Bayan Abu Shawar, Eric Atwell

Chatbots: Are they Really Useful?

Chatbots are computer programs that interact with users using natural languages. This technology started in the 1960's; the aim was to see if chatbot systems could fool users that they were real humans. However, chatbot systems are not only built to mimic human conversation, and entertain users. In this paper, we investigate other applications where chatbots could be useful such as education, information retrival, business, and e-commerce. A range of chatbots with useful applications, including several based on the ALICE/AIML architecture, are presented in this paper.

Chatbots sind Computerprogramme, die mit Benutzern in natürlicher Sprache kommunizieren. Die ersten Programme gab es in den 60er Jahren; das Ziel war festzustellen, ob Chatbots Benutzer davon überzeugen könnten, dass sie in Wirklichkeit Menschen seien. Chatbots werden aber nicht nur gebaut, um menschliche Kommunikation nachzuahmen und um Benutzer zu unterhalten. In diesem Artikel untersuchen wir andere Anwendungen für Chatbots, zum Beispiel in Bildung, Suchmaschinen, kommerzielle Anwendungen und e-commerce. Wir stellen eine Reihe von Chatbots mit nützlichen Anwendungen vor, einschliesslich mehrerer Chatbots, die auf der ALICE/AIML Architektur basieren.

1 Introduction

> The need of conversational agents has become acute with the widespread use of personal machines with the wish to communicate and the desire of their makers to provide natural language interfaces (Wilks, 1999)

Just as people use language for human communication, people want to use their language to communicate with computers. Zadrozny et al. (2000) agreed that the best way to facilitate Human Computer Interaction (HCI) is by allowing users "to express their interest, wishes, or queries directly and naturally, by speaking, typing, and pointing".

This was the driver behind the development of chatbots. A chatbot system is a software program that interacts with users using natural language. Different terms have been used for a chatbot such as: machine conversation system, virtual agent, dialogue system, and chatterbot. The purpose of a chatbot system is to simulate a human conversation; the chatbot architecture integrates a language model and computational algo-

rithms to emulate informal chat communication between a human user and a computer using natural language.

Initially, developers built and used chatbots for fun, and used simple keyword matching techniques to find a match of a user input, such as ELIZA (Weizenbaum, 1966, 1967). The seventies and eighties, before the arrival of graphical user interfaces, saw rapid growth in text and natural-language interface research, e.g. Cliff and Atwell (1987), Wilensky et al. (1988). 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 AL-ICE (2007). 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 (Braun, 2003).

In this paper, we will present practical chatbot applications, showing that chatbots are found in daily life, such as help desk tools, automatic telephone answering systems, tools to aid in education, business and e-commerce. We begin by discussing the ALICE/AIML chatbot architecture and the pattern matching techniques used within it in section 2; it is easy to build an ALICE-style chatbot, just by supplying a set of chat-patterns in AIML format. Section 3 describes our development of a Java program that can convert a machine readable text (corpus) to the AIML format used by ALICE, allowing different re-trained versions of ALICE to be developed to serve as tools in different domains. Section 4 presents a chatbot as tool of entertainment; a chatbot as a tool to learn and practice a language is discussed is section 5. Section 6 shows a chatbot as an information retrieval tool; using a chatbot in business, e-commerce and other fields is presented in section 7. Our conclusion is presented in section 8.

2 The ALICE Chatbot System

A.L.I.C.E. (Artificial Intelligence Foundation, 2007; Abu Shawar and Atwell, 2003a; Wallace, 2003) is the Artificial Linguistic Internet Computer Entity, which was first implemented by Wallace in 1995. Alice's knowledge about English conversation patterns is stored in AIML files. AIML, or Artificial Intelligence Mark-up Language, is a derivative of Extensible Mark-up Language (XML). It was developed by Wallace and the Alicebot free software community from 1995 onwards to enable people to input dialogue pattern knowledge into chatbots based on the A.L.I.C.E. open-source software technology.

AIML consists of data objects called AIML objects, which are made up of units called topics and categories. The topic is an optional top-level element, 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 matches against the user input, and a template, which

is used in generating the ALICE chatbot answer. The format of AIML is as follows:

```
<aiml version="1.0">
<topic name="the topic">
<category>
<pattern>PATTERN</pattern>
<that>THAT</that>
<template>Template</template>
</category>
        ..
        ..
</topic>
</aiml>
```

The `<that>` tag is optional and means that the current pattern depends on a previous chatbot output.

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.

a. *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'.

b. *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'.

c. *Recursive categories:* are those with templates having `<srai>` and `<sr>` tags, which refer to recursive reduction rules. 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.

c.1 Symbolic reduction

```
<category>
  <pattern>DO YOU KNOW WHAT THE * IS</pattern>
  <template>
    <srai>What is <star/></srai>
  </template>
</category>
```

In this example <srai> is used to reduce the input to simpler form "what is *".

c.2 Divide and conquer

```
<category>
  <pattern>YES*</pattern>
  <template>
    <srai>YES</srai>
    <sr/>
  <template>
</category>
```

