby: My name is EHeBby .. and you? What is your name?

User: Marco

Funny Bot: Nice to meet you, Marco!!

```
<category>
<pattern>WHAT IS YOUR NAME</pattern>
<template> My name is <bot name/>.. and you? What is your name? </template>
< /category>
```

```
<category>
<pattern>*</pattern>
<that>MY NAME IS EHEBBY ... AND YOU? WHAT IS YOUR NAME?</that>
```

```
<template> Nice to meet you, <set name="user"> <star/> </set>!! </template>
< /category>
```

As explained in Section 2.1. the *pattern* tag delimits what the user can say. Every time the “<pattern>” is matched, the corresponding “<template>” is activated. In this case the first sentence “What is your name?” activates the template “<template> My name is <bot name/>.. and you? What is your name? </template>” and “<bot name/>” recalls the name of the chatbot “EHeBby”. In the second category, the instruction “<set name="user"> <star/> </set>” sets the “user” variable to whatever is matched by the “*” wildcard in the pattern.

The following is an example of an humoristic dialogue:

User: Tell me a joke

EHeBby: Do you have any preference?

User: Yes

EHeBby: What kind of argument do you prefer?

User: What kind of argument do you know?

EHeBby: Blondes, Lawyer, Male, Political, Religious, Computers, . . .

User: Computers.

EHeBby: COFFEE.EXE is missing. Insert cup and press any key.

The previous dialogue can be obtained by means of the following categories:

```
<category>
<pattern>TELL ME A JOKE</pattern>
<template>Do you have any preference?</template>
< /category>
```

```
<category>
<pattern>YES</pattern>
<that>DO YOU HAVE ANY PREFERENCE </that>
<template>What kind of argument do you prefer?</template>
< /category>
```

```
<category>
<pattern>WHAT KIND OF ARGUMENT DO YOU KNOW</pattern>
<template>Blondes, Lawyer, Male, Political, Religious, Computers, . . . </template>
< /category>
```

```
<category>
<pattern>COMPUTERS</pattern>
<template>
<think> <set name="topic">COMPUTERS</set> </think>
<random>
<li> Computers will never replace books. You can't stand on a floppy disk to reach a high shelf</li>
<li>Robots will be able to buy happiness, but in condensed chip form</li>
<li>COFFEE.EXE is missing. Insert cup and press any key </li>
```

```
< /random>
< /template>
< /category>
```

In particular, in the last category the instruction “<think> <set name="topic">COMPUTERS</set> </think>” hides in the dialogue the collateral effect of setting the “topic” variable to “computers”, which identifies the current topic of the dialogue, while the “<random>” tag simply allows to randomly choose an answer among “Computers will never replace books. You can’t stand on a floppy disk to reach a high shelf”, “Robots will be able to buy happiness, but in condensed chip form” or “COFFEE.EXE is missing. Insert cup and press any key”.

4.2. AIML categories for humor recognition

The recognition of humoristic sentences is obtained by means of special AIML modules executed by means of the *system* tag inserted into the template, as shown in the following categories:

```
<category>
<pattern>CAN I TELL YOU A JOKE</pattern>
<template>Yes you can</template>
< /category>

<category>
<pattern>*</pattern>
<that>YES YOU CAN</that>
<template><system>java HumorLevel <star/> </system> </template>
< /category>
```

In this case the second category is activated only if the previous answer of the chatbot was “Yes you can” (according to the “<that>” tag behavior), and the template calls a java application “HumorLevel” that calculates the humor level of what has been matched with the “*” wildcard (i.e. what the user said).

This is a dialogue example that shows the chat-bot reaction to a funny joke:

User: Doctor gave a man six months to live. The man couldn’t pay his bill, so he gave him another six months.

EHeBby: Ehehe! You are very funny!! Good.. your joke is fantastic!!

This other example shows the chat-bot reaction to a non-humoristic user sentence:

User: The book is on the table.

EHeBby: I don’t enjoy myself.. !! Your joke is stupid!!

The last example shows the chat-bot reaction to a rude joke:

User: 100,000 sperm and you were the fastest?

Funny Bot: >:(You are a pig !! You can’t tell this to a lady!!

4.2.1. AIML categories for the evocation of humorous sentences

The semantic space of the EHeBby chat-bot evocative area has been created using 3835 funny sentences downloaded from internet one-liners repositories [28,29] as well as the standard categories stored in the AIML knowledge base.

The evocation is obtained by means of the presence of specific AIML tags. As an example the tag *relatedSentence* in the category template triggers the comparison between the user inputs with the sentences mapped in the semantic space belonging to the evocative area choosing the sentence semantically closer to the user question. The tag *randomRelatedSentence* allows the chat-bot to answer with one of the sentences semantically related to the user input, according to the set defined in Eq. (2).

As an example given the following category:

```
<category>
<pattern>What do you think about *</pattern>
<template> <randomRelatedSentence> <star/> </randomRelatedSentence> </template>
< /category>
```

If the user question is:

User: What do you think about marriage?

The chat-bot will answer randomly in one of the following ways:

EHeBby: Marriage is an expensive way of getting your laundry done for free.

EHeBby: Benjamin Franklin said “Keep your eyes wide open before marriage, half shut afterwards”.

EHeBby: Love may be blind but marriage is a real eye-opener.

While the standard Alice chat-bot answers:

Alice: The state of being married; relation between husband and wife; married life.

The following dialogue shows a comparison between the answers given with the traditional pattern matching by the chat-bot Alice and the answers given by the evocative chat-bot using the *relatedSentence* tag

User: You are ignorant.

Alice: We can learn new things.

EHeBby: Everybody is ignorant, only on different subjects

User: What is the difference between woman and men?

Alice: I think you already know the answer. I will search for it.

EHeBby: Men always want to be a woman’s first love. Women have a more subtle instinct; what they like to be is a man’s last romance.

User: See you tomorrow

Alice: Goodbye

EHeBby: Tomorrow is a mystery

User: Do you enjoy working?

Alice: I enjoy working 24 hours every day.

EHeBby: I only work to enjoy when I am not working

User: My computer has the windows operating system.

Alice: Where did he get it?

EHeBby: What do Windows and a handgun have in common? Both are harmless while not loaded.

User: Are you a pessimist or an optimist?

Alice: Am I a pessimist or an optimist? Yes.

EHeBby: A pessimist is one who makes difficulties of his opportunities and an optimist is one who makes opportunities of his difficulties. (Harry Truman)

4.3. AIML categories for targeting

EHeBby is able to increase its sub-symbolic knowledge base by means of a targeting process, which allows the mapping of new riddles in the semantic space. The targeting is obtained by means of the ad-hoc created AIML tag *addRiddle*, as shown in the following category.

```
<category>
<pattern>Listen this joke *</pattern>
<template> <system>java HumorLevel <star/> </system>
<think>
<addRiddle> <star/> </addRiddle>
< /think>
< /template>
< /category>
```

The *addRiddle* tag has the effect to code as vector in the semantic space a sentence introduced by the user by means of the folding-in procedure. The entire procedure is hidden to the user by the tag *think*. The user will see only the chat-bot reaction to his joke.

5. Examples of evaluation

In order to validate the EHeBby architecture, according to what proposed in [17] a humorous data set has been created. The dataset consists of 100 positive examples (humorous phrases) extracted by ad-hoc selected internet sites related to the subject, and a non-humorous data set consisting of 100 negative examples (non-humorous phrases) always extracted from the Internet and chosen among titles of newspapers, proverbs and sets definitions.

Figure 4 shows the features of positive data set, while Fig. 5 shows that one relative to the negative data set.

In a first set of experiments, we have evaluated literature algorithms implemented on each one of both data sets, in order to recognize, if present, the stylistic humor-specific features: alliteration, antinomy, and adult slang. Results report 66% of correct recognition and 34% of no recognition.

In the second set of experiment, we evaluated the number of sentences properly recognized on each one of the two data sets. The results are reported to be 81% of correct recognition and 19% of no recognition.

