
# Building an AIML Chatter Bot Knowledge-Base Starting from a FAQ and a Glossary

Giovanni De Gasperis

Dipartimento Ingegneria Elettrica e dell'Informazione, Università degli Studi dell'Aquila

giovanni.degasperis@univaq.it

Keywords: chatter bot human computer interaction

## Abstract

Chatter bots are software programs that emulate human conversation and can show human-like conversational behavior in limited knowledge domain if properly crafted. AIML, Artificial Intelligence Markup Language, is a well known XML derived language to build chatter bot knowledge bases, in a context of case-based reasoning and textual pattern matching algorithms.
A design methodology will be explained based on a novel algorithm to automatically generate AIML knowledge bases starting from a frequently asked question free text file and a glossary of terms.
A generated demonstrator chatter bot using the Italian language will be shown.

## 1 Introduction

In the '60s-'70s the first conversational agents were implemented, ELIZA (1966) and PARRY (1972) (Güzeldere & Franchi, 1995). They were based on the recognition of keywords or phrases given in input and on a set of corresponding pre-arranged and pre-programmed answers in output, so that the conversation could be considered intelligent.

Modern multimodal user interfaces (Pirrone *et al.*, 2008) comprise a pseudo-natural language interpreter which emulates human conversation in order to make the user feel comfortable during information retrieval. The conversation is carried out by a chatter bot (Wikipedia, 2009) based on AIML knowledge-base and AIML interpreter (AIML, 2005). The knowledge-base (KB) of a chatter bot conversational agent is made of pairs of patterns and templates, which can be linked together semantically and/or recursively by means of srai connections.

A.L.I.C.E., in its various versions, is the most famous generalist English-speaking chatter bot. Nowadays A.L.I.C.E. uses pattern-matching techniques which are similar to those used by ELIZA since 1966.

Among the different Alicebots, recently Wallace announced the SpellBinder (Wallace, 2009) web service by which a chatter-bot knowledge base can be generated using transcripts of movie characters, assimilating their personalities and ways of talking. A funny and interesting fake James T Kirk is available to interact with, born from all the transcripts of the original Star Trek TV serial.

The model can be related to case-based reasoning semantic networks (Smid, 2002) and resolved using textual pattern-matching algorithms (Wallace, 2007). The KB could be represented graphically using a graph as shown in figure 1.

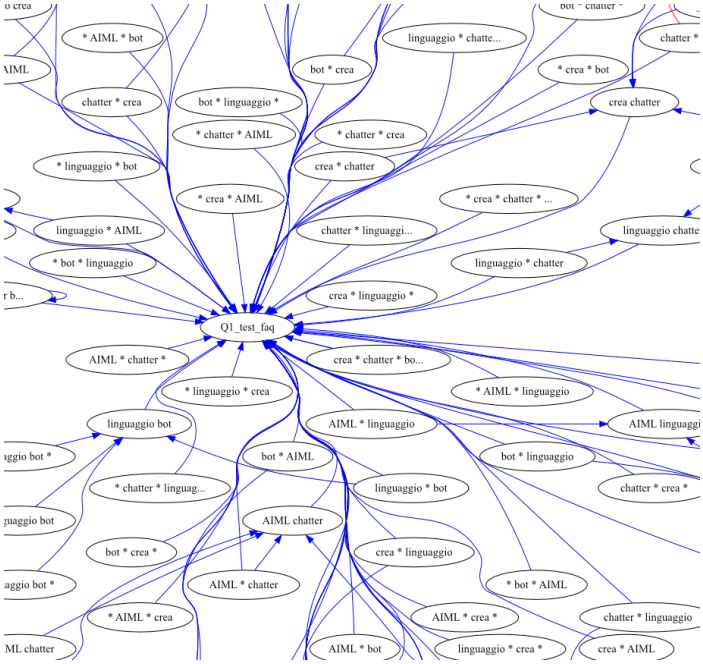


Fig. 1 - Particular of the KB representation graph centered on the question Q1. In the graph nodes are patterns (P) and templates (T), and edges are P-T associations and T-P semantic recursions. The graph represents the P-T generated by FAQ and glossary files explained in the paper.

## 2 Knowledge-Base Design Methodology

Frequently asked question sets define semantically the knowledge domain given to the chat bot. In the form of text file they are easy to write and to obtain because they are often available at many web sites. Text glossaries are, instead, less common, but can be derived from many available online resources, thanks to the ability to relate given terms with the knowledge domain.

All the knowledge related to the domain of interest is explicitly and implicitly included in FAQ and glossary files. So making available a method to extract all the possible knowledge from FAQ and glossary files could be an important starting point for the process to generate an answering software expert about the same knowledge domain, as shown in figure 2.

