AIML Knowledge Base Construction from Text Corpora

Chapter · January 2012

DOI: 10.1007/978-3-642-29694-9_12

Giovanni De Gasperis
Università degli Studi dell'Aquila

Isabella Chiari
Sapienza University of Rome

AIML Knowledge Base Construction from Text Corpora

Giovanni De Gasperis[1], Isabella Chiari[2] and Niva Florio[1]

[1] Dipartimento di Ingegneria e Scienze dell'Informazione, Matematica,
Università degli Studi dell'Aquila, L'Aquila, Italy
(giovanni.degasperis,niva.florio)@univaq.it
[2] Dipartimento di Scienze documentarie, linguistico-filologiche e geografiche,
Università degli Studi di Roma "La Sapienza", Roma, Italy
isabella.chiari@uniroma1.it

Abstract. Text mining (TM) and computational linguistics (CL) are computationally intensive fields where many tools are becoming available to study large text corpora and exploit the use corpora for various purposes. In this chapter we will address the problem of building conversational agents or chatbots from corpora for domain-specific educational purposes. After addressing some linguistic issues relevant to the development of chatbot tools from corpora, a methodology to systematically analyze large text corpora about a limited knowledge domain will be presented. Given the Artificial Intelligence Markup Language as the "assembly language" for the artificial intelligence conversational agents we present a way of using text corpora as seed from which a set of "source files" can be derived. More specifically we will illustrate how to use corpus data to extract relevant keywords, multiword expressions, glossary building and text patterns in order to build an AIML knowledge base that could be later used to build interactive conversational systems. The approach we propose does not require deep understanding techniques for the analysis of text.
As a case study it will be shown how to build the knowledge base of an English conversational agent for educational purpose from a child story that could answer question about characters, facts and episodes of the story. A discussion of the main linguistic and methodological issues and further improvements is offered in the final part of the chapter.

1 Introduction: Turing test and conversational agents

This work is aimed at understanding how it is possible to analyze the text of a fictional book in order to create an artificial agent that can answer questions about the same content. The critical step is to build a knowledge base in a format usable for a conversational system that should sustain information exchange at a human level, i.e. using natural language. We proposed a natural language interpreter/generator, without deep understanding of the text, or phrase structure analysis.

The chapter looks at historical evolution of such conversational systems, trying to concentrate on application in education. We then propose a specific educational application, choosing a children novel book as text source being the reference corpus.

The educational application we illustrate is focused on specific domain and purpose chatbot design, is corpus-based, but does not require deep understanding techniques to be applied to the knowledge base. We further propose some possible future improvements regarding the ability to take synonymy and semantic inference into the answer retrieval system. Section 1 illustrates the historical background of chatbot systems design with reference to Turing imitation game proposal. We present some major experiences in chatbot construction and their approach, specifically focusing on chatbots built for educational purposes. The end of the Section proposes and overview of AIML (Artificial Intelligence Markup Language), the markup language chosen for the construction of our chatbot system. Section 2 addresses the main linguistic issues posed by corpus selection and exploitation, glossary building and FAQ construction from corpora and provides examples from our chatbot case study, the Talking Cricket, answering questions about the child story "*Adventures of Pinocchio*."
Section 3 is dedicated to the design approach from a software engineering point of view: requirements, input set definition, implementation and testing. We then discuss the overall proposed methods in section 4. Section 5 provides readers with a brief illustration of selected tools and resources useful to build similar chatbots.

1.1 Turing test as a regulative idea

In his 1950 paper *Computing Machinery and Intelligence* Alan Turing described for the first time the imitation game, later best known as the Turing test. In the imitation game an interrogator engages in communication with a man and woman, situated in separate rooms. The interrogator poses questions to discover who is the woman between the two. A variant of the game substitutes the woman with a machine. The objective of the game is to observe if the interrogator, by asking questions, is able to assess which is the machine. The interrogator is free to ask questions on any topic in order to pursue his task. The machine passes the test if it is able to perform not worst than the man in the man-woman game.

With the imitation game Turing explicitly replaces the question "Can machines think?", which he considers meaningless, with another sort of questions "Are there imaginable digital computers which would do well in the imitation game?".

The debate that started from the 1950 paper has been so intense and fruitful in its philosophical, theoretical, methodological aspects - to last up to today. The Turing test posed challenges to Artificial Intelligence (AI), but also to the philosophical debate on intelligence, thinking, meaning and reasoning, focusing its approach on performance evaluation. The roots of the imitation game can be further traced in XVIIth century thought, especially in Descartes, but also in Leibniz, as pointed out by N. Chomsky (2008). What seems truly interesting in the light of current computational linguistics trends is that the Turing test poses language, natural language, in its spontaneous register of conversation, to the core of the evaluation of man-machine interaction. The Turing test can be considered a sort of regulative idea, as it guides and grounds empirical investigation in intelligent and conversational system design and testing.

A deep overview of the manifold issues Turing paper raised is given in three recent publications: The Turing test: the elusive standard of artificial intelligence (Moor 2003); The Turing test: verbal behavior as the hallmark of intelligence (Shieber 2004); Parsing the Turing test: philosophical and methodological issues in the quest for the thinking computer (Epstein 2008).

Among the many questions raised by the imitation game one of the most obvious has been to test its discriminative power by creating conversational agents able to face such a challenge. From the mid-Nineties onwards the wide accessibility to the world wide web and advances in various fields of computational linguistics and artificial intelligence lead to a larger demand for computer programs capable of interacting with users. Those applications tend to be centred in specific domains and thus restricted in tasks performed and in the extension of their knowledge bases. Still some chatbot applications aim at broader scopes and can aim at challenging humans in unrestricted Turing tests.

As an acknowledgment of the interest in these applications in 1990, the Loebner Prize proposed a competition for artificial intelligence chatbots to contest in an unrestricted Turing test (Mauldin 1994). On the same path is to be seen the recent Jeopardy human-versus-machine match won by IBM question answering (QA) computing system Watson in 2011.

1.2 Chatbot early history and general developments

Chatbots (or chatter-bots, conversational agents, dialogue systems) are applications that simulate human conversation through a textual interaction between a human user providing the input and the agent that responds to it (answering or making questions). The first attempt at building a conversational agent was ELIZA, designed in the late Sixties by Joseph Weizenbaum (1966) as a fictitious psychotherapist engaging in a conversation with users. ELIZA used a keyword matching technique in order to perform her task.

Similar to ELIZA is PARRY (Colby et al. 1971), the paranoid bot, implemented in MLISP (meta-lisp), who also dialogued with ELIZA creating the first machine-machine conversation log. Immediately after, care has been taken to provide these chatbots of a more user-friendly interface, focusing on the development of text and natural language interfaces (Shawar and Atwell 2007), as in Cliff and Atwell (1987) and Wilensky et al. (1988).

A rebirth of research on chatbot architecture has begun since the '80s (Shawar and Atwell 2007a). For example, Hutchens (1996, 1998) implemented MegaHal, a chatbot that produces its answers thanks to the mechanism of the Markov chain, which bases its answers on what the user and the conversational agent have previously said in their conversation. CONVERSE (Batacharia et al. 1999) uses various tools of computational linguistics (text parser, spell checker, etc.)

to analyze and understand user questions, while linguistic databases and dictionaries are used by a CONVERSE special module to understand user questions and generate answers. Another very successful example of chatbot is A.L.I.C.E. (`http://alice.pandorabots.com`), the Artificial Linguistic Internet Computer Entity, a general purpose "unrestricted" conversational agent using pattern-matching and AIML, or Artificial Intelligence Markup Language, a derivative of Extensible Mark-up Language (XML). A.L.I.C.E. has about 50,000 categories manually coded by a community of about 500 editors (Wallace 2009). Jabberwocky (Pirner 2007) is able to answer questions on the homonymous poem by Lewis Carroll. This chatbot has a learning mechanism based on its interaction with human users; the mechanism that uses pattern matching is similar to that of ELIZA, while its knowledge base is not contained in AIML files, but in simple text files containing a particular template for questions and answers.

