Different measurements metrics to evaluate a chatbot system

Bayan Abu Shawar
IT department
Arab Open University
[add]
b_shawar@arabou-jo.edu.jo

Eric Atwell
School of Computing
University of Leeds
LS2 9JT, Leeds-UK
eric@comp .leeds.ac.uk

Abstract

A chatbot is a software system, which can interact or “chat” with a human user in natural language such as English. For the annual Loebner Prize contest, rival chatbots have been assessed in terms of ability to fool a judge in a restricted chat session. We are investigating methods to train and adapt a chatbot to a specific user’s language use or application, via a user-supplied training corpus. We advocate open-ended trials by real users, such as an example Afrikaans chatbot for Afrikaans-speaking researchers and students in South Africa. This is evaluated in terms of “glass box” dialogue efficiency metrics, and “black box” dialogue quality metrics and user satisfaction feedback. The other examples presented in this paper are the Qur'an and the FAQchat prototypes. Our general conclusion is that evaluation should be adapted to the application and to user needs.

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 1999a). Human machine conversation as a technology integrates different areas where the core is the language, and the computational methodologies 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 such as Eliza (Weizenbaum 1966), PARRY (Colby 1999b), CONVERSE (Batacharia etc 1999), ALICE[1]. Chatbots have been used in different domains such as: customer service, education, web site help, and for fun.

Different mechanisms are used to evaluate Spoken Dialogue Systems (SLDs), ranging from glass box evaluation that evaluates individual components, to black box evaluation that evaluates the system as a whole McTear (2002). For example, glass box evaluation was applied on the (Hirschman 1995) ARPA Spoken Language system, and it shows that the error rate for sentence understanding was much lower than that for sentence recognition. On the other hand black box evaluation evaluates the system as a whole based on user satisfaction and acceptance. The black box approach evaluates the performance of the system in terms of achieving its task, the cost of achieving the task in terms of time taken and number of turns, and measures the quality of the interaction, normally summarised by the term ‘user satisfaction’, which indicates whether the user “ gets the information s/he wants, is s/he comfortable with the system, and gets the information within acceptable elapsed time, etc.” (Maier et al 1996).

The Loebner prize[2] competition 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”.

ALICE (Abu Shawar and Atwell 2003) is the Artificial Linguistic Internet Computer Entity, first

[1] http://www.alicebot.org/

[2] http://www.loebner.net/Prizef/loebner-prize.html

Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings, pages 89–96,
NAACL-HLT, Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

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.

In this paper we present other methods to evaluate the chatbot systems. ALICE chtabot system was used for this purpose, where a Java program has been developed to read from a corpus and convert the text to the AIML format. The Corpus of Spoken Afrikaans (Korpus Gesproke Afrikaans, KGA), the corpus of the holy book of Islam (Qur'an), and the FAQ of the School of Computing at University of Leeds[3] were used to produce two KGA prototype, the Qur'an prototype and the FAQchat one consequently.

Section 2 presents Loebner Prize contest, section 3 illustrates the ALICE/AIMLE architecture. The evaluation techniques of the KGA prototype, the Qur'an prototype, and the FAQchat prototype are discussed in sections 4, 5, and 6 consequently. The conclusion is presented in section 7.

2 The Loebner Prize Competition

The story began with the "imitation game" which was presented in Alan Turing's paper "Can Machine think?" (Turing 1950). The imitation game has a human observer who tries to guess the sex of two players, one of which is a man and the other is a woman, but while screened from being able to tell which is which by voice, or appearance. Turing suggested putting a machine in the place of one of the humans and essentially playing the same game. If the observer can not tell which is the machine and which is the human, this can be taken as strong evidence that the machine can think.

Turing's proposal provided the inspiration for the Loebner Prize competition, which was an attempt to implement the Turing test. The first contest organized by Dr. Robert Epstein was held on 1991, in Boston's Computer Museum. In this incarnation the test was known as the Loebner contest, as Dr. Hugh Loebner pledged a $100,000 grand prize for the first computer program to pass the test. At the beginning it was decided to limit the topic, in order to limit the amount of language the contestant programs must be able to cope with, and to limit the tenor. Ten agents were used, 6 were computer programs. Ten judges would converse with the agents for fifteen minutes and rank the terminals in order from the apparently least human to most human. The computer with the highest median rank wins that year's prize. Joseph Weintraub won the first, second and third Loebner Prize in 1991, 1992, and 1993 for his chatbots, PC Therapist, PC Professor, which discusses men versus women, and PC Politician, which discusses Liberals versus Conservatives. In 1994 Thomas Whalen (Whalen 2003) won the prize for his program TIPS, which provides information on a particular topic. TIPS provides ways to store, organize, and search the important parts of sentences collected and analysed during system tests.

However there are sceptics who doubt the effectiveness of the Turing Test and/or the Loebner Competition. Block, who thought that "the Turing test is a sorely inadequate test of intelligence because it relies solely on the ability to fool people"; and Shieber (1994), who argued that intelligence is not determinable simply by surface behavior. Shieber claimed the reason that Turing chose natural language as the behavioral definition of human intelligence is "exactly its open-ended, freewheeling nature", which was lost when the topic was restricted during the Loebner Prize. Epstein (1992) admitted that they have trouble with the topic restriction, and they agreed "every fifth year or so … we would hold an open-ended test - one with no topic restriction." They decided that the winner of a restricted test would receive a small cash prize while the one who wins the unrestricted test would receive the full $100,000.

Loebner in his responses to these arguments believed that unrestricted test is simpler, less expensive and the best way to conduct the Turing Test. Loebner presented three goals when constructing the Loebner Prize (Loebner 1994):

- "No one was doing anything about the Turing Test, not AI." The initial Loebner Prize contest was the first time that the Turing Test had ever been formally tried.
- Increasing the public understanding of AI is a laudable goal of Loebner Prize. "I believe that this contest will advance AI and

[3] http://www.comp.leeds.ac.uk

serve as a tool to measure the state of the art.”
- Performing a social experiment.

The first open-ended implementation of the Turing Test was applied in the 1995 contest, and the prize was granted to Weintraub for the fourth time. For more details to see other winners over years are found in the Loebner Webpage[4].

In this paper, we advocate alternative evaluation methods, more appropriate to practical information systems applications. We have investigated methods to train and adapt ALICE to a specific user’s language use or application, via a user-supplied training corpus. Our evaluation takes account of open-ended trials by real users, rather than controlled 10-minute trials.

3 The ALICE/AIML chatbot architecture

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; it 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 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 structure of AIML is shown in figure 1.

```
< 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 bot output.

Figure 1. AIML format

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. Three types of AIML categories are used: *atomic category*, are those with patterns that do not have wildcard symbols, _ and *; *default categories* are those with patterns having wildcard symbols * or _. The wildcard symbols match any input but can differ in their alphabetical order. For example, given input ‘hello robot’, if ALICE does not find a category with exact matching atomic pattern, then it will try to find a category with a default pattern; The third type, *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.

The knowledge bases of almost all chatbots are edited manually which restricts users to specific languages and domains. We developed a Java program to read a text from a machine readable text (corpus) and convert it to AIML format. The chatbot-training-program was built to be general, the generality in this respect implies, no restrictions on specific language, domain, or structure. Different languages were tested: English, Arabic, Afrikaans, French, and Spanish. We also trained with a range of different corpus genres and structures, including: dialogue, monologue, and structured text found in the Qur’an, and FAQ websites.

The chatbot-training-program is composed of four phases as follows:

- Reading module which reads the dialogue text from the basic corpus and inserts it into a list.
- Text reprocessing module, where all corpus and linguistic annotations such as overlapping, fillers and others are filtered.
- Converter module, where the pre-processed text is passed to the converter to consider the first turn as a pattern and the

[4] http://www.loebner.net/Prizef/loebner-prize.html

second as a template. All punctuation is removed from the patterns, and the patterns are transformed to upper case.

- Producing the AIML files by copying the generated categories from the list to the AIML file.

An example of a sequence of two utterances from an English spoken corpus 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>
```

