Die Modellierung von Turn-taking in einem korpusbasierten Chatbot / Modelling turn-taking in a corpus-trained chatbot

Bayan Abu Shawar and Eric Atwell

School of Computing

University of Leeds

bshawar@comp.leeds.ac.uk ; eric@comp.leeds.ac.uk

Typ des Beitrags/Type of the paper Workshop

Die Modellierung von Turn-taking in einem korpusbasierten Chatbot / Modelling turn-taking in a corpus-trained chatbot

Bayan Abu Shawar and Eric Atwell

A chatbot system is a conversational agent that interacts with a user turn by turn using natural language. To simulate human chat, we developed a java program to convert a machine readable text (corpus) to the ALICE-AIML chatbot linguistic knowledge representation format. An important problem in corpus-trained chatbot development is how to model dialogue turn-taking. In the ALICE-AIML representation, the corpus is mapped onto a list of rules or Categories, pairing Pattern (user input) with appropriate Template (system response). We present different turn-taking models, depending on the type of corpus: spoken dialogue transcripts such as the Corpus of Spoken Afrikaans and the spoken part of the British National Corpus; partly-structured monologue such as the verses of the Qur'an; and texts showing clear turn-taking, such as Loebner Prize contest transcripts, and Frequently Asked Questions (FAQ) websites. Our turn-taking model is simple enough to be generic and learnable from this wide range of training corpora; yet is adequate to give reasonable user satisfaction in user trials.

Ein Chatbot ist ein Agent, der mit einem Benutzer in natürlicher Sprache interagiert, wobei der Benutzer und der Agent sich mit der Ein/Ausgabe abwechseln (turn-taking). Um menschliche Unterhaltung zu simulieren, entwickelten wir ein Java Programm, das einen maschinenlesbaren Text (Korpus) in das Wissensrepräsentationsformat des ALICE-AIML Chatbots umwandelt. Ein wichtiges Problem in der Entwicklung eines Chatbots, der auf Korpora trainiert wird, ist die Modellierung des Turn-taking im Dialog. In der ALICE-AIML Repräsentation wird der Korpus auf eine Liste von Regeln oder Kategorien abgebilded, die ein Muster (Benutzereingabe) mit einem geeigeneten Templage (Systemantwort) paaren. Wir stellen verschiedene Turn-taking Modelle vor, die vom Korpustyp abhängig sind: transkribierte Dialoge wie der Korpus des gesprochenen Afrikaans und der gesprochene Teil des British National Corpus; teilstrukturierete Monologe wie die Verse des Koran; und Texte, die klares Turn-taking Verhalten aufweisen, wie die transkribierten Texte des Loebner Preis Wettbewerbs, und FAQ Internetseiten. Unser turn-taking Modell ist einfach genug, um allgemein anwendbar auf und lernbar von verschiedenen Trainingkorpora zu sein; dennoch ist es beschreibungsadequat genug, um Zufriedenheit der Benutzer in Benutzerexperimenten zu garantieren.

1. Introduction

"Before there were computers, we could distinguish persons from non-persons on the basis of an ability to participate in conversations. But now, we have hybrids operating between person and non persons with whom we can talk in ordinary language." (Colby 1999). Human machine conversation as a technology integrates language and the computational methodologies to facilitate communication between users and computers using natural language.

A related term to machine conversation is the chatbot, a conversational agent that interacts with users turn by turn using natural language. Different chatbots or human-computer dialogue systems have been developed using text communication starting from ELIZA (Weizenbaum 1966, 1967) that simulates a psychotherapist, then PARRY (Colby 1973) which simulates a paranoid patient. "Colby regarded PARRY as a tool to study the nature of paranoia, and considered ELIZA as a potential clinical agent who could, within a time-sharing framework, autonomously handle several hundred patients an hour." (Guzeldere and Franchi 1995).

Nowadays many chatbots are available online and are used for different purposes such as: MIA (aitools.org 2004) which is a German advisor on opening a bank account; Sanelma (MUMMI 2004), a fictional female to talk with in a museum that provides information related to specific works of art; Cybelle (Cybelle 2004), a female guide to cyberspace and Agentland; and AskJevees (2004), a web-based search engine.

