

Universität Hamburg
Department Informatik
Dept. Informatik – Knowledge Technology, WTM

Chatbots: Review, sentiment analysis and a comparison human-chatbot

Seminar Paper

Bio - Inspired Artificial Intelligence

Miguel Angel Robles Urquiza — Rafael Ruz Gomez

Matr.Nr. 7023522 — 7019703

migueurquiza@gmail.com — rafaroco96@gmail.com

24-01s-2018

Abstract

In this paper we are going to do a research on the automatic learning of the chatbots, focus in sentimental chatbots. First, we will study what is a chatbot, studying its origins (E.L.I.Z.A) and seeing another famous bots as A.L.I.C.E. and Cleverbot. Once this is done, we will see what are the different types of chatbots that exist today, seeing why we need them, analyzing their behavior with the feelings, analyzing a tool that allows us to know the polarity of the messages of the Twitter social network, one of the most famous social networks used in this decade. We will also study how using tools such as Pattern, a sentimental library made in Python, we can build our own program that detects the positivity or negativity of a given message.

Once we have introduced the different types of chatbots that exist, we will analyze how similar is the language that a robot can get to know with the language of a human person. We will see how the Turing Test are able to rank the different existing chatbots, thus giving rise to the Loebner Awards, an annual competition that rewards the robot that can make a human being believe that it is not.

Contents

1	Introduction	2
2	A review of chatbots	3
2.1	E.L.I.Z.A	3
2.2	A.L.I.C.E.	6
2.3	Other interesting Chatbots	8
3	Types of chatbots	8
3.1	Sentimental-chatbots	8
3.2	Assistant-chatbots	12
4	Comparison between chatbots and humans	13
4.1	Loebner prizes	15
5	Conclusion	16
	Bibliography	17

1 Introduction

Technology is changing the way we communicate. Let's start by defining what chatbots are. "*Chatbots are computer programs that interact with users using natural languages.*" [SA07]

A chatbot is a conversational bot whose objective is to interact with the user, asking question, giving answers or even initiating new topics of conversation. [HZY07]

A chatbot is programmed with its own information and receive together the necessary information, reformatting and presenting it in a way that meets the needs of the user. How do they respond accurately? Most chatbots solve this problem by using Case Based Reasoning(CBR). The CBR is able to use the previously experienced specific knowledge, that is, specific problematic situations (cases). Therefore, look for a similar previous case and its reused in the new problematic situation, adapting it to the appropriate conditions and getting to its solution. [Kol14]. There are also other ways to make a chatbot works, for example, using patterns. The program is programmed with response patterns that associate what the user enters with answers stored in a database. X being a variable, the program would identify it as a key phrase "I have X". One pattern could be to have the program respond to the user "I also have X".

Usually, the artificial intelligence in a chatbots is built using concepts from Natural Language Interaction (NLI). NLI is the technology that allows computers and humans to communicate using natural language (the way we speak to other humans) instead of using a set of programming codes. The advantage of NLI processing is the additional ability to use the phrasing (verbs, nouns, adjectives, etc.) from the input to supply an answer that is more sensitive to the intent of the question.

Modern commercial chatbots, such as those developed with LingubotTM [Cre06] technology, offer sophisticated development environments allowing the building of intelligent conversational agents with complex, goal driven behaviour. In Lingubots both the words and the grammatical structure of the users input are analysed using customised templates. This facilitates the development of a user model, which is used in conjunction with the conversational context and specific words in the dialogue to determine the chatbots response. Responses might include further conversation with the user, reading or writing to external systems (for instance to open a web page or update a database), or a combination of these. This rich range of responses allows for intelligent conversation with the user, and provides the ability to steer the user back to the task in hand if they stray from the designated discussion content for too long.

The biggest difficulty that resides in chatbots is in distinguishing when a phrase is context dependent or not. When it comes to independent sentences the bot should not look for the information in the sustained dialogue, they are common phrases when the answers to the questions are prepared. But when they are not dependent, the bot must be able to have previously understood the context of the conversation, to be able to respond appropriately.

Originally, chatbots only responded to written text. In the last decade chatbots became more versatile and included speech synthesis and recognition, and affective state detection and responses. One example is ,[CO12] .