In more recent years, the attempts to solve the problem concerning the construction of chatbot knowledge base and its representation have been various. Agostaro et al. (2005) proposes a LSA-bot. The knowledge base of this chatbot is created with the mechanism of Latent Semantic Analysis (LSA) by which a corpus of documents is mapped to a data-driven conceptual space. In the course of a conversation between this LSA-bot and a human user, the sentences of user input are mapped into the same conceptual space; at this point the chatbot is able to answer due to a binding mechanism based on the level of similarity between the question of the user and the LSA-bot knowledge base. Augello et al. (2009) use the LSA to create a semantic space integrated with a semi-structured knowledge base. The knowledge of their chatbot is composed of an AIML file, DBpedia and Wikipedia: through the AIML categories the conversational agent is able to find a match between the user input sentence and DBpedia resources; if it finds no correspondence, it looks for the answer in the LSA-based semantic space representing the knowledge contained in Wikipedia.

Spoken language conversation samples are the source of approaches that extract data from spoken dialogue corpora or large reference corpora (Shawar and Atwell 2003b, 2005), in (Shawar and Atwell 2003b) the authors describe how to implement a corpus-based chatbot like ELIZA or ALICE. They implement a Java program that transforms a corpus of plain texts into pattern and template in the AIML format; the corpus they use is the Dialogue Diversity Corpus, that is made up of links to different corpora of dialogues on multiple subjects (e.g. physics, algebra, astronomy, etc.) transcribed by hand. According to Wu, Wang, Li and Li (Wu et al 2008), the mechanism of acquisition of knowledge proposed by Shawar and Atwell (Shawar and Atwell 2003b) is not suitable to restricted domains of knowledge, but is only apt to be used for the acquisition of common sense knowledge. Additionally, this approach is based on a manually trascribed training corpus. For these reasons, Wu and the other authors use threads of online forums as a training corpus, further translated automatically into AIML format. These kind of conversations are thus considered suitable for automatic conversion as they hold a question-answer structure similar to the structure of the AIML template and, moreover, the dialogues in forum threads tend to be about a restricted domain of knowledge.

Now online we can find a large number of professional and non professional chatter-bots, some of which can be found at Pandorabots (`http://www.pandorabots.com`), a webservice, where it is possible to test chatbot prototypes host them. Recent experimentations try to combine traditional conversational agent design techniques with machine learning in order to avoid the largely manual approach typical of content supply (Shawar and Atwell 2003a, 2005; Chantarotwong 2006). Some recent tools try to derive information directly from the web (Ueno et al. 2010; Augello et al. 2009).

Latest developments in conversational agents, especially those aimed at entertainment and simulation games, are associated to speech synthesis and multidimensional avatars and talking heads and embodied agents, capable of emotion expressions and gesture (Vrajitoru 2006; Augello et al. 2011; Santos-Perez et al. 2011).

1.3 Chatbot applications for educational purpose

Applications of chatbots vary from tutoring, e-commerce, information retrieval, as helpdesk tools, customer support, automatic answering systems and human digital assistants. One of the most prolific, best-documented and useful application concerns the use of chatbots in education.

Early in 2003, Wallace, Tomabechi and Aimless envisaged "*chatterbots acting as talking books for children*", chatter-bots for foreign language instruction, and teaching in general (Wallace et al. 2003). Since then, the use of these conversational agents as educational tools has been further explored (Kerly et al. 2007; Kelly at al. 2009). They are seen as learning companions (Eynon et al. 2009), (Jia et al. 2003), helpers (Feng et al. 2006; Vieira et al. 2004), tutors (De Pietro and Frontera 2005; Kim et al. 2002) and as pedagogical agents (Veletsianos et al. 2010), but, as in the case of other learning environments supported by software agents, they are mostly used in relation to specific domains of learning (e.g. FREUDBOT and SOFIA).

FREUDBOT (Heller et al. 2005) is a chatbot who mimics the character of Sigmund Freud, who speaks to learners in the first person about Freudian theories and life episodes. The content of this chatbot is written in AIML and its developer has added some particular control features: for example, if FREUDBOT does not know the answer to a question, it admits not to know the topic and asks the learner to provide additional information or to change the subject of the conversation. SOFIA (Knill et al. 2004) is a calculus chatbot, a conversational agent that helps students to solve general mathematical problems. The main mathematical knowledge of SOFIA is contained in plain text files that are converted to AIML thanks to Perl scripts; this knowledge base is used to produce glossaries of mathematical definitions and algorithms and a help for students. This chatbot is able to communicate with other mathematical agents such as Pari, web resources as Wikipedia and a computer algebra system via its web interface in order to solve algebra problems.

Pirrone et al. (2008) describe an Intelligent Tutoring System (ITS), that, thanks to a particular cognitive architecture, is able to enrich its knowledge by interaction with users and by external web resources, representing structured and unstructured information in the same framework. The knowledge base of this ITS is made up of an ontology that is integrated with other information from Wikipedia, Wordnet, folksonomies and learning documents provided by the teacher and collected in a dedicated database. Through the mechanism of LSA, the knowledge base of the chatbot is represented in a semantic space, where the various documents are connected with the symbolic representation of the topic of conversation. Also the student input sentences are represented via LSA in order to assess the level of similarity between the user questions and the chatbot knowledge base. To provide its answers, this ITS needs an AIML file that describes the typical structure of the interaction in natural language. This tutorbot holds a student model, that describes the assessment level of each student, and, according to it, the chatbot searches the knowledge base and provides the appropriate material for each student; moreover, during conversation, the chatbot is able to evaluate the improvements of the user and to update its model.

Some of the applications of chatbots are focused on language learning and practice (Shawar and Atwell 2007a). Jia proposes several chatbot intended as systems for teaching English as a foreign language (Jia 2003, 2004, 2009). He describes a chatbot based on keywords like a *"partner learning of foreign language"* (Jia 2003) for Chinese university and college students who speak fluent English. This chatbot is based on AIML files and on pattern-matching techniques, like ELIZA. This type of chatbot, according to Jia, does not perform satisfactorily because it is simply based on the mechanics of keywords, completely ignoring grammar and semantics.

A few years later, Jia proposes a chatbot (Jia 2004, 2009) based on the representation of grammatical structures that underly natural language sentences. CSIEC (Computer Simulation in Educational Communication) is an English learning companion that *"generates communicative response according to the user input, the dialogue context, the user and its own personality knowledge, common sense knowledge, and inference knowledge"* (Jia 2009). Jia introduces also a NLML (Natural Language Markup Language), an XML dialect designed for annotating natural language texts. NLML annotation can be written manually thanks to a GUI pattern-template editor, or automatically generated. In the latter case, an English parser analyzes natural language texts and the resulting grammatical structure is converted into NLML. Then the NLML parser translates NLML annotation into NLOMJ (Natural Language Object Model in Java), that is into the objects that represent the grammar elements in the rules. NLDB (Natural Language Database) contains the NLOMJs, and other tables containing all the chatbot knowledge: a *"table direct-response"*, that, according to Jia, *"must be done by an author who is good at English grammar and dialogue generation"* and an inference rule table, while for the semantic knowledge the system takes advantage of WordNet. In a chatting session with students, the input text is parsed and transformed first in NLML and then into NLOMJ. CSIEC has two answer mechanisms. Thanks to a pattern-matching mechanism, CSIEC searches for the correct direct answers in NLDB in the *"table direct-response"* and in the first case the answer is produced just taking into account the student input. A second type of answer is produced using the GTE (Generation of Textual Entailment) mechanism makes inference on a text thanks to an inference rule table contained in NLDB. At the end the CR (Communication Response) mechanism generates the answer, taking into account user input and the knowledge in NLDB.

Most applications of chatbots as foreign language learning tools assume students to possess a good level of proficiency of the second language in order to interact with them, and learners can mostly have a general conversation with them. On the contrary in (De Gasperis and Florio 2012), authors describe two chatbots designed for students who have no fluency in the second language. The first conversational agent is intended for English word spelling improvement and proposes to students exercises that are typical of a foreign language text book. Students have English chat sessions with their virtual tutor, checking their conversation, proposing to learners the correct sentence in case they produce an incorrect one. The second is an expert fable chatbot that can answer to student question about a fable that learners have to read. This conversational agent is indented to check and improve learners reading comprehension of a text in a foreign language. Both chatbots are ALICE-like based, but their AIML knowledge base is automatically generated form plain text documents such as FAQ, glossary, multiwords, keywords and stopwords files as described in this chapter.

1.4 AIML language and architecture

AIML is the Artificial Intelligence Markup Language defined by Richard Wallace (Wallace 2009) aimed at describing lexical knowledge bases for conversational agents. It is derived from XML with the purpose of describing units of textual knowledge called categories:

```
<category>
   <pattern>WHO IS PINOCCHIO</pattern>
      <template>
         Pinocchio in the main character in the book:
         Adventures of Pinocchio, written by Carlo Collodi.
      </template>
</category>
```

Here is described the exact text pattern **WHO IS PINOCCHIO** to be matched in a question that has to produce as output the text between the $< template >$ tags.

