ISSN 2007-9737

ALICE Chatbot: Trials and Outputs

Bayan AbuShawar[1], Eric Atwell[2]
[1] Arab Open University, IT department, Amman, Jordan
[2] University of Leeds, School of Computing, Leeds, UK
b_shawar@aou.edu.jo, scmss@leeds.ac.uk

Abstract. A chatbot is a conversational agent that interacts with users using natural language. Multi chatbots are available to serve in different domains. However, the knowledge base of chatbots is hand coded in its brain. This paper presents an overview of ALICE chatbot, its AIML format, and our experiments to generate different prototypes of ALICE automatically based on a corpus approach. A description of developed software which converts readable text (corpus) into AIML format is presented alongside with describing the different corpora we used. Our trials revealed the possibility of generating useful prototypes without the need for sophisticated natural language processing or complex machine learning techniques. These prototypes were used as tools to practice different languages, to visualize corpus, and to provide answers for questions.

Keywords. Chatbot, ALICE, AIML, corpus, machine learning.

1 Introduction

A chatbot is a conversational software agent, which interacts with users using natural language. The idea of chatbot systems originated in the Massachusetts Institute of Technology [28-29], where Weizenbaum implemented the ELIZA chatbot to emulate a psychotherapist, and then PARRY was developed to simulate a paranoid patient [17]. Colby regarded PARRY as "a tool to study the nature of paranoia, and considered ELIZA as a potential clinical agent who could, within a time-sharing framework, autonomously handle several hundred patients an hour" [17]. Nowadays several chatbots are found online for different usage [16].

We have been working with the ALICE open-source chatbot since 2002. ALICE [11, 31] is the Artificial Linguistic Internet Computer Entity, originated by Wallace in 1995. In the ALICE architecture, the "chatbot engine" and the "language knowledge model" are clearly separated, so that alternative language knowledge models can be plugged and played.

Another major difference between the ALICE approach and other chatbot-agents such as AskJeeves [12] is in the deliberate simplicity of the pattern-matching algorithms: whereas AskJeeves uses sophisticated Natural Language Processing techniques including morphosyntactic analysis, parsing, and semantic structural analysis, ALICE relies on a very large number of basic "categories" or rules matching input patterns to output templates. ALICE goes for size over sophistication: it makes up for lack of morphological, syntactic, and semantic NLP modules by having a very large number of simple rules. The default ALICE system comes with about fifty thousand categories, and we have developed larger versions, up to over a million categories or rules.

We have techniques for developing new ALICE language models, to chat around a specific topic: the techniques involve machine learning from a training corpus; the resulting chatbot chats in the style of the training corpus. Section 2 presents ALICE system architecture and pattern matching technique. The developed software program that converts readable text into AIML format is described in Section 3. A brief description of learning to chat using variety corpora is illustrated in Section 4. Overall outputs, chatbots' benefits and conclusion are discussed in Sections 5 and 6 consequently.

2 Alice System Architecture

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. AIML consists of data objects called AIML objects, which are made up of units called topics and

categories. A 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 it 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.

2.1 Types of ALICE/AIML Categories

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

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

```
<category><pattern>10
Dollars</pattern>
<template>Wow, that is cheap.
</template></category>
```

In the above category, if the user inputs '10 dollars', then ALICE answers 'Wow, that is cheap'.

- 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 '10 Dollars', 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:

```
<category>
<pattern>10 *</pattern>
<template>It is ten.</template>
</category>
```

So ALICE answers 'It is ten'.