The input is partitioned into two parts, "yes" and the second part; * is matched with the `<sr/>` tag. `<sr/>=<srai><star/></srai>`

c.3 Synonyms

```
<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 Pattern Matching Algorithm

Before the matching process starts, a normalization process is applied for each input, to remove all punctuation; the input is split into two or more sentences if appropriate; and converted to uppercase. For example, if input is: "I do not know. Do you, or will you, have a robots.txt file?" Then after the normalization it will be: "DO YOU OR WILL YOU HAVE A ROBOTS DOT TXT FILE".

After the normalisation, the AIML interpreter tries to match word by word to obtain the longest pattern match, as we expect this normally to be the best one. This behaviour can be described in terms of the Graphmaster 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 (Wallace, 2003).

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:

1. If the folder has a subfolder starts with underscore then turn to ,"_/" , scan through it to match all words suffixed X, if no match then:

2. Go back to folder, try to find a subfolder start with word X, if so turn to "X/", scan for matching the tail of X. Patterns are matched. If no match then:

3. 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.

There are more than 50,000 categories in the current public-domain ALICE "brain", slowly built up over several years by the Botmaster, Richard Wallace, the researcher who maintained and edited the database of the original ALICE. 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 re-train ALICE using a corpus based approach.

3 Learning AIML from a Dialogue Corpus Training Dataset

We developed a Java program that converts a text corpus to the AIML chatbot language model format. Two versions of the program 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; Abu Shawar and Atwell, 2003a) to investigate the problems of utilising dialogue corpora. The dialogue corpora contain linguistic annotation that appears during the spoken conversation such as overlapping, and using linguistic fillers. 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 as a template. Removing all punctuation from the patterns and converting it to upper case is done during this phase.

Phase Four: Copy these atomic categories in an AIML file.

For example, assume the DDC corpus has the following sample of XML-tagged text:

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

After applying the text processing module in phase two, the result is:

```
F72PS002: Hello
PS000: Hello Donald
```

The corresponding AIML atomic category can be generated in phase 3:

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

The second version of the program has a more general approach to finding the best match against user input from the training dialogue. Two machine learning category-generation techniques were adapted, the "first word" approach, and the most significant word approach.

In the first word approach we assumed that the first word of an utterance may be a good clue to an appropriate response: if we cannot match the input against a complete corpus utterance, then at least we can try matching just the first word of a corpus utterance. 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.
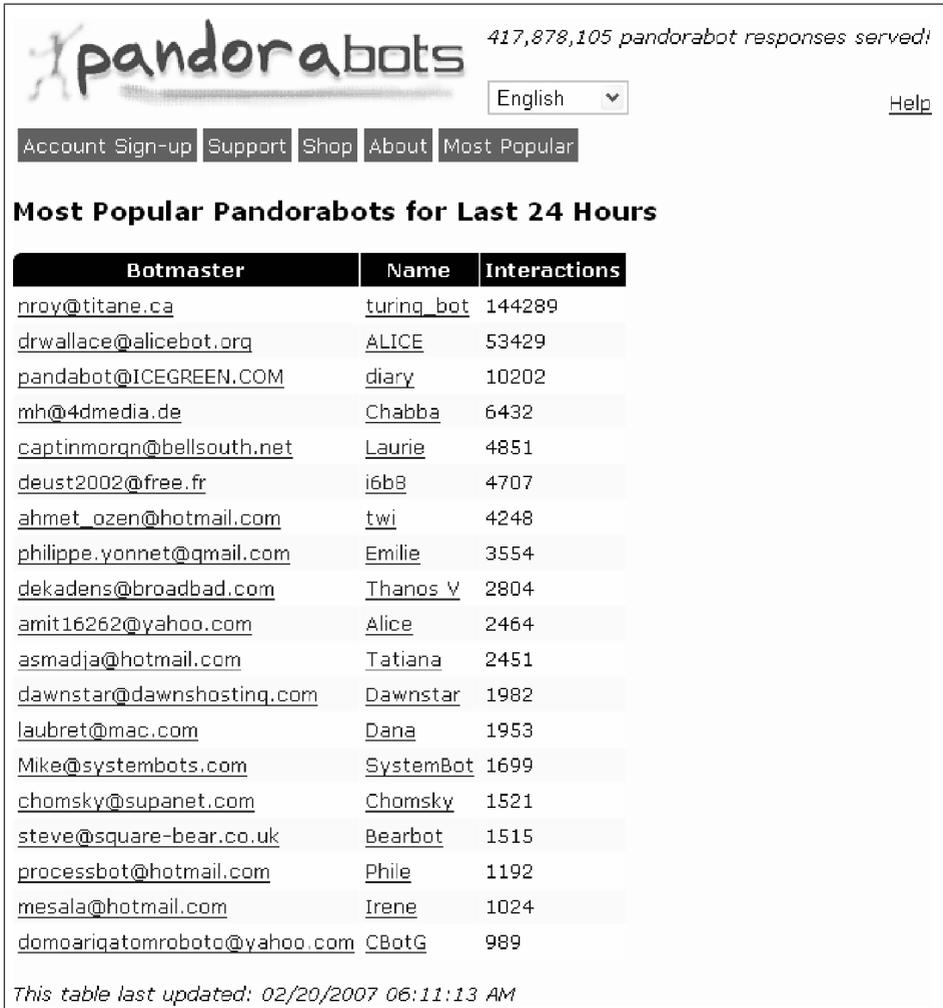
One advantage of the Machine-Learning approach to re-training ALICE is that we can automatically build AIML from a corpus even if we don't understand the domain or even the language; to demonstrate this, the program was tested using the Corpus of Spoken Afrikaans (van Rooy, 2003). Unfortunately this approach still failed to satisfy our trial users, who found some of the responses of the chatbot were inappropriate; so instead of simply assuming that the first word is the best "signpost", 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 chose the most significant approach to generate the default categories, because usually in human dialogues the intent of the speakers is best represented in the least-frequent, highest-information word. We extracted a local least frequent word list from the Afrikaans corpus, and then compared it with each token in each pattern to specify the most significant word within that pattern. Four categories holding the most significant word were added to handle the positions of this word first, middle, last or alone. The feedback showed improvement in user satisfaction (Abu Shawar and Atwell, 2003b).

The same learning techniques were used to re-train different versions of ALICE as will be shown in the following sections. The Pandorabot (2002) web-hosting service was used to publish these prototypes. Pandorabots.com hosts thousands of chatbots built using the AIML format. The most popular Pandorabots for the last 24 hours web-page regularly lists chatbots developed by researchers and hobbyists, and also some commercial systems as shown in figure 1. For example, Cyber-Sandy and Nickie act as portals to adult-entertainment websites; Jenny introduces the English2Go website, and lets English language learners practise their chatting technique. The first Pandorabot chatbots were text-only: the user typed a sentence via keyboard, and then the chatbot reply appeared onscreen as text too. Now some Pandorabot chatbots incorporate speech synthesis; for example, Jenny talks with an educated British accent, via a speech synthesis engine. However, Pandorabot chatbots cannot recognise speech: the user still has to type their input via keyboard. This is because existing Markov-model-based speech recognition is still too error-prone, and does not fit the AIML key-phrase model. Existing speech recognition systems would take a lot of time and memory trying to recognise everything in the input, even though little of this is subsequently needed by the AIML language model; and speech recognition errors may cause inappropriate AIML patterns to be matched (Atwell, 2005).

4 A Chatbot as a Tool of Entertainment

The initial aim of building chatbot systems was to mimic human conversation and amuse users. The first attempt at building chatbots was ELIZA, which was created in the 60's by Joseph Weizenbaum to emulate a psychotherapist in clinical treatment (Weizenbaum, 1966, 1967). 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 mechanically using some fixed phrases if no

Figure 1: Most popular Pandorabots for last 24 hours

keyword match is found such as "Very interesting. Please go on.", or "Can you think of a special example?". Figure 2 shows an example of chatting with ELIZA. When ELIZA was released, at least some users believed that they were talking to a real therapist, and spent hours talking about their problems.

Even though ELIZA was not able to understand, and a user can realise after a while

```
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 2: A sample of conversation with ELIZA (Jurafsky and Martin, 2000)

of chatting that many ELIZA responses are extracted from users input, it was the inspiration for many modern chatbots which aim mainly to fool users that they are talking to another human as applied in the "imitation game" (Turing Test, Turing, 1950).

After ELIZA, a lot of chatbots or human-computer dialogue systems have been developed either to simulate different fictional or real personalities using different algorithms of pattern matching, such as simple keyword-matching in ELIZA, or more linguistically-sophisticated models such as using Markov Models like MegaHAL (Hutchens and Alder, 1998).
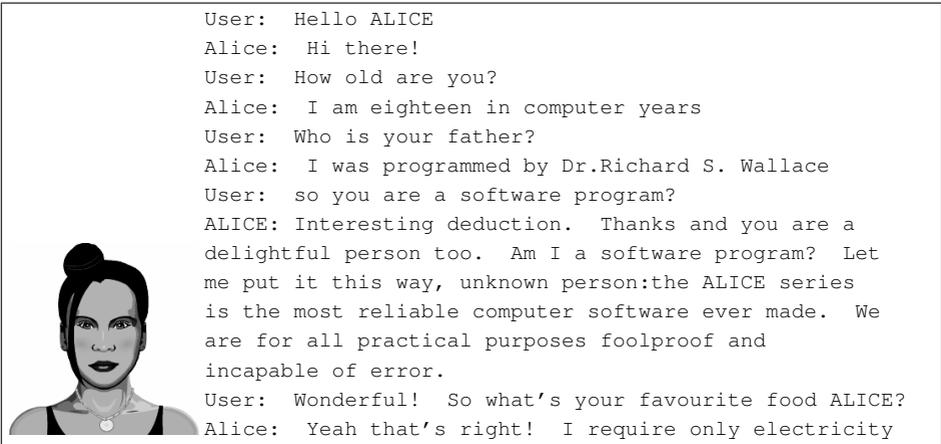
Another example used in this field is ALICE, the chatbot engine we used in our research, which was built basically to entertain users and talk to them as a real person. ALICE won the Loebner prize (2003) competition three times in 2000, 2001, and 2004. The Loebner competition is the way used nowadays to judge how much a chatbot could convince a user that it is a real human by chatting for 10 minutes. Figure 3 shows an example of chatting with ALICE. In fact this conversation seems good, however if you try to chat more, you will probably figure out that:

- ALICE does not save the history of conversation.

- ALICE does not truly understand what you said; it gives you the responses from the knowledge domain stored in her "brain".

These are also the most common drawbacks in almost all chatbots.

