Chatbots and natural automated language

A comparison between first word and most significant word search

MATILDA VALLERYD VALLERYD@KTH.SE
THERESE ASKLING TASKLING@KTH.SE

SUPERVISOR: GABRIEL SKANTZE

Stockholm 2014

DD143X: Degree Project in Computer Science
School of Computer Science and Communication
Kungliga Tekniska Högskolan

Abstract

This study aims to determine if there is any difference in the perceived naturalness of chatbots implemented with either a first word search- or a most significant word search-algorithm. To this end two versions of the same chatbot were implemented using parsed movie dialogue used as a knowledge base and evaluated using methods developed for chatbot competitions. The results of the study were inconclusive and no statistically certain difference between the two implementations was found.

Sammanfattning

Syftet med denna studie är att avgöra om det är någon skillnad i hur naturlig en chatbot upplevs vara beroende på om den är implementerad med en första ord- eller mest signifikanta ord-algoritm. För att undersöka detta implementerades två versioner av samma chatbot, nyttjande dialog från film som databas, och dessa utvärderades med hjälp av en metod som används till chatbot tävlingar. Resultaten av studien var ofullständiga och ingen statistiskt säkerställd skillnad kunde påvisas.

Contents

1 Introduction 3

1.1 Purpose . . . 3

1.2 Problem statement . . . 3

2 Background 4

2.1 The importance of chatbots . . . 4

2.2 The turing test . . . 4

2.3 Implementation of Chatbots . . . 4

2.3.1 Knowledge Base . . . 4

2.3.2 A.L.I.C.E and AIML . . . 5

2.4 Evaluation of chatbots . . . 6

2.4.1 The Loebner prize and other competitions . . . 6

2.4.2 Different types of evaluation . . . 6

2.4.3 Manual or automatic detection . . . 6

2.4.4 Comparing chatbots . . . 6

2.5 Deficiencies of chatbots . . . 7

3 Method 8

3.1 Implementation . . . 8

3.1.1 Knowledge base . . . 8

3.1.2 Subtitle parser . . . 9

3.1.3 Creating and parsing the AIML files . . . 9

3.1.4 First word implementation . . . 9

3.1.5 Most significant word implementation . . . 10

3.1.6 Sentence comparison . . . 10

3.2 Evaluation . . . 10

3.2.1 Data collection . . . 10

3.2.2 Data preparation . . . 11

4 Results 11

4.1 Data 11

4.2 Statistical analysis 12

5 Discussion **13**

5.1 Analysis of the results 13

5.2 Overall discussion 13

6 Conclusion **13**

1 Introduction

Chatbots, also known as chatterbots [3], are a type of software application in the computational linguistics discipline used to converse with a user in written, natural language. [20] They can be used in chatrooms, on internet forums or blogs, as online support for webpages, built into desktop applications and basically anywhere on a computer where written text is available for a wide range of possible uses. They can be used for beneficial purposes, ie virtual doctors, [2] entertainment, [5]information retreival, [21] or malicious reasons, ie spamming and phishing.

The concept of a chatbot is not limited to any single use but generally work in a similar manner; some predetermined event triggers the bot and the bot responds by writing out an appropriate message selected by its AI. The triggering event can be used to categorize chatbots into four types, periodic; a bot posting at fixed intervals usually used for spamming, random; similar to periodic but uses random intervals to disguise themselves from automatic bot detectors, responder; triggered by the occurrence of certain key words in other users posts, and reply bots; the most well known type [16]. Reply bots are triggered by a user directly interacting with it and are probably what most people think of when hearing the term chatbot. The main focus of this paper will be on this type of bot.

1.1 Purpose

Most reply bots depend on a database of human dialogue, where a certain user input gives a specified reply. When this database grows large, it becomes time consuming to go through every sentence as a way of finding the closest match. Two very common approaches to dealing with this is to search the database using either the first word or the most significant word in the user input. [22] [**?**] The most significant word being the word which occurs the least amount of times in the entire database. While both approaches are quite common, very few studies have been done in regards to their impact on the naturalness of a chatbot.