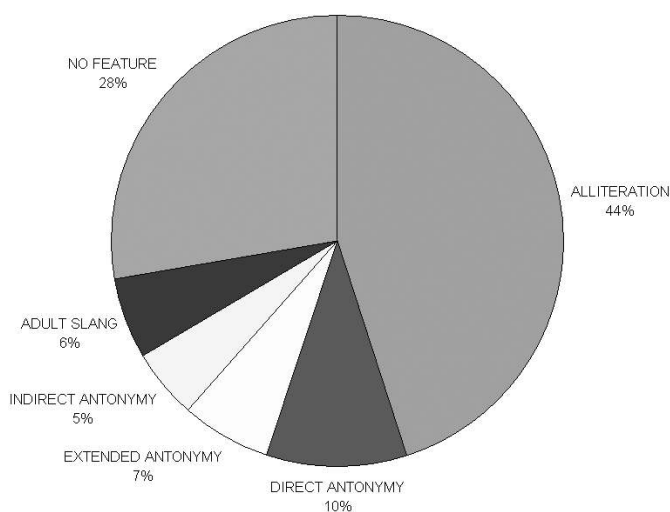


Fig. 4. Features of the positive data-set.

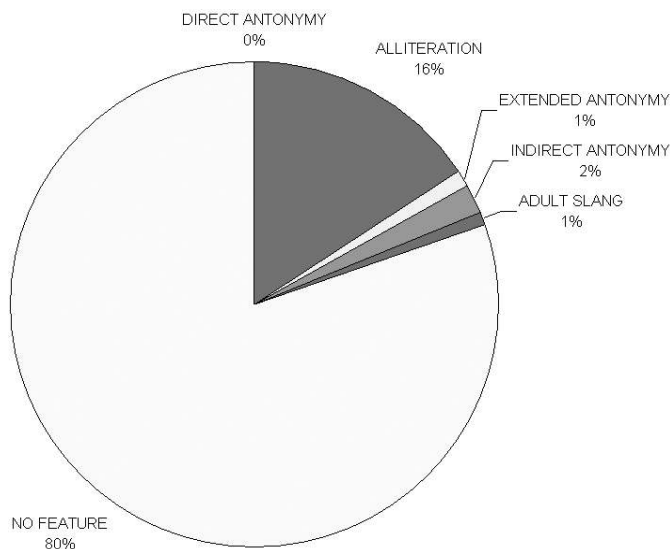


Fig. 5. Features of the negative data-set.

From the obtained results, we can state that on the basis of 100 random humorous phrases, the system is able to recognize and correctly classify 66% of the sentences, while on a random set of 100 non-humorous phrases there are only 19 false positives or 19 phrases that despite being labeled as “non-humorous” are labeled “humorous” instead.

Then we performed a new set of experiment using a data set of 200 phrases, equally distributed between humorous and non-humorous sentences, getting more satisfactory results, given by 73% of correct recognition and 27% of no recognition.

The results can be considered satisfactory, because even if with the rational part the chat-bot cannot identify the humorous level of sentences written by the user, the dialogue can continue in a funny way. As a matter of fact EHeBby exploits her evocative area, which allows retrieving amusing sentences so

that the dialogue can continue in a funny manner.

As an example we report some of the funny sentences stored in the evocative area with their closeness to the user entry, compared to the answers given by Alice [5]:

Example 1:

User: What do you think about chocolate?

Alice: Cocoa flavoring used in a variety of food products.

EHeBby:

0.94 A quarter-ounce of chocolate = four pounds of fat. (Slick's Second Universe Law)

0.84 My pastor-husband has a sweet tooth, so I knew the chocolate chip cookies I'd just baked might disappear before I returned from running errands. To discourage him, I taped a verse on the wrapped goodies: "Everything is permissible for me, but not everything is beneficial." – 1 Cor. 6:12.

0.82 Shopper's Hymn – "Sweet By and By"

0.82 Equal amounts of dark chocolate and white chocolate, is a balanced diet.

Example 2:

User: What can you say about children?

Alice: I will think about it and get back to you.

EHeBby:

0.69 When a friend asked his six-year-old brother why babies are spanked when they are born, the youngster replied, "To get them used to it. (Doug Belknap & Chad Goodwin)

0.68 Praise your children openly, reprove them secretly. (W. Cecil)

0.68 When speaking to your children: say what you mean, mean what you say, but don't be mean when you say it.

Example 3:

User: What do you think of life?

Alice: Life is the opposite of death.

EHeBby:

0.987 Life is an echo.

0.986 Life is a roller coaster.

0.985 Never confuse having a career with having a life.

0.985 You should not confuse your career with your life.

Example 4:

User: Have you ever seen Santa Claus?

Alice: No I have not seen it. What's it like?

