Using corpora in machine-learning chatbot systems

Bayan Abu Shawar and Eric Atwell
University of Leeds

A chatbot is a machine conversation system which interacts with human users via natural conversational language. Software to machine-learn conversational patterns from a transcribed dialogue corpus has been used to generate a range of chatbots speaking various languages and sublanguages including varieties of English, as well as French, Arabic and Afrikaans. This paper presents a program to learn from spoken transcripts of the Dialogue Diversity Corpus of English, the Minnesota French Corpus, the Corpus of Spoken Afrikaans, the Qur'an Arabic-English parallel corpus, and the British National Corpus of English; we discuss the problems which arose during learning and testing. Two main goals were achieved from the automation process. One was the ability to generate different versions of the chatbot in different languages, bringing chatbot technology to languages with few if any NLP resources: the corpus-based learning techniques transferred straightforwardly to develop chatbots for Afrikaans and Qur'anic Arabic. The second achievement was the ability to learn a very large number of categories within a short time, saving effort and errors in doing such work manually: we generated more than one million AIML categories or conversation-rules from the BNC corpus, 20 times the size of existing AIML rule-sets, and probably the biggest AI Knowledge-Base ever.

Keywords: chatbot, dialogue, AIML, Artificial Intelligence, Machine Learning, French, Afrikaans, Arabic, Qur'an, British National Corpus, lemmatised and unlemmatised lists.

1. Introduction

Corpora have been widely used by linguists to develop and refine "language models", descriptions of lexis, grammar, dialogue, etc. Language models can

International Journal of Corpus Linguistics 10:4 (2005), **489–516.**
ISSN 1384–6655 / E-ISSN 1569–9811

also be automatically extracted or machine-learnt from corpora, to drive language analysis systems; for example, machine-learning of Part-of-Speech taggers from PoS-tagged corpora (Atwell 1983; Atwell et al. 2000a); machine-learning to automatically cluster words in a corpus into grammatical classes (Atwell & Drakos 1987; Hughes & Atwell 1994); machine-learnt grammar checkers (Atwell 1983, 1987); machine learning of parsers from parsed corpus treebanks (Atwell 1983, 1988, 1996; Atwell et al. 1984), adapting speech recognition models for English language learners (Atwell et al. 2000b, 2003).

Chatbot technology 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, Cicekli 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." Chatting (1) shows a sample of chatting with ELIZA.

Chatting (1). A sample of conversation with ELIZA (from Jurafsky & Martin 2000)

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.

Since that time, several other chatbots or human-computer dialogue systems have been developed using text communication such as MegaHAL (Hutchens 1996), CONVERSE (Batacharia et al. 1999), ELIZABETH (Abu Shawar & Atwell 2002), HEXBOT (HEXBOT 2004) and ALICE (ALICE 2002). Chatbots have been used in different domains such as: customer service, education, web site help, and for fun.

However, these chatbots are restricted to the linguistic knowledge that is manually "hand-coded" in their files. To save the time and effort of encoding such knowledge and to develop a chatbot that simulates a human dialogue, we developed a Java program to convert a dialogue transcript text corpus to AIML format: Artificial Intelligence Markup Language, the ALICE chatbot rule-format (see Section 2). In order to retrain ALICE, we used a range of corpora to create several different experimental versions of ALICE, speaking different varieties of English, as well as French, Afrikaans, Arabic and bilingual chatbots. This paper illustrates the ability of our program to learn a linguistic knowledge base of more than one million categories or rules, extracted from the British National Corpus (BNC) spoken transcriptions. The approach seemed straightforward at the outset, but we encountered problems and drawbacks; we discuss these and propose potential directions for further research.

The ALICE chatbot engine and its AIML knowledge representation formalism are presented in Section 2. Section 3 outlines our initial attempts to learn AIML files from English, French, Afrikaans and Arabic corpora; we explain how feedback from users of our initial machine-learnt chatbots led us to develop more sophisticated versions of the learning algorithm. Section 4 examines the British National Corpus and the problems which arose when converting the BNC spoken transcripts to the AIML format. The latest version of the AIML-learning program tackles the BNC problems; the necessary modifications are discussed in Section 5. The results and conclusions are in Sections 6 and 7 respectively.

2. The ALICE chatbot engine

A.L.I.C.E. (ALICE 2002; Wallace 2003) is the Artificial Linguistic Internet Computer Entity, first implemented by Wallace in 1995. ALICE 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), developed by Wallace and the Alicebot free software community during 1995–2000 to enable people to input dialogue pattern knowledge into chatbots based on the ALICE 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 units of knowledge in AIML. Each category is a rule for matching input to 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 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, as shown in Section 4.1.4. 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* have patterns that do not have wildcard symbols _ or *, e.g.:

```
<category><pattern>Hello Alice</pattern>
<template>Hi, who are you?</template></category>
```

In the above category, if the user inputs 'Hello Alice', then ALICE answers 'Hi, who are you?'. An atomic category only fires if the human input is an exact word-for-word match for the pattern; this can be used to encode formulaic conversation openers, for example.

b. *Default categories* have patterns including wildcard symbols * or _. The wildcard symbols match any input but they differ in their alphabetical order. Assuming the input 'Hello robot', if this does not match a category with an atomic pattern, then it will try to find a category with a default pattern such as:

```
<category><pattern>Hello *</pattern>
<template>Hi there</template> </category>
```

So ALICE answers 'Hi there'. The wildcard symbol allows the category to match a wider range of possible human inputs.

c. *Recursive categories* have templates including <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 and misspellings 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 human input "Do you know what the * is?" to a simpler form "what is *"; this is then recursively fed back into ALICE as replacement for the original input, allowing other categories to match.