The purpose of this paper is to further the field of chatterbots by evaluating the experienced difference between two chatterbots implemented using these two different search algorithms.

1.2 Problem statement

Is there a measurable difference in how natural a chatbot is perceived depending on its implementation?

Two versions of a chatbot will be implemented using the same database, then evaluated and compared in regards to their naturalness. One version will use a first word pattern search-algorithm, where the database is indexed after the first words of a phrase, the other will use a most significant-word search algorithm, where every word in a phrase is indexed and a match is found using the rarest word in the input.

This study aims to determine if there is a measurable difference between these two types of implementations regarding the naturalness of their respective responses to input.

2 Background

2.1 The importance of chatbots

Computers are not always easy to communicate with and over the course of computer science history a lot of effort, time and money has been spent on making it easier. Programming languages, both high- and low-level, and graphic user interfaces were introduced for this purpose and today the possibility of using multimodal technology for even greater ease is being explored [12] but in general humans and computers do not speak the same language.

Chatbots are made to understand and mimic human language and are therefore a great tool for abridging these differences in communication. They make computers more accessible for users unaccustomed to the regular means by which we are invited to explore them and possible applications can be found in a range of different fields, such as as Siri, the personal assistant software made by Apple [10], or different customer service chatbots [7]. [22]. For this quality the field of chatbots and automated language deserves to be further explored theoretically, its uses more realised and its implementation improved upon.

2.2 The turing test

Can machines think? In 1950, Alan Turing published a paper named “Computing Machinery and Intelligence” wherein he proposed that this question be answered with another question, one which is more easily answered. He wanted to know if computer generated script could become indistinguishable from human language. To this end, he presented modifications for a known game called “the Imitation Game” where a participant would try and deduce which of two conversations he/she was having was with a man and a woman respectively. Turing suggested that a computer capable of fooling the participant into thinking it was human in such a game would be as close to “thinking” as a computer could get. This version of the game has in later texts been dubbed a “Turing Test”. [19].

The initial question, whether passing the Turing Test implies thought or not, has since then lost relevance but the test itself continues to challenge software developers around the world. As of yet, no program has officially been declared by the academic world to have beaten the turing test despite several annual competitions [8] [4] [9] and a devoted community.

2.3 Implementation of Chatbots

2.3.1 Knowledge Base

An important part of any chatbot is its database of dialogue, its knowledge base. It needs to contain a vast amount of phrases, both as cues to match the user input to and as suitable responses to each of these cues. As such the rapidly increased available disc-size has improved the quality of chatbots immensely. Today a bigger issue is writing good algorithms for fast retrieval of a suitable answer within these vast databases and a tradeoff needs to be made between size and speed. [18]

A great amount of work needs to go into a chatbot's knowledge base to be able to create a bot with personality, knowledge related to its intended use and enough answers to simulate a real conversation. These qualities are important when implementing a good chatbot as they are highly regarded in the chatbot community. [8] [4] [9]. This may take up the largest portion of the work in creating a chatbot. However, thanks to the internet it is easy to get started. There are numerous libraries for this purpose available online as well as other mediums intended for other uses but which are still useful. [1]. One such medium could be movie scripts or movie subtitles as used in this study to create the knowledgebase for both versions of the chatbot.

2.3.2 A.L.I.C.E and AIML

One of the most famous chatbots is the Artificial Linguistic Internet Computer Entity known as A.L.I.C.E, and it is the brainchild of Dr. Richard Wallace. Dr. Wallace created the first version of A.L.I.C.E in 1995 using SETL, a mathematical programming language which utilizes the concept of sets [15]. A.L.I.C.E has since then been implemented using a variety of programming languages, including Java and C/C++. Three of these implementations have managed to win the Loebner prize, in 2000, 2001, and 2004. [24]

A.L.I.C.E utilizes an algorithm known as the graphmaster, which is similar to first word search. In the graphmaster, the bot's knowledge base is represented as a graph with the root of the graph having approximately 2000 branches. Each branch is for each unique first word among the roughly 40000 patterns in A.L.I.C.E's database. [24]