As computing technology and the underlying language processing software progresses, we can expect to see potentially exponential growth in the delivered complexity of chatbots. Already, they have come a long way from their roots in systems that were more about fun, flirtation or simple "chat". So, we are going to go deep in the roots of chatbots

2 A review of chatbots

Before we get to how sentimental analysis works in sentimental chatbots, we need to talk about the origins of chatbots. Let's see the first of all, created in the decade of the 60s, E.L.I.Z.A. and we will also see in depth the chatbot A.L.I.C.E. created years later in the decade of the 90s. We are going to see what they did and how they worked

2.1 E.L.I.Z.A

The need to create virtual machines that speak as if they were human beings born in the decade of the 60s. E.L.I.Z.A was born, by the hand of Joseph Weizenbaum, professor emeritus of Informatics at the Massachusetts Institute of Technology. At that time, this chatbot was created to emulate a psychotherapist in clinical treatment. The objective of this teacher when he created the program was simple and made use of keyword matching. A keyword is searched in the input, and if this keyword is in the database, a previously programmed response is answered, and if this does not happen, the program returns a random phrase to continue obtaining information. For example, if the input includes the keyword "mother", E.L.I.Z.A can respond "Tell me more about your family" [SA07].

E.L.I.Z.A is basically composed of a dictionary of pronouns, a list of matches and 3 functions.[Vei66] The dictionary of pronouns is basically responsible for entering the correct pronouns in the answers given by the machine, that is, if the user enters "my father" as input, the machine returns a response in which it says "your father. The list of matches is responsible for linking keywords that the user enters with possible answers related to that keyword. For example, if the user enters the word "child", the machine will probably respond: "Did you have close friends as a child?" or "Did the other children sometimes tease you?"." [Vei66]

Now we are going to see how the functions work. We have implemented in Python, with the help of [Thi17], the E.L.I.Z.A chatbot, in order to understand more easily how the first chatbot of all times worked. Let's do the analisis with the help of some images with the program code.

2.1.1 Functions

- Reflect

```
def reflect(fragment):
    tokens = fragment.lower().split()
    for i, token in enumerate(tokens):
        if token in reflections:
            tokens[i] = reflections[token]
    return ' '.join(tokens)
```

Figure 1: Reflect

The first function that appears to us is called reflect. We iterate through the list of tokens and, if the token exists in our dictionary, we replace it with the value from the dictionary.

- Analyze

```
def analyze(statement):
    for pattern, responses in psychobabble:
        match = re.match(pattern, statement.rstrip(".!"))
        if match:
            response = random.choice(responses)
            return response.format(*[reflect(g) for g in match.groups()])
```

Figure 2: Analyze

The second function that appears is called analyze. We look for if any word in the dictionary match with any word in the user's statement, from which we have stripped the final punctuation. If we find a match, we choose a response template randomly from the list of possible responses associated with the matching pattern. If not, we ask for more information.

- Main

```
def main():  
    print "Hello. How are you feeling today?"  
  
    while True:  
        statement = raw_input("> ")  
        print analyze(statement)  
  
        if statement == "quit":  
            break
```

Figure 3: Main

Finally, we have the main function that runs the program. First, we show the initial sentence on the screen, to begin the conversation with the user. From there and until the user does not want to stop, we continue having a conversation with him and analyzing his answers with the function presented in figure 2.

- Example

Here is an example of the output of the program. We can see exactly what we are referring to in section X. We have given E.L.I.Z.A an entry with the word "child", and it has given us using pattern matching, a response that fits into the conversation. Then we have given E.L.I.Z.A a phrase with words that it does not have in its database and what it has done has been to return a random phrase from the list to continue obtaining information.

```
urqui@urqui:~/GIT/BAI$ python eliza.py  
Hello. How are you feeling today?  
> Very good, I feel like a child today  
Did the other children sometimes tease you?  
> No, I was happy when I was a child  
How do you think your childhood experiences relate to your feelings today?  
> They are very related  
I see. And what does that tell you?
```

Figure 4: Example

We see that the program designed by Mr. Joseph Weizenbaum was a very simple program that response according to previously created templates. With the creation of this program we could already see the first brushstrokes of artificial intelligence in a chatbots.

2.2 A.L.I.C.E.

If we are talking about chatbots, we can not forget to talk about A.L.I.C.E.. A.L.I.C.E. is the Artificial Linguistic Internet Computer Entity, first implemented by Wallace in 1995.