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 (where <sr/> is a short form for <srai><star/></srai>). Each of the two parts is then recursively fed back into ALICE to match against separate categories; then the two separate responses are joined together to produce the ALICE reply.

c.3 *Synonyms and misspellings*

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

The misspelled input 'halo' is mapped to another form 'hello', which has the same meaning and should match against another category when recursively fed back into ALICE.

2.2 ALICE Interpreter

The AIML interpreter tries to find the longest match between the human user input and the stored knowledge. The Interpreter does not have a record of the dialogue so far, so each ALICE response is based solely on the most recent human input. Furthermore, ALICE has no model of the overall goals of the dialogue, or what the user is "on about" beyond the most recent input. This may seem implausibly simple to a linguist; but chatbot developers have found that in practice a lot of conversations have little discourse structure, this "naïve" approach actually works much of the time as long as ALICE has a wide range of categories to respond to a wide range of possible human inputs. There are more than 50,000 categories in the current public-domain ALICE "brain", added by the Botmaster over several years. The AIML format was deliberately kept simple to make it easy to author new rules or categories; other public-domain chatbots such as Elizabeth (Abu Shawar & Atwell 2003a, c) or HEXBOT (HEXBOT 2004) claim to be readily adaptable and extendable, but their more complex rule format makes it much harder to author extensions or new chatbot rules. However, all these categories are manually "hand-coded", which even in AIML is time-consuming, and restricts adaptation to new discourse-domains and new languages. Chatting (2) shows a sample of chatting with the hand coded version of ALICE. In the following sections we will present the automation process we developed, to train ALICE using a corpus based approach.

Chatting (2). A sample of conversation with ALICE (from the AI movie website)

Emily:	My name is Emily
Chatbot:	OK, I will call you Emily
Emily:	Are you really Alice?
Chatbot:	I won't say "yes" or "no" for now. For real
Emily:	Who created you?
Chatbot:	I was created by Dr Richard S. Wallace
Emily:	Do you know the chatbot Alice?
Chatbot:	Is there only one chatbot. Ask the open directory about it
Emily:	Have you seen the movie AI?
Chatbot:	Have you told many people that your favorite movie is a you?
Emily:	Can you explain?
Chatbot:	There is no more information to clarify.
Emily:	Do you know Steven Spielberg?
Chatbot:	Not a robot. Probably a human. Maybe a celebrity. Would you like to know more?
Emily:	I know. I was wondering if you knew.
Chatbot:	How do you know it? I have never considered it. I answer a lot of silly questions.
Emily:	You are actually a guest on the AI website. Do you know that?
Chatbot:	Thanks for telling me that I am actually a guest on the AI website
Emily:	What do you like to talk about?
Chatbot:	My favorite subject is artificial intelligence and robots.
Emily:	What is artificial intelligence?
Chatbot:	Artificial intelligence is the branch of engineering and science devoted to constructing machines that think. Would you like to know more?

3. Learning to chat in a range of languages

We developed Java software to extract AIML categories from a corpus text-file. Different corpora were used (displayed in Table 1) and a range of increasingly sophisticated learnt patterns were examined. The learning techniques range from primitive literal matches to corpus utterances, generating atomic categories which require an exact match to human user input, to more complicated patterns involving identification of the most significant words in an utterance, generating default categories with wildcards (Abu Shawar & Atwell 2003c).

Table 1. The corpora used to train our prototype chatbot-learning program

Corpus	Number of words	Size (KB)
Minnesota French Corpus	25,761	428
Corpus of Spoken Afrikaans	30,793	317
Qur'an in Arabic	85,229	779
Qur'an in English	175,626	955

3.1 Learning from the Dialog Diversity Corpus of English

The first version learnt simple pattern+template categories, where each utterance or turn in the dialogue was taken as a pattern to match the user input, and the subsequent or following utterance became the template for the chatbot answer. The program is composed of four phases: reading the dialogue from the corpus, and inserting it in a vector; applying a text-reprocessing module to remove all unnecessary annotations; passing over the converter module, which considers each turn as a pattern and its successor as a template; And finally saving these categories in an AIML file. This version was tested using samples of the English-language Dialogue Diversity Corpus (DDC) (Mann 2002). The DDC is a collection of links to different dialogue corpora in different fields where each corpus has its own annotation format. These annotated texts are transcribed from recorded dialogues between more than two speakers. Abu Shawar and Atwell (2003a) detail problems encountered, summarised as follows:

a. No standard formats to distinguish between speakers, or for linguistic annotations.
b. Extra-linguistic annotations were used.
c. Long turns and monologues.
d. Irregular turn taking (overlapping).
e. More than two speakers.
f. Scanned text-image not converted to text format.

Unfortunately most of these problems also occur in other corpora, which necessitates changing the filtering process to meet the difference in the corpora format. Figure 1 shows samples of the DDC corpora. The figure illustrates some of the above problems: speaker turns are marked "S1:", "S2:" etc in MICASE, but by more complex XML tags in ICE; MICASE uses XML-like tags for extralinguistic annotations like "<SS LAUGH>" or "<ROTATES CEILING>", these tags must be ignored.

MICASE corpus: Michigan Corpus of Academic Spoken English (Mann 2002) is a 1.8-million-word collection of transcripts of academic speech events recorded at the University of Michigan.

S1: circumpolar stars. So if I keep my pointer there, [S2: oh] <ROTATES CEILING> everything else moves and we all get sick. <SS LAUGH> and we go backwards in time. And that's even more fun.
S2: make it go really really fast.

<SS LAUGH>

S1: okay so that's how the sky is going to move, a couple of other things that we can do in here, um, this is a presentation of, the, grid, that we use to divide the sky, so these lines that run, north south what do we call those?
S3: declination

ICE-Singapore: International Corpus of English, Singapore English (Nelson 2002), has one million words.