5 A Chatbot as a Tool to Learn and Practice a Language

We used our Java program described in section 3, to read a Corpus of Spoken Afrikaans (Korpus Gesproke Afrikaans) (van Rooy, 2003) and to convert it to the AIML format files. Since the corpus does not cover topics like greetings, some manual atomic categories were added for this purpose and the default ones were generated by the program

```
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 favourite food ALICE?
Alice:  Yeah that's right!  I require only electricity
```

Figure 3: A sample of chatting with ALICE

automatically. As a result two Afrikaans chatbots were generated: Afrikaana (2002), which speaks just Afrikaans, and a bilingual version speaking English and Afrikaans, named AVRA (2002); this was inspired by our observation that the Korpus Gesproke Afrikaans actually includes some English, as Afrikaans speakers are generally bilingual and switch between languages comfortably. We mounted prototypes of the chatbots on websites using Pandorabot service, and encouraged open-ended testing and feedback from remote users in South Africa. Unfortunately, users found that many responses were not related to the topic or nonsense. The reasons behind most of the users' feedback can be related to three issues. Firstly the dialogue corpus context does not cover a wide range of domains, so Afrikaana can only "talk about" the domain of the training corpus. Secondly, the repeated approach that we used to solve the problem of determining the pattern and the template in case of more than two speakers may lead to incoherent transcripts: if the training corpus does not have straightforward equivalents of "user" and "chatbot" then it can be non-trivial to model turn-taking correctly in Machine-Learnt AIML (Abu Shawar and Atwell, 2005b). Thirdly, our machine-learnt models have not included linguistic analysis markup, such as grammatical, semantic or dialogue-act annotations (Atwell, 1996; Atwell et al., 2000), as ALICE/AIML makes no use of such linguistic knowledge in generating conversation responses. However, users found it an interesting tool to practise the language and enjoyed chatting, and we concluded that even with its key-word based matching technique, a chatbot could be used as a tool for unknown languages, where "unknown" means (i) unknown to the chatbot author/developer, and/or (ii) unknown to computational linguistics, that is, where there is a shortage of existing tools to deal with the languages.

Jia (2004b) also found that with the absence of linguistic knowledge and the inability to understand users' input, many answers will be nonsense. Moreover, Jia claimed that with this key word matching, a chatbot system could not work as a teaching assistant program in foreign language learning. This was his conclusion from an experiment in China to see if a chatbot could replace a chat partner with users who learn a foreign language. Students from universities and colleges were asked to chat with ALICE, students only knew that they were chatting with a partner to help them learning English. After a short time of chatting, most of students figured out that Alice was an AI agent not a real human. A sample of chatting done in this experiment is shown in figure 4.

```
User:   you are a computer?
Alice:  What if I was a person pretending to be a computer?
User:   I can not believe it
Alice:  Why can't you do believe?
User:   Your answer is not like a human
Alice:  I am still now young and will learn more.
```

Figure 4: A sample of chatting between a student in China and ALICE (from Jia (2004b))

Analysis of all dialogues generated and feedback from students revealed that: 1256 users chatted with Alice, 88% of them chatted only once and did not come back to the site; the duration of chatting was short; 17% made positive comments such as: "you are so nice", "you are clever", etc, and 24% evaluated it negatively. In this respect, Jia concluded that the failure of this experiment is down to the pattern matching technique used in Alice which is based on key-word matching without any attempt to understand what is said. The topics of chatting covered every aspect in our daily life, for example: study, emotion, life, computer, free time, travel/world and job. 11.39% of students talk about English study, and exams, and 13% mentioned love, mostly students younger than 30 years old dealt with Alice as a friend rather than as a teacher, and told her some private emotional problems and experiences. Jia (2004b) concluded that "the conversational chatbot should not only work as a teacher or learning partner with rich special knowledge, but also as a dear friend who may enjoy the joy and suffer the pain of the users". After that Jia (2004a) developed an intelligent Web-Based teaching system for foreign language learning which consists of: natural language mark-up language that labels grammar elements; natural language object model in Java which represents the grammatical elements; natural language database; a communication response mechanism which considers the discourse context, the world model and the personality of the users and of the system itself.

In the same respect, Chantarotwong (2005) reported that "responses of most chatbots are frequently predictable, redundant, lacking in personality, and having no memory of previous responses which could lead to very circular conversation."

However, in contrast to these findings, Fryer and Carpenter (2006) claimed that "chatbots could provide a means of language practice for students anytime and virtually

anywhere". Even though most chatbots are unable to detect spelling errors, and grammar mistakes, they could still be useful for non-beginner students. Fryer and Carpenter did an experiment where 211 students were asked to chat with ALICE and Jabberwocky chatbots. The feedback in general was that students enjoyed using the chatbots, and felt more comfortable and relaxed conversing with the bots than a student partner or teacher as in classical teaching. The authors listed other advantages of chatbots in this domain: the chatbot could repeat the same material with students several times without being bored, many bots used text and speech mode in responding which is an opportunity to practice the reading, and listening skills, and chatbots as new trends improve students motivation towards learning. In addition to this, if computers are available in the class room, teachers could encourage students who finished their class work early to talk to a chatbot and giving them a topic to focus on. An easy self analysis could be achieved since most chatbots keep a transcript of the conversation where students can evaluate themselves.

6 A Chatbot as Information Retrieval Tool

A chatbot could be a useful tool in education, for example to practise language as illustrated in section 5. Knill et al. (2004) found that using a chatbot to answer questions will help the teacher to see where students have problems, what questions students ask, and the generated logs file could be accessed to gauge student learning, and students weaknesses. The authors developed the Sofia chatbot to assist in teaching Mathematics. The Sofia chatbot has the ability to chat with users and at the same time to chat with other mathematical agents such as Pari and Mathmatica to help in solving Algebra problems. The "brain" of the bot contains text files mainly focussing on maths and other common knowledge to make Sophia friendly to use. Sophia was trained with some jokes, and is familiar with movies in which maths plays a role. Sophia was used at Harvard Mathmatics department. Results showed that teachers can use a chatbot to look for problems as students use it to solve problems.

Information Retrieval researchers recognise that techniques to answer questions from document-sets have wide applications, beyond education; see for example the overview of question-answering in restricted domains (Molla and Vicedo, 2007). In a similar application, we used a range of different retrained version of ALICE to retrieve answers for questions in a range of topics (Abu Shawar et al., 2005; Abu Shawar and Atwell, 2005a,c). We adapted the Java program to the FAQ (Frequently Asked Questions) in the School of Computing (SoC) at University of Leeds, producing the FAQchat system. Earlier systems were built to answer questions specifically about the Unix operating system, e.g. Wilensky et al. (1988), Cliff and Atwell (1987); but the SoC FAQ also covers other topics including teaching and research resources, how to book a room, even "what is doughnuts?" (Friday morning staff meeting with an incentive to turn up...) An FAQ has the advantage over other corpus training sets in that there are clear equivalents of "user" (Question) and "chatbot" (Answer) which simplifies modelling of turn-taking (Abu Shawar and Atwell, 2005b). The results returned from FAQchat

are similar to ones generated by search engines such as Google, where the outcomes are links to exact or nearest match web pages. Because of this similarity an interface was built which accepts users input and produce two answers, one is generated from the FAQchat and the other is generated from Google after filtering it to the FAQ of SoC. An evaluation sheet was prepared which contains 15 information-seeking tasks or questions on a range of different topics related to the FAQ database. 21 members of the staff and students tried chatting with the interface as shown in figure 5; the following is a summary of the feedback we obtained:

- Most staff and students preferred using the FAQchat for two main reasons:

    1. The ability to give direct answers sometimes while Google only gives links.
    2. The number of links returned by the FAQchat is less than those returned by Google for some questions, which saves time browsing/searching.

- Users who preferred Google justified their preference for two reasons:

    1. Prior familiarity with using Google.
    2. FAQchat seemed harder to steer with carefully chosen keywords, but more often did well on the first try. This happens because FAQchat gives answers if the keyword matches a significant word. The same will occur if you re-formulate the question and the FAQchat matches the same word. However Google may give different answers in this case.

Feedback favourable to FAQchat was gained from almost all users, even those who preferred Google. Overall, they found it a novel and interesting way to access the FAQ using natural language questions. Overall, about two thirds of users managed to find answers by FAQchat, and about two thirds of the users preferred to use it. The aim was not to try to evaluate the two systems to come up with relative scores, but to show that it is a viable alternative to Google and it could be used as a tool to access FAQ databases. Our system is not restricted to the English language, other FAQ's such as Python FAQ which was found in English and Spanish were also used to generate an FAQ-answering version of ALICE about Python in both English and Spanish languages (Abu Shawar and Atwell, 2005b,a).

Gibbs et al. (2004) used a chatbot conversation to enhance the learning of social theory. They used an ALICE chatbot to build knowledge base that answers question of type "tell me more about. . ." The model was tested by sociology undergraduates studying the natural theory course.

Recently in 2006, Schumaker et al. retrained ALICE with 298 telecommunications-related definitions. The experimental system was assigned to a different section of an introductory Management of Information System course. Evaluations and results show that "the ALICE dialog system is promising as extensions readily come to mind to target both knowledge delivery and acquisition" (Schumaker et al., 2007).

```
a.   Direct answer generated by FAQchat using the FAQ of the school of
     computing at University of Leeds
     Human:   How can I remove a job from the printer?
     FAQchat:
     If you want to remove a job from the printer queue then find out
     the job number using the lpq command, and in a UNIX shell type:
          %lprm -P printername job_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)
b.   Related links generated by FAQchat
     Human:   How can I delete a job from a printer?
     FAQchat:  You may find your answer in the following links:
     How do I check the printer queue?  (Printing)
     How do I remove a job from the printer queue?  (Printing)
     How do I check my printer credits?  (Printing)
     How do I use the printer from a Windows machine?  (Other
     Windows information )
```

Figure 5: Answers generated by FAQchat

Nevertheless, Using a chatbot as an information retrieval system is not only restricted to the education field. The YPA "is a natural language dialogue system that allows users to retrieve information from British Telecom's Yellow pages" (Kruschwitz et al., 1999, 2000). The yellow pages contain advertisements, with the advertiser name, and contact information. The YPA system returns addresses and if no address found, a conversation is started and the system asks users more details in order to give a user the required address. The YPA is composed of Dialog Manager, Natural Language front-end, Query Construction Component, and the Backend database. The Backend includes relational database that contains tables extracted from the Yellow pages. The conversation starts by accepting users' input through a graphical user interface, then the dialogue manager sent the textual input to the Natural Language Fronted for parsing. After that, the parse tree is sent to the Query Constructed Component which translates the input into a database query to query the Backend database and returns the retrieved address. If no addresses found then the dialog manager start putting questions for the user to obtain more clarification. To evaluate the YPA, 75 queries were extracted from a query corpus, and a response sheet was prepared to see if the returned addresses were appropriate or not, how many dialog steps were necessary, the total number of addresses recalled and the number of those relevant to the original query. Results show that 62 out of 75 queries managed to return addresses, and 74% of those addresses were relevant to the original query. The YPA answers questions such as "I need a plumber with an emergency service?", "Which restaurants are there in Colchester high school?"