The biggest revolution which A.L.I.C.E has brought to the chatbot community is the Artificial Intelligence Markup Language or AIML for short. In the 2013 Loebner contest, three of the top four chatbots were implemented using AIML. AIML is a variation of XML, which means that the chatbot's knowledge base is stored similarly to XML, using tags. The most basic tag in AIML is the category tag. The category tag contains at least two other tags, pattern and template. The pattern tag contains the sentence that the chatbot will try to match with the user input. If a match is found, the template tag in the same category as the pattern contains the reply to give to the user. AIML also supports recursion, context, and conditional branching. [14] [23]

As an example, this is how a category in AIML may look.

```
<category>
  <pattern>HELLO</pattern>
  <template>Hi there!</template>
</category>
```

The pattern in this case is the word "hello", and if that word matches the user's input the template, "hi there!", will be returned as the bot's response.

2.4 Evaluation of chatbots

2.4.1 The Loebner prize and other competitions

Despite the many difficulties finding a universal evaluation method, programmers have not stopped trying to create chatbots natural enough to fool humans and comparing them to other chatbots. To that end there are a number of international competitions open for anyone with a chatbot, regardless of its application or implementation, such as the Chatterbot Challenge [4] and Robo Chat Callange [9]. The oldest and most prestigious of these is the Loebner prize introduced by Hugh Loebner in 1990 as a way to further scientific fields related to AI, human-computer interaction and naturalness in automated language. [8]

2.4.2 Different types of evaluation

Evaluating humanness or naturalness is not so easily quantifiable and depends on each individual's definition of the terms. What one person might consider natural is not necessarily what anyone else does and therefore theoretical evaluation of these traits is understandably difficult. In practice however there are some techniques that can be applied since no bot has truly been mistaken for human.

2.4.3 Manual or automatic detection

The approach to distinguishing bots from humans is very different depending on if the detection is manual or automatic but both give good results with different strengths and weaknesses.

Automatic detection is used to filter out malicious bots on the internet by detecting repetition, the presence of certain phrases belonging to known bots or statistical analysis techniques such as measuring response time. [16] Computers are excellent at these types of operations and can detect most bots much quicker than humans but there are some well made bots capable of fooling the automatic countermeasures.

Manual detection consumes much more time and resources than automatic but since humans by definition are better at evaluating humanness than machines are it is harder for a bot to fool. This type of detection is less useful for bot screening in online forums but provides a much more interesting challenge, namely the Turing Test.

Though this type of evaluation is very popular, it has been criticised by some. Other methods have been suggested [22] but none have yet reached the same support as the original which remains the superior method of evaluation.

2.4.4 Comparing chatbots

While it is possible to distinguish between chatterbots and humans [22] it is not as easy determening which of several bots are the best, or in this case, most natural. For this purpose automatic testing is practically useless but manual testing is difficult partly because they can be so different from each other and partly because perceived naturalness is a personal opinion. Hence in competitions there is usually a panel of several judges whos combined thoughts form the result.

The loebner prize uses a true turing test but most others use a more cost effective method by asking all the participating bots a set number of predetermined questions and then letting a panel of judges evaluate the quality of each answer according to guidelines. [8]

2.5 Deficiencies of chatbots

Chatbots have come far in the last 20 years but still have far to go before they can mimic perfectly flawed human language because whichever detection method is used, automatic or manual, there are still many ways to distinguish between human users and chatbots, some mentioned in the last subsection.

As of yet, no bot has been good enough to win the Loebner grand prize gold medal or even the silver one so a lot of improvement will have to be done on the subject before robots and computers will be as sophisticated as in science fiction. Some of the breakthroughs needed will come through advancement in hardware, for example, better hard disk drive performance such as more disk space and faster access time. This has increased the capacities of databases and therefore raised the limit of possible responses a chatbot can give which results in more natural language but it still cannot be compared to the number of possible phrases a human can interpret and respond with.