<$A>
<ICE-SIN:S1A-099#35:1:A>
Uhm okay lah
<ICE-SIN:S1A-099#36:1:A>
Bearing up lah
<$B>
<ICE-SIN:S1A-099#37:1:B>
Ah hah
<$A>
<ICE-SIN:S1A-099#38:1:A>

Ya I mean I don't really feel comfortable talking about it over the phone so when I see you I'll tell you about it lah

Figure 1. Samples of MICASE and ICE-Singapore subcorpora in the Dialog Diversity Corpus

3.2 Learning from the Minnesota French Dialogue Corpus

A great attraction of the Machine Learning approach is that a learning system used on English corpora should be readily applicable to corpora in other languages. The chatbot-training mechanism does not "understand" the dialogue, it simply treats it as a sequence of character-string-matches, and the character-strings could be in any language. To test this, we applied the same program to the Minnesota French dialogue corpus (Kerr 1983); this required chang-

MINNESOTA CORPUS
SESSION I, TAPE 1, SIDE A

Christine= C.; Martine=M.; Evelyne=E.; unknown speaker=u.s.

C.	Peut-être il faut, un divan oui
M.	Vous allez [faire quoi] ce dimanche
u.s.	[/inaudible/]
M.	J'en avais un à dîner [l'autre fois euh]
E.	[Un divan?] Pourquoi tu vas pas euh, euh tu sais près de, tu connais Como, la, Como Avenue? Bon et là il y a c'est c'est l'Armée du Salut quelque chose et des fois y a de très très [jolies choses]
C.	[ah mm]
E.	et ça par exemple ça dépend ce que tu aimes mais il avait un divan imitation anc- enfin pas an- ancien tu vois avec du bois, le devant ici en bois, et ici vraiment bien et c'est pas cher. Tu devrais de temps en temps y aller. Parce que nous on aller regarder aussi, bon enfin maintenant on a tout ce qui faut mais, euh c'est pas cher du tout et puis c'est pas c'est pas
C.	[mm mm]

Figure 2. Sample of the Minnesota French corpus

ing the pre-processing text since it has its own specific annotations, illustrated in Figure 2. This figure shows that speaker turns are marked differently from MICASE and ICE: a single letter abbreviation of the speaker's name. The figure also shows overlapping is encoded via position of the text: for example, as Martine finishes her second utterance "... l'autre fois euh", Evelyn interrupts with "Un divan? ...". We were able to call on a number of French speakers in our Computer Vision and Language Laboratory to test the French chatbot.

3.3 Learning from the Corpus of Spoken Afrikaans

Our Machine Learning approach should be usable on languages with little or no existing chatbots or other language processing technology; and on languages which we do not speak ourselves, or have ready access to native-speaker informants. The only requirement is a corpus of spoken dialogue in the language in question. Gerhardt van Huysteen and Bertus van Rooy of Potschefstroom University suggested Afrikaans as a suitable language for our next trials, as they were able to give us access to the recently-collected Corpus of Spoken Afrikaans (Van Rooy 2003).

Our revised version of the learning program, Afrikaans.java, has a more general approach to finding the best match against user input from the training dialogue. Two machine learning 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.

The first word approach was tested using the Corpus of Spoken Afrikaans, illustrated in Figure 3. Speaker turns are encoded in yet another way: turns start with an XML "<sprekerN>" tag, and must also end with a matching "</sprekerN>" closing tag. Overlaps are also encoded via XML tags: <oorvleuel> and closing tag "</oorvleuel>". Unfortunately this first word approach still failed to satisfy our trial users, so we looked 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 word 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. The program calculates the Afrikaans corpus word-frequency list, and then a comparison is run against each token in each pattern to find the least frequent word with 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 & Atwell 2003b).

A restructuring module was added in this version to map all patterns with the same response to one form, and to transfer all repeated patterns with different templates to one pattern with a list of alternative responses.

3.4 Learning from the Arabic and Arabic-English Qur'an

This version was updated to generate 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 trans-

<spreker2> is dit (lag) hoe gaan dit met Franna </spreker 2>
<spreker1> Franna </spreker1>
<spreker2> het Franna weer drie gedruk </spreker2>
<spreker1> nee hy't Donderdag twee gedruk <oorvleuel>
<spreker1> en Din∼ </spreker1>
<spreker2> teen wie't hulle </spreker2> </oorvleuel> </spreker1>
<spreker2> gespeel </spreker2>
<spreker1> teen <fil> uh uuhm </fil> Proteapark </spreker1>
<spreker2> *a+ gewen </spreker2>
<spreker1> vyf-en-vyftig nul en Dinsdag het hulle agt-en-tagtig nul gewen </spreker1>

Figure 3. Sample of spoken Afrikaans corpus

mission 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. We assumed that if an input is an ayya, then the reply will be the next ayya. Children often learn the Qur'an in this way: the teacher cites an ayya, and the learner must recite the following ayya. So, our chatbot could be a novel tool to help learn the Qur'an. Two chatbot versions were created: the first accepts Arabic input and responds with the Arabic verse(s) (see Abu Shawar & Atwell 2004a). To help non-Arabic speakers (including one of the authors!) to understand the meaning of the interactions, the second version was retrained with a parallel Arabic-English version of the Qur'an; it also accepts English input and responds with both Arabic and English verse(s) (see Abu Shawar & Atwell 2004b). Figure 4 shows samples of the English and Arabic sooras of the Qur'an.

4. Chatbot-Learning from the British National Corpus

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). We wanted to investigate the possibility of using machine learning to extract a much larger set of AIML files: in theory, the chatbot-learning program can learn millions of categories given an appropriate dialogue corpus. We selected the BNC cor-

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 practise secret arts;
113/5 And from the mischief of the envious one as he practises envy.

(113) قلفـلا ةروس
بِسْمِ اللهِ الرَّحْمَنِ الرَّحِيمِ
{113/1} الفلق قُلْ أعُوذُ برَبِّ
{113/2} مِن شَرِّ مَا خَلَقَ
{113/3} وَمِن شَرِّ غَاسِقٍ إِذَا وَقَبَ
{113/4} وَمِن شَرِّ النَّفَّاثَاتِ فِي الْعُقَدِ
{113/5} وَمِن شَرِّ حَاسِدٍ إِذَا حَسَدَ

Figure 4. Samples of the Arabic and English versions of the Qur'an

pus to train our program, the largest dialogue corpus readily available. Abu Shawar and Awell (2005) present two uses of the BNC: to automatically generate the largest AIML model ever; and to use chatbots trained on specific subsets of the BNC to "animate" or illustrate the type of English used with a specific domain or speaker-type. The British National Corpus (BNC 2002) is a collection of text samples amounting to over 100 million words, extracted from 4124 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. Herring (1996) argues that computer mediated communication (CMC) "is typed, and hence like writing, but exchanges are often rapid and informal, and hence more like spoken conversation", and Grondelaers et al. (2003) describe the language of Internet Relay Chat (IRC) as an example of "spoken language in written form"; so we decided to retrain ALICE using the BNC spoken transcripts.

4.1 The BNC spoken dialogue transcripts

The spoken dialogue transcripts amount to 10 million words, and can be divided into two parts: a demographic part, involving transcriptions of spontaneous natural conversations between families, friends, and so forth, and the context-governed part, containing transcriptions recorded in educational, informative, business, leisure, institutional, and public events (Crowdy 1994). Each corpus file starts with a long Header section, containing details of source, speakers, etc. In the transcript Body, the dialogue consists of a series of utterances or speaker-turns, marked at start and end by <u> and </u> tags. Each utterance tag also includes a speaker number (anonymised, eg F72PS002). Within a text sample, all sentences are tagged <s> and numbered; and each word is preceded with a CLAWS Part-of-Speech tag, e.g. ITJ = interjection, PUN = punctuation-mark, NP0 = singular proper name. An example of a sequence of two utterances is:

```
<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>
```

Stripped of XML markup, this is simply an opening to a conversation:

F72PS002: Hello
PS000: Hello Donald

The corresponding AIML atomic category can be generated:

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

However, the translation process from BNC format to AIML is not as simple as it might seem to be on the surface. A range of problems emerged during the translation process, which will be discussed in the following subsections.

4.1.1 *More than two speakers*

Since the number of participants in chatbot dialogue is two, the user and the program, a dialogue corpus recorded between two parties would be most appropriate; we could then train the chatbot to mimic the part of one or other of the participants. However, the BNC recorded conversations covered a wide range of domains and often involved more than two speakers. When there are several dialogue participants, we cannot simply identify one participant as taking the place of the chatbot, so we cannot just follow one speaker in training. Instead, we assume that every utterance, by any speaker, is a candidate pattern, and the subsequent utterance, regardless of speaker, is the corresponding template. The resultant AIML merges the contributions of all speakers.

4.1.2 *Unclear sections in utterances*

Since all spoken samples are transcribed from recorded speech, some parts of the utterances are unclear. The BNC uses the <unclear> tag to mark these, as in the following example:

```
<u who=PS000>
<s n="5"><unclear> <w AT0>a <w NN1>minute<c PUN>.
</u>
<u who=PS100 ><unclear ></u>
<u who=F72PS000>
<s n="6"><w CJC>And <w DTQ>what <w VBB>are <w PNP>they<c
PUN>?
</u>
```

Stripped of XML markup, this becomes:

> PS000: ??? a minute
> PS100: ???
> F72PS000: And what are they?

A problem with the unclear turn is that it might be a response to a previous utterance, or it might introduce a new idea, which the next speaker responds to. In the translation to AIML we cannot decide if the unclear turn is a pattern or a template. To solve this problem we decided to remove the unclear turns. There are two approaches to elimination, either before or after the converter module maps pairs of successive utterances into pattern+template categories. The difference between them is as follows.

Assume that there are four speakers denoted by (spk) and the sequence of turns is: spk1→ spk2→ unclear→ spk3→ spk4.

The first approach is to omit the unclear turn itself before converting the transcript. In this case we will have the following sequence of utterances: spk1→ spk2→ spk3→ spk4. The conversion process will generate three categories: (spk1→ spk2), (spk2→ spk3), (spk3→ spk4).

The second approach is to omit the unclear after the conversion. After considering each pair as a pattern and a template, we have: (spk1, spk2), (spk2, unclear), (unclear, spk3), (spk3, spk4). Then any pair containing the unclear is excluded, so we will have two categories left.

The second approach completely sidesteps the problem by deleting the two cases where the unclear is a pattern or a template, and this means avoiding the category where spk3 is a response to spk2, which did not actually happen during the conversation. However, it is arguably possible to consider spk3 as a possible response to spk2, even if it did not really happen in this sequence, as at least the utterance is still a continuation of the conversation. As our goal was to generate a large set of categories automatically, we decided to adopt the first approach.

4.1.3 *Overlapping utterances*

Overlapping represents the case where more than one speaker was active at the same time; this occurs during human conversation but not during chatbot interactions. The BNC corpus transcribed the overlapping turns using an alignment map tag <align> to synchronise points within a spoken text, declared at the start of the division or text concerned; and the pointer tag <ptr target> points to the identifier which was synchronised. The following example illustrates this problem:

<u who=w0014>
<s n=00011><w AJ0>Poor <w AJ0>old <w NP0>Luxembourg'<w VBZ>s
<w AJ0-VVN>beaten<c PUN>.
<s n=00012><w PNP>You <w PNP>you<w VHB>'ve <w PNP>you<w
VHB>'ve
<w AV0>absolutely <w AV0>just<w VVN>gone <w AV0>straight
<ptr target=P1> <w PRP>over <w PNP>it <ptr target=P2> </u>
<u who=w0001>
<s n=00013><ptr target=P1> <w PNP>I <w VHB>haven<w XX0>'t<c
PUN>.
<ptr target=P2/> </u>
<u who=w0014>
<s n=00014><w CJC>and <w VVN>forgotten <w AT0>the <w AJ0>poor
<w AJ0>little<w NN1>country<c PUN>. </u>

The equivalent in a more human-readable format is:

W0014: Poor old Luxembourg's beaten. You, you've, you've absolutely just gone straight over it ...
W0001: (interrupting) I haven't.
W0014: . . . and forgotten the poor little country.