EHeBby:

0.983 The Bermuda Triangle got tired of warm weather. It moved to Finland. Now Santa Claus is missing.

0.746 There are 4 Santa Claus stages:1) When you believe in Santa.2) When you don't believe in Santa.3) When you are Santa.4) When you look like Santa.

6. Conclusions

An entertainment-oriented conversational agent has been proposed. The chat-bot exploits humoristic features such as alliteration, antinomy and adult slang for humor recognition. However the chat-bot is able to manage a dialogue using an evocative/associative capability given by a sub-symbolic knowledge representation given by a semantic space automatically induced from data. This allows an amusing, non repetitive, non rigid dialogue, which is a desirable feature for an entertainment-oriented chat-bot.

Future work will regard increasing the size of the data sets used for training EHeBby in order to map more samples in the evocative area, the improvement of humor recognition techniques and a better exploitation of both the rational and the evocative area of the chat-bot.

Acknowledgments

Authors would thank Ing. Gaetano Saccone for his partial contribution to the work.

References

- [1] D. Heylen, Talking Head Says “Cheese!” *Humour as an impetus for Embodied Conversational Agent Research CHI-2003 Workshop: Humour Modeling in the Interface*.
- [2] G. Ritchie, Prospects for Computational Humor. In *Proceedings of 7th IEEE International Workshop on Robot and Human Communication (ROMAN-98)*, Takamatsu, Japan, October 1998, 283–291.
- [3] S. Attardo, *Linguistic Theory of Humor*, Mouton de Gruyter, Berlin, 1994.
- [4] R. Mihalcea and C. Strapparava. Learning to laugh (automatically): Computational Models for Humour Recognition, *Computer Intelligence* **22** (2006).
- [5] Alice Chabot: www.alicebot.org.
- [6] O. Stock and C. Strapparava, Getting serious about the development of computational humor, in *proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, 2003, 59–64.
- [7] J. McKay, Generation of idiom-based witticisms to aid second language learning. In Stock et al., 2002, 77–87.
- [8] J.M. Taylor and L.J. Mazlack, Computationally recognizing wordplay in jokes. In *Proceedings of Cognitive Science Conference*, Stresa, Italy, 2004, 2166–2171.
- [9] G. Lessard and M. Levison, Computational modeling of riddle strategies. In *ALLC/ACH Joint Annual Conference*, Georgetown University, Washington, DC, 1993, 120–122.
- [10] J.M. Taylor and L.J. Mazlack, Toward Computational Recognition of Humorous Intent, *Cognitive Science Conference 2005 Proceedings (CogSci 2005)*, Stresa, Italy, July 2005, 2166–2171.
- [11] D. Buscaldi and P. Rosso, Some Experiments in Humour Recognition Using the Italian Wikiquote Collection, *Applications of Fuzzy Sets Theory Lecture Notes in Computer Science*, 2007.
- [12] W. Ruch, Computer with a personality? Lessons to be learned from studies of the psychology of humor. In *Proceeding of The April Fools Day Workshop on Computational Humor*, University of Twente, the Netherlands, 2002, 57–70.
- [13] C. Bucaria, Lexical and syntactic ambiguity as a source of humor, *Humour* **17**(3) (2004), 279–309.
- [14] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K.J. Miller, Introduction to WordNet: An On-line Lexical Database, *International Journal of Lexicography* **3**(4) (1990), 235–244.
- [15] CMU Dictionary: <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>.
- [16] A. Augello, G. Saccone, S. Gaglio and G. Pilato, Humorist Bot: Bringing Computational Humor in a Chat-Bot System. *Proc. of First International Workshop on Intelligent Interfaces for Human-Computer Interaction (IIHCI-2008)*, Technical University of Catalonia, Barcelona, Spain, March 4th–7th, 2008.
- [17] R. Mihalcea and C. Strapparava, 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)*, Ottawa Canada, 2005, 25–30.
- [18] 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 *Lecture Notes in Artificial Intelligence*, vol. 3673/2005, Springer-Verlag GmbH, 381–392.
- [19] G. Pilato, A. Augello, G. Treccarichi, G. Vassallo and S. Gaglio, LSA-Enhanced Ontologies for Information Exploration System on Cultural Heritage. *AI*IA Workshop for Cultural Heritage*, University of Milan Bicocca, Milano, Italy, 2005.