Practical applications and evaluation are key issues in Language Engineering: Cunningham (1999) characterises Language Engineering in terms of "…its focus on large-scale practical tasks and on quantitative evaluation of progress, and its willingness to embrace a diverse range of techniques". The Loebner prize competition (Loebner 03) has been used to evaluate machine conversation chatbots. The Loebner Prize is a Turing test, which evaluates the ability of the machine to fool people that they are talking to human. In essence, judges are allowed a short chat (10 to 15 minutes) with each chatbot, and asked to rank them in terms of "naturalness". A.L.I.C.E (Wallace 2003) is the Artificial Linguistic Internet Computer Entity, started by Wallace in 1995. Different versions of Alice won the Loebner prize contest, a modern Turing test competition, three times in 2000, 2001, and recently in 2004. Other Alice-style AIML-based chatbots have been developed; for example, Speak2me (speak2me 2004) is another chatbot used to assist learning English language through

chatting. The chatbot accepts the input in English text mode, and returns the reply in both textual and synthesised voice media. This tool is used to practise and improve conversation skills.

All these chatbots are restricted to the language and to the linguistic knowledge which is manually handcrafted. In our research we have implemented a Java program to convert a readable text corpus to the AIML format to retrain ALICE. For this to work properly, the corpus-training program must model turn-taking appropriately: it must know which parts of the training corpus to learn chatbot turns from, and which parts correspond to user turns.

In this paper, we present several turn-taking models corresponding to different corpora structure. Section 2 presents the A.L.I.C.E chatbot system and the AIML linguistic knowledge representation formalism. Section 3 describes the Java program that converts a readable text to AIML format, and the training dialogue corpora used. Section 4 shows that the approach can be readily extended to corpora in exotic languages, such as Arabic; and to a monologue corpus. In section 5 dialogues generated from FAQ databases and Loebner Prize transcripts are discussed. Section 6 gives conclusions and ideas for future work.

2. The ALICE chatbot system

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

ALICE (Alice 2000, Abu Shawar and Atwell 2003a,b, Wallace 2003) is the Artificial Linguistic Internet Computer Entity, which was 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). It was 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 A.L.I.C.E. open-source software technology.

AIML is explained in (Abu Shawar and Atwell 2003b) as follows. 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 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.

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 simply recursive artificial intelligence and symbolic reduction. Recursive categories have many applications: symbolic reduction that reduces complex grammatical forms to simpler ones; divide

and conquer that splits an input into two or more subparts, and combines the responses to each; and dealing with synonyms by mapping different ways of saying the same thing to the same reply. For example:

<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.

The AIML interpreter tries to match word by word to obtain the longest pattern match, as this is normally the best one. This behaviour can be described in terms of a 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.

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

3. System implementation and dialogue corpus training datasets

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,c) 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 fours 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.

We applied the same program version to a French dialogue corpus (Kerr 1983), which required changing the text pre-processing module since it has its own specific annotations; the problematic difference was NOT the change from English to French, but the changes in mark-up conventions.

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 frequent 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 program was tested using an Afrikaans corpus (Van Rooy 2002). Unfortunately this approach still failed to satisfy our trial users, so we look for the word in the utterance with the highest "information content", the word that is most specific to this utterance compared to other utterances in the corpus. This should be the word that has the lowest frequency in the rest of the corpus. We choose the most significant approach to generate the default categories, because usually in human dialogues the intent of the speakers is 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 technique was used with other corpora such as the British National Corpus (BNC 2002) where more than one million categories were generated in different domains, such as food, sport, and London teenager conversations.

Figures 1 and 2 illustrate the dialogue generated between human and the trainable chatbot generated from the BNC and the Korpus Gesprooke Afrikaans.

Figure (1): Sample of a dialogue generated between a real human and the sport BNC chatbot.

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

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

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

Human: Do you like it?

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

Figure 2: Sample of a dialogue generated between a real human and the Afrikaans chatbot.

Human: hallo
Bot: Goeiemore

Human: wat is jou naam?
Bot: My naam is Karike. Wat is jou naam?

Human: my naam is louis
Bot: Nice om te ontmoet, louis!

Human: waar woon jy?
Bot: hy werk vandag

Human: jy praat allemintige klomp stront karike
Bot: Hoe gaan dit?

In these two samples, the program managed to generate dialogue, even though it does not have a sophisticated dialogue model.