Categories can also be linked together where there is a semantic common base by means of the SRAI construct:

```
<category>
   <pattern>WHO IS THE PINE-WOOD MARIONETTE</pattern>
      <template>
         <srai>WHO IS PINOCCHIO</srai>
      </template>
</category>
```

effectively realizing a link between the two categories:

Fig. 1. Two categories linked by the SRAI AIML relation

An important feature of the AIML pattern is that wildcards can be used '_' and '*', the first with higher priority over the latter. For example a more general category can be used for all "WHO IS" questions, obtained combining the wildcards, the SRAI construct and the <star/> element:

```
<pattern> WHO IS * </pattern>
<template>
   <srai>
      WHO_QUESTION <star/>
   </srai>
</template>
```

<star/> is a monolithic tag that copies the text selected by the position of the wild card.

SRAI constructs can be linked in a tree like structure in order to reduce lexical forms to a common root.

In Fig. 2 many categories are generated combining wildcards, singular and plural lexemes, all linked to the same root concept made by the lemma BIRD. This tree-like connection could also be used with synonyms and verbs forms so that user input text can be reduced to a sequence of lemmas.

AIML has a much more complex TAG set and possible parameters (Wallace 2009), such as temporal memory, variables, topics, etc.., but we found that for the purpose of this chapter, to automatically generate AIML knowledge bases starting from text corpora, it is sufficient to consider only the following elements:

1. categories
2. SRAI construct
3. wildcards

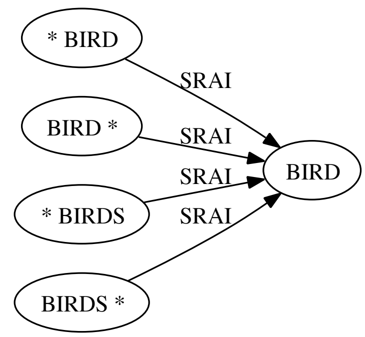

Fig. 2. Many categories linked to the same root element by SRAI relations

Once all the AIML categories have been aggregated into **.aiml** files, they can be used with an AIML interpreter so that a human can engage in a conversation with it. AIML files are also incremental, they can be added to the chatter-bot memory as needed, supplying it with the respective knowledge.

For example, the explanation of words that can be derived by a glossary can be accumulated in single **.aiml** file.

2 Corpus based chatbots and linguistic issues

In the last few years a number of tools have been developed combining the chatbot architecture with corpora. Spoken language corpora can be used to provide new topics and naturally occurring linguistic samples for the dialogue in general purpose chatbots, while domain-specific corpora can be used to provide a knowledge base that can be queried using natural language and interacting by means of the conversational agent.

General problems of objective and domain specificity still apply to knowledge bases constituted by corpora. Chatbots modeled on spoken language conversation and large reference corpora can produce interesting results from the point of view of the naturalness of the dialogue turns (Shawar and Atwell 2005), but still cannot avoid the problem of inconsistencies, illogical dialogue sequences and unreliability of information sources.

Furthermore it has become overtly clear that general purpose chatbots pose problems radically different than those of task-specific and domain-specific , i.e. restricted, chatbots. While the use of spoken conversational routines and natural spontaneous language are capital for the design of general purpose chatbots, for chatbots built to respond to narrower objectives such features are not considered so relevant. The idea of simulating a human being and eventually deceiving the user has now been completely removed from the scene, focusing more on coherence in question answering systems built for specific purposes on restricted domains.

2.1 The corpus

A corpus, intended as a collection of texts, can be used in chatbot creation for many purposes, depending on the kind of text included and on the specific objective of the conversational agent application. It can be used to train a chatbot, as in the case of spoken corpora containing spontaneous natural conversation (Shawar and Atwell 2003a), and it can be used as a knowledge base to build conversation and information structure for a dialogue system.

There are many digital libraries that contain free and out of copyright restriction texts. The best known multilingual digital libraries are: Internet Archive (`http://www.archive.org`) containing 3,209,736 books of all genres in various formats and Project Gutemberg (`http://www.gutenberg.org`) which offers about 38,000 books, mostly fictional works. Depending on the kind of corpus needed for the chatbot project different pre-processing procedures will be required, such as eliminating unneeded information, links, coding, etc.

For our case study we chose to develop a prototype of a conversational agent for educational purposes. We selected children books as our source texts and decided to opt for the well known novel **Adventures of Pinocchio** by Carlo Collodi. We used the English translation of Pinocchio by Carol Della Chiesa. The electronic text is freely available at the Project Gutemberg website (`http://www.gutenberg.org`). The chatter-bot built was named : "The Talking Cricket".

When building a conversational agent from corpora, especially if intended for educational purposes, it is necessary to determine how to use the corpus and how to integrate corpus data with other data that might be useful for the application. In our case-study we chose to use the text itself as the domain for the chatbot and also as the main corpus for the Talking Cricket chatbot question-answering system.

When dealing with literary text and building a chatbot to develop textual under-standing and analysis in learners, it is important to consider using different sources as knowledge bases for the chatbot:

1. the original text itself as a primary source
2. summaries of the text, manually or automatically generated
3. para-textual and inter-textual information (from essays, literary criticism, external information on the work fortune and historical information on author/s, etc.).

2.2 Building a glossary: linguistic issues

A glossary is useful in chatbot design in order to cope with the possibility that learners (whether natives or second language learners) might need linguistic explanations of terms and words occurring in the text that is the object of the chatbot domain. The glossary is thus intended to contain significant words or multiwords occurring in the chatbot conversation and/or in the source text that the user might not know. Thus the glossary will provide explanation on the meaning of those words. Since words are not all equally well known by people bearing different education degrees, it is appropriate to evaluate the objectives of each chatbot application in order to define which threshold to set for glossary inclusion.

Glossary candidates are the words that we presume that the chatbot user might not know and thus might need to find the meaning of. We can select those words manually, but in this case we would certainly hazard on the user knowledge. The best way to select glossary candidates is to presume little knowledge in the final user, in order to prevent the case in which the user poses a definition question and the chatbot does not have the correct answer.

The fist step in selecting glossary candidate entries is to develop a procedure to eliminate non relevant vocabulary, e.g. mainly extremely high frequency words that presumably are well known to native speakers. The necessity of eliminating non relevant vocabulary comes from the aim at economizing on glossary entries to be coded in the system.

A large portion of the vocabulary of every language, containing about 2,000 words, among the most frequent in the language, are commonly known by native speakers having primary education. The top 2,000 words of a reference corpus generally contain most function words (*and, it, on, to, the*, etc.) and very common content words (*to do, to make, man, home*, etc.). Those words, not only are commonly known, but also have a very high coverage rate on any text of a specific language (Guiraud 1960), and are generally called the "fundamental vocabulary" of a given language. It is very unlikely that a native speaker will need explanation for those terms.

To further economize on glossary candidates it is possible to restrict the word list to nouns and verbs, extracted both from the corpus list and the reference corpus list in order to obtain the most salient possible words to be included in the glossary.

Thus a good practice is to eliminate from the glossary candidate list the most common lemmas of the language. In order to do so we need:

1. a lemmatized frequency list of a reference corpus (lemma list A) for the language we are processing
2. a lemmatized frequency list of the corpus (lemma list B) we intend to use for the chatbot.

Finding a lemmatized frequency list of a reference corpus

A reference corpus is a very large corpus, balanced by genres (containing texts from written and spoken language), aimed at representing most variation registers and text types of that language. The golden standard is generally set to 100 million words.

Frequency data on usage of words in a reference corpus are generally given in a lemmatized form. This means that all running words that occur inflected in texts, such as *abandoned, loves, men, does,* etc., are grouped in lemmas that represent the general lexical class for that form, cumulating frequencies of all inflected forms pertaining to that lexeme.

The usual format for a lemmatized frequency list is the following:

```
rank, frequency, lemma, word-class
```

For the English language a very often used reference corpus is the British National Corpus (`http://info.ox.ac.uk/bnc`), which we used in our case study. It is also possible to use other similar resources such as the American National Corpus (`http://americannationalcorpus.org`), or existing reference corpora for other languages, if a lemmatized version of the vocabulary of the corpus is given.