The overlapping interruption problem is similar to the unclear section problem: both could impinge on the dialogue and affect what was said afterwards, but both require special handling to map onto pattern+template pairs. In our earlier system, learning from French and Afrikaans corpora, we simply ignored the all utterances involved, i.e. the interruption and also the interrupted utterances (the Qur'an does not have overlaps and interruptions, avoiding this problem for our Arabic chatbot). This meant we lost potential categories; the above example would just be skipped. For the BNC learning model, since the overlapping turn is separated, we treat it as a new turn; this gives us two pattern+template categories for the above example.

4.1.4 *Using abbreviations*

The encoders used some enclitics in writing the recorded speech as: "I'd", "he'll", "John's", and so on. A problem arises in converting such abbreviations to the AIML patterns. We have to remove all punctuations from the pattern to be accepted by the ALICE interpreter. To date our machine-learnt models have not included linguistic analysis markup, such as grammatical, semantic or dialogue-act annotations (Atwell 1996; Atwell et al. 2000a, b), as ALICE/AIML makes no use of such linguistic annotations in generating conversation responses. It cannot distinguish if "s" is an abbreviation of "is" or "has" or a possessive. We decided to remove all punctuations without expanding the enclitic. Even though a sentence such as "I'd like" will be mapped into "I d like", this is still compatible with our approach of the most significant word, since whether "d" denotes had or would, it will not be the most significant word in the sentence. The "n't" abbreviation was the only one replaced by "not", so "haven't" becomes "have not" instead of being "haven t" which is a different word with different meaning, which might be erroneously selected as the most significant word.

4.1.5 *Using character entity references*

Some transcripts included foreign words including accented charaters, encoded using HTML character entity references, such as "à", "Ö", and so on. Unfortunately these non-standard letters raised problems during compilation of the AIML files, and furthermore could not match input from a UK English keyboard. So, all entity references were replaced with the corresponding unaccented characters. For example: "à" is mapped to "a", and "Ö" to "O".

4.1.6 *Linguistic annotations*

The spoken transcripts include markup of paralinguistic phenomena such as: voice quality (whispering, laughing, etc.), non-verbal but vocalised sounds (coughs, humming noises), non-verbal and non-vocal events (animal noises, passing lorries), significant pauses (silence) and speech management phenomena (truncation, false starts). These phenomena might be of interest for other purposes, but the auditory features will not occur when chatting with a computer via keyboard and screen text. So we removed all linguistic annotations including the POS tags. For example:

<u who=PS21K>
<s n="37"><w CRD>forty <w NN0>percent <w PRF>of <w DPS>her <w NN1>time
<w CJS>because <w PNP>she <w VDZ>does <w PNP>it <w AV0>so <w AV0>quickly <vocal desc=laugh> <w CJC>but <w UNC>er <w ITJ>oh </u>

The utterance stripped of XML markup will be:

PS21K: forty percent of her time because she does it so quickly but er oh

4.1.7 *Long monologues*

We followed the BNC partitioning into utterances, even though sometimes the transcribers marked an utterance as running over several sentence-boundaries. For example:

<u who=F72PS000>
<s n="29"><w PNP>You <w VDB>do<c PUN>?
<s n="30"><w AV0>Well <w PNP>you <w VBB>are <w AV0>very <w AJ0>fortunate <w NN0>people<c PUN>.
<s n="31"><w CJC>But <w PNI>none <w PRF>of <w PNP>you <w VM0>will <w VVI>know <w DPS>my <w NN1>friend <w AV0>over here <w DTQ>whose <w NN1>name <w VBZ>is <w NP0>Donald<c PUN>. </u>

Stripped of XML markup, this is equivalent to:

F72PS000: You do? Well you are very fortunate people. But none of you will know my friend over here whose name is Donald.

The program merges all of the sentences to form one string starting with <u who..> and ends with </u>. This generates a very long turn; this is not normally found in computer-human chatting. The alternative would be to artificially treat each <s> as a separate turn, splitting the above into 3 pseudo-utterances; but we decided that, as our aim is to investigate the utility of a corpus for machine-learning, we should follow the boundaries set out in the corpus rather than reinterpret them.

4.2 Adapting the learning software to the BNC

We modified the Afrikaans.java system to cope with the BNC samples:

1. Using the lemmatised BNC frequency list (Kilgarriff 1996) in extracting the least frequent words.
2. Modifying the algorithms to handle the BNC-specific annotations and problems discussed above.
3. The large AIML file learnt from the BNC proved too big for the default ALICE engine to handle, so we had to find a work-around.

4.2.1 *The BNC frequency list*

"A central fact about a word is how common it is. The more common it is, the more important it is to know it." (Kilgarriff 1996: 135). Kilgarriff argues that language learners should be taught the commonest words first, so they understand them and know how to use them. Kilgarriff echoed Zipf's observation that the most common words dominate real use.

Kilgarriff extracted two word-frequency lists from the BNC, the lemmatised and unlemmatised list. The lemmatised frequency list includes 6,318 words with more than 800 occurrences in the whole 100-million-word BNC. The frequency of verbal words and its nominal are generated separately, where the count of the verb is the sum of counts of all instances for each verbal, so the frequency of verbal 'aim' will count 'aims', 'aiming', and 'aimed'. In contrast, the unlemmatised list counts the frequency for each verb-form separately. The unlemmatised list gives the frequency, the word, the PoS, and finally the number of files the word occurs in, as illustrated below:

6187267	*the*	at0	4120
2941444	*of*	prf	4108
2682863	*and*	cjc	4120
2126369	*a*	at0	4113
1812609	*in*	prp	4109

The program starts by reading the frequency list and mapping it into a vector named "bnc_freq". The next step is to read the file name from the index and adding all utterances into a vector named "dialogue". Now the same phases of version (2) are used as follows: the dialogue vector elements are filtered, reiterated, and prepared to originate pattern and template sequentially. Then the dialogue vector is re-structured where all different patterns with the same template are categorised as <srai> categories, and all different templates related to the same pattern are grouped as an atomic category with random list. After that all categories are copied into an AIML file. Finally the process is repeated again by accessing the index and selecting the next transcript to be read.

The reading process involves two aspects:

1. Extracting the word and its frequency, disregarding the POS and the number of files in which the word occurs.
2. Ignoring numbers and any non-orthographic words such as "in-spite-of"; non-orthographic words will not be found in the AIML pattern, especially after removing all punctuations.

The extracted pair <word, frequency> is inserted into the "bnc_freq" vector. The vector will be used later on to obtain the frequency of each token in the pattern.

Some BNC spoken tokens were not found in the unlemmatised list, such as "huhuhuhu".; in such cases the token itself is considered as the least frequent word. Since the BNC spoken transcripts are annotated with part-of-speech tags, we used these tags to filter the meaningful words to be used as the first word or least frequent words: *wh*-question-words, prepositions, and pronouns are not considered. This modification improves the matching process and we record better user satisfaction than before.

4.2.2 *Text normalization for BNC files*

The BNC-specific format used to annotate dialogues required changes in the filtering process, including removal of unnecessary linguistic annotations. We modified the normalisation module as follows:

1. Removing the unclear turns.
2. Deeming the overlapping turns as separate ones. The overlap is referenced as an individual turn in the BNC corpus, and since we want to maximise the number of categories, we consider it as a turn rather than eliminating it as in earlier versions of the program.
3. Replacing enclitics and abbreviations, e.g. "n't" with " not".
4. Replacing the character entity references with normal alphabetic characters.

The preparation phase began by considering the first element in the vector as a pattern and the second as a template. After removing all punctuation from the pattern, the first word of each pattern is used to create a new default category holding the first word followed by star, which represents the first word approach. After that, the pattern is tokenised, and the "bnc_freq" vector generated in module one is scanned to extract the frequency for each token in the pattern. The generated list is sorted by frequency in ascending order, and the first token is considered the most significant word (least frequent one). The process continues by generating four categories: atomic category holding the least frequent word only, and another three default categories holding the least frequent word in the first, middle, and last of the sentence. Then the restructuring phase is executed and the final categories are written to an AIML file.

4.2.3 *The problems in scaling up ALICE to very large AIML files*

During the program run, the execution terminated many times due to an "out of memory storage" problem. This problem related to the large size of some files, around 2MB. To solve this problem, the large files were distributed into several smaller files. This allowed us to load up the BNC-learnt AIML into an ALICE chatbot hosted at the Padorabot website; but this proved to be only a temporary solution, see below.

4.3 Results and evaluation

After nearly ten days of running the program, 1,153,129 categories were generated. This number is 20 times bigger than any existing chatbot: the large public-domain AIML category-file developed over several years (Wallace 2003) contains about 50,000 categories.

We tried to create a new version of ALICE using the Pandorabot web-hosting service (Pandorabot 2003) to make our chatbot available for use over the World Wide Web. After the successful compilation, and a little chatting, we received a message from the Pandorabot that the big size of our dialogue model was causing major problems for the server and to other user chatbots; so we had to take the BNC version offline again.

The evaluation of the AIML-learning system can be tackled into three directions:

1. Dialogue efficiency in terms of matching type.
2. Dialogue quality metrics based on response type.
3. Users' satisfaction assessment based on an open-ended request for feedback.

From the dialogue efficiency and quality we aim to measure the success of the first and most significant word approaches. We had previously measured the quality and efficiency of the Afrikaans dialogue generated between users and Afrikaans chatbot version (for more details see Abu Shawar & Atwell 2003b); we concluded that the most significant word approach increases the ability of finding answers to general user input. In order to measure the quality of each response, we classified the responses into three types: reasonable, weird but reasonable, or nonsensical. A human Afrikaans speaker and teacher evaluated the dialogue; for this evaluator, it seemed that "nonsensical" responses are more likely than reasonable or understandable but weird answers.

User satisfaction plays a critical point in evaluation; following feedback from users who tried chatting with the Afrikaans version, we expanded the AIML pattern matching using the first and least-frequent word approaches.

Using the BNC version, we relied on user satisfaction to evaluate the system. The first impressions from users were disappointing; Chatting (3) is a sample of dialogue between a user and the chatbot where we tried to analyse the reasons behind users' disappointment.

Chatting (3). A sample of chatting with the BNC chatbot version

User:	Hi
Chatbot:	good morning. Now young lady what can I do for you today?
User:	Let's talk.
Chatbot:	Yeah
User:	Do you like dancing?