The limited number of phrases provides one of easiest ways to tell a bot from a human, repetition; repetition of vague answers when no better answer can be given; using the exact wording multiple times; giving the same response to multiple questions on the same topic; these are today one of the biggest flaws with chatbots.

There are of course limitations due to the difficulty of writing a good AI resulting in answers that have nothing to do with the triggering phrase and conversations lacking an overarching thread of logic.

Another limitation that probably will be focused on more in the future is the ability - or lack thereof - to interpret media other than written language. [8] This trait is used in a famous countermeasure against malicious bots called CAPTCHA. [6] The test has been dsecribed as a reverse Turing Test [13] as it is a computer trying to decide if the user is a human by showing a picture containing letters. A bot cannot process such images and therefore fail the test.

3 Method

3.1 Implementation

3.1.1 Knowledge base

As mentioned previously, one of the most timeconsuming parts of creating a chatbot is developing its knowledge base. One method to get around this problem is to use dialogue from movies instead of creating dialogue from scratch. To be able to do this there are two different options on how to obtain the movie dialogue, either use movie scripts or movie subtitles, both of which are easily accessible on the internet.

A movie script usually looks something like this:

”Andy finds a table occupied by Red and his regulars, chooses a spot at the end where nobody is sitting. Ignoring their stares, he picks up his spoon – and pauses, seeing something in his food. He carefully fishes it out with his fingers.

It’s a squirming maggot. Andy grimaces, unsure what to do with it. BROOKS HATLEN is sitting closest to Andy. At age 65, he’s a senior citizen, a longstanding resident.

BROOKS
You gonna eat that?

ANDY
Hadn’t planned on it.

BROOKS
You mind?

Andy passes the maggot to Brooks. Brooks examines it, rolling it between his fingertips like a man checking out a fine cigar. Andy is riveted with apprehension.

BROOKS
Mmm. Nice and ripe.”

The dialogue is divided very nicely, it is very clear which character is saying what. Unfortunately, it is also filled with enviromental descriptions and information about the character, this makes it very difficult when trying to extract the dialogue automatically.

The same scene as movie subtitles looks something like this:

”242
00 : 20 : 19, 160 − − > 00 : 20 : 21, 240
Are you going to eat that?
243
00 : 20 : 21, 880 − − > 00 : 20 : 24, 080
I hadn’t planned on it.
244
00 : 20 : 25, 079 − − > 00 : 20 : 26, 280
Do you mind?

245
$00:20:34,519 -- > 00:20:35,839$
That's nice and ripe."

While it is unclear who is saying what, the format of the subtitles makes it quite easy to extract the dialogue. One can then assume that every sentence is a reply to the sentence before.

3.1.2 Subtitle parser

To create the AIML documents for the knowledge base the dialogue was extracted from movie subtitle files. A total of 200 subtitles files were used from a wide variety of genres. The large amount of files was to try to counteract the randomness of movie dialogue and the variety in genres to be reasonabe sure that most topics of conversation were represented in the database.

All the subtitle files used were SubRip files. SubRip is a software which extracts subtitles from movies and videos. The files have the benefit of being formated the same, making it easy to automatically remove non-useful data and extract the dialogue. [11]

3.1.3 Creating and parsing the AIML files

The AIML files were both created and parsed using the Document Object Model, or DOM, API for XML. DOM is one of the most used APIs for XML and works by viewing the XML documents in a object-oriented way. [25] Due to the nature of the dialogue used as knowledge base, only the category, pattern, and template tags were used when creating the AIML files. Any other tags would have required manipulation of the dialogue in a way that was quite difficult to automate. The information in the pattern tag was normalized before written to the AIML document. This normalization was done by removing all characters except for letters and numbers.

3.1.4 First word implementation

The algorithm for first word pattern matching was implemented in a similar fashion to the graphmaster used by A.L.I.C.E. The knowledge database was indexed by each unique first word among the AIML patterns. Each first word was linked to a list of every pattern in the database beginning with that word, with each pattern being mapped to one or more templates used as replies.