A sample of the most common words in English (BNC), ordered by inverse frequency, will be similar to the following: 1, 6187267, the, det; 2, 4239632, be, v; 3, 3093444, of, prep; 4, 2687863, and, conj; 5, 2186369, a, det; 6, 1924315, in, prep, etc. The further step is to extract from the list the top ranked verbs and nouns. We will now have a lemmatized frequency list of the so-called fundamental vocabulary of the reference corpus (lemma list A).

Lemmatizing the corpus and extracting glossary candidates

In order to build a lemmatized frequency list of the corpus (lemma list B) we need a specific tool called a POS-tagger or a lemmatizer. The POS tagger performs the operation of analysing the corpus and tagging all inflected forms with its part-of-speech (or word-class) in the context the word occurs, and associating it to its reference lemma.

There are many ways of performing this tasks and many available tools, depending on the language needed. A freely available tool that performs POS tagging for English and many other languages is **TreeTagger** (Schmid 1994).

Once the corpus has been tagged for lemma and part of speech all we need to do is building a lemmatized frequency list that summarizes data on lemma, word-class and number of occurrences in the corpus in order to compare it to the reference corpus lemmatized list.

If the tagset used for the reference corpus and that used for the corpus are different, they need to be converted in a common coding in order to make the matching possible. In our case the reference corpus BNC has a larger tagset than that used by TreeTagger for the corpus (e.g. the BNC tags nouns NN0: Common noun, neutral for number; NN1: Singular common noun, NN2: Plural common noun, while TreeTagger uses only two tags NN: for Singular or mass nouns and NNS for plural nouns, verb tagging is more divergent) so it was necessary to re-code both lemma list in a common tagset.

From the corpus list (lemma list B), sorted by inverse frequency, we filter and re-move the reference corpus list (lemma list B) by matching both lemma and POS in both lists and obtain a list of candidate for the glossary of the chatbot

Table 1. Example of a POS-tagged text output

Word	POS	Lemma
how	WRB	how
it	PP	it
happened	VVD	happen
that	IN/that	that
Mastro	NP	Mastro
Cherry	NP	Cherry
carpenter	NN	carpenter
found	VVD	find
a	DT	a
piece	NN	piece
of	IN	of
wood	NN	wood

(lemma list C). It is very important to match both lemma and word-class since several lemmas exist in different word-classes (e.g. *love* as noun and *love* as verb).

The lemma list (C) now contains relevant words that occur in the corpus, but not words that appear banal and common to a native speaker.

Depending on the specific objective for the glossary, it is possible to remove from the list all hapax, words that occur only once in the corpus. To evaluate this option it is important to fully assess tasks and objectives of the chatbot application and to manually screen hapax. In our prototype we did not remove hapax from the glossary list since the corpus was not so large and rarely occurring nouns and verbs were considered relevant to the global understanding of the work by our future users.

Glossary entries and definitions

When building a glossary we might not need only common words (especially verbs and nouns) occurring in the corpus but we also need proper nouns and named entities (in our case study: Pinocchio, Geppetto, Antonio, Polendina, Tunny, Melampo, Harlequin, Cherry, Eugene, John, Pulcinella, Alidoro, Romeo, Rosaura, Medoro), all of which can be extracted from lemma list C (by selecting noun, verb and proper noun POS tags).

The next step is to build the glossary by associating the lemma list C to glossary definitions. Depending on the specific goals of the chatbot we will choose the appropriate source for our definitions. In our case we provided brief definition for proper nouns (character nouns) occurring in the corpus from the Wikipedia page of Adventures of Pinocchio and used Wordnet (Fellbaum 1998, 2005) definitions for all the remaining words. It is also possible to link glossary entries to Wiktionary (`http://en.wiktionary.org`) or other available sources. Along with the lemma list we need to collect multiword expressions. It is not always necessary to include multiword expressions in the glossary, but it is very important if those multiwords constitute keywords or named entities (such as Mastro Cherry, Mastro Antonio in our case study).

2.3 Corpus based FAQ

In this section we will present some of the most relevant issues in designing the FAQ for chatbot creation from text, keyword, multiword selection, and grammatical and semantic problems that might arise in the design process.

General issues

The fist problem to address in building FAQs from corpora is how to use the corpus, whether to use it as a knowledge base or as a machine learning tool. In the first case the corpus will be analyzed combining manual and automatic means in order to select portions of the text to be included in the FAQ. In the latter case the corpus itself is to be considered the source of both questions and answers for the FAQ and of the categorization of topics and categories in AIML. In our paper we have chosen to adhere to the former approach, thus FAQ building is text-centred but task definition and question selection has been conducted (semi)manually.

The texts of the corpus have been used as the main source for the answers in a bottom up procedure. After choosing portions of text to be selected as answers, a keyword analysis has been made on each piece of text in order to build questions for the FAQ and keywords to be matched with the user input.

The goal is to build a set of questions (Q) and answers (A) that are related to the main content of the source corpus. For example, from the first chapter a selection of relevant sections of the text is generated, manually or automatically. We preferred to select paragraphs manually in order not to introduce further complexity in the set of software tools needed to achieve a reasonable result. Starting from these selections of the first chapter, the most significant sentences are selected as possible answers. Then each answer has been associated to possible questions including their formal variants.

At this point it is important to evaluate the opportunity of introducing the processing of inflected forms (e.g. *loved, did, noses*, etc.) and synonyms (e.g. *story, novel, book*, etc.) to be able to cope with input that is presented in a different textual form from that of the keywords present in the source texts, but ultimately to be considered equivalent in the question answering procedure. Choosing to take into account inflected forms and synonyms largely depends on the language of the chatbot and on the specific aims for the application.

Another relevant issue is that of incomplete, ill-formed or ungrammatical input. When designing the FAQ it is important to consider the fact that for multiple reasons it is common for the user to pose question that seem incomplete or far from a well-formed standardized questions. Thus it is important in addressing the problem of input analysis to use an approach that allows the user freedom in the wording of questions and that allows deviating and unusual grammatical rules. Pattern- or keyword matching techniques and careful selection of stopwords generally make it possible to solve this problem, as in our case study.

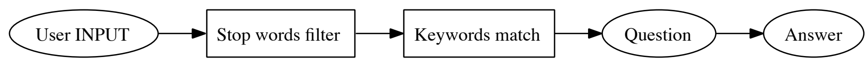

Fig. 3. Ideal workflow from the user input to the right answer.

Stopwords can have a great impact on the efficiency of the final AIML categories matching. Using a combination of wildcards and SRAI the stop words can be carefully filtered out from the user input, so that ideally only a list of significant lemmas from the user input is matched with a list of significant lemmas of a question, so to find the right answer, as shown in Fig. 3.

A further problem is the scope and level of generality of the question and answer relationship. Using corpora for FAQ building tends to focus question/answer scope to local ranges, because small bits of texts, often single sentences, are chosen as relevant answers in FAQs, following a general characteristic of spontaneous conversation.

```
Human: Why was Mastro Cherry called that way?
The Talking Cricket: His real name was Mastro Antonio, but
 everyone called him Mastro Cherry, for the tip of his nose
 was so round and red and shiny that it looked like a ripe cherry.
```

But in the case of applications that are aimed at general knowledge querying and for text analysis in educational contexts it is also relevant to introduce a broader and global level of question answering that might include summarized information.

A further relevant aspect is the selection of keywords to be used both for input analysis and pattern-matching. The selection of keywords can be done manually or automatically depending of objectives of the application and extension of the corpus chosen. Associated to this step is the relevance of multiword expressions identification and processing of inflected forms and semantic aspects.

Pattern-matching techniques generally do not adequately respond to the complexity of texts produced in languages like those of the Romance or Semitic families that possess a rich morphology and thus present the same lexeme inflected in multiple textual forms. Keyword selection and input matching strongly rely on the capability of processing a large number of forms (e.g. any Italian verb has about 50 different textual forms that can be reduced to the same lemma).

Precision in associating the user input with the correct answer relies on the capability of using this grammatical information in the matching process. Since our case study is in the English language, that is not particularly rich in morphology compared to other languages, we decided to stick to traditional pattern-matching techniques.

The same problem arises with the use of synonyms (*e.g. marionette, puppet*) or co-referent expressions (e.g. Mastro Cherry, Mastro Antonio, the carpenter), but also with hyperonyms and hyponyms (*e.g. the novel, the book; the cat, the animal*).

In future developments of the Talking Cricket prototype we intend to use Wordnet information on inflections and sets of synonyms to allow a better management of keywords and input matching.