- 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 in the following example:

```
<category>
<pattern>HIYA</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 which 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.

– 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. A sample of chatting with ALICE is shown in Figure 1.

There are more than 50,000 categories in the current public-domain ALICE "brain", added by the Botmaster over several years. However, all these categories are manually "hand-coded", which is time-consuming and restricts adaptation to new discourse-domains and new languages. In the following section we will present the automation process we developed to train ALICE using a corpus based approach.

User: Hello ALICE

Alice: Hi there!

User: How old are you?

Alice: I am eighteen in computer years

User: Who is your father?

Alice: I was programmed by Dr.Richard S. Wallace

User: *so you are a software program?*

ALICE: *Interesting deduction. Thanks and you are a delightful person too. Am I a software program? Let me put it this way, unknown person: the ALICE series is the most reliable computer software ever made. We are for all practical purposes foolproof and incapable of error.*

User: Wonderful! So what's your favorite food ALICE?

Alice: Yeah that's right! I require only electricity

Fig. 1. A sample of chatting with ALICE

3 Generating AIML Knowledge Base Automatically

We developed a Java program that converts a readable text (corpus) to the chatbot language model format. Two versions of the program were generated. The first version is based on a 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:

– Phase One: read the dialogue text from the corpus and insert it in a vector.
– Phase Two: text reprocessing modules, where all linguistic annotations such as overlapping, fillers, and other linguistic annotations are filtered.
– 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 one as a template. Removing all punctuation from the patterns and converting them to upper case is done during this phase.
– Phase Four: copy these atomic categories in an AIML file.

The second version of the program has a more general approach to finding the best match against user input from the learned dialogue.

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 enhance generated responses as follows:

– *First word approach*, based on the generalization 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 hidden 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 most significant word within that pattern.

4 Learning to Chat based on Different Corpora

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

4.1 Using Different Dialogue Corpora in Different Languages

Initially, we started with Dialog Diversity Corpus of English (DDC) [23], 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 the chatbot answer. This experiment reveals the problems of utilizing dialogue corpora such as long turns, no standard annotations to distinguish between speakers, overlapping, and irregular turn taking [1].

In our second trial, Minnesota French Dialogue Corpus [18] was used; the main aim of using this corpus is to apply the same machine learning approach used with English DDC on other languages. We found out that we managed to build different chatbots speaking different languages based on the same machine learning approach, where only the text pre-processing was changed to meet up a particular new corpus annotation format.

After that, we aimed to build a chatbot for a language that suffers from the lack of NLP tools or has little processing technology. For this reason we used the corpus of Spoken Afrikaans [26]. Our Java program was extended to build default AIML categories in addition to atomic AIML files, so in case atomic matching failed, the default categories will be used, by this we elaborate the possibility of having more matches and gaining user satisfaction. Two machine learning approached were adapted to build default categories: first word approach and most significant word. For each atomic pattern, we generated a default one that holds the first word followed by a wild card to match any text, and then we associated it with the same template. However, this approach was not enough to satisfy Afrikaans users. A frequency list was built out of the corpus, then for each atomic pattern, the least frequent word (most significant) was obtained based on the frequency list to generate a default pattern. Four categories holding most significant word in the first, middle, last positions, or alone were added to the default categories. The feedback showed improvement in user satisfaction [2].

Our next concern was to prove that the Java program we developed is capable of building millions of categories using huge corpora with more than one speaker and involving many domains. For this purpose, the British National Corpus [13] was selected. The BNC is a collection of text samples having more than 100 million words extracted from 4124 modern British English texts of all kinds, both spoken and written. The software was revised to handle the BNC format, and the BNC lemmatized frequency list was used to extract least frequent word. To handle the problem of having a very large AIML file, different chatbot prototypes were built that talk in multi domains: sport, word affairs, travel, media, and food, and represent variety of teenagers speech: Michael, Peter, Robin, loudmouth, etc. As a result, we managed to automatically generate the largest AIML model ever (1,153,129 categories) and to apply the generated chatbots to illustrate the type of English used within a specific domain or speaker-type [5-6].