When searching for a match, the user's input was first normalized in exactly the same manner as the AIML patterns. The first word of the input was then used to find the list of patterns beginning with that word. Every pattern sentence in that list was compared to the input to find the closest possible match, and the template belonging to that match. In the case when had matching pattern more than one template mapped to it, one was chosen at random. The template was then returned to the user as the reply.

3.1.5 Most significant word implementation

Unlike the bot using first word pattern matching, the bot using most significant word pattern matching was implemented by indexing all the words of all the patterns in the knowledge base. Each word was then linked to the list of every pattern in the database which contained that word. Just like with the first word algorithm, every pattern was mapped to one or more template.

Starting the search for the closest match, the user's input is first normalized and then analysed to find the the most significant word in the sentence. The most significant word being the one word which occurs the least amount of times in the database. This word is used to get the list of all the patterns containing that word. All the patterns in the list are compared to the input sentence, and the closest match is returned along with the template mapped to that pattern. If more than one template was mapped to the template, one was chosen at random and given as a reply to the user.

3.1.6 Sentence comparison

The algorithm used for comparing the user input to a potential match was a version of the Levenshtein distance algorithm. The Levenshtein algorithm measures how many edits are required in order to turn one sentence into another. The edits in question being insertions, deletions, or substitutions. The difference between the two sentences increases as the amount of edits needed increases. [17]

3.2 Evaluation

To achieve the purpose of this paper a method was needed to evaluate the naturalness of the two different implementations.

3.2.1 Data collection

As motivated in the Evaluation of Chatbots section in the Background chapter, the quality of chatbots needs to be estimated by a number of human judges, both to evaluate naturalness as well as avoid relying on opinion. However, this type of evaluation is not an easy task. This issue has been solved by some competitions [robo chat, chatterbox] by developing an easy to use judging system and inviting multiple people to participate. The data collection part of the evaluation in this study is based on this system as it suits both the general criteria for evaluating chatbots, and the particular needs of this study, well. Some modifications have been made to better accommodate the fact that there are only two bots. An extra feature was also added in which participants also judge the quality of each bots match to the input question which was added to evaluate the significance of the results.

According to this concept, a questionnaire was prepared 25 predetermined questions asked of each bot along with the answer and the input match that each chatbot gave to it. The quantity of questions, 25, was chosen as it was higher than the average competition, 15, but still reasonably low so that participants would be able to finish the questionnaire in a reasonable amount of time. The questions themselves were a random selection from previous competition questions. Users then determined which, if either, of the two bots gave the better match to the input and most natural answer on a scale from 1 to 5 where 1 corresponded to “A is a lot better” and 5 to “B is a lot better”.

A large quantity of people were invited to partake in the experiment by completing the questionnaire and data was collected from their answers to form the results.

3.2.2 Data preparation

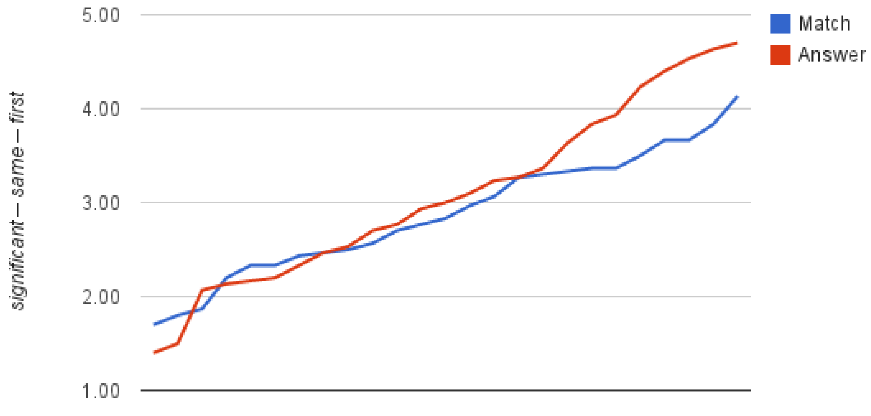
Results of the questionnaire was compiled into several diagrams visualising the distribution of the participants answers. Statistical analysis was performed on the results using the Wilcoxon signed-rank test to determine if they were statistically certain. The purpose of this was to determine if any observed difference between the two algorithms could be the result of natural variations. Only if there was a 5% or less chance of getting the observed results when there is in reality no difference will be counted as statistically certain difference. [26]

4 Results

4.1 Data

In the study, a total of 31 participants provided their opinions on the input matchings and the given answers on 25 questions. The results are presented in the following diagrams.