As [SA02] says, A.L.I.C.E. (Artificial Linguistic Internet Computer Entity), first implemented by Wallace in 1995, is a robot that you can make chatting with it. A.L.I.C.E. knowledge is stored in AIML files. AIML is an abbreviation of The Artificial Intelligent Mark up Language that is a derivative of Extensible Mark up Language(XML). Let's see A.L.I.C.E. deeply:

2.2.1 AIML

As we can see in [Wal03], the categories in AIML are the fundamental units of knowledge. A category is at least two more elements, but it can has optional elements. Basic elements are the patterns and templates, which are usually coded in that order (first the pattern and second the template) because patter would be the question and template would be the answer. The optional context portion of the constant category of variants, called <that>and <topic>. The <that>tag appears within the category, and its pattern must match the last statement of the robot. Remembering a last statement is important if the robot asks a question. The <topic>tag appears outside the category and brings together a group of categories.

2.2.2 Preparation for Pattern Matching

Before starting pattern matching procedures each input to the AIML interpreter must pass through two processes:

1. Normalization Process:

In this step, we intend to retain most of the input information while we adapt that entry to be able to use it well. The recognition of punctuation marks is used mostly and the letters are capitalized. Here we have a table with some examples of this process step by step.

"I like that. :)"	"I like that. "	"I like that"	"I LIKE THAT"
"I don't have. Do you have?"	"I do not have. Do you have?"	"I do not have"	"I DO NOT HAVE"
		"Do you have?"	"DO YOU HAVE"

2. Producing input path from each sentence.

This path has the following form:

Input <That>Tvalue<Topic>Pvalue, where:

- Input is the normalized input
- <That>previous bot answer, normalized in the same way as input
- Tvalue is the previous bot answer if exist and * if it doesn't exist.
- <Topic>is an optional top level element that contains categories and has a name attribute assign to it, in order to allow interpreter to prefer responses. dealing with that topic.
- Pvalue is the topic name if exists and * if it doesn't exist.

2.2.3 Pattern Matching Behaviour

AIML interpreter try to match word by word in order to get the largest pattern matching, that is, the best one. It uses the Graphmaster. The Graphmaster consists of a collection of nodes called Nodemappers. These Nodemappers map the branches from each node. The branches are either single words or wildcards.

Algorithm :

1. Does the Nodemapper contain the key "_" ?
2. If so, search the sub graph rooted at the child node linked by "_" .
3. Try all remaining suffixes of the input following "X" to see if one matches.
4. If no match was found, try:
5. Does the Nodemapper contain the key "X" ?
6. If so, search the sub graph rooted at the child node linked by "X", using the tail of the input (the suffix of the input with "X" removed).
7. If no match was found, try:
8. Does the Nodemapper contain the key "*" ?
9. If so, search the sub graph rooted at the child node linked by "*" .
10. Try all remaining suffixes of the input following "X" to see if one matches.
11. If no match was found, go back up the graph to the parent of this node, and put "X" back on the head of the input.

For completeness there should also be a terminal case: If the input is null (no more words) and the Nodemapper contains the `template` key, then a match was found. Halt the search and return the matching node. If the root Nodemapper contains a key "" and it points to a leaf node, then the algorithm is guaranteed to find a match.*

2.3 Other interesting Chatbots

- In [VL15] a sub symbolic chatbot is presented that uses machine learning. It appears that it can handle the general cases of conversation, even questions it didnt train upon. The strength of the model is that it can be trained end-to-end and thus requires much fewer hand-crafted rules. The authors state that its answers are sometimes on the short side, and that slight variance in semantics can result in inconsistent answers. Another issue we have with such a methodology is that its a completely opaque process.
- First launched on the internet by Rollo Carpenter in 1997, Cleverbot is a web service that offers the possibility of interacting by chat message with artificial intelligence in real time. It is an intelligent social robot capable of talking about any topic, but its nature does not allow it to carry out processes such as: remembering long conversations or contextualizing; necessary for interaction and conversation.[Mol13]

In [Mol13] we can read that since 1991, computer programs capable of simulating a human conversation have been awarded with the Loebner prizes. These, are granted those programs that manage to hire an examiner, who is isolated from everything, and who is in front of a computer; which through written text, dialog with him. This test is known as Turing Test [Nao96], and it is the origin of many web services, programs and applications that can be offered to the public, stories such as search assistants, anti spam systems or human understanding interfaces. Of the services of chatbot, or conversational robot of public access existing at present, Cleverbot -www.cleverbot.com- one of the best known and most "human" that exist in the market. And one of the few that has managed to pass the Turing Test with a score higher than 50 %. This implies that in front of its human counterpart, more than half of examiners mistakenly determined that a person hidden behind a computer was the chatbot, and that the robot was a person.