Chatbot:	No. I ve. I ve just. All I ve done is because Erm because that is a function of that.
User:	Tell me about dancing:
Chatbot:	The dances. Oh they were just great fun the dances all met in the school. It was the nearest to us and the next one was the Glen Ayloch school. But oh we d we d a lot of fiddlers in those days and a girl at the Piano. And then when things got a wee bitty better we used to have the Camerons up from Kilray from Kerry to dance to the. Play at the dances. And of course the night of the. Gathering Ball it was just packed tight with people hardly, the school would hardly hold them all. And then they put up a big tent in the. Beside the school where you went out for supper er it was that S s.
User:	You had a dancing party at your school?
Chatbot:	That s good.
User:	I like swimming
Chatbot:	Mhm.
User:	Do you like swimming?
Chatbot:	No. I ve. I ve just. All I ve done is because Erm because that is a function of that.
User:	Talk to you later.
Chatbot:	Oh. It is still fine.
User:	Bye bye
Chatbot:	Right cheerio now.

A careful look at this example dialogue provides us with an insight into some remaining problems with the pattern matching process. Some responses seem meaningless, because the response is taken from a specific BNC text and this chat is not in the same context. Other responses show that we need to improve our pattern-extraction process. For example, when the user input starts with "do you like...", the same response will be generated regardless of the object. This is because it matches the first word approach before the most significant one. To solve these problems we improved the first word approach by considering the first word scanned in the pattern, which is not a question, a pronoun, or a preposition. The first "meaningful" word in the utterance will be considered instead; also the least "meaningful" word will be considered as least frequent word. To concentrate on certain contexts, we extracted specific domains from

Table 2. BNC subdomain specialist versions

Chatbot	Atomic		Default		Total
	Categories	File size (KB)	Categories	File size (KB)	Categories
Sport	3283	974	7913	1,820	11196
World affairs	3120	983	8756	1,886	11876
Travel	640	314	1636	575	2276
Media	1061	491	3126	1,210	4187
Food	503	93	1125	168	1628

Table 3. The BNC-trained London Teenager and Loudmouth chatbots

Chatbot version	Atomic		Default		Total
	Categories	File size (KB)	Categories	File size (KB)	Categories
Michael	7021	1,044	14914	1,905	21935
Jessica	4261	787	10069	1,554	14330
Peter	5912	939	12878	1,771	18790
Robin	4726	673	14580	1,774	19306
Samantha	6058	788	18283	1,695	24341
Loudmouth1	8114	1,301	17827	2,368	25941
Loudmouth2	8383	1,217	16971	2,130	25354

the BNC and used them to develop different versions of the chatbot as shown in Table 2.

Table 3 lists another five chatbots generated from London teenager transcripts, trained to talk like London teenagers; and two "Loudmouth" chatbots, trained with "interesting" BNC files containing exemplary swearing, as suggested by participants in the CORPORA e-mail discussion list.

Chatting (4) illustrates the dialogue between a user and the Peter chatbot.

Chatting (4). A sample of dialogue with Peter chatbot version

Human: hello
Peter: Hello, hello see it sounds sort of like like you're listening to the tape, but it
Human: do you like music?
Peter: Oh are you going home or are you gonna be back?
Human: I'm going home, what about you?
Peter: Oh well they're bloody playing football again are not they, fucking poofters No, I bet they're playing football with the dikes

Human:	are you going to watch a football game?
Peter:	Are you sure?
Human:	I'm asking?
Peter:	Erm
Human:	what about football?
Peter:	Fuck off
Human:	This is rude.
Peter:	Yes of course yes Right

5. Conclusions

Software capable of imitation of human communication behaviour is a fundamental goal of Natural Language Processing (NLP) research. This goal has led researchers to develop machine conversations systems, or chatbots. However, most chatbots require linguistic knowledge to be hand coded, and are restricted to the language spoken by the developer. We developed a java program that uses a corpus-based approach to train ALICE. The program was used to create different versions of ALICE speaking varieties of English, French, Afrikaans, and Arabic languages. We modified the program to use the British National Corpus spoken transcripts. We were able to develop two learning techniques, the first word and the most significant word approaches, which were successful in learning 1,153,129 categories extracted from the BNC corpus. Two goals were achieved from the automation process: the possibility of generating different versions in different languages, bringing chatbot technology to languages with few if any NLP resources; and the ability to learn a very large number of rules (categories) within a short time, saving effort and errors in doing such work manually. The conversation rules are automatically derived from text, without need of mark-up or linguistic tagging; for example the PoS tagging or sociolinguistic speaker information in BNC files was not needed or used. This means chatbots can be derived from any dialogue transcripts, even untagged corpora. Our Afrikaans chatbot has been acknowledged at Potscheftsroom University as a groundbreaking example of emerging Afrikaans NLP technology; and our BNC-trained chatbot has learnt a set of rules which is larger than any other NLP knowledge base, and is probably the largest AI rule-based system ever.

It is hard to evaluate "accuracy" or "relevance" of chatbot responses, since there is no simple automated metric of "relevance". All our chatbots were made available on the Padorabots.com website for public access and testing, and we elicited some subjective feedback from users. The Afrikaans and Arabic Qur'an chatbots drew mainly favourable feedback; but the London Teenager and Loudmouth chatbots seemed to impress less, users found some responses not just rude but incoherent. Perhaps one lesson is that corpus-trained chatbots should be seen to be "useful" to be appreciated.

References

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

Abu Shawar, B. & Atwell, E. (2004a). An Arabic chatbot giving answers from the Qur'an / *Un chatbot arabe qui donne des reponses du Coran*. In B. Bel & I. Marlien (Eds.), *Proceedings of TALN2004: XI Conference sur le Traitement Automatique des Langues Naturelles* (Volume 2, pp. 197–202). ATALA.

Abu Shawar, B. & Atwell, E. (2004b). Accessing an Information system by chatting. In F. Meziane & E. Metais (Eds.), *Natural Language Processing and Information Systems: Proceedings of NLDB04* (pp. 407–412). Berlin: Springer-Verlag.