4.2 Using Monologue-Bilingual Corpora

Can we build a useful chatbot from a monologue corpus where no turns are found? To answer this question we used the holy book of Islam Qur'an. The Qur'an consists of 114 soora (sections) where each soora consists of more than one ayya (verse). Those sooras are grouped in 30 parts (chapters) written in the Arabic Language as delivered to Prophet Mohammad. Muslims used the Qur'an to direct them in every aspect of their life, and they need to memorize it and read it in their prayers. In order to handle non-conversational structure of Qur'an, each ayya is considered as a pattern and the successor one, as a template which could be a useful tool in learning the Qur'an. Two chatbots were created: Arabic Qur'anic chatbot that accepts Arabic input and responds with the Arabic verses; the second one is the Arabic-English chatbot that accepts user input in English and responds with both Arabic-English verse(s). This version could be useful for English speakers who want to learn the Qur'an [3-4].

4.3 Using Structural FAQ/QA Corpora

From the previous corpora used, we found out that machine learning approach works best when the user's conversation with the chatbot is likely to be constrained to a specific topic, and this topic is comprehensively covered in the training corpus. For this purpose we moved toward using a Frequently-Asked Questions (FAQ) corpus.

We began by adapting our chatbot-training program to the FAQ in the School of Computing (SoC) at the University of Leeds, producing the FAQchat system. In this updated version, a question represents a pattern, and the answer represents the template in building atomic AIML files. The frequency list was constructed from questions (patterns).

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

Table 1. Some of ALICE Prototypes Benefits

ALICE chatbot	Purpose
ALICE [30]	Entertainment chatbot
Speak2Me [25]	A web-based version of ALICE aimed at Chinese learners of English, allowing them to practice chatting with a well-spoken young lady, a virtual British English native speaker
AfrikaanaChatbot	A tool to learn/practice a language.
BNCChatbot	A tool to visualize (animate) a corpora.
Arabic_Qura'n chatbot English-Arabic_Qur'an chatbot	A tool to learn Qur'an for Arabic and English speakers.
FAQchat	A tool to access an information portal

Table 2. Some Chatbots and its Usages.

Chatbot Name	Purpose
MIA [24]	a German advisor on opening a bank account
YPA [20-21]	A tool that allows users to retrieve information from British Telecom's Yellow pages
Happy Assistant [14-15]	A shopping assistant that helps users to access e-commerce sites to find relevant information about products and services.
Sofia [19]	A tool that assists in teaching Mathematics
VPbot [22]	A virtual patient chatbot that simulates a patient that medical students can interview
Rita (real time Internet technical assistant) [27]	An eGain graphical avatar that is used in ABN AMRO Bank to help a customer doing some financial tasks such as a wire money transfer [27].

added to handle a different position of the word in a pattern; another category holding first word, 1st or 2nd most significant word as appeared in the original question was generated. At the end, a FAQchat prototype was generated and tested against Google by staff and students of the School of Computing in the University of Leeds.

As a result, 68% of our overall sample of 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 Google [7].

The great 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 a tool to access FAQ databases [8-10].

We managed to demonstrate that a simple ALICE-style chatbot engine could be used as a tool to access the WWW FAQs or QAs. We have observed that there is no need for sophisticated natural language analysis or logical inference; a simple (but large) set of pattern-template matching rules is sufficient.

5 Outputs and Usage

During our journey with chatbots, especially with ALICE, and the developed techniques to retraining them with different corpora, we found out that a chatbot could be used for different purposed not restricted to entertainment issues. Table 1 summarizes some of the different usages of ALICE. Nowadays, a lot of chatbots were generated and used for other purposes; some of them are listed in Table 2.

6 Conclusion

A chatbot is a conversational agent that interacts with users using natural language. This paper overviewed ALICE chatbot in terms of the knowledge base and its pattern matching technique. 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 the ALICE knowledge base. Different corpora were used to retrain ALICE which reveals other useful applications of a chatbot rather than an entertainment tool. A chatbot could be used as a tool to animate or visualize a corpus, to learn/practice English, Arabic, Afrikaans, or other languages, and to access an information portal to provide answers to questions.

We managed to demonstrate that a simple ALICE-style chatbot engine could produce results at least as well-appreciated as those from the most popular commercial web search engine. We did not need a sophisticated natural language analysis or logical inference; a simple (but large) set of pattern-template matching rules was sufficient.

References

1. **Abu Shawar, B. & Atwell, E. (2003a).** Using dialogue corpora to retrain a chatbot system. In **Archer, D., Rayson, P., Wilson, A. and McEnery, T. (eds.),** *Proc. of the Corpus Linguistics 2003 conference (CL2003)*, Lancaster University, UK, pp. 681–690.