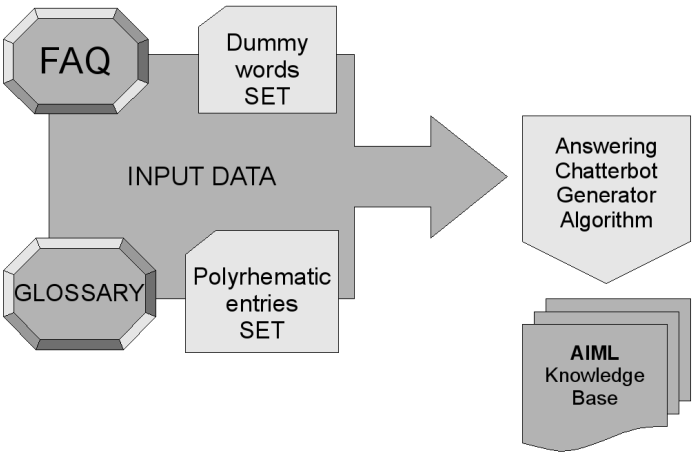


Fig 2 - Workflow of the chatter bot generation process

Given the FAQ text file TFAQ,in a format where a question is associated to the relative answer, and the glossary file TGLO, in a format where a glossary term is associated to the respective definition, in order to obtain an accurate

answering chatter bot at the end of the generating process, the following steps need to be accomplished:

1. Definition of the set of dummy words

Depending on the knowledge domain:

2. Definition of the set of domain specific polyrhematic entries
3. Definition of the chatter bot generation algorithm

where dummy means words used only as syntactic structure holders, but with zero or low semantic levels; polyrhematic entries are sentences made by a sequence of words which relate to a single semantic entity. The two sets are used in combination to filter out not meaningful words before building the pattern set for the AIML generation. Computational linguistics techniques can be used in order to cut the set of words; they are useful to identify dummy words, which usually correspond to the most frequent words that can be calcutated given a set of spoken language corpora, in order not to insert them in the AIML database.

## 2.1 INPUT data definition

Data is organized as follows:

FILE FAQ $F_{in}$:

```
Q <space> <question text>
A <space> <answer text>
```

FILE Glossary $G_{in}$:

```
G <space> <item text (polyrhematic>)
D <space> <definition text>
```

Set of dummy words $D_{in}$:
A text file with a word for each line.

Polyrematic entries set $P_{in}$:
A text file with a polyrhematic entry (few words `s`) for each line.

# 3 Chatter bot generator algorithm

Given the input provided by $\{F_{in},G_{in},D_{in},P_{in}\}$, the algorithm output will be a 1.0.1 compliant AIML(XML) file $Z_{out}$ graph (AIML, 2005). The AIML generated output can be later used as the knowledge base of the question answering chatter bot, being processed by any AIML interpreter/reasoner.

## 3.1 Generation of FAQ-AIML

The generator algorithm has been developed in Python programming language resulting in about 500 lines of code. The main steps can be summarized as follows:

Algorithm 1: Main AIML Generation Algorithm

F1. extract all the relevant categories lists from FAQ questions $F_{in}$
F2. calculate possible branches
F3. extract the answers
F4. generate AIML, i.e $Z_{out}$ graph

### *3.1.1 Detailed steps of F1*

A single category, such as it is defined in the AIML, is a couple of patterns-templates. The patterns need to coincide with one or more words taken from the question Fin so that they can be found in the user question and matched to

the proper answer of the FAQ file, as listed in Algorithm 2.

Algorithm 2: Generation of AIML categories

```
define D_w as the dummy words set
define P_w as the polyrhematic entries
FOR alla questions q in FAQ F_in DO
        build list L  of meaningful words w_i
                (i.e. Filter out all w_i in D_w and use w_i in P_w)
         initialize an emtpy category list C
        FOR all words w_i in L DO
                append w_i 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 S_c
END FOR
```

### 3.1.2 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 the category lists C_1  in S_c DO
        A_i is to be the answer whose question Q_i generated C_1
        FOR all categories c_i in C_1 DO
                append the integer i to the OUT[c_i] 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.

### 3.1.3 Detailed steps of F4

This method finally generates the FAQ AIML file, trying to catch all of the meaningful words from the user sentence and matching them with the meaningful words of the FAQ questions. It generates SRAI recursions as defined by the AIML 1.0.1 standard (AIML, 2005).

## 3.2 Generation of GLOSSARY-AIML

The generation of AIML Glossary is straightforward: for each glossary item

filtered with dummy and polyrhematic sets a definition is associated with the relative glossary definition.