After the reading and the text processing phase, the text becomes:

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

The corresponding AIML atomic category that is generated from the converter modules looks like:

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

As a result different prototypes were developed, in each prototype, different machine-learning techniques were used and a new chatbot was tested. The machine learning techniques ranged from a primitive simple technique like single word matching to more complicated ones like matching the least frequent words. Building atomic categories and comparing the input with all atomic patterns to find a match is an instance based learning technique. However, the learning approach does not stop at this level, but it improved the matching process by using the most significant words (least frequent word). This increases the ability of finding a nearest match by extending the knowledge base which is used during the matching process. Three prototypes will be discussed in this paper as listed below:

- The KGA prototype that is trained by a corpus of spoken Afrikaans. In this prototype two learning approaches were adopted. The first word and the most significant word (least frequent word) approach;
- The Qur'an prototype that is trained by the holy book of Islam (Qur'an): where in addition to the first word approach, two significant word approaches (least frequent words) were used, and the system was adapted to deal with the Arabic language and the non-conversational nature of Qur'an as shown in section 5;
- The FAQchat prototype that is used in the FAQ of the School of Computing at University of Leeds. The same learning techniques were used, where the question represents the pattern and the answer represents the template. Instead of chatting for just 10 minutes as suggested by the Loebner Prize, we advocate alternative evaluation methods more attuned to and appropriate to practical information systems applications. Our evaluation takes account of open-ended trials by real users, rather than artificial 10-minute trials as illustrated in the following sections.

The aim of the different evaluations methodologies is as follows:

- Evaluate the success of the learning techniques in giving answers, based on dialogue efficiency, quality and users' satisfaction applied on the KGA.
- Evaluate the ability to use the chatbot as a tool to access an information source, and a useful application for this, which was applied on the Qur'an corpus.
- Evaluate the ability of using the chatbot as an information retrieval system by comparing it with a search engine, which was applied on FAQchat.

4 Evaluation of the KGA prototype

We developed two versions of the ALICE that speaks Afrikaans language, Afrikaana that speaks only Afrikaans and AVRA that speaks English and Afrikaans; this was inspired by our observation that the Korpus Gesproke Afrikaans actually includes some English, as Afrikaans speakers are generally bilingual and "code-switch" comfortably. We mounted prototypes of the chatbots on websites using Pandorabot service[5], and encouraged

[5] http://www.pandorabots.com/pandora

open-ended testing and feedback from remote users in South Africa; this allowed us to refine the system more effectively.

We adopted three evaluation metrics:

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

4.1 Dialogue efficiency metric

We measured the efficiency of 4 sample dialogues in terms of atomic match, first word match, most significant match, and no match. We wanted to measure the efficiency of the adopted learning mechanisms to see if they increase the ability to find answers to general user input as shown in table 1.

Matching Type	D1	D2	D3	D4
Atomic	1	3	6	3
First word	9	15	23	4
Most significant	13	2	19	9
No match	0	1	3	1
Number of turns	23	21	51	17

Table 1. Response type frequency

The frequency of each type in each dialogue generated between the user and the Afrikaans chatbot was calculated; in Figure 2, these absolute frequencies are normalised to relative probabilities.

No significant test was applied, this approach to evaluation via dialogue efficiency metrics illustrates that the first word and the most significant approach increase the ability to generate answers to users and let the conversation continue.

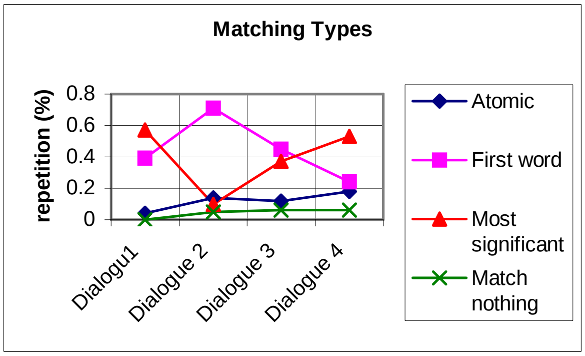

Figure 2. Dialogue efficiency: Response Type Relative Frequencies

4.2 Dialogue quality metric

In order to measure the quality of each response, we wanted to classify responses according to an independent human evaluation of “reasonableness”: reasonable reply, weird but understandable, or nonsensical reply. We gave the transcript to an Afrikaans-speaking teacher and asked her to mark each response according to these classes. The number of turns in each dialogue and the frequencies of each response type were estimated. Figure 3 shows the frequencies normalised to relative probabilities of each of the three categories for each sample dialogue. For this evaluator, it seems that “nonsensical” responses are more likely than reasonable or understandable but weird answers.

4.3 Users' satisfaction

The first prototypes were based only on literal pattern matching against corpus utterances: we had not implemented the first word approach and least-frequent word approach to add “wildcard” default categories. Our Afrikaans-speaking evaluators found these first prototypes disappointing and frustrating: it turned out that few of their attempts at conversation found exact matches in the training corpus, so Afrikaana replied with a default “ja” most of the time. However, expanding the AIML pattern matching using the first-word and least-frequent-word approaches yielded more favorable feedback. Our evaluators found the conversations less repetitive and more interesting. We measure user satisfaction based on this kind of informal user feed back.

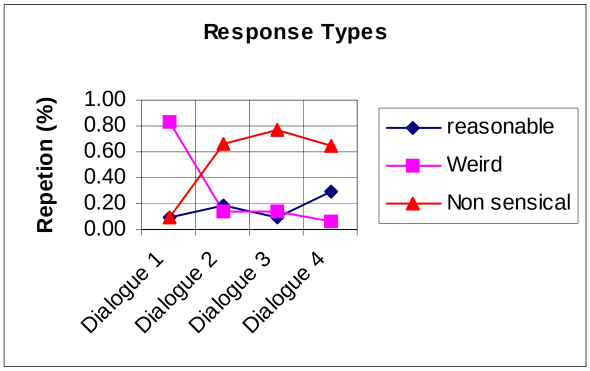

Figure 3. The quality of the Dialogue: Response type relative probabilities

5 Evaluation of the Qur'an prototype

In this prototype a parallel corpus of English/Arabic of the holy book of Islam was used, the aim of the Qur’an prototype is to explore the problem of using the Arabic language and of using a text which is not conversational in its nature like the Qur’an. The Qur’an is composed of 114 soora (chapters), and each soora is composed of different number of verses. The same learning technique as the KGA prototype were applied, where in this case if an input was a whole verse, the response will be the next verse of the same soora; or if an input was a question or a statement, the output will be all verses which seems appropriate based on the significant word. To measure the quality of the answers of the Qur’an chatbot version, the following approach was applied:

1. Random sentences from Islamic sites were selected and used as inputs of the English/Arabic version of the Qur’an.
2. The resulting transcripts which have 67 turns were given to 5 Muslims and 6 non-Muslims students, who were asked to label each turn in terms of:

- Related (R), in case the answer was correct and in the same topic as the input.
- Partially related (PR), in case the answer was not correct, but in the same topic.
- Not related (NR), in case the answer was not correct and in a different topic.

Proportions of each label and each class of users (Muslims and non-Muslims) were calculated as the total number over number of users times number of turns. Four out of the 67 turns returned no answers, therefore actually 63 turns were used as presented in figure 4.

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

- The different interpretation of the answers. The Qur’an uses traditional Arabic language, which is sometimes difficult to understand without knowing the meaning of some words, and the historical story behind each verse.
- The English translation of the Qur’an is not enough to judge if the verse is related or not, especially given that non-Muslims do not have the background knowledge of the Qur’an.

Using chatting to access the Qur’an looks like the use of a standard Qur’an search tool. In fact it is totally different; a searching tool usually matches words not statements. For example, if the input is: “How shall I pray?” using chatting: the robot will give you all ayyas where the word “pray” is found because it is the most significant word. However, using a search tool[6] will not give you any match. If the input was just the word “pray”, using chatting will give you the same answer as the previous, and the searching tool will provide all ayyas that have “pray” as a string or substring, so words such as: ”praying, prayed, etc.” will match.

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

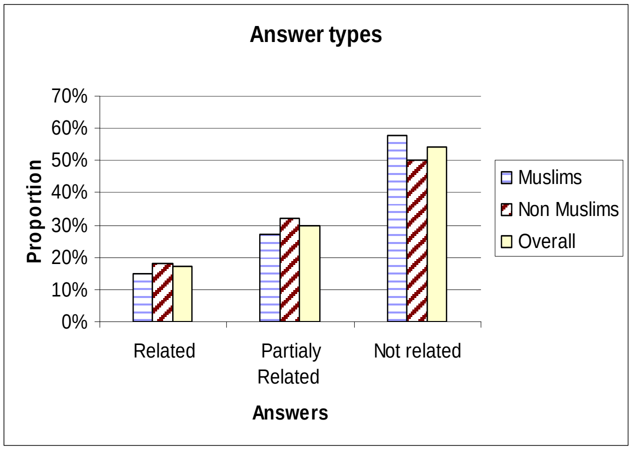

Figure4. The Qur’an proportion of each answer type denoted by users

6 Evaluation of the FAQchat prototype

To evaluate FAQchat, an interface was built, which has a box to accept the user input, and a button to send this to the system. The outcomes ap-

[6] http://www.islamicity.com/QuranSearch/

pear in two columns: one holds the FAQchat answers, and the other holds the Google answers after filtering Google to the FAQ database only. Google allows search to be restricted to a given URL, but this still yields all matches from the whole SoC website (http://www.comp.leeds.ac.uk) so a Perl script was required to exclude matches not from the FAQ sub-pages.

An evaluation sheet was prepared which contains 15 information-seeking tasks or questions on a range of different topics related to the FAQ database. The tasks were suggested by a range of users including SoC staff and research students to cover the three possibilities where the FAQchat could find a direct answer, links to more than one possible answer, and where the FAQchat could not find any answer. In order not to restrict users to these tasks, and not to be biased to specific topics, the evaluation sheet included spaces for users to try 5 additional tasks or questions of their own choosing. Users were free to decide exactly what input-string to give to FAQchat to find an answer: they were not required to type questions verbatim; users were free to try more than once: if no appropriate answer was found; users could reformulate the query.

The evaluation sheet was distributed among 21 members of the staff and students. Users were asked to try using the system, and state whether they were able to find answers using the FAQchat responses, or using the Google responses; and which of the two they preferred and why.

Twenty-one users tried the system; nine members of the staff and the rest were postgraduates. The analysis was tackled in two directions: the preference and the number of matches found per question and per user.

6.1 Number of matches per question

The number of evaluators who managed to find answers by FAQchat and Google was counted, for each question.

Results in table 2 shows that 68% overall of our sample of users managed to find answers using the FAQchat while 46% found it by Google. Since there is no specific format to ask the question, there are cases where some users could find answers while others could not. The success in finding answers is based on the way the questions were presented to FAQchat.

Users /Tool	Mean of users finding answers		Proportion of finding answers	
	FAQchat	Google	FAQchat	Google
Staff	5.53	3.87	61%	43%
Student	8.8	5.87	73%	49%
Overall	14.3	9.73	68%	46%

Table 2: Proportion of users finding answers

Of the overall sample, the staff outcome shows that 61% were able to find answers by FAQchat where 73% of students managed to do so; students were more successful than staff.

6.2 The preferred tool per each question

For each question, users were asked to state which tool they preferred to use to find the answer. The proportion of users who preferred each tool was calculated. Results in figure 5 shows that 51% of the staff, 41% of the students, and 47% overall preferred using FAQchat against 11% who preferred the Google.