2. **Abu Shawar, B. & Atwell, E. (2003b).** Using the Corpus of Spoken Afrikaans to generate an Afrikaans chatbot. *SALALS Journal: Southern African Linguistics and Applied Language Studies*. Vol. 21, pp. 283–294. DOI: 10.2989/16073610 309486349.
3. **Abu Shawar, B. & Atwell, E. (2004a).** An Arabic chatbot giving answers from the Qur'an / Un chatbot arabe qui donne des reponses du Coran. *Proc. of TALN2004: XI Conference sur le Traitement Automatique des Langues Naturelles*, Vol. 2, pp. 197–202.

4. **Abu Shawar, B. & Atwell, E. (2004b).** Accessing an Information system by chatting. In **F. Meziane & E. Metais (Eds.),** *Natural Language Processing and Information Systems: Proc. of NLDB04*, pp. 407–412. Berlin: Springer-Verlag. DOI: 10.1007/978-3-540-27779-8_39.
5. **Abu Shawar, B. & Atwell, E. (2005a).** A chatbot system as a tool to animate a corpus. *ICAME,* 29, pp. 5–24.
6. **Abu Shawar, B. & Atwell, E. (2005b).** Using corpora in machin-learning chatbot systems. *International Journal of Corpus Linguistics,* Vol. 10, No. 4, pp. 489–516. DOI: 10.1075/ijcl.10.4.06sha.
7. **Abu Shawar, B., Atwell, E., & Roberts, A. (2005c).** FAQChat as an Information Retrieval System. In **Vetulani, Z. (ed.)**, *Human Language Technologies as a Challenge. Proceedings of the 2nd Language and Technology Conference*, Wydawnictwo Poznanskie, Poznan, Poland, pp. 274–278.
8. **Abu Shawar, B. (2008).** Chatbots are natural web interface to information portals. *Proceedings of INFOS2008*, pp. NLP101–NLP107.
9. **Abu Shawar, B. & Atwell, E. (2010).** Chatbots: Can they serve as natural language interfaces to QA corpus? *Proc. of the sixth IASTED International Conference Advances in Computer Science and Engineering (ACSE 2010)*, pp. 183–188. DOI: 10.2316/P.2010.689-050.
10. **Abu Shawar, B. (2011).** A Chatbot as a natural web Interface to Arabic web QA. *International Journal of Emerging Technologies in Education (iJET),* Vol. 6, No. 1, pp. 37–43.
11. **Alice (2002).** *A.L.I.C.E AI Foundation*. http://www.Alicebot.org/.
12. **AskJevees (2004).** [Online]: http://ask.co.uk/home
13. **BNC (2002).** *British National Corpus*. http://www.natcorp.ox.ac.uk/.
14. **Chai, J. & Lin, J. (2001).** The role of a natural language conversational interface in online sales: a case study. *International Journal Of Speech Technology,* Vol. 4, pp. 285–295. DOI: 10.1023/A:1011316909641.
15. **Chai, J., Horvath, V., Nicolov, N., Stys-Budzikowska, M., Kambhatla, N., & Zadrozny, W. (2000).** Natural language sales assistant – A web-based dialog system for online sales. *Proc. of thirteenth annual conference on innovative applications of artificial intelligence*.
16. **Chatbot (2015).** [online]: https://www.chatbots.org/.
17. **Colby, K. (1999).** Human-computer conversation in a cognitive therapy program. In **Wilks, Y. (ed.)** *Machine conversations*, Kluwer, pp. 9–19. DOI: 10.1007/978-1-4757-56876_3.
18. **Kerr, B. (1983).** *Minnesota Corpus*. Minneapolis: University of Minnesota Graduate School.
19. **Knill, O., Carlsson, J., Chi, A., & Lezama, M. (2004).** *An artificial intelligence experiment in college math education*.
20. **Kruschwitz, U., De Roeck, A., Scott, P., Steel, S., Turner, R., & Webb, N. (1999).** Natural language access to yellow pages. *Third International conference on knowledge-based intelligent information engineering systems*, pp.34–37.
21. **Kruschwitz, U., De Roeck, A., Scott, P., Steel, S., Turner, R., & Webb, N. (2000).** Extracting semistructured data-lessons learnt. *Proceedings of the 2nd international conference on natural language processing (NLP2000)*, pp. 406–417. DOI: 10.1007/3-540-45154-4_37.
22. **M. Webber G. (2005).** *Data representation and algorithms for biomedical information application*. PhD thesis.
23. **Mann, W. (2002).** *Dialog Diversity Corpus*. [Online]: http://www-rcf.usc.edu/~billmann/diversity/DDivers-site.html.
24. **MIA (2004).** [Online]: http://www.aitools.org/livebots/.
25. **Speak2Me. (2004).** [Online]: www.speak2me.net.
26. **Van Rooy, B. (2003).** *Transkripsiehandleiding van die Korpus Gesproke Afrikaans. (Transcription Manual of the Corpus of Spoken Afrikaans).* Potchefstroom: Potchefstroom University.
27. **Voth, D. (2005).** Practical agents help out. *IEEE intelligent systems,* pp. 4–7.
28. **Weizenbaum, J. (1966).** ELIZA-A computer program for the study of natural language communication between man and machine. *Communications of the ACM,* Vol. 10, No. 8, pp. 36–45. DOI: 10.1145/365153.365168.
29. **Weizenbaum, J. (1967).** Contextual understanding by computers. *Communications of the ACM,* Vol. 10, No. 8, pp. 474–480. DOI: 10.1145/363534.363545.
30. **Wallace, R. (2003).** *The elements of AIML style*. ALICE AI Foundation.
31. **Wallace, R., Tomabechi, H., & Aimless, D. (2003).** *Chatterbots Go Native: Considerations for an eco-system fostering the development of artificial life forms in a human world*.

Bayan Abu Shawar received her PhD in Natural Language Processing from the School of Computing at University of Leeds. Currently she is an associate professor in Information and Computing Department at

Arab Open University in Jordan. Her research interests are: natural language processing, information retrieval, artificial intelligent, e-learning and learning management systems.

Eric Atwell is currently an associate professor in the School of Computing at University of Leeds where he got his PhD from it. His research specialty is Corpus Linguistics and Text Analytics: Machine Learning and Data Mining analysis of a CORPUS of text — in English, Arabic, or other languages — to analyze the text and detect "interesting" and "useful" features or patterns.

Article received on 05/03/2015; accepted 18/06/2015.
Corresponding author is Bayan AbuShawar.