Algorithm 4: Generation of output graph Zout and final AIML

```
FOR all questions Q_i DO
        given the category list C_1 generated from Q_i
        let T_a be the AIML SRAI template containing the answer text
        FOR all categories C_1 in C_1 DO
                IF C_1 is a combination of two words THEN
                        generate all possible edges (SRAI) to T_a
                ELSE IF C_1 has just one branch THEN
                        generate an edge (SRAI) to T_a
                        IF C_1 is just a single word THEN
                                generate an edge (SRAI) to glossary item
                        END IF
                END IF
        END FOR
END FOR
```

## 4 AIML ouput

The generation of the chatter bot AIML includes generation from the FAQ file and from the glossary file. Also a reduced AIML needs to be generated guided by the dummy words set, so that each sentence the user enters during the conversation will be reduced to a sequence of meaningful words that can be matched with the FAQ AIML and GLOSSARY AIML knowledge-base, in order to search the right answer to the asked question.

## 5 Case study of an Italian language answering bot

The case study is designed to be simple in order to keep small sizes of data, especially after graph visualization. It is made up of two questions Q1 and Q2.

A simple chatter bot can be generated using the following input data file:

### 5.1 FAQ

**Q** Come si crea un chatter-bot con il linguaggio AIML?

**A** Prima devi imparare il linguaggio e i suoi principali costrutti. Ovviamente devi prima aver imparato l'XML perche' ne e' un suo derivato. Tra i costrutti sintattici pi potenti ti suggerisco lo SRAI, che ti permette di connettere coppie pattern-template tra loro, come ad esempio nel caso dei sinonimi. Poi devi costruire bene la base di conoscenza in AIML e darla in pasto a pandorabots.com.

**Q** Come si usa un chatter-bot?

**A** Per usare il chatter-bot devi inviare i file AIML presso un server ove sia residente un interprete AIML. Quindi devi editare una pagina HTML che contenga i campi codificati per la domanda e la risposta verso l'utente finale.

## 5.2 Glossary file

**G** chatter bot

**D** un chatter-bot e' un software che simula una conversazione in linguaggio naturale

**G** AIML

**D** AIML, Artificial Intelligence Markup Language, e' il linguaggio per la definizione di una tipologia di chatter bot introdotto nel 2000 da Richard Wallace

**G** Alice

**D** E' il miglior chatter-bot conosciuto basato su AIML

**G** Richard Wallace

**D** Dr. Richard Wallace e' l'inventore del linguaggio AIML e del chatter bot Alice

## 5.3 Generated knowledge base

Given the 5 text lines of the FAQ file, the output comprises 195 categories. The glossary file comprises 19 categories, made up of the original 4 glossary entries and a combination of meaning asking words. The categories cannot be shown here for lack of space. They can be requested to the author by email. The generated chatter bot is available online.

## Conclusions

A methodology to develop automatically an AIML answering chatter bot to FAQ has been shown. Possible applications in e-learning could be to facilitate interaction with the user or user navigation through teaching material by means of a human digital assistant through a speaking avatar. For example, in a typical distance learning session the learning module contents can be summarized in a glossary and FAQs, the student can use the online learning material in a conventional way, but he/she can also interact with the digital assistant, implemented through the above-mentioned methodologies; so the student can ask it free text questions about those contents, if they can be expressed through a system of question and answer.

Other possible applications can be developed in personal robotics and

pseudo natural language systems that need to interact in multimodal ways (Pirrone & Cannella, 2008).

# REFERENCES

Pirrone R., Cannella V., R.G. (2008), *Gaiml: A new language for verbal and graphical interaction in chatbots*. In: International Conference on Complex, Intelligent and Software Intensive Systems, 2008, 715–720.

Many Authors (2009), *Chatterbot entry*. http://en.wikipedia.org/wiki/Chatterbot (November 2009) Wikipedia.

AIML 1.0.1 reference (2005), http://www.alicebot.org/TR/2005/WD-aiml (2005) ALICE Artificial Intelligence Foundation.

Wallace R. (2009), *Pandorabots announces the availability of bespoke pandorabots spellbinder service*. Web page (October 2009) Pandorabots.com, http://pandorabots.com/pandora/pics/spellbinder/index.html.

Smid K. P.I. (2002), *Conversational virtual character for the web*. In: Proceedings of Computer Animation 2002, Geneva, Swtizerland (2002) 240.

Wallace R. (2007), *AIML pattern matching simplified*. http://www.alicebot.org/documentation/matching.html (2007) ALICE Artificial Intelligence Foundation.

Shawar B.A., Atwell E. R. A. (2008), *Faqchat as an information retrieval system*. http://www.comp.leeds.ac.uk/andyr/research/papers/ltc05-faqchat.pdf (2008) FAQchat.

De Gasperis G. (2009), *Italian generated example chatter-bot*. http://www.pandorabots.com/pandora/talk?botid=f0a3e607de36aa16 (2009) Hosted on pandorabots.com.

Güzeldere G., Franchi S. (1995), *Dialogues with colorful personalities of early AI*. http://www.stanford.edu/group/SHR/4-2/text/dialogues.html (24 July 1995).