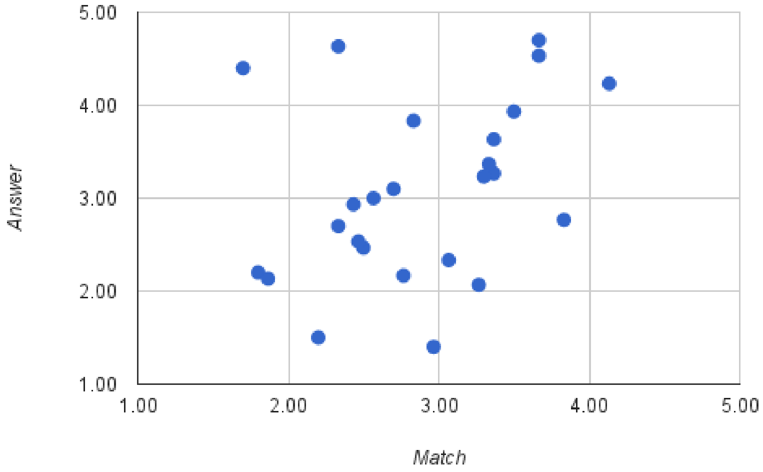
Figure 1: Distribution of match quality and answer naturalness



This diagram shows the distribution of the average opinion of the participants on the quality of the match to the user input and the naturalness of the bots given answer. While the results here are sorted and the lines not comparable to each other, it is a lucid presentation of the results for each of the two evaluation points regarding which of the bots was the better.

In the diagram, a line closer to 1 - lower - represents that the significant word search yielded better results while a line closer to 5 - higher - represents that the first word search was the superior. Close to the 3 shows them as about equal.

Figure 2: Correlation between the perceived quality of the input match and output answer



In this diagram each dot represents a question and its position in the diagram shows its average match quality and answer naturalness.

In the diagram, a line closer to 1 - lower and to the left - represents that the significant word search yielded better results while a line closer to 5 - higher and to the right - represents that the first word search was the superior. Close to the middle shows them as about equal.

4.2 Statistical analysis

A two-tailed Wilcoxon signed-rank test was done on the results to conclude if they were statistically certain at 5%. The null hypothesis used is that the median difference between the versions are the same.

Answer	**Match**
N=24	N=25
Z-value = -0.3571	Z-value = -0.8341
p-value = 0.71884	p-value = 0.40654
Significance 5%	Significance 5%

The z-values represents the number of standard deviations the respective results are from their average. The p-values, calculated from the z-values, are the respective probabilities that a result is at least as extreme as the observed ones would be obtained.

5 Discussion

5.1 Analysis of the results

Looking at the diagram of the result distribution it is clear that both lines are about equally above and below the middle as well as equally distributed among the intermediate values suggesting that there is no difference between the two versions in either input quality of answer naturalness.

Analysing the scatter pattern of the correlation between match and answer reveals that a clear line should form diagonally if a good input match resulted in a natural answer. The result here can only very loosely be described as a line and as such it indicates that a good match does not result in a natural answer. This is a source of error for this study.

In the statistical analysis; comparing the p-values of both the input match quality and answer naturalness we find that both are well above the 5%-significant level at 72% and 41% respectively. In other words, there is a 72% or 41% chance that any observed results will be at least as extreme as these ones.