The Talking Cricket FAQ design.
The Talking Cricket (`http://glottoweb.org/talkingcricket`) uses two kinds of sources as knowledge bases: the English translation of Adventures of Pinocchio and the Wikipedia page on the work to provide minimal para-textual information for the chat-bot.

The Talking Cricket mainly uses the exact wording of the original text to answer question about episode and story development, while using external sources to inform the user on the author, characters and textual analysis. The general approach has privileged local scope questions and answers since the text has been chosen to be the main base to be questioned. So questions are designed in order to be answered by a small self-contained portion of the source text, without any manual rewriting.

```
Q Why did Mastro Cherry sing?
A The poor fellow was scared half to death, so he tried to
  sing a gay song in order to gain courage.
```

Summaries of chapters, provided in the source text itself, are used for broader scope question/answers, regarding story development of each chapter.

```
Q What happens in the first chapter?
A How it happened that Mastro Cherry, carpenter, found
  a piece of wood that wept and laughed
  like a child.
```

Broader scope question/answer typologies are extracted from the Wikipedia page on the novel and give information about the book that are not contained in the source text itself, but that can be useful to young users of the educational chatbot we designed.

```
Q Who is the author of the book?
A The Adventures of Pinocchio is a novel
  for children by Italian author Carlo Collodi
```

```
Q Who is Carlo Collodi?
A Carlo Lorenzini (November 24, 1826 $-$ October
  26, 1890), better known by the pen name Carlo Collodi, was an
  Italian children's writer known for the world-renowned fairy
  tale novel, The Adventures of Pinocchio.
```

At the moment the prototype of the Talking Cricket is able to chat about the first chapter of the book and about external para-textual information. Further versions of the Talking Cricket prototype will include automatically generated summaries of episodes in the text in order to answer to general aspects of the story development.

3 Steps for chatter-bot generation

The following sequence diagram shows the overall procedure needed to build the restricted AIML FAQ based chatter-bot the Talking Cricket shown in this chapter, but it can be used in general on any other source text corpus:

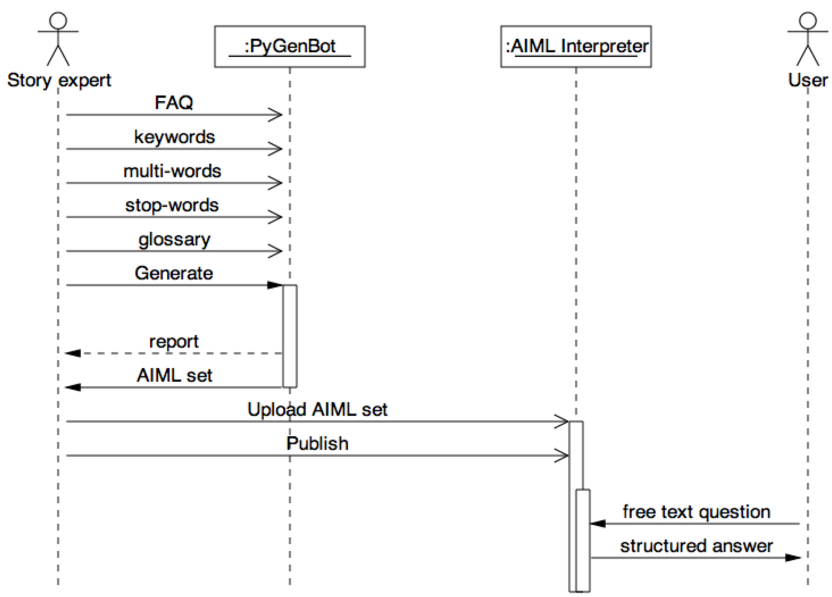

Fig. 4. Sequence diagram to generate the FAQ chatter-bot lexical knowledge base

In a language learning context, the story expert is the educator and the user is the student. The story expert/educator has to produce text files as knowledge seeds in order to generate the lexical knowledge base about the story. No programming skills are required, files can be edited writing in free text form, just taking care of a straightforward format to separate questions from answers (FAQ), or item from definitions (glossary). Only after the lexical knowledge base is generated in the form of AIML files, it can be uploaded to an online AIML interpreter, if the learning application is web based, or locally saved to a folder where standalone interpreter can read them. Then the user/student can interact in real time with the interpreter which matches user text input with AIML categories to return an answer. The user/student can use free text, as far as there are no typing errors and she/he keeps using lexemes being part of the lexical set used in the FAQ questions and the glossary items. The AIML agent will reply a structured text, not free, meaning that the output text it is just copied from the answers of the FAQ set, or the glossary definitions where applicable.

3.1 Requirements for the chatter-bot

The overall design of the Talking cricket chatter-bot is based on requirements that are needed to keep the application up-to-date with current conversational agents technology:

1. restricted knowledge domain
2. human computer interaction requirements:
   (a) textual interface
   (b) free text from the user
   (c) structured answer text from the chatter-bot
3. no temporal memory
4. resolution of lexical ambiguity by finite number of choices
5. no answer transparency

Requirement n.1 implies that chatter-bot will not answer general knowledge questions, but only on the specific subject defined by the FAQ and the glossary. This also means that this kind of chatter-bot cannot be considered at the unrestricted Turing test level.

Requirement n. 2 is related to the simplified implementation, but it could feasibly evolve in future version in a semantic network representation that could allow some form of text re-generation of the FAQ answers.

Requirement n.3 implies that chatter-bot output is independent from any previous interaction.

Requirement n.4 implies that in case of multiple possible answers derived from the user input processing, the chatter-bot should explicitly ask for a more detailed question to overcome the ambiguity.

Requirement n.5 implies that in case no answer is available for a given user input, the chatter-bot should produce a standard output to let the user be aware that the given knowledge base is insufficient to give a proper answer.

3.2 Input set definition

The story expert needs to analyze the given text corpus and derive from it units of lexical knowledge that can seed the generation of the chatter-bot knowledge base.

This is a-priori work made by the chatter-bot designer. This can be supported by several computational linguistic tools now available, but still is not completely automated.

Data is organized as follows:

1. a FAQ file F, frequently asked questions, is a free text file composed of several units of FAQ-knowledge:

   ```
   Q <the question phrase> | {Q <alternative version>}
   A <the answer phrase> | {A <alternative version>}
   ```

   for as many units as needed to cover the restricted knowledge domain.
2. a glossary file G, where important keywords and or multi-word expressions will be listed with their free text definition:

   ```
   G <the glossary item> | {G <alternative version>}
   D <the glossary item definition>
   ```

   the glossary item can be a single word or a multi word.
3. a keywords file K, just listed one on each text line
4. a multiwords file M, just listed as many as needed on each text line
5. a stopwords file S, listed all of the non meaningful words, like articles, prepositions, adjectives, adverbs and other forms, mostly taken from the question text in the FAQ set.

Text files <F, G, K, M, S> can be directly typed by the story expert using a simple text editor; this set makes the input of the PyGenBot software package, as seen in the previous sequence diagram.

3.3 Chatter-bot lexical knowledge base construction

The generator algorithm has been developed (De Gasperis, 2010) in Python programming language resulting into about 500 lines of code. The program has been named **PyGenBot**.

The main steps of the generation algorithm can be summarized as the following:

```
Algorithm 1: Main AIML Generation Algorithm

F0. merge user defined multiwords and character names
    from the glossary
F1. extract all the relevant category lists from FAQ questions
F2. calculate possible branches from each category
F3. extract the answers
F4. generate AIML output set linking categories to answers
```

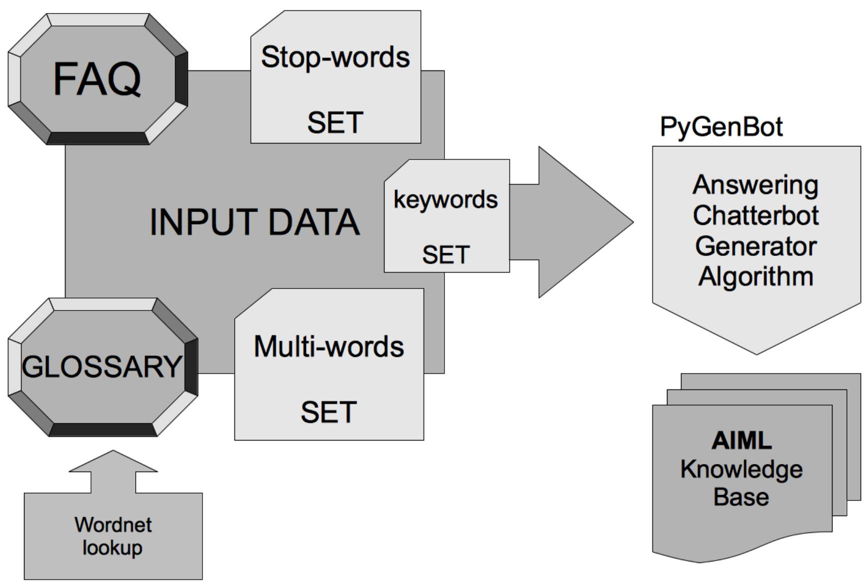

Fig. 5. Workflow from input data set to AIML output.