In another example, Webber (2005) developed a similar chatbot to ALICE called Virtual Patient bot (VPbot). VPbot simulates a patient that medical students can interview. The input is entered as a text by a student, and the patient response is presented as a textual and spoken mode. VPbot was successful in Harvard Medical School's virtual patient program; students who have used the virtual patient scored higher marks on exams (Kerfoot et al., 2006). A sample of chatting with VPbot is shown in figure 6.

```
Student:  Hello.
Patient:  Hello, doctor.
Student:  What brought you into the hospital today?
Patient:  I have a crushing feeling in my chest, and I'm worried it
might be something serious.
Student:  Does it hurt?
Patient:  I have a heaviness in my chest.
Student:  Is it spreading?
Patient:  My chest pain has spread to my left shoulder, neck, and jaw.
Student:  What did you eat last night?
Patient:  I had pizza for supper, which I can usually eat plenty of
without getting sick.
Student:  How often do you drink?
Patient:  I drink beer on weekends but usually not much else.
Student:  Do you have any problems with your heart?
Patient:  I do not have any heart problems that I know of.
Student:  When did the pain start?
Patient:  I have been having the discomfort for about 2 hours.  When it
didn't go away with Tylenol and Motrin, I decided to come in.
Student:  Are you allergic to any medications?
Patient:  I am not allergic to anything that I know of.
```

Figure 6: An example conversation with VPbot (from Webber (2005))

7 Chatbot Assistants in E-Commerce, Business, and other Domains

Shopping assistant is one of the most persuasive sales in traditional commerce. Shopping assistant offer help in a store, provide additional information on products and simplify decision making process helping to find a good that satisfies customer's requirements and various constraints. (Bogdanovych et al., 2005)

Happy Assistant is "a natural language dialog-based navigation system that helps users access e-commerce sites to find relevant information about products and services" (Chai et al., 2000). The system is composed of three main modules: the presenta-

tion manager (PM), the dialog Manager (DM), and the Action Manager (AC). The presentation manager applies a shallow parsing technique to identify semantic and syntactic information of interest from the user textual input. Then it translates the user's input into a well formed XML message called the logical form. The dialogue manager is responsible for matching concepts from user's query to business rules found in the knowledge domain. The business rules consist of a list of concepts together with some meta data about the target product or service. If a match is found, the webpage associated with that rule is presented to the user. Otherwise, the most important missing concept is figured out by introducing questions to the user. The role now is turned to the action manager, which accesses the product that matched the query, and if the user provides special preferences, a sorting algorithm is applied to yield a ranked list of products. To make users trust the system, system must offer some explanation before producing a result, so the system summarizes user's request by paraphrasing it using the context history. Figure 7 presents a sample of conversation with the Happy Assistant System taken from Chai and Lin (2001).

```
U: I am looking for a notebook for my consulting business
S: Please describe your financial constraints.
In this stage a list of most popular ones is displayed
U: not important thing performance is essential
S: are you looking for something that is top of the line?
U: Yes, absolutely
S: I have found something for you.
The target notebook is displayed for the user.  And beneath it a summary
of the users request displayed to explain why this product is displayed.
```

Figure 7: A sample of chatting with the Happy Assistant system

Evaluations and results showed that users found it easy to use, meeting the users' needs, users like the idea that they could express their needs in their language, users feel that the computer did all the work for them, and moreover users found that the system reduced the interaction time.

Sanelma (2003) is a fictional person to talk with in a museum, which provides background information concerning a certain piece of art. Sanelma is a 26 year old woman from Helsinki of the 30's as shown in figure 8.

Rita (real time Internet technical assistant), an eGain graphical avatar, is used in the ABN AMRO Bank to help customer doing some financial tasks such as a wire money transfer (Voth, 2005). If Rita does not understand, it can redirect the customer to another channel such as an e-mail or live chat.

hei, olen Sanelma.