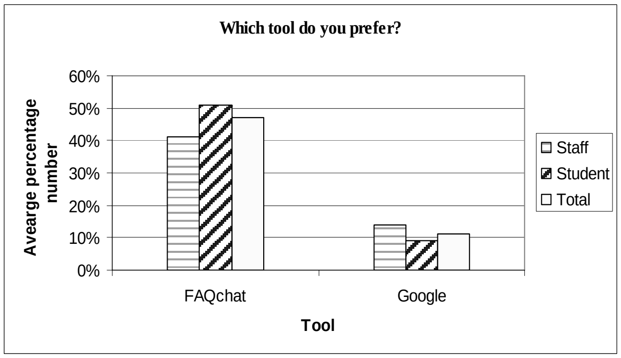

Figure5. Proportion of preferred tool

6.3 Number of matches and preference found per user

The number of answers each user had found was counted. The proportions found were the same. The evaluation sheet ended with an open section inviting general feedback. The following is a summary of the feedback we obtained:

- Both 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 reformulate the question and the FAQchat matches the same word. However Google may give different answers in this case.

To test reliability of these results, the t=Test were applied, the outcomes ensure the previous results.

7 Conclusion

The Loebner Prize Competition has been used to evaluate the ability of chatbots to fool people that they are speaking to humans. Comparing the dialogues generated from ALICE, which won the Loebner Prize with real human dialogues, shows that ALICE tries to use explicit dialogue-act linguistic expressions more than usual to re enforce the impression that users are speaking to human.

Our general conclusion is that we should NOT adopt an evaluation methodology just because a standard has been established, such as the Loebner Prize evaluation methodology adopted by most chatbot developers. Instead, evaluation should be adapted to the application and to user needs. If the chatbot is meant to be adapted to provide a specific service for users, then the best evaluation is based on whether it achieves that service or task

References

Abu Shawar B and Atwell E. 2003. Using dialogue corpora to retrain a chatbot system. In *Proceedings of the Corpus Linguistics 2003 conference*, Lancaster University, UK, pp681-690.

Batacharia, B., Levy, D., Catizone R., Krotov A. and Wilks, Y. 1999. CONVERSE: a conversational companion. In Wilks, Y. (ed.), *Machine Conversations*. Kluwer, Boston/Drdrecht/London, pp. 205-215.

Colby, K. 1999a. Comments on human-computer conversation**.** In Wilks, Y. (ed.), *Machine Conversations*. Kluwer, Boston/Drdrecht/London, pp. 5-8.

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

Epstein R. 1992. Can Machines Think?. *AI magazine*, Vol 13, No. 2, pp80-95

Garner R. 1994. The idea of RED, [Online], http://www.alma.gq.nu/docs/ideafred_garner.htm

Hirschman L. 1995. The Roles of language processing in a spoken language interface. In Voice Communication Between Humans and Machines, D. Roe and J. Wilpon (Eds), National Academy Press Washinton, DC, pp217-237.

Hutchens, J. 1996. *How to pass the Turing test by cheating*. [Onlin], http://ciips.ee.uwa.edu.au/Papers/, 1996

Hutchens, **T., Alder, M.** 1998. Introducing MegaHAL. [Online], http://cnts.uia.ac.be/conll98/pdf/271274hu.pdf

Loebner H. 1994. In Response to lessons from a restricted Turing Test. [Online], http://www.loebner.net/Prizef/In-response.html

Maier E, Mast M, and LuperFoy S. 1996. Overview. In Elisabeth Maier, Marion Mast, and Susan LuperFoy (Eds), *Dialogue Processing in Spoken Language Systems*, , Springer, Berlin, pp1-13.

McTear M. 2002. Spoken dialogue technology: enabling the conversational user interface*. ACM Computing Surveys.* Vol. 34, No. 1, pp. 90-169.

Shieber S. 1994. Lessons from a Restricted Turing Test. *Communications of the Association for Computing Machinery*, Vol 37, No. 6, pp70-78

Turing A. 1950. Computing Machinery and intelligence. *Mind* 59, 236, 433-460.

Weizenbaum, J. 1966. ELIZA-A computer program for the study of natural language communication between man and machine. *Communications of the ACM.* Vol. 10, No. 8, pp. 36-45.

Whalen T. 2003. My experience with 1994 Loebner competition, [Online], http://hps.elte.hu/~gk/Loebner/story94.htm