4. Dealing with an exotic language, and a monolog corpus

In this section the Qur'an, the holy book of Islam, is used as an example of the Arabic language, and a text which is not conversational in its nature. The Arabic language has 28 consonants and three vowels: "a", "I", "u" that can be short or long. Vowels can be omitted in Arabic texts, resulting in either a vowelled text such as the language of the Qur'an or an unvowelled text such as newspapers or most published text. The Qur'an text is available via the Internet; and in principle the Qur'an provides guidance and answers to religious and other questions.

This version was updated to generate Arabic AIML files extracted from the Qur'an, the holy book of Islam. 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 a verse, then the reply will be the next verse. Also we wanted the chatbot to reply to user input with relevant quotations from the qur'an; our turn taking model assumed every verse was a template for both user input and chatbot response, again using every "turn" (verse) twice in training.

Two chatbots version were created: the first accepts Arabic input and responds with the Arabic verse(s) see (Abu Shawar and Atwell 2004a). The

second version accepts English input and responds with Arabic and English verse(s), see (Abu Shawar and Atwell 2004b). Figure 3 shows samples of the English-Arabic chatting dialogues with the Qur'an.

Figure 3: Samples of chatting with the Qur'an

a. Samples of chatting with the Arabic Qur'an where the input is a verse and the output is the next one

Human: تَفْعَلُونَ يَا أَيُّهَا الَّذِينَ آمَنُوا لِمَ تَقُولُونَ مَا لَا
Qur'an28-30: مَا لَا تَفْعَلُونَ (سورة الصف)3 كَبُرَ مَقْتًا عِندَ اللَّهِ أَن تَقُولُوا

Human: تَفْعَلُونَ كَبُرَ مَقْتًا عِندَ اللَّهِ أَن تَقُولُوا مَا لَا
Qur'an28-30: فِي سَبِيلِهِ صَفًّا كَأَنَّهُم بُنيَانٌ مَّرْصُوصٌ (سورة الصف)4 إِنَّ اللَّهَ يُحِبُّ الَّذِينَ يُقَاتِلُونَ

Human: كَأَنَّهُم بُنيَانٌ مَّرْصُوصٌ إِنَّ اللَّهَ يُحِبُّ الَّذِينَ يُقَاتِلُونَ فِي سَبِيلِهِ صَفًّا
Qur'an28-30: اللَّهِ إِلَيْكُمْ فَلَمَّا زَاغُوا أَزَاغَ اللَّهُ قُلُوبَهُمْ لِقَوْمِهِ يَا قَوْمِ لِمَ تُؤْذُونَنِي وَقَد تَّعْلَمُونَ أَنِّي رَسُولُ وَإِذْ قَالَ مُوسَى يَهْدِي الْقَوْمَ الْفَاسِقِينَ (سورة الصف)5 وَاللَّهُ لَا

b. Samples of chatting with the English/Arabic version of the Qur'an

Human: peace and justice in Islam
Qur'an14-114: {Drawn Up In Ranks (As-Saaffat)} 181 And Peace on the apostles! وَسَلَامٌ عَلَى الْمُرْسَلِينَ (سورة الصافات)181{Ornaments Of Gold (Az-Zukhruf)} 89 But turn away from them, and say "Peace!" But soon shall they know! وَقُلْ سَلَامٌ فَسَوْفَ يَعْلَمُونَ (سورة الزخرف)89 فَاصْفَحْ عَنْهُمْ

Human: Is iti true that Allah is described as being totally free from any defects or flaws
Qur'an14-114: {The Letter Qaf (Qaf)} 6 Do they not look at the sky above them?- How We have made it and adorned it, and there are no flaws in it? أَفَلَمْ يَنظُرُوا إِلَى السَّمَاء فَوْقَهُمْ كَيْفَ بَنَيْنَاهَا وَزَيَّنَّاهَا وَمَا لَهَا مِن فُرُوجٍ (سورة ق)6

Human: is it right that our protector is Allah?
Qur'an14-114: {Muhammad (Muhammad)} 11 That is because Allah is the Protector of those who believe, but those who reject Allah have no protector. مَوْلَى الَّذِينَ آمَنُوا وَأَنَّ ذَلِكَ بِأَنَّ اللَّهَ الْكَافِرِينَ لَا مَوْلَى لَهُمْ (سورة محمد)11{The Morning Star (At-Tariq)} 4 There is no soul but has a protector over it. إِن كُلُّ نَفْسٍ لَّمَّا عَلَيْهَا حَافِظٌ (سورة الطارق)4

5. Dealing with training texts with explicit turn-taking

We have retrained the chatbot using transcripts of previous Loebner Prize competitions, where user and chatbot turns are clearly marked; in this training dataset, it is clear which parts of the text correspond to human input, and which parts provide chatbot replies. The resulting chatbot is illustrated in figure 4.