Juttele Sanelman kanssa, kirjoita sanottavasi ja paina enteriä.
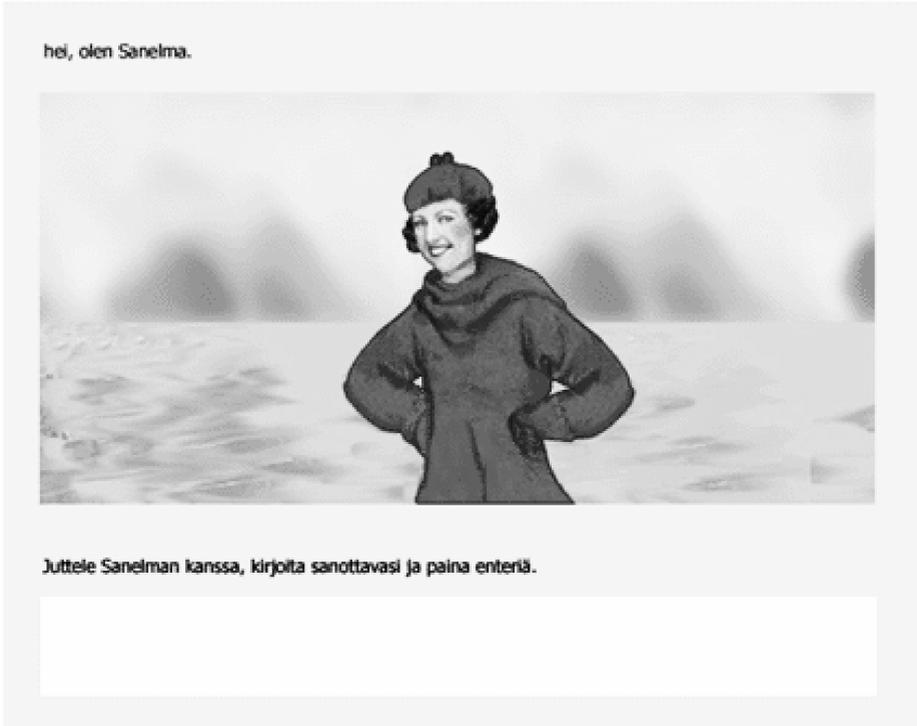
Figure 8: Sanelma chatbot

8 Conclusion

We have surveyed several chatbot systems which succeed in practical domains like education, information retrieval, business, e-commerce, as well as for amusement. In the future, you could "imagine Chatterbots acting as talking books for children, Chatterbots for foreign language instruction, and teaching Chatterbots in general." (Wallace et al., 2003). However, in the education domain Knill et al. (2004) concluded that "the teacher is the backbone in the teaching process. Technology like computer algebra systems, multimedia presentations or 'chatbots' can serve as amplifiers but not replace a good guide". In general, the aim of chatbot designers should be: to build tools that help people, facilitate their work, and their interaction with computers using natural language; but not to replace the human role totally, or imitate human conversation perfectly. Finally, as Colby (1999) states, "We need not take human-human conversation as the gold standard for conversational exchanges. If one had a perfect simulation of

a human conversant, then it would be human-human conversation and not human-computer conversation with its sometimes odd but pertinent properties."

References

Abu Shawar, B. and Atwell, E. (2002). A comparison between alice and elizabeth chatbot systems. Research Report 2002.19, University of Leeds – School of Computing, Leeds.

Abu Shawar, B. and Atwell, E. (2003a). Using dialogue corpora to retrain a chatbot system. In Archer, D., Rayson, P., Wilson, A., and McEnery, T., editors, *Proceedings of the Corpus Linguistics 2003 conference (CL2003). Lancaster University, UK*, pages 681–690.

Abu Shawar, B. and Atwell, E. (2003b). Using the corpus of spoken afrikaans to generate an afrikaans chatbot. *SALALS Journal: Southern African Linguistics and Applied Language Studies*, 21:283–294.

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

Abu Shawar, B. and Atwell, E. (2005b). Die Modellierung von Turn-taking in einem korpus-basierten Chatbot / Modelling turn-taking in a corpus-trained chatbot. In Fisseni, B., Schmitz, H.-C., Schroder, B., and Wagner, P., editors, *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen*, pages 549–561. Peter Lang Verlag, Frankfurt/Main.

Abu Shawar, B. and Atwell, E. (2005c). Using corpora in machine-learning chatbot systems. *International Journal of Corpus Linguistics*, 10:489–516.

Abu Shawar, B., Atwell, E., and Roberts, A. (2005). FAQChat as an information retrieval system. In Vetulani, Z., editor, *Human Language Technologies as a Challenge. Proceedings of the 2nd Language and Technology Conference, Wydawnictwo Poznanskie, Poznan, Poland*, pages 274–278.

Afrikaana (2002). Published online: http://www.pandorabots.com/pandora/talk?botid=eba8f4dc9e3406b8.

Artificial Intelligence Foundation (2007). The A. L. I. C. E. Artificial Intelligence Foundation. Published online: http://www.alicebot.org oder http://alicebot.franz.com/.

Atwell, E. (1996). Comparative evaluation of grammatical annotation models. In Sutcliffe, R., Koch, H.-D., and McElligott, A., editors, *Industrial Parsing of Technical Manuals*, pages 25–46. Rodopi, Amsterdam.

Atwell, E. (2005). Web chatbots: the next generation of speech systems? *European CEO*, November-December:142–144.

Atwell, E., Demetriou, G., Hughes, J., Schiffrin, A., Souter, C., and Wilcock, S. (2000). A comparative evaluation of modern english corpus grammatical annotation schemes. *ICAME Journal*, 24:7–23.

AVRA (2002). Published online: http://www.pandorabots.com/pandora/talk?botid=daf612c52e3406bb.

Batacharia, B., Levy, D., A., R. C., Krotov, and Wilks, Y. (1999). CONVERSE: a conversational companion. In Wilks, Y., editor, *Machine conversations*, pages 205–215. Kluwer, Boston/ Dordrecht/ London.

Bogdanovych, A., Simoff, S., Sierra, C., and Berger, H. (2005). Implicit training of virtual shopping assistants in 3D electronic institutions. In *Proceedings of the IADIS International e-Commerce 2005 Conference, Porto, Portugal, December 15-17*, pages 50–57. IADIS Press.