5.2 Overall discussion

As suggested by the results, a good input match did not correlate well to a natural answer. This leads us to question the quality of the knowledgebase used in the study, that is knowledgebase based on movie scripts. While movie scripts are a fast way to gain a vast amount of data within a reasonable time frame there are still many alterations needed before they might be applied to this sector of application with good results. This is a source of error in this study as the poor quality of knowledgebase might have influenced the results more than anticipated and left the implementation only as a minor factor in the results. For this reason we believe that the study would have to be repeated using a better knowledgebase to get more dependable results.

6 Conclusion

Based on the results in this study we are unable to conclude that there is any statistical difference between first word search and most significant word search algorithms in either the quality of the input match or the naturalness of the answers.

References

[1] Alicebot language libraries. http://www.alicebot.org/downloads/sets.html. Accessed April 13, 2014.

[2] Chatbot for diabetic patient. http://www.slideshare.net/harshitg3/chatbot-for-diabetic-patient#. Accessed March 4, 2014.

[3] Chatterbot. http://en.wikipedia.org/wiki/Chatterbot. Accessed April 13, 2014.

[4] Chatterbotchallenge. http://www.chatterboxchallenge.com/. Accessed March 4, 2014.

[5] Cleverbot. http://www.cleverbot.com/. Accessed March 4, 2014.

[6] Cleverbot. http://www.cleverbot.com/. Accessed March 4, 2014.

[7] Customer service chatbots. http://www.chatbots.org/applications/customer_service/. Accessed April 13, 2014.

[8] Loebner prize. http://www.loebner.net/Prizef/loebner-prize.html. Accessed March 4, 2014.

[9] Robochatchallenge. http://www.robochatchallenge.com/. Accessed March 4, 2014.

[10] Siri. http://en.wikipedia.org/wiki/Siri. Accessed April 13, 2014.

[11] Subrip. http://en.wikipedia.org/wiki/SubRip. Accessed April 13, 2014.

[12] Tobi. http://www.tobii.com/. Accessed March 4, 2014.

[13] Who made that captcha. http://www.nytimes.com/2014/01/19/magazine/who-made-that-captcha.html?_r=1&module=ArrowsNav&contentCollection=Magazine&action=keypress&region=FixedLeft&pgtype=article. Accessed April 13, 2014.

[14] das Graças Bruno Marietto, M., de Aguiar, R. V., de Oliveira Barbosa, G., Botelho, W. T., Pimentel, E., dos Santos França, R., and da Silva, V. L. Artificial intelligence markup language: A brief tutorial. *CoRR abs/1307.3091* (2013).

[15] Dewar, R. B. The setl programming language. 1979.

[16] Gianvecchio, S., Xie, M., Wu, Z., and Wang, H. Measurement and classification of humans and bots in internet chat. In *Proceedings of the 17th Conference on Security Symposium* (Berkeley, CA, USA, 2008), SS'08, USENIX Association, pp. 155–169.

[17] Gilleland, M. Levenshtein distance, in three flavors, 2009.

[18] Mitkov, R., Orasan, C., and Evans, R. The importance of annotated corpora for nlp: the cases of anaphora resolution and clause splitting, 1999.

[19] Oppy, G., and Dowe, D. The turing test. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., spring 2011 ed. Stanford, 2011.

[20] Schubert, L. Computational linguistics. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., spring 2014 ed. 2014.

[21] Shawar, B. A. Chatbots are natural web interface to information portals.

[22] Shawar, B. A., and Atwell, E. Different measurements metrics to evaluate a chatbot system. In *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies* (Stroudsburg, PA, USA, 2007), NAACL-HLT-Dialog '07, Association for Computational Linguistics, pp. 89–96.

[23] Wallace, R. The elements of aiml style. *Alice AI Foundation* (2003).

[24] Wallace, R. The anatomy of a.l.i.c.e. In *Parsing the Turing Test*, R. Epstein, G. Roberts, and G. Beber, Eds. Springer Netherlands, 2009, pp. 181–210.

[25] Wang, Jinyu; Brewton, S. Making xml technology easier to use, 2005.

[26] Woolson, R. F. *Wilcoxon Signed-Rank Test.* John Wiley Sons, Inc., 2007.