Automatic Extraction of Chatbot Training Data from Natural Dialogue Corpora

Bayan AbuShawar, Eric Atwell
IT department; School of Computing
Arab Open University; University of Leeds
Amman, Jordan; Leeds, UK
b_shawar@aou.edu.jo; eric@comp.leeds.ac.uk; scmss@leeds.ac.uk

Abstract

A chatbot is a conversational agent that interacts with the users turn by turn using natural language. Different chatbots or human-computer dialogue systems have been developed using spoken or text communication and have been applied in different domains such as: linguistic research, language education, customer service, web site help, and for fun. However, most chatbots are restricted to knowledge that is manually “hand coded” in their files, and to a specific natural language which is written or spoken.

This paper presents the program we developed to convert a machine readable text (corpus) to a specific chatbot format, which is then used to retrain a chatbot and generate a chat which is closer to human language. Different corpora were used: dialogue corpora such as the British National Corpus of English (BNC); the holy book of Islam Qur’an which is a monologue corpus where verse and following verse are turns; and the FAQ where questions and answers are pair of turns. The main goal of this automation process is the ability to generate different chatbot prototypes that spoke different languages based on corpus.

Keywords: Chatbot, ALICE, AIML, Corpus

1. Introduction

Human machine conversation is a new technology integrating different areas where the core is the language, and the computational methodologies, which aim to facilitate communication between users and computers via natural language. A related term to machine conversation is the chatbot, which is a conversational agent that interacts with users turn by turn using natural language; chatbots have been applied in different domains such as: linguistic research, language education, customer service, website help, and for fun. The purpose of a chatbot system is to simulate a human conversation; the chatbot architecture integrates a language model and computational algorithms to emulate informal chat communication between a human user and a computer using natural language.

The idea of chatbot systems originated in the Massachusetts Institute of Technology (Weizenbaum 1966, 1967), where Weizenbaum implemented the Eliza chatbot to emulate a psychotherapist. The idea was simple and based on keyword matching. The input is inspected for the presence of a keyword. If such a word is found, the sentence is mapped according to a rule associated with the keyword; if not, a connected free remark, or under certain conditions an earlier transformation, is retrieved. For example, if the input includes the keyword “mother”, ELIZA can respond “Tell me more about your family”. This rule is inspired by the theory that mother and family are central to psychological problems, so a therapist should encourage the patient to open up about their family; but the ELIZA program does not really ‘understand’ this psychological strategy, it merely matches the keyword and regurgitates a standard response. To keep the conversation going, ELIZA has to produce responses which encourage the patient to reflect and introspect, and this is done mechanistically using some fixed phrases if no keyword match is found such as “Very interesting. Please go on.”, or “Can you think of a special example?” The next major program was PARRY (Colby 1973, 1999). In contrast to ELIZA, instead of simulating a psychotherapist, PARRY modelled a paranoid patient during an interview with his therapist. Saygin, Cicelki, and Akman (2000) noticed that, “Both ELIZA and PARRY use certain tricks to be able to successfully perform in conversations. ELIZA directs the conversation away from herself by asking questions. ELIZA uses parts of the user’s input in the output questions and seems to be following the conversations. In addition to these techniques, PARRY has little stories to tell and tend to insert these in the conversation.” Figure 1 shows a sample of chatting with ELIZA.

The seventies and eighties, before the arrival of graphical user interfaces, saw rapid growth in text and natural-language interface research. Since that time, a range of new chatbot architectures have been developed, such as: MegaHAL (Hutchens, 1996), CONVERSE (Batacharia et al., 1999), ELIZABETH (Abu Shawar and Atwell, 2002),
HEXBOT (2004) and ALICE (2002). With the improvement of data-mining and machine-learning techniques, better decision-making capabilities, availability of corpora, robust linguistic annotations/processing tools standards like XML and its applications, chatbots have become more practical, with many commercial applications. In this paper, ALICE/AIML chatbot architecture and the pattern matching techniques used within it are presented in section 2. Section 3 illustrates our development of a Java program that can convert a machine readable text (corpus) to the AIML format used by ALICE. A description of the automatic extraction of training data from different corpora is explained in section 4. The results an conclusion are presented in sections 5 and 6 consequently.