Detailed steps of F1

A single category, such as it is defined in the AIML, is a couple of pattern-template. The pattern need to coincide with one or more words taken from the question text so that they can be matched in the user question and linked to the proper answer of the FAQ file, as listed in Algorithm 2.

```
Algorithm 2: Generation of AIML categories

define Dw as the stopwords set
define Pw as the multiwords entries set
define Kw as the keywords entries set
FOR all questions q in FAQ file DO
    build list L of meaningful words/multiwords wi from q
    (i.e. filter out all wi in Dw and use wi in Pw
     taken as a sequence or in Kw)
    initialize an emtpy category list C
    FOR all words/multiwords wi in L DO
        append wi in C combined with all the others taken 2 by 2
    END FOR
    build a category list M with all the meaningful words found in q
    append C and M to category list set Sc
END FOR
```

Detailed steps of F2.

This method, shown in algorithm F2, is needed to calculate all the possible outgoing branches from a category that can lead to different answers. This will be used later as information to generate the AIML code, as shown in algorithm 3.

```
Algorithm 3: Extraction of categories branches

Let OUT be the output dictionary map indexing
  a category to a list of integers
FOR all categories list Cl in Sc DO
    let Ai be the answer which question Qi has generated Cl
    FOR all categories ci in Cl DO
        append the integer i to the OUT[ci] list
    END FOR
END FOR
return OUT
```

In the implementation, the powerful dictionary data structure as defined in the Python language, here $OUT[< category >]$ is crucial during the calculation of the categories branches.

Detailed steps of F4.
This method finally generates the FAQ AIML file, trying to catch all of the meaningful word from the user sentence and matching them with the meaningful words sequences from the FAQ questions. It uses SRAI recursions as defined by the AIML 1.0.1 standard [3].

Generation of GLOSSARY-AIML.
The generation of Glossary AIML takes into account the list of most significant lexemes selected by the linguistic analysis; for each glossary item, its definition is manually selected from Wordnet.

```
Algorithm 4: Generation of final AIML

FOR all questions Qi DO
    given the category list Cl generated from Qi
    let Ta be the AIML SRAI template containing the answer text
    FOR all categories ci in Cl DO
        IF ci is a combination of two words THEN
            generate all possible edges (SRAI) to Ta
        ELSE IF ci has just one branch THEN
            generate an edge (SRAI) to Ta
            IF ci is just a single word
               and is a glossary item THEN
               generate an edge (SRAI) to the glossary definition
            END IF
        ELSE IF ci has multiple branches to several answers THEN
               generate an edge (SRAI) to a phrase
               asking the user to be more specific.
        END IF
    END FOR
END FOR
```

AIML ouput.
The generation of the chatter-bot AIML includes code generation from the FAQ file and from the glossary file. Also a reduction AIML is needed so that it could filter out the stopwords set, during user interaction, so that each sentence the user enters during the conversation will be reduced to a sequence of meaningful words that can be matched with a FAQ AIML/GLOSSARY AIML knowledge-base unit, in order to best match the right answer to the asked question.

3.4 Chatter-bot testing

In (Shawar and Atwell 2007b), the authors illustrate different chatbot evaluation methods, and describe three different metrics they used to evaluate three different chatbot prototypes (see (Shawar and Atwell 2007b) for more details) and at the end they conclude that *"evaluation should be adapted to the application and to user needs"*. As regards the Talking Cricket chatbot, a dialogue efficiency metrics has been used for evaluating it, measuring the correctness of the responses given by the chatbot. For this purpose different quality levels has been defined:

1. Level 0: the Talking Cricket answers correctly to all questions in the FAQ file
2. Level 1: the Talking Cricket gives the correct answer for at least 50% of questions that have semantic variations with respect to the questions in the FAQ file
3. Level 2: the Talking Cricket gives the correct answer for all the question that presents semantic variants of all questions with respect to the FAQ questions.

The testing phase demonstrate the Talking Cricket has reached Level 1. Firstly the Talking Cricket has been tested by submitting to it all the FAQ question and it answered correctly in all cases, as the following example of conversation shows:

```
Human: Who found the piece of wood?
talking_cricket: Mastro Cherry, carpenter, found a piece of wood
 that wept and laughed like a child.

Human: What does the carpenter's nose look like?
talking_cricket: His real name was Mastro Antonio, but everyone
 called him Mastro Cherry, for the tip of his nose was so round
 and red and shiny that it looked like a ripe cherry.

Human: Why is Mastro Cherry filled with joy?
talking_cricket: As soon as he saw that piece of wood, Mastro
Cherry was filled with joy.

Human: What is the book ending?
talking_cricket: After long months of working for the farmer and
 supporting the ailing Geppetto (Mastro Cherry) he goes to town
 with what money he has saved (40 pennies to be exact) to buy
 himself a new suit. He meets the snail, who tells him that the
 Blue-haired Fairy is ill and needs money. Pinocchio instantly
 gives the snail all the money he has, promising that he will help
 his mother as much as he is helping his father. That night, he
 dreams he is visited by the Fairy, who kisses him. When he wakes
 up, he is a real boy at last. Furthermore, Pinocchio finds that
 the Fairy left him a new suit and boots, and a bag which
 Pinocchio thinks is the forty pennies he originally loaned to
```

```
 the Blue Fairy. The boy is shocked to find instead forty freshly
 minted gold coins. He is also reunited with Geppetto, now healthy
 and resuming woodcarving. They live happily ever after.

Human: Who is Pinocchio?
talking_cricket: Pinocchio is a naughty, pine-wood marionette who
 gains wisdom through a series of misadventures which lead him to
 becoming a real human as re-ward for his good deeds.

Human: What is firewood?
talking_cricket: a fire fed by pieces of wood.

Human: What does it mean mistake?
talking_cricket: an understanding of something that is not correct;
 "he wasn't going to admit his mistake";
 "make no mistake about his intentions";
 "there must be some misunderstanding--I don't have a sister".
```

Then, questions that have semantic variations with respect to the questions in the FAQ file are made to the Talking Cricket and it has answered correctly to the 50% of the case, as the following chat session example demonstrates:

```
Human: What kind of piece of wood mastro cherry found?
talking_cricket: Mastro Cherry, carpenter, found a piece of wood
that wept and laughed like a child.

Human: To what fruit does the nose of Mastro Antonio look like?
talking_cricket: His real name was Mastro Antonio, but everyone
 called him Mastro Cherry, for the tip of his nose was so round
 and red and shiny that it looked like a ripe cherry.

Human: What does the carpenter wish to make with the piece of
 wood?
talking_cricket: He mumbled half to himself: "I shall use it to
 make the leg of a table.".

Human: For what reason is Mastro Cherry filled with joy?
talking_cricket: As soon as he saw that piece of wood, Mastro
 Cherry was filled with joy.
```

The following example shows some cases where the chatbot did not provide correct replies and admits to not knowing the answer:

```
Human: What is the reason for the joy of Mastro Antonio?
talking_cricket: No glossary information about " the reason for
 the joy of Mastro Antonio", maybe... the reason for the joy of
 Mastro Antonio.
```

```
Human: What is the meaning of firewood?
talking_cricket: No glossary information about " the meaning of
 firewood", maybe... the meaning of firewood.

Human: what is the real name of mastro cherry?
talking_cricket: No glossary information about " the real name
 of mastro cherry", maybe... the real name of mastro cherry.
```

In few cases the talking Cricket has given an incorrect answer:

```
Human: What happened when Mastro Cherry sees the piece of wood?
talking_cricket: As he was about to give it the first blow, he stood
 still with arm up-lifted, for he had heard a wee, little voice say in a
 beseeching tone: "Please be careful! Do not hit me so hard!".
```

If users ask question very different from the FAQ questions, as the Level 2 requires, in most cases the chatbot do not provide the correct answers:

```
Human: Why did Mastro Cherry want to hit the piece of wood
 with an axe?
talking_cricket: He mumbled half to himself: "I shall use it
 to make the leg of a table.".

Human: What instilled courage in mastro antonio?
talking_cricket: Sorry, I
 do not understand written in this form.
 Can you ask it in other words, please?
```

4 Discussion

This chapter gives an overview of past and present research on chatbot design and illustrates procedures in order to build a corpus-based conversational agent that takes into account some basic linguistic issues. More specifically we have discussed glossary candidates selection, FAQ construction and presented tools to automatically compile AIML from pre-processed text documents. We also introduced a simple quality level grid of FAQ chatter-bots that can help to evaluate qualitatively their performance, counting the number of correct answers, i.e. the error in respect to a desired behaviour. The lower the quality level, more the behaviour of the chatter-bot is dependent on lexical adherence to the text sources of the FAQ. The quality raises if the chatter-bot exhibits behaviour that suggests its capability of taking into account lexical variation within the same semantic set, that should not depend on the exact match with the source text (by using keyword match o searching over the multiple textual forms related to the same concept). As more machine learning algorithm will be introduced the quality of such FAQ chatter-bots can definitively be enhanced so they can be used proficiently in real world applications.