Abu Shawar, B. & Atwell, E. (2003a). Using dialogue corpora to retrain a chatbot system. In D. Archer, P. Rayson, A. Wilson & T. McEnery (Eds.), *Proceedings of the Corpus Linguistics 2003 conference (CL2003)* (pp. 681–690), UCREL Technical Paper 16. Lancaster: Lancaster University.

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, 21*, 283–294.

Abu Shawar, B. & Atwell, E. (2003c). Machine Learning from dialogue corpora to generate chatbots. *Expert Update Journal, 6* (3), 25–30.

Abu Shawar, B. & Atwell, E. (2002). *A comparison between Alice and Elizabeth chatbot systems*. School of Computing research report 2002.19. Leeds: University of Leeds.

ALICE (2002). *A.L.I.C.E AI Foundation website* (http://www.Alicebot.org/ or http://Alicebot.franz.com/)

Atwell, E. (1996). Machine Learning from corpus resources for speech and handwriting recognition. In J. Thomas & M. Short (Eds.), *Using corpora for language research: Studies in the honour of Geoffrey Leech* (pp. 151–166). Harlow: Longman.

Atwell, E. (1988). Transforming a Parsed Corpus into a Corpus Parser. In M. Kytö, O. Ihalainen & M. Risanen (Eds.), *Corpus Linguistics, Hard and Soft: Proceedings of the ICAME 8th International Conference on English Language Research on Computerised Corpora* (pp. 61–70). Amsterdam: Rodopi.

Atwell, E. (1987). How to detect grammatical errors in a text without parsing it. In B. Maegaard (Ed.), *Proceedings of EACL: the Third Conference of European Chapter of the Association for Computational Linguistics* (pp. 38–45). New Jersey: ACL.

Atwell, E. (1983). Constituent Likelihood Grammar. *ICAME Journal, 7*, 34–67.

Atwell, E. & Drakos, N. (1987). Pattern Recognition Applied to the Acquisition of a Grammatical Classification System from Unrestricted English Text. In B. Maegaard (Ed.), *Proceedings of EACL: The Third Conference of European Chapter of the Association for Computational Linguistics* (pp. 46–54). New Jersey: ACL.

Atwell, E., Howarth, P. & Souter, C. (2003). The ISLE corpus: Italian and German spoken learner's English. *ICAME Journal, 27*, 5–18.

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

Atwell, E., Howarth, P., Souter, C., Baldo, P., Bisiani, R., Pezzotta, D., Bonaventura, P., Menzel, W., Herron, D., Morton, R. & Schmidt, J. (2000b). User-Guided System Development in Interactive Spoken Language Education. *Natural Language Engineering journal: Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering, 6* (3–4), 229–241.

Atwell, E., Leech, G. & Garside, R. (1984). Analysis of the LOB Corpus: progress and prospects. In J. Aarts & W. Meijs (Eds.), *Corpus Linguistics: Proceedings of the ICAME 4th International Conference on the Use of Computer Corpora in English Language Research* (pp. 40–52). Amsterdam: Rodopi.

Batacharia, B., Levy, D., Catizone, R., Krotov, A. & Wilks, Y. (1999). CONVERSE: a conversational companion. In Y. Wilks (Ed.), *Machine conversations* (pp. 205–215). Boston/Dordrecht/London: Kluwer.

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

Colby, K. (1999). Human-computer conversation in a cognitive therapy program. In Y. Wilks (Ed.), *Machine conversations* (pp. 9–19). Boston/Dordrecht/London: Kluwer.

Colby, K. (1973). Simulation of belief systems. In R. Schank & K. Colby (Eds.), *Computer models of thought and language* (pp. 251–286). San Francisco: Freeman.

Crowdy, S. (1994). Spoken corpus transcription. *Literary and Linguistic Computing, 9* (1), 25–28.

Grondelaers, S., Speelman, D. & Geeraerts, D. (2003). A corpus-based approach to informality: The case of Internet chat. In D. Archer, P. Rayson, A. Wilson & T. McEnery (Eds.), *Proceedings of the Corpus Linguistics 2003 conference (CL2003)* (p. 264). UCREL Technical Paper 16. Lancaster: Lancaster University.

Herring, S. (1996). Introduction. In S. Herring (Ed.), *Computer-mediated communication: linguistics, social and cross-cultural perspectives* (pp. 1–10). Amsterdam: John Benjamins.

HEXBOT (2004). *HEXBOT chatbot website*. (http://www.hexbot.com/)

Hughes, J. & Atwell, E. (1994). The automated evaluation of inferred word classifications. In A. Cohn (Ed.), *Proceedings of ECAI'94: 11th European Conference on Artificial Intelligence* (pp. 535–540). Chichester: John Wiley.

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

Jurafsky, D. & Martin, J. (2000). *Speech and Language Processing*. Prentice Hall.

Kerr, B. (1983). *Minnesota Corpus*. Minneapolis: University of Minnesota Graduate School.

Kilgarriff, A. (1996). Putting Frequencies in the Dictionary. *International Journal of Lexicography, 10* (2), 135–155.

Mann, W. (2002). *Dialog Diversity Corpus website.* (http://www-rcf.usc.edu/∼billmann/diversity/DDivers-site.htm

Nelson, G. (2002). *International Corpus of English: The Singapore Corpus user manual.* (http://www-rcf.usc.edu/∼billmann/diversity/ICE-SIN_Manual.PDF)

Pandorabot (2003). *Pandorabot chatbot hosting website.* (http://www.pandorabots.com/pandora)

Saygin, A., Cicekli, I. & Akman, V. (2000). Turing test: 50 years later. *Minds and Machines, 10* (4), 463–518.

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

Wallace, R. (2003). *The elements of AIML style.* ALICE AI Foundation.

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

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.

Author's address

Bayan Abu Shawar and Eric Atwell
School of Computing, University of Leeds
Leeds LS2 9JT, England
bshawar@comp.leeds.ac.uk
eric@comp.leeds.ac.uk