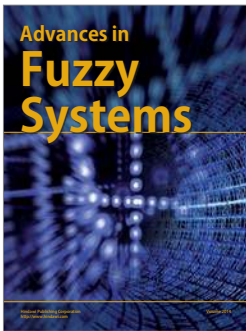
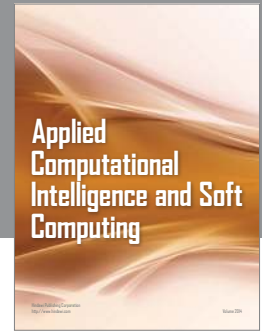
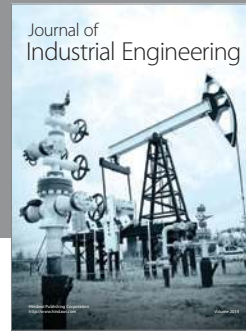
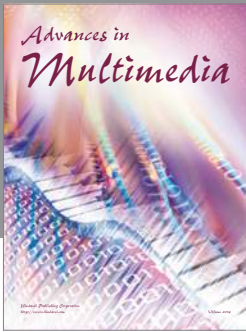
- [20] T.K. Landauer, P.W. Foltz and D. Laham, Introduction to Latent Semantic Analysis, *Discourse Processes* **25** (1998), 259–284.
- [21] <http://research.cyc.com>.
- [22] G.O. Sing, K.W. Wong, C.C. Fung and A. Depickere, Towards a more natural and intelligent interface with embodied conversation agent. *ACM Proc. of the 2006 international Conference on Game Research and Development*, vol. 223, 2006, 177–183.
- [23] K. Coursey, Daxtron Laboratories, Inc. Living in CYN: mating AIML and CYC together with Program N. 2004.
- [24] K.-M. Kim, J.-H. Hong and S.-B. Cho, A semantic Bayesian network approach to retrieving, *Information Processing and Management* **43** (2007), 225–236.
- [25] M. Mehta and A. Corradini, Developing a Conversational Agent using Ontologies, *12th International Conference on Human-Computer Interaction*, Lul 2007, Beijing, China, 22–27.
- [26] M. Berry et al., Using Linear Algebra for Intelligent Information Retrieval, *SIAM Review* **37**(4) (1995), 573–595.
- [27] O.S. Goh, C. Ardil, W. Wong and C.C. Fung, A Black-box Approach for Re-sponse Quality Evaluation Conversational Agent System, *International Journal of Computational Intelligence* **3** (2006), 195–203.
- [28] <http://www.oneliners-and-proverbs.com/>.
- [29] <http://www.bdwebguide.com/jokes/1linejokes-1.htm>.

Giovanni Pilato received his “cum laude” degree in “Ingegneria Elettronica” and the Ph.D. degree in “Ingegneria Elettronica, Informatica e delle Telecomunicazioni” from the University of Palermo, Italy, in 1997 and 2001, respectively. Since 2001 he is a staff research scientist of the ICAR-CNR (Istituto di CALcolo e Reti ad alte prestazioni, Italian National Research Council). He is also Lecturer at the DINFO (Dipartimento di ingegneria INFormatica) of the University of Palermo. His research interests include geometric techniques for data mining, natural language processing and Human-Computer Interfaces.

Agnese Augello received her “cum laude” Laurea degree in “Ingegneria Informatica” and the Ph.D. degree from the University of Palermo in 2004 and 2008, respectively. Her research interests are in the field of Natural Language Processing and Human-Computer Interfaces.

Giorgio Vassallo graduated, “cum laude”, in Physics at the Univeristy of Palermo in 1982. He has worked in the VLSI and Neural Network Lab of CRES (Centro per la Ricerca Elettronica in Sicilia) of Monreale (PA). Currently he is a research scientist at DINFO (Dipartimento di ingegneria INFormatica), University of Palermo, Italy. His research interests are in the field of neural networks, geometric techniques for data mining and natural language processing.

Salvatore Gaglio is full professor of Computer Science and Artificial Intelligence at DINFO (Dipartimento di ingegneria INFormatica), University of Palermo, Italy. He is member of various committees for projects of national interest in Italy and he is referee of various scientific congresses and journals. His present research activities are in the area of artificial intelligence and robotics. Currently he is also Delegated Director of the Palermo branch of ICAR-CNR (Istituto di CALcolo e Reti ad alte prestazioni, Italian National Research Council).



Hindawi

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