Human: Men are all alike.
ELIZA: In what way?
Human: They’re always bugging us about something.
ELIZA: Can you think of a special example?
Human: Well, my boyfriend made me come here.
ELIZA: Your boyfriend made you come here?
Human: He says I’m depressed much of the time.
ELIZA: I’m sorry to hear that you are depressed much of the time.
Human: It’s true. I’m unhappy.
ELIZA: Do you think coming here will help you not to be unhappy?
Human: I need some help, that much seems certain.

Figure 1. A sample of conversation with ELIZA (from Jurafsky and Martin 2000)

2. ALICE Architecture

We have worked with the ALICE open-source chatbot initiative. ALICE is the Artificial Linguistic Internet Computer Entity (ALICE 2002, Wallace 2003), originated by Wallace in 1995. A sample of chatting with ALICE is shown in figure 2. ALICE stores knowledge about English conversation patterns in AIML files. AIML, or Artificial Intelligence Mark-up Language, is a derivative of Extensible Mark-up Language (XML). It was developed by the Alicebot free software community during 1995-2000 to enable people to input dialogue pattern knowledge into chatbots based on the ALICE free software technology.

User: Hello, my name is George and I am the human.
ALICE: Hi George.
User: Would you like to play a game?
ALICE: Sure.
User: Ok then. I will email it to you by use of the CD drive.
ALICE: What hobbies do you enjoy?
User: I like cinema.
ALICE: What types of films?
User: All.
ALICE: What movie did you see this year that you liked?
User: I don't have time for movies.
ALICE: I thought you liked cinema.
User: I did, but then you started to say stupid things.

Figure 2. A sample of conversation with ALICE (from Loebner prize contest 2004)

AIML consists of data objects called AIML objects, which are made up of units called topics and categories as shown in figure 3. The topic is an optional top-level element, it has a name attribute and a set of categories related to that topic. Categories are the basic unit of knowledge in AIML. Each category is a rule for matching an input and converting to an output, and consists of a pattern, which represents the user input, and a template, which implies the ALICE robot answer. The AIML pattern is simple, consisting only of words, spaces, and the wildcard symbols _ and *. The words may consist of letters and numerals, but no other characters. Words are separated by a single space, and the wildcard characters function like words. The pattern language is case invariant. The idea of the pattern matching technique is based on finding the best, longest, pattern match.

<aiml version="1.0">
<topic name="the topic">
<category>
<pattern>USER INPUT</pattern>
<template>Chatbotanswer</template>
</category></topic></aiml>

Figure 3. The AIML format
2.1 Types of ALICE/AIML Categories

There are three types of the AIML categories: atomic categories, default categories, and recursive categories.

Atomic categories are those with patterns that do not have wildcard symbols, _ and *, e.g.:

```xml
<category>
  <pattern>WHAT IS 2 AND 2</pattern>
  <template> It is 4 </template>
</category>
```

In the above category, if the user inputs ‘What is 2 and 2’, then ALICE answers ‘it is 4’.

Default categories are those with patterns having wildcard symbols * or _. The wildcard symbols match any input but they differ in their alphabetical order. Assuming the previous input WHAT IS 2 AND 2, if the robot does not find the previous category with an atomic pattern, then it will try to find a category with a default pattern such as:

```xml
<category>
  <pattern>WHAT IS 2 *</pattern>
  <template><random>
    <li>Two.</li>
    <li>Four.</li>
    <li>Six.</li>
  </random>
</template>
</category>
```

So ALICE will pick a random answer from the list. The _ works on the same manner but if first words are missed: `<template> _4 </template>` so any template end with 4 will match.

Recursive categories are those with templates having `<srai>` and `<sr>` tags, which refer to simply recursive artificial intelligence, and symbolic reduction. Recursive categories have many applications: symbolic reduction that reduces complex grammatical forms to simpler ones; divide and conquer that splits an input into two or more subparts, and combines the responses to each; and dealing with synonyms by mapping different ways of saying the same thing to the same reply as the following example:

```xml
<category>
  <pattern>HALO</pattern>
  <template><srai>Hello</srai>
</template></category>
```

The input is mapped to another form, which has the same meaning.