One of the major weaknesses in existing chatbot systems is the lack of memory. Chatbots generally engage in conversations that cannot be appreciated throughout turns. It has been often pointed out that this lack in the ability of keeping track of the conversational turns and their development is an element that reveals the unnaturalness of the communicative interchange, because in introduces inconsistency and incoherence in the turn sequences. This is especially witnessed in general-purpose chatbots, such as Alice. Users tend to engage in short exchanges and tend not to come back to the tool after the first try in 88% of the cases (Jia 2004). Some attempts at taking into account at least a form of short-term memory have been made (Vrajitoru 2003), but this issue is far from being properly addressed. There are cases in which the input provided by the user can be associated to multiple answers. In this case, if short term memory is introduced to keep track of the turn sequences, it is possible to conceive a further intervention of the chatbot to ask questions to the user in order to give the appropriate answer.

As we have pointed out (2.3.1) further improvements can address the problem of dealing with languages with rich morphology and with introducing systematically (and automatically) the semantic properties of keywords, such as synonyms, hyperonyms and co-reference. In a future development of the Talking Cricket prototype we mean to integrate Wordnet information on these aspects to improve the automatic generation of keywords and of input pattern-matching.

A further issue regards the automatic alignment of Wordnet definition with glossary entries. At the moment the correct word sense for each of the glossary entries has been selected manually, but when the corpus source of the chatbot is a larger text, manual alignment may not be a feasible option. The easiest solution could be to integrate the whole Wordnet lemma entry letting the user disambiguate the specific word sense present in the text; the hardest, and most challenging option, is to evaluate the possibility of automatizing at least partly the word sense disambiguation task by relying on ontologies, especially when the domain of the application is restricted.

5 Tools and resources

Here we refer to useful tools and examples available online, mostly free or open source, that can be used to approach the text corpora study in order to produce good linguistic based chatter-bots and to understand their evolution.

Chatter-bot hosting.

Making chatbots and let them "live" online can be done in several different ways. First of all the chatter-bot need to be hosted in some kind of server. The most common way is uploading AIML files on an online chatbot hosting server where users can chat online with their own conversational agent. Some commercial and free chatbot hosting service are:

- Pandorabots (`http://www.pandorabots.com`)
  Pandorabots is a chatbot hosting web service; allows anyone to develop their own chatbots manually writing the AIML files, were the chatbots can also be published. Users can upload their AIML knowledge files on Pandorabots server. It is free as long as the interaction keeps traffic lower than a given threshold. Chatbots hosted on Pandorabots can be integrated on web pages, in Second Life, in online games and on instant messaging service, respond to email or in forum threads, appear in social networks as Twitter and Facebook and run on mobile smart-phones applications.
- AI-Buddy (`http://www.ai-buddy.com`)
  AI-buddy is a chatbot hosting service that enables users to create chatbots for AIM, AOL, the web, email and mobile phones. It provides a set of tool to create chat-bots, as for example a bot editor. It also offers a free plan for low traffic chabots.

AIML interpreters.
Otherwise a chatter-bot master can use some other open source AIML interpreter, install on its own virtual private server and let it answer question to online users:

- Program D (`http://aitools.org/Program_D`)
  Program D is an AIML bot engine implemented in Java, easy to configure and runs in a GUI application.
- Program W (`http://programw.sourceforge.net`)
  Program W is an AIML interpreter implemented in Java that extends ProgramD, adding new AIML tags that allow user to create chatbots able to question the WordNet lexical dictionary.
- PyAIML (`http://pyaiml.sourceforge.net`)
  PyAIML, also known as Program Y, is an AIML interpreter implemented with Python, developed as an extension to the AIML chatbot Howie.
- Program E (`http://sourceforge.net/projects/programe`)
  Program E is an AIML application written in PHP for running chatbots. It uses a MySQL database where AIML files have to be uploaded. It provides an AIML rule engine and HTML, Flash and XML-R chat interfaces.

Selected online tools.
Here follows a list of some of the tools mentioned in this chapter.

The Talking Cricket (`http://glottoweb.org/talkingcricket`)

"*The Adventures of Pinocchio*" story expert (limited to the first chapter and some meta knowledge) developed for this chapter.

British National Corpus (http://info.ox.ac.uk/bnc)

The British National Corpus (BNC) is a 100 million word corpus, containing samples of written (90) and spoken (10) language from a wide range of sources, designed to represent a wide cross-section of British English from the later part of the 20th century. The corpus is encoded according to the Guidelines of the Text Encoding Initiative (TEI). Data and documentation (lemma lists, forms list, corpus composition, etc.) is freely available at the following address: ftp://ftp.itri.bton.ac.uk/bnc.

A detailed version of the frequency data can be found in Leech et al. 2001. The lemmatized frequency list for the 6,318 words with more than 800 occurrences in the BNC is called lemma.al [123kb], is a space-separated text and can be found at:
ftp://ftp.itri.bton.ac.uk/bnc/lemma.al.

Project Gutemberg (http://www.gutenberg.org)

Project Gutemberg is a library of free electronic versions of printed books, founded in 1971 by Michael Hart. The texts available are free because they are of public domain, they are never been covered by copyright, or copyright restriction have lapsed. The library offers also some copyrighted texts given permission from authors to this form of publication. In 2011 Project Gutemberg collection contains 33,000 eBooks, most of which are in English.

TreeTagger (http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger)

TreeTagger was developed by Helmut Schmid at the Institute for Computational Linguistics of the University of Stuttgart. The tool is language independent and performs POS-tagging on a large number of languages such as German, English, French, Italian, Dutch, Spanish, Bulgarian, Russian, Greek, Portuguese, Chinese, Swahili, Latin, Estonian. It can be used for other languages if provided with a manually tagged training corpus. It is available for Sparc, Linux and Windows PC and Mac.

WordNet (http://wordnet.princeton.edu)

Wordnet is a free and open source lexical database of the English language. Wordnet contains semantic and cognitive information on nouns, verbs,adjectives and adverbs grouped in sets of synonyms (synsets). The synsets are further interlinked by means of conceptual-semantic and lexical relations. Wordnet can be freely browsed but can also be downloaded and used for multiple application objectives. For chatbot improvements it can be used in word definition association for glossary building and for synonym sets to be used in generating questions for the FAQ.

References

Agostaro, F., Augello, A., Pilato, G., Vassallo, G., Gaglio, S.: A Conversational Agent Based on a Conceptual Interpretation of a Data Driven Semantic Space. Advances in Artificial Intelligence (AI*IA 2005), 3673. (2005) 381–392

Augello, A., Vassallo, G., Gaglio, S., Pilato, G.: A Semantic Layer on Semi-Structured Data Sources for Intuitive Chatbots. Complex, Intelligent and Software Intensive Systems, International Conference on. (2009) 760–765

Augello, A., Gambino, O., Cannella, V., Pirrone, R., Gaglio, S., Pilato, G.: An Emotional Talking Head for a Humoristic Chatbot. Applications of Digital Signal Processing. InTech. (2011)

Batacharia, B., Levy, D., Catizone, R., Krotov, A., Wilks, Y.: CONVERSE: a conversational companion, Kluwer Iternational Series in Engineering and Computer Science. Kluwer Academic Publishers Group. (1999) 205–216

Chantarotwong, B.: The learning chatbot. Ph.D. Thesis. UC Berkeley School of Information. (2006)

Chomsky, N.: Turing on the "Imitation game". In: Epstein, Robert, Gary Roberts and Grace Beber. Parsing the Turing test: philosophical and methodological issues in the quest for the thinking computer. Springer, New York. (2008) 103–106

Colby, K. M., Weber, S., Hilf, F.D.: Artificial Paranoia. Artificial Intelligence, 2. (1971) 1, 1–15.