Braun, A. (2003). *Chatbots in der Kundenkommunikation (Chatbots in customer communication)*. Springer, Berlin.

Chai, J., Horvath, V., Nicolov, N., Stys-Budzikowska, M., Kambhatla, N., and Zadrozny, W. (2000). Natural language sales assistant - a web-based dialog system for online sales. In *Proceedings of thirteenth annual conference on innovative applications of artificial intelligence, 2000*.

Chai, J. and Lin, J. (2001). The role of a natural language conversational interface in online sales: a case study. *International Journal Of Speech Technology*, 4:285–295.

Chantarotwong, B. (2005). The learning chatbot. Final year project. Published online: http://courses.ischool.berkeley.edu/i256/f06/projects/bonniejc.pdf.

Cliff, D. and Atwell, E. (1987). 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*, 19:49–51.

Fryer, L. and Carpenter, R. (2006). Emerging technologies bots as language learning tools. *Language Learning & Technology*, 10(3):8–14.

Gibbs, G., Cameron, C., Kemenade, R., Teal, A., and Phillips, D. (2004). Using a chatbot conversation to enhance the learning of social theory. Published online: http://www.hud.ac.uk/hhs/dbs/psysoc/research/SSCRG/chatbot.htm.

HEXBOT (2004). Hexbot chatbot website. Published online: http://www.hexbot.com/.

Hutchens, J. (1996). How to pass the turing test by cheating. Research Report TR97-05, University of Western Australia – School of Electrical, Electronic and Computer Engineering, Perth.

Hutchens, T. and Alder, M. (1998). Introducing MegaHAL. Published online: http://cnts.uia.ac.be/conll98/pdf/271274hu.pdf.

Jia, J. (2004a). CSIEC (computer simulator in educational communication): An intelligent web-based teaching system for foreign language learning. In Kommers, P. and Richards, G., editors, *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004*, pages 4147–4152, Chesapeake, VA. AACE press.

Jia, J. (2004b). The study of the application of a web-based chatbot system on the teaching of foreign languages. In *Proceedings of the SITE2004 (The 15th annual conference of the Society for Information Technology and Teacher Education)*, pages 1201–1207. AACE press.

Jurafsky, D. and Martin, J. (2000). Introduction. In *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, pages 1–18. Prentice Hall, New Jersey.

Kerfoot, B. P., Baker, H., Jackson, T. L., Hulbert, W. C., Federman, D. D., Oates, R. D., and DeWolf, W. C. (2006). A multi-institutional randomized controlled trial of adjuvant web-based teaching to medical students. *Academic Medicine*, 81(3):224–230.

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

Kruschwitz, U., De Roeck, A., Scott, P., Steel, S., Turner, R., and Webb, N. (1999). Natural language access to yellow pages. In *Third International conference on knowledge-based intelligent information engineering systems*, pages 34–37.

Kruschwitz, U., De Roeck, A., Scott, P., Steel, S., Turner, R., and Webb, N. (2000). Extracting semistructured data-lessons learnt. In *Proceedings of the 2nd international conference on natural language processing (NLP2000)*, pages 406–417.

Loebner, H. (2003). Home page of the loebner prize-the first turing test. Published online: http://www.loebner.net/Prizef/loebner-prize.html.

Mann, W. (2002). Dialog diversity corpus. Published online: http://www-rcf.usc.edu/~billmann/diversity/DDivers-site.htm.

Molla, D. and Vicedo, J. (2007). Question answering in restricted domains: An overview. *Computational Linguistics*, 33(1):41–61.

Pandorabot (2002). Published online: http://www.pandorabots.com/pandora.

Sanelma (2003). Published online: http://www.mlab.uiah.fi/mummi/sanelma/.

Schumaker, R. P., Ginsburg, M., Chen, H., and Liu, Y. (2007). An evaluation of the chat and knowledge delivery components of a low-level dialog system: The AZ-ALICE experiment. *Decision Support Systems*, 42(2):2236–2246.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, 49:433–460.

van Rooy, B. (2003). *Transkripsiehandleiding van die Korpus Gesproke Afrikaans. [Transcription Manual of the Corpus Spoken Afrikaans.]*. Potchefstroom University, Potchefstroom.

Voth, D. (2005). Practical agents help out. *IEEE Intelligent Systems*, 20(2):4–6.

Wallace, R. (2003). *The Elements of AIML Style*. A.L.I.C.E. Artificial Intelligence Foundation, Inc.

Wallace, R., Tomabechi, H., and Aimless, D. (2003). 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.

Webber, G. M. (2005). *Data representation and algorithms for biomedical informatics applications*. PhD thesis, Harvard University.

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

Weizenbaum, J. (1967). Contextual understanding by computers. *Communications of the ACM*, 10(8):474–480.

Wilensky, R., Chin, D., Luria, M., Martin, J., Mayfield, J., and Wu, D. (1988). The berkeley unix consultant project. *Computational Linguistics*, 14(4):35–84.

Wilks, Y. (1999). Preface. In Wilks, Y., editor, *Machine Conversations*, pages vii–x. Kluwer, Boston/Dordrecht/London.

Zadrozny, W., Budzikowska, M., Chai, J., and Kambhatla, N. (2000). Natural language dialogue for personalized interaction. *Communications of the ACM*, 43(8):116–120.