Figure 4: Sample of chatting with the chatbot trained with Loebner Prize transcripts

Human: what are you?
Bot: That's a good question - but perhaps a little bit too short in words mm! are you a tree frog? mm!

Human: I am not a tree frog, are you???
Bot: mm! are you a tree frog? mm!

Human: NO I am not a tree frog
Bot: We all know the line that the liberal sixties were to blamefor the escalating crime rate Do you accept that or haveanother theory?I'm not too great with names, but are you Elliott? mm! are you a tree frog? mm!

Human: at least you've stopped talking about tree frogs
Bot: mm! It would be best to answer that for yourself Do you think its likely that i'm trying to act human? mm!

Human: no you're being a tree frog fanatic
Bot: Please tell me something about your childhood, sometimes I wish i was a systems analyst mm! are you a tree frog? mm!

Another apparently straightforward case is FAQs: Frequently Asked Questions websites can be used to train chatbots. In these versions the chatbot expects user input similar to FAQ questions, and replies with FAQ answers, or with lists of links to FAQ answers. FAQs have clear turn-taking structure, and are widely used in computing so are a good source of training corpuses.

Three versions of FAQ chatbots were generated: Ask me (2004) which is trained using the FAQ of the School of Computing (SoC) at the University of

Leeds; other versions are Python advisors in both English and Spanish (2004), trained with the English and Spanish versions of the online Python FAQ website. Figure 5 illustrates these three FAQ-chatbot versions.

Figure 5: Samples of questions and answers generated with the FAQ chatbot versions

---

a. Using FAQ of the School of Computing at the University of Leeds

Human: How to remove a job from the printer?
Ask me:

If you want to remove a job from the printer queue then find out the job number using the 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. Using the Python FAQ website in English

Human: what is a class ?
English_Python_FAQ:

A class is the particular object type created by executing a class statement. Class objects are used as templates to create instance objects, which embody both the data (attributes) and code (methods) specific to a datatype.

A class can be based on one or more other classes, called its base class(es). It then inherits the attributes and methods of its base classes. This allows an object model to be successively refined by inheritance. You might have a generic `Mailbox` class that provides basic accessor methods for a mailbox, and subclasses such as `MboxMailbox`, `MaildirMailbox`, `OutlookMailbox` that handle various specific mailbox formats.

c. Using the Python FAQ website in Spanish

Human: que es python?

Spanish_Python_FAQ:

Python es un lenguaje de programacin interpretado, interactivo y orientado a objetos Incorpora mdulos, excepciones, tipado dinamico, tipos de datos dinamicos de muy alto nivel, y clases Python combina un remarcable poder con una sintaxis muy clara Tiene interfaces a muchas llamadas al sistema y bibliotecas, asi como tambien a varios sistemas de ventanas, y es extensible en C o C++ Tambien es utilizable como un lenguaje de extensin para aplicaciones que necesiten interfaces programables Finalmente, Python es portable, corre en muchas variantes de Unix, en la Mac, y en PCs bajo MS-DOS, Windows, Windows NT, y OS/2

Para descubrir mas, comience con la Guia de Python para el Principiante

6. Conclusions and future developments

A chatbot is a conversational agent that interacts with users turn by turn using natural language. A Java program was implemented to convert a machine readable text (corpus) to the AIML chatbot linguistic knowledge representation format. For this to work properly, the corpus-training program must model turn-taking appropriately: it must know which parts of the training corpus constitute chatbot turns and which correspond to user turns. We investigated different corpora structure where the turn taking sometimes is obvious and clear such as: FAQ databases, Loebner Prize transcripts. Sometimes the turn taking is not so obvious such as dialogue corpora like BNC, KGA where there are more than two speakers; so who plays the role of the input, and who plays the role of the output is not obvious. Another challenge was an Arabic chatbot, trained with the Qur'an, the holy book of Islam. Also, as the Qur'an is non conversational in its nature, the turn taking is not obvious; to overcome this problem, a reiteration process was used for each turn, to play the role of input in one category and the role of output in the next one.

Even without adapting any complicated dialogue models, the systems managed to generate dialogue that satisfied some (but not all) users. We conclude that our simple turn-taking model, based on a large set of Categories containing pattern and template, is sufficient for at least some applications. More generally, dialogue systems developers should consider this simple approach, before assuming a more sophisticated linguistic model is needed.

We are looking into applications of the chatbot in education, and one challenge is re-use of existing teaching texts and materials to train the chatbot. It is not clear how to model turn-taking in all cases: if the training text includes

transcripts of student-teacher dialogues (as in the Dialogue Diversity Corpus (Man 2002)) or something akin to Frequently Asked Questions (and their answers) then we can readily model turn taking; but in a transcript of a lecture, or a generic textbook, the "dialogue" is really a monologue without turn-taking: we have to devise a way to artificially impose a turn taking structure on the text before it can be used in chatbot-training. Hopefully we can gain some good ideas for this at the Workshop on turn-taking, and this resultant Proceedings!

8. References

Abu Shawar, B. and Atwell, E. (2003a). Using dialogue corpora to retrain a chatbot system. Proceedings of the Corpus Linguistics 2003 conference (CL2003), Lancaster University UK. pp. 681-690.

Abu Shawar, B. and Atwell, E. (2003b). Using the corpus of Spoken Afrikaans to generate an Afrikaans chatbot. Southern African Linguistics and Applied Language Studies. Vol. 21, pp. 283-294.

Abu Shawar, B. and Atwell, E. (2003c). Machine Learning from dialogue corpora to generate chatbots. Expert Update journal, Vol 6, No 3, pp 25-29

Abu Shawar, B. and Atwell, E. (2004a). An Arabic chatbot giving answers from the Qur'an. In Bel, B & Marlien, I (editors) Proceedings of TALN04: XI Conference sur le Traitement Automatique des Langues Naturelles, Volume 2, pp. 197-202, ATALA.

Abu Shawar, B. and Atwell, E. (2004b). Accessing an information system by chatting in: Meziane, F & Metais,E (editors) Natural Language Processing and Information Systems, pp. 407-412 Springer-Verlag. 2004.

Aitools.org. (2004). http://www.aitools.org/livebots/

Alice (2002). A.L.I.C.E AI Foundation , http://www.Alicebot.org/

Ask me (2004). http://www.pandorabots.com/pandora/talk?botid=c666f74d9e357bda

AskJevees. (2004). http://ask.co.uk/home

BNC (2002). http://www.natcorp.ox.ac.uk/

Colby, K. (1973). Simulation of Belief systems. In Schank and Colby (Eds.) Computer Models of Thought and Language. Freeman, San Francisco, CA.

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

Cunningham, H. (1999). A definition and short history of Language Engineering. Journal of Natural Language Engineering, 5 (1), Pp 1-16.

Cybelle. (2004). AgentLand.com

English_Python. (2004). http://www.pandorabots.com/pandora/talk?botid=9be545433e356551

Güzeldere, G. and Franchi, S. (1995). Dilaogue with colourful personalities of early ai". In Constructions of the Mind, SEHR, volume 4, issue 2. [On line]: http://www.stanford.edu/group/SHR/4-2/text/toc.html

Kerr, B. (1983). Minnesota Corpus. University of Minnesota, Minneapolis, USA.

Loebner, H. (2003), "Home Page of the Loebner Prize-The First Turing Test", http://www.loebner.net/Prizef/loebner-prize.html

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

MUMMI (2004). A Concept for chatbot:"Sanelma" building engaging relationship between the work of art and the exhibition visitor. http://www.mlab.uiah.fi/mummi/sanelma/

Spanish_Python. (2004). **http://www.pandorabots.com/pandora/talk?botid=a503ce04be35655e**

Speak2ME. (2005). www.speak2me.net

Van Rooy, B. (2002). Transkripsiehandleiding van die Korpus Gesproke Afrikaans. [Transcription Manual of the Corpus Spoken Afrikaans.] Potchefstroom: Potchefstroom University.

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

Weizenbaum, J. (1966). ELIZA-A computer program for the study of natural language communication between man and machine, Communications of the ACM, Vol. 10, No. 8, pp36-45.

Weizenbaum, J. (1967). Contextual understanding by computers, Communications of the ACM, Vol. 10, No. 8, pp474-480

Wilks, Y. (1999). Preface. In Wilks, Y. (eds.) Machine conversations. Kluwer.