Cliff, D., Atwell, E.: Leeds Unix Knowledge Expert: a domain-dependent Expert System generated with domain-independent tools. BCS-SGES: British Computer Society Specialist Group on Expert Systems journal, 19. (1987) 49–51

De Gasperis, G.: Building an AIML Chatter Bot Knowledge-Base Starting from a FAQ and a Glossary. JE-LKS. Journal of e-Learning and Knowledge Society, 2. (2010) 79–88

De Gasperis, G., Florio, N.: Learning to read/type a second language in a chatbot enhanced environment. Proceedings of ebTEL - 2012: International Workshop on evidenced-based Technology Enhanced Learning, University of Salamanca, 28-30 March 2012. Accepted for publication. (2012)

De Pietro, O. and Frontera, G.: TutorBot: An Application AIML-based for Web-Learning. Advanced Technology for Learning, ACTA Press, Advanced Technology for Learning, 2, 1. (2005)

Epstein, R., Roberts, G., Beber, G.: Parsing the turing test : philosophical and methodological issues in the quest for the thinking computer. Springer, New York. (2008)

Eynon, R., Davie, C., Wilks, Y.: The Learning Companion: an Embodied Conversational Agent for Learning, WebSci09: Society On-Line. Conference on. (2009)

Fellbaum, C.: WordNet : an electronic lexical database. MIT Press, Cambridge, Mass. (1998)

Fellbaum, C.: WordNet and wordnets. In K. Brown, editors, Encyclopedia of Language and Linguistics. Elsevier, Oxford. (2005) 665–670

Feng, D., Shaw, E., Kim, J., Hovy, E.: An intelligent Discussion-bot for answering student queries in threaded discussions. In Proceeding of the International Conference on Intelligent User Interfaces IUI. (2006) 171–177

Guiraud, P.: Problmes et mthodes de la statistiquelinguistique. Presses universitaires de France, Paris. (1960)

Heller, B., Procter, M., Mah, D., Jewell, L., Cheung, B.: Freudbot: An investigation of chatbot technology in distance education, In Proceedings of the World Conference on Multimedia, Hypermedia and Telecommunication. (2005)

Hutchens, J.L.: How to pass the Turing test by cheating. School of Electrical, Electronic and Computer Engineering research report TR97-05. Perth: University of Western Australia. (1996)

Hutchens, J.L., Alder, M.D.: Introducing MegaHAL. Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning. (1998) 271–274

Jia, J.: The study of the application of a keywords-based chatbot system on the teaching of foreign languages, Arxiv preprint cs/0310018. (2003)

Jia, J.: The study of the application of a web-based chatbot system on the teaching of foreign languages. Proceedings of Society for Information Technology and Teacher Education International Conference 2004. Edited by Ferdig, R.E., Crawford, C., Carlsen, R., Davis, N., Price, J., Weber, R., Willis, D.A. (2004) 1201–1207

Jia, J.: CSIEC: A computer assisted English learning chatbot based on textual knowledge and reasoning, Knowledge-Based Systems, 22, 4. (2009) 249–255

Kerly, A., Hall, P., Bull, S.: Bringing chatbots into education: Towards natural language negotiation of open learner models. Know.-Based Syst., Elsevier: 20, 2. (2007) 177–185

Kerry, A., Ellis, R., Bull, S.: Conversational Agents in E-Learning, Ap-plications and Innovations in Intelligent Systems XVI. (2009) 169–182

Kim, Y.G., Lee, C.H., Han, S.G.: Educational Application of Dialogue System to Support e-Learning, Association for the Advancement of Computing in Education (AACE). (2002)

Knill, O., Carlsson, J., Chi, A., Lezama, M.: An artificial intelligence experiment in college math education, Preprint available at http://www. math. harvard. edu/˜ knill/preprints/sofia. Pdf. (2004)

Leech, G., Rayson, P., Wilson, A.: Word frequencies in written and spoken English: based on the British National Corpus. Longman, London. (2001)

Mauldin, M.L.: Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. AAAI '94 Proceedings of the twelfth national conference on Artificial intelligence, 1. (1994) 16–21

Moor, J.: The Turing test: the elusive standard of artificial intelligence. Kluwer Academic Publishers, Dordrecht, Boston, 6. (2003) 273

Pirner, J.: The beast can talk. Pdf. Published online: http://www.abenteuermedien.de/jabberwock/how-jabberwock-works.pdf. (2012) Accessed February 2012

Pirrone, R., Cannella, V., Russo, G.: Awareness mechanisms for an intelligent tutoring system, in Proc. of 23th Association for the Advancement of Artificial Intelligence (2008)

santosetal2011] Santos-Prez, M., Gonzlez-Parada, E., Cano-Garca, J.: AVATAR: An Open Source Architecture for Embodied Conversational Agents in Smart Environments In J. Bravo, R. Hervs and V. Villarreal, eds, Ambient Assisted Living, Springer Berlin / Heidelberg. (2011) 109–115

Schmid, H.: Probabilistic Part-of-Speech Tagging Using DecisionTrees. Paperpresented to the Proceedings of International Conference on New Methods in Language Processing. (1994)

Shawar, B.A., Atwell, E.: Using dialogue corpora to train a chatbot. In Archer, D., Rayson, P., Wilson, A., McEnery, T., editors, Proceedings of the Corpus Linguistics 2003 conference. Lancaster University. (2003) 681–690

Shawar, B.A., Atwell, E.: Machine Learning from dialogue corpora to generate chatbots, In Expert Update journal: 6, 3. (2003) 25–29

Shawar, B.A., Atwell, E.: A chatbot system as a tool to animate a corpus. ICAME J., 29. (2005) 5–24

Shawar, B.A., Atwell, E.: Chatbots: are they really useful? LDV Forum, 22. (2007) 29–49

Shawar, B.A., Atwell, E.: Different measurements metrics to evaluate a chatbot system, In Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies. (2007) 89–96

Shieber, S.M.: The Turing test : verbal behavior as the hallmark of intelligence. MIT Press, Cambridge, Mass. (2004)

Turing, A.M.: Computing machinery and intelligence. Mind, 59. (1950) 433–460

Ueno, M., Mori, N., Matsumoto, K.: Novel Chatterbot System Utilizing Web Information. Distributed Computing and Artificial Intelligence. (2010) 605–612

Veletsianos, G., Heller, R., Overmyer, S., Procter, M.: Conversational agents in virtual worlds: Bridging disciplines. Wiley Online Library, British Journal of Educational Technology, 41, 1. (2010) 123–140

Vieira, A. C., Teixeria, L., Timteo, A., Tedesco, P., Barros, F.: Analyzing online collaborative dialogues: The OXEnTCH-Chat , IEEE, In Proceedings of the Intelligent Tutoring Systems 7th International Conference. (2004) 72–101

Vrajitoru, D.: Evolutionary sentence building for chatterbots. GECCO 2003 Late Breaking Papers. (2003) 315–321

Vrajitoru, D.: NPCs and Chatterbots with Personality and Emotional Response. Computational Intelligence and Games 2006 IEEE Symposium. (2006) 142–147

Wallace, R.S., Tomabechi, H., Aimless, D.: Chatterbots Go Native: Considerations for an eco-system fostering the development of artificial life forms in a human world. Published online: http://www.pandorabots.com/pandora/pics/chatterbotsgonative.doc. (2003) Accessed February 2012.

Wallace, R.S.: The Anatomy of A.L.I.C.E. In R. Epstein, G. Roberts and G. Beber, editors, Parsing the Turing Test. Springer Netherlands. (2009) 181–210

Weizenbaum, J.: ELIZA A computer program for the study of natural language communication between man and machine. Communications of the ACM, 10, 8. (1966) 36-45

Wilensky, R., Chin, D.N., Luria, M., Martin, J., Mayfield, J., Wu, D.: The Berkeley UNIX consultant project. MIT Press. Computational Linguistics, 14, 4. (1988) 35–84

Wu, Y., Wang, G., Li, W., Li, Z.: Automatic Chatbot Knowledge Acquisition from Online Forum via Rough Set and Ensemble Learning, IEEE, Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on. (2008) 242–246.