2.2 ALICE/AIML Pattern Matching Technique

The AIML interpreter tries to match word by word to obtain the longest pattern match, as this is normally the best one. This behavior can be described in terms of the Graphmaster as shown in figure 4. A Graphmaster is a set of files and directories, which has a set of nodes called nodemappers and branches representing the first words of all patterns and wildcard symbols. Assume the user input starts with word X and the root of this tree structure is a folder of the file system that contains all patterns and templates; the pattern matching algorithm uses depth first search techniques:

- If the folder has a subfolder starting with underscore then turn to “_/”, scan through it to match all words suffixed X, if no match then:
- Go back to folder, try to find a subfolder starts with word X, if so turn to “X/”, scan for matching the tail of X, if no match then:
- Go back to the folder, try to find a subfolder start with star notation, if so, turn to “*/”, try all remaining suffixes of input following “X” to see if one match. If no match was found, change directory back to the parent of this folder, and put “X” back on the head of the input. When a match is found, the process stops, and the template that belongs to that category is processed by the interpreter to construct the output.

The above paragraph describes how internally ALICE search a response for the user input, how to match between user input and the stored knowledge in AIML brain. Users does not know what knowledge is there, but whatever the user input is, ALICE will try to find the longest pattern match based on lexical matching. In the following section we will clarify how we implement a Java program to read from any corpus and convert it into AIML format then extend ALICE knowledge by the generated categories.

Figure 4. A Graphmaster that represents ALICE brain
3. Automatic Generation of AIML Categories

We developed a Java program that converts the readable text (corpus) to the chatbot language model format. The aim of this software is to create ALICE knowledge base automatically and based on specific corpus or domain. Then extend current knowledge of ALICE with the new generated files.

Two versions of the program were generated. The first version is based on simple pattern template category, so the first turn of the speech is the pattern to be matched with the user input, and the second is the template that holds the robot answer. Usually the dialogue corpora contain linguistic annotation that appears during the spoken conversation such as overlapping, and using some linguistic filler. To handle the linguistic annotations and fillers, the program is composed of four phases as follows:

1. Phase One: Read the dialogue text from the corpus and insert it in a vector.
2. Phase Two: Text reprocessing modules, where all linguistic annotations such as overlapping, fillers and other linguistic annotations are filtered.
3. Phase Three: converter module, where the pre-processed text is passed to the converter to consider the first turn as a pattern and the second as a template. Removing all punctuation from the patterns and converting it to upper case is done during this phase.
4. Phase Four: Copy these atomic categories in an AIML file.
5. Phase Five: Building a frequency list of patterns lexical. This list will be used to obtain the first and second most significant words (least frequent words) from each utterance.
6. Phase Six: Building the default category file. AIML pattern-matching rules, known as “categories”, are created. There are two possible types of match: input matches a complete pattern so atomic categories will be matched; or input matches 1st or 2nd most significant word in the user input (least frequent words).

After building the atomic files in phase 4, the program is adopted to a more general approach to finding the best match against user input from the learned dialogue. In case no exact matching is found the default categories are built to give a close answer based on significant keywords: first word and most significant ones.

A restructuring module was added to map all patterns with the same response to one form, and to transfer all repeated pattern with different templates to one pattern with a random list of different responses. Two machine learning approaches were adapted to build default categories (phase six) as follows:

- First word approach, based on the generalisation that the first word of an utterance may be a good clue to an appropriate response: if we cannot match the whole input utterance, then at least we can try matching just the first word. For each atomic pattern, we generated a default version that holds the first word followed by wildcard to match any text, and then associated it with the same atomic template.
- Most significant word approach, we look for the word in the utterance with the highest "information content", the word that is most specific to this utterance compared to other utterances in the corpus. This should be the word that has the lowest frequency in the rest of the corpus. We choose the most significant approach to generate the default categories, because usually in human dialogues the intent of the speakers is hiding in the least-frequent, highest-information word. We extracted a local least frequent list from the corpus, and then compared it with each token in the pattern to specify the first most significant word within that pattern. Later on, the second most significant word were also used in conjunction with first word and first most significant word to obtain the best pattern match. Once may argue that significant word could misspelled, in this case a default category will be built but it will be matched if user input have the same misspelled word which will be rare.

The program was enhanced to handle different format and structure for three main types of corpora as follows:

- Dialogue corpora: where each corpus has its own annotations, so filtering process will differ. And the first utterance is considered as a pattern and the next one as a template (response).
- Monologue corpora: represented by the holy book of Islam the Qur’an where each verse is considered as a pattern and the next one as a template.
- FAQ corpora: where the question represents the pattern and the answer represents the template.

4. Applying the Program on Multi Corpora

During the enhancement and evolving to our system, we tried different types of corpora: human dialogue transcripts, monologue and structural one (FAQs, QA). In this section a brief discussion of all corpora used and how our software was evolved are presented.

4.1 Human Dialogue Transcripts

Two versions of the system were initially developed. The first version is based on simple pattern template category, so the first turn of the speech is the pattern to be matched with the user input, and the second is the template that holds the robot answer. This version was tested using the English-language Dialogue Diversity Corpus (DDC, Mann, 2002). This corpus is a collection of links to different dialogue corpora in different fields, where each corpus has its own annotation format. After text re-processing and filtering, the Java program was simple and considered each utterance as a pattern, and its successor as a template that represents chatbot answer. This experiment reveals the
problems of utilising dialogue corpora such as: long turns; no standard annotations to distinguish between speakers, overlapping and irregular turn taking, and using linguistic fillers. (Abu Shawar and Atwell 2003a). Unfortunately most of these problems also occur in other corpora, which necessitate changing the filtering process to meet the difference in the corpora format. Figure 5 shows a sample of DDC and its equivalent atomic category.

```
<who=F72PS002>
<s n="32"><w ITJ>Hello<c PUN>.</w>
</u>
<who=PS000>
<s n="33">Hello <w NP0>Donald<c PUN>.</w>
</u>

The corresponding AIML atomic category is:

<class>
<pattern>HELLO</pattern>
<template>Hello Donald</template>
</category>
```

Figure 5. A sample of DDC turn and its equivalent atomic category

To prove that our system can be used with other dialogue corpora, the Minnesota French Dialogue Corpus (kerr 1983) was used. One advantage of the Machine-Learning approach to re-training ALICE is that we can automatically build AIML from a corpus even if we don’t understand the domain or even the language; to demonstrate this, the program was tested using the Corpus of Spoken Afrikaans (van Rooy, 2003). The new chatbot that speaks Afrikaans was published on line using Pandorabot service\(^1\), and we encouraged open-ended testing and feedback from remote users in South Africa; this allowed us to refine the system more effectively. We adopted three evaluation metrics (AbuShawar and Atwell, 2003b, 2007).

- **Dialogue efficiency in terms of matching type:** whether a user input is matched an atomic category, or first word, and most significant ones. For this purpose four sample of dialogues were analyzed. Outputs illustrates that the first word and the most significant approach increase the ability to generate answers to users and let the conversation continue.

- **Dialogue quality metrics based on response type:** where the responses of the four dialogue sample are classified according to an independent human evaluation of “reasonableness”: reasonable reply, weird but understandable, or nonsensical reply. We gave the transcript to an Afrikaans-speaking teacher and asked her to mark each response according to these classes. The number of turns in each dialogue and the frequencies of each response type were estimated. For this evaluator, it seems that “nonsensical” responses are more likely than reasonable or understandable but weird answers.

- **Users' satisfaction assessment based on an open-ended request for feedback:** expanding the AIML pattern matching using the first-word and least-frequent-word approaches yielded more favorable feedback. Our informants found the conversations less repetitive and more interesting.

It took several years for the Alice Botmaster to accumulate the 50,000 categories in the current public-domain set of AIML files (Wallace 2003). The British National Corpus (BNC 2002) was selected to test if our system is capable to generate more than one million categories extracted from the BNC. The British National Corpus (BNC) (Aston and Burnard 1998) is a collection of text samples amounting to over 100 million words, extracted from 4,124 modern British English texts of all kinds, both spoken and written. The corpus is annotated using SGML (XML-like) mark-up, including CLAWS Part-of-Speech category of every word. All annotations are marked between <angle brackets>. The corpus is partitioned into two types: the spoken and the written transcripts. In order to generate versions of ALICE which talk like humans, we decided to retrain ALICE using the BNC spoken transcripts comprising 10 million words. A range of different chatbot systems were generated using the program that converts the BNC to AIML format; this enabled us to animate and explore the language of various BNC sample texts. 1,153,129 categories were created in total, and divided in files representing a range of domains such as: sport, food, travel and media. By the BNC prototypes the chatbot could be used a tool to visualise a large corpus. Figure 6 shows a sample of chatting with BNC prototype in the sport domain.

**Human:** Hi, do you like sport?
**Chatbot:** Hi there. No I’m not much into sports. I like solo sports like rollerblading and skiing though.

**Human:** Do you know anything about football?
**Chatbot:** Any of several games played with an inflated leather ball by two teams on a field with goals at each end, the object being to get the ball across the opponents' goal.

**Human:** What about women's football?
**Chatbot:** I think football seems to, to bring out the worst in a lot of people and that's the part of it that I do not like.

\(^1\)http://www.pandorabots.com/pandora
**Human:** Do you like it?

**Chatbot:** and er, I think until women's football take it on board all the way erm, they're not really gonna stand up with the men they have to become, referees, they have to become coaches, and more and more of these people erm, taking responsibility I go along and watch women's football and get really infuriated to see a man out there refereeing the game, and even in a very condescending manner! Instead of carding someone he'll put his arm round her and say Do not! It's not very nice to do that! Do not do that again!

**Human:** It is an enjoyable game.

**Chatbot:** Aha Aha It is a case is not it, that if you want to be a er a professional woman footballer, certainly there's more opportunities on the continent than there are in Britain, is, is is that the case? Yes?

**Human:** I watch it regularly.

**Chatbot:** Because if football’s on it has to be watched! It’s like a a religion! Fi

---

**4.2 Holy Qur’an Text**

In order to see if we can extract AIML files from a corpus text which is not a dialogue where no turns are there, another version of the system was updated to generate an Arabic AIML files extracted from the Qur’an, the holy book of Islam. Moslems believe the Arabic text is a faithful transcription of the infallible words of God relayed through the angel Gabriel to the prophet Mohammed, who memorised the entire monologue to pass on verbally. Mohammed’s successors transcribed the message to simplify transmission and avoid corruption, but every Moslem should aim to memorise it, in original Arabic, and to use the Qur’an to guide every aspect of their lives. The Qur’an consists of 114 sooras, which could be considered as sections, grouped into 30 parts (chapters). Each soora consists of more than one verse (Ayya). These ayyas are sorted, and must be shown in the same sequence. The AIML-learning system was revised to handle the non-conversational nature of the Qur’an. Aversion of English/Arabic was built where in this case if an input was a whole verse (Ayya), the response will be the next verse of the same soora; or if an input was a question or a statement, the output will be all verses which seem appropriate based on the significant word. As ample of chatting with the Qur’an chatbot is presented in figure 7.

**THE DAYBREAK, DAWN, CHAPTER NO. 113**

With the Name of Allah, the Merciful Benefactor, the Merciful Redeemer

113/1 Say: I seek refuge with the Lord of the Dawn
113/2 From the mischief of created things;
113/3 From the mischief of Darkness as it overspreads;
113/4 From the mischief of those who practice secret arts;
113/5 And from the mischief of the envious one as he practices envy.

**Figure 7. Samples of the Arabic and English versions of the Qur’an**

To measure the quality of the answers of the Qur’an chatbot version, the following approach was applied:

1. Random sentences from Islamic sites were selected and used as inputs of the English/Arabic version of the Qur’an.
2. The resulting transcripts which have 67 turns were given to 5 Muslims and 6 non-Muslims students, who were asked to label each turn in terms of:
   - Related (R), in case the answer was correct and in the same topic as the input.
   - Partially related (PR), in case the answer was not correct, but in the same topic.
   - Not related (NR), in case the answer was not correct and in a different topic.

Proportions of each label and each class of users (Muslims and non-Muslims) were calculated as the total number over number of users times number of turns. Four out of the 67 turns returned no answers, therefore actually 63 turns.

In the transcripts used, more than half of the results were not related to their inputs. A small difference can be noticed between Muslims and non-Muslims proportions. Approximately one half of answers in the sample were not related from non-Muslims’ point of view, whereas this figure is 58% from the Muslims’ perspective. Explanation for this includes:

- The different interpretation of the answers. The Qur’an uses traditional Arabic language, which is sometimes difficult to understand without knowing the meaning of some words, and the historical story behind each verse.
- The English translation of the Qur’an is not enough to judge if the verse is related or not, especially given that non-Muslims do not have the background knowledge of the Qur’an.
Using chatting to access the Qur'an looks like the use of a standard Qur'an search tool. In fact it is totally different; a searching tool usually matches words not statements. For example, if the input is: “How shall I pray?” using chatting: the robot will give you all ayyas where the word “pray” is found because it is the most significant word. However, using a search tool\(^2\) will not give you any match. If the input was just the word “pray”, using chatting will give you the same answer as the previous, and the searching tool will provide all ayyas that have “pray” as a string or substring, so words such as: “praying, prayed, etc.” will match.

Another important difference is that in the search tool there is a link between any word and the document it is in, but in the chatting system there is a link just for the most significant words, so if it happened that the input statement involves a significant word(s), a match will be found, otherwise the chatbot answer will be: “I have no answer for that”. (Abu Shawar and Atwell 2004a, 2004b).

### 4.3 FAQ Websites

Corpus Linguists have shown increasing interest in possibilities of using the Web as corpus source, so we decided to try using FAQs as pseudo-dialogue to train chatbots. In these versions the chatbot expects user input similar to FAQ questions, and replies with FAQ answers, or with lists of links to possible FAQ answers. FAQs have clear turn-taking structure, and are widely used in computing, so are a good source of training corpus data. The system was adapted such that the question represents a pattern, and the answer represents the template in building atomic AIML files. The frequency list was constructed from questions (patterns).

A Frequently Asked Questions document or FAQ is a structured database; most of the “data-cleaning” problems found with dialogue corpora (such as overlapping, and more than two speakers) are not found in the FAQs. Moreover almost all HTML tags embedded in the FAQ file are recognised by the ALICE interpreter, because the AIML definition allows HTML tags to be embedded within templates. The questions and answers were extracted directly from the HTML files of the FAQ.

Different categories are added to extend the chance of finding answers, where the answer is either a set of links in case most significant words are found in more than one question or a direct answer in the instance where only one match was found. In addition to first word and most significant word (1st), we extracted second most significant one (2nd) (least frequent words). For each significant word, four default categories were added to handle different position of word in pattern; another category holding first word, 1st or 2nd most significant as appeared in original question was generated.

At the end a FAQchat prototype was generated and tested against Google by nine staff members and twelve students at School of Computing in University of Leeds. As a result, 68% overall of our sample of users (21 users) managed to find answers using the FAQchat; while 46% found it by Google. Since there is no specific format to ask the question, there are cases where some users could find answers while others could not. In terms of preferences, 51% of the staff, 41% of the students, and 47% overall preferred using FAQchat against 11% who preferred the Google (Abu Shawar et al., 2005).

Two example English FAQ chatbots were generated which are trained using: the FAQ of the School of Computing (SoC) at the University of Leeds; and the public-domain Python programming language FAQ website. The Python FAQ website has been translated into other languages, so we were able to generate a Spanish-speaking Python advisor. Figure 8 illustrates the questions and answers generated with these FAQ chatbots.

<table>
<thead>
<tr>
<th>a. Using the FAQ of the School of Computing at the University of Leeds</th>
</tr>
</thead>
</table>
| **Human**: How to remove a job from the printer?  
**Ask me:** If you want to remove a job from the printer queue then find out the job number using the `lpr help` command, and in a UNIX shell type: `lprm -P printernamejob_no`  
You can only remove jobs that you have sent to the printer yourself - apart from Support that is.  
For more information look at: [How do I remove a job from the printer queue? (Printing)](http://www.islamicity.com/QuranSearch/) |

<table>
<thead>
<tr>
<th>b. Using the FAQ of the Python programming language website</th>
</tr>
</thead>
</table>
| **Human**: what is a class?  
**English_Python_FAQ**: A class is the particular object type created by executing a class statement. Class objects are used as templates to create instance objects, which embody both the data (attributes) and code (methods) specific to a datatype. A class can be based on one or more other classes, called its base class(es). It then inherits the attributes and methods of its base classes...... |

---

\(^2\)http://www.islamicity.com/QuranSearch/
The success with using chatbot as a tool to answer SoC FAQs encouraged us to try other FAQs, or Questions Answers (QA) corpora to investigate the possibility of using a chatbot as a tool to access an information portal without the need for sophisticated natural language processing or logical inference. In 2008, an open ended FAQChat was built where the knowledge base was extracted from multiple FAQs: Perl, Linux and Python. In 2010, TREC09 QA track was used to retrain ALICE and in 2011 Arabic QA corpora was used. Overall User trials with AskJeeves, Google and generated chatbot demonstrate that chatbot is a viable alternative, and in fact many users prefer it to Google as tool to access FAQ databases (Abu Shawar 2008, 2011), (AbuShawar and Atwell 2010). We managed to demonstrate that simple ALICE-style chatbot engine could be used as a tool to access the WWW FAQs, or QAs. No need for sophisticated natural language analysis or logical inference; a simple (but large) set of pattern-template matching rules will be sufficient.

5. Results and Outputs

The automatic system was evolved within each corpora used, the system was able to generate AIML training data for ALICEchatbot to serve in different domains. For example, the number of training data that were extracted automatically from the BNC corpus is 1,153,129 which is 20 times bigger than any existing chatbot: the large public domain. Some of the different usages of ALICE, and some statistics about the number of categories that were built automatically are present in tables 1 and 2 consequently.

### Table 1. A sample of services that can be provided by ALICE chatbot

<table>
<thead>
<tr>
<th>ALICE chatbot</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALICE (Wallace 2003)</td>
<td>Entertainment chatbot</td>
</tr>
<tr>
<td>Speak2Me(2004)</td>
<td>A web-based version of ALICE aimed at Chinese learners of English, allowing them to practise chatting to a well-spoken young lady, a virtual British English native speaker</td>
</tr>
<tr>
<td>AfrikaanaChatbot</td>
<td>A tool to learn/practice a language.</td>
</tr>
<tr>
<td>BNCChatbot</td>
<td>A tool to visualize (animate) a corpora.</td>
</tr>
<tr>
<td>Arabic/English Qur'an chatbot</td>
<td>A tool to learn Qur'an for Arabic and English speakers.</td>
</tr>
<tr>
<td>FAQchat</td>
<td>A tool to access an information portal</td>
</tr>
</tbody>
</table>

### The number of AIML training data obtained from BNC domains

<table>
<thead>
<tr>
<th>BNC domains</th>
<th>Atomic Categories</th>
<th>Default Categories</th>
<th>Total Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>File size (KB)</td>
<td>File size (KB)</td>
<td></td>
</tr>
<tr>
<td>Sport</td>
<td>3283</td>
<td>7913</td>
<td>11196</td>
</tr>
<tr>
<td>World affairs</td>
<td>3120</td>
<td>8756</td>
<td>11876</td>
</tr>
<tr>
<td>Travel</td>
<td>640</td>
<td>1636</td>
<td>2276</td>
</tr>
<tr>
<td>Media</td>
<td>1061</td>
<td>3126</td>
<td>4187</td>
</tr>
<tr>
<td>Food</td>
<td>503</td>
<td>1125</td>
<td>1628</td>
</tr>
</tbody>
</table>

### The number of AIML training data obtained from the Qur’an

<table>
<thead>
<tr>
<th>The Holy book of Islam the Qur’an</th>
<th>Atomic Categories</th>
<th>Default Categories</th>
<th>Total Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>File size (MB)</td>
<td>File size (MB)</td>
<td></td>
</tr>
<tr>
<td>22,950</td>
<td>11,51</td>
<td>44.2</td>
<td>79,299</td>
</tr>
</tbody>
</table>

### The number of AIML training data obtained from the Arabic FAQ Websites

<table>
<thead>
<tr>
<th>Arabic FAQ domains</th>
<th>Atomic Categories</th>
<th>Default Categories</th>
<th>Total Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mother&amp;Kids</td>
<td>164</td>
<td>2,057</td>
<td>2,221</td>
</tr>
<tr>
<td>Blood_disease</td>
<td>97</td>
<td>1,292</td>
<td>1,389</td>
</tr>
<tr>
<td>Blood_charity</td>
<td>69</td>
<td>960</td>
<td>1,029</td>
</tr>
<tr>
<td>Health&amp;fasting</td>
<td>25</td>
<td>318</td>
<td>343</td>
</tr>
<tr>
<td>Teeth Care issues</td>
<td>42</td>
<td>614</td>
<td>656</td>
</tr>
</tbody>
</table>

Table 2. A sample of AIML training data obtained from different corpora
6. Conclusion

A chatbot is a conversational agent that interacts with users using natural language. An illustration of ALICE chatbot in terms of: Knowledge base and its pattern matching technique is presented. The main lack in ALICE and other chatbots is the manual developing of its knowledge, we managed to build a software program that reads from a corpus and converts it to ALICE knowledge base. This paper presents the algorithm of the automation process and machine learning techniques used to extract training data from a corpus. Different corpora were used to retrain ALICE that reveals other usefulness for of chatbot rather than an entertainment tools. A chatbot could be used as a tool to animate or visualize a corpus, a tool to learn/practice English, Arabic, Afrikaans or other languages, and a tool to access an information portal to provide answers to questions.

7. References


BNC (2002). British National Corpus website http://www.natcorp.ox.ac.uk/


Chatbot. 2015. [online]: https://www.chatbots.org/