3 Types of chatbots

3.1 Sentimental-chatbots

Why would we want to add sentiment analysis to our chatbot? The reasons can be several, like adjusting better to the topic of the conversation, getting more natural answers from our chatbot or maybe just develop a chatbot to tell us something to make us feel better, someone to be with in bad times or someone to share happiness with.

But before we start talking about sentimental chatbots, let's talk about sentiment analysis.

3.1.1 Sentiment analysis

The sentiment analysis is an important factor to take into account in any company, either to adjust our marketing strategy or to improve customer service, for example. Therefore it is also an important factor for any chatbot that works in a company, for example functioning as a chatbot for customer service.

Opinion mining is usually called also sentiment analysis. According to Bing Liu from the University of Illinois:

"Given a set of evaluative text documents that contain opinions (or sentiments) about an object, opinion mining aims to extract attributes and components of the object that have been commented on in each document $d \in D$ and to determine whether the comments are positive, negative or neutral" [lu08].

In other words, we have some documents containing subjective information and our goal is to evaluate this information in order to assign a sentiment or positiveness value to each comment or document.

As B. Pang and L. Lee say, there are several problems in the sentiment analysis [PL08]:

- First of all, we have to determine which documents or webpages have relevant information for sentiment analysis, that's opinions, reviews, etc.
- Then it comes the problem of which portions of the documents or the information we are analyzing are useful for the sentiment analysis.
- After this we have to determine whether the opinion or information expresses a positive or negative feeling, or directly with what feeling it corresponds. In some pages like Amazon it can be easier since in the comments section, each comment is associated with a product evaluation that usually reflects the user's agreement and therefore the degree of positivity of the comment, but in other occasions in which we have no clue can be complicated.
- Finally we must internally represent in our chatbot or sentiment analysis program that feeling, since not in all the web pages or documents in which we are given a previous valuation the same measurement system is used (Amazon uses a 5 stars system, Yahoo answers uses a like-dislike system, etc) and we must normalize these values.

There are many tools that allow us or help us to analyze sentiment easily. Most of them are convolutional neural networks trained with huge datasets.

For example, there are already developed free tools that we can use directly in our sentiment analysis like **"pattern.en"**.

"The pattern.en module contains a fast part-of-speech tagger for English (identifies nouns, adjectives, verbs, etc. in a sentence), sentiment analysis, tools for English verb conjugation and noun singularization & pluralization, and a WordNet interface." [pyta]



Figure 5: Illustration of the patter.en python module working.

Let's take a deeper look into the sentiment section of this module. This module has two functions for sentyment analysis [pytb]:

- `sentiment(sentence)`: It's the main function for sentiment analysis in this module . As we can read in the webpage [pytb], it gives us two values:
 - `polarity`: It takes values from -1.0 to 1.0. It tell us if the sentence is positive or negative. The algorithm is based on the adjectives of the sentence.
 - `subjectivity`: It takes values from 0.0 to 1.0.
- `positive(s, threshold=0.1)`: It returns True if the given sentence's polarity is above the threshold. We can use this function to detect stronger ir lighter sentiments giving different values to the threshold. *"Accuracy is about 75% for movie reviews"* [pytb] so we can see that there are good tools free-to-use in our sentiment analysis.

In another example, Cicero Nogueira dos Santos and Maira Gatti propose a **deep convolutional neural network for sentiment analysis of short texts** [GBB11].

In this case the neural network takes a sentence as an input and returns a score for each sentiment label in a set.

In order to do that, every word is converted into a vector, which is composed of two sub-vectors:

1. Word-level: It captures syntactic and semantic information.
2. Character-level: It captures morphological and shape information. s

Once we have the sentence converted into word-level and character-level embeddings, the next step they do in CharSCNN consists in extracting a sentence-level representation, but there are two main problems: sentences have different sizes and important information can appear at any position in the sentence.

To solve this problem they use a second convolutional layer: *"This layer produces local features around each word in the sentence and then combines them using a max operation to create a fixed-sized feature vector for the sentence."*[GBB11].

Finally, to finish their neural network architecture, this feature vector of the sentence is processed by two neural network layers, which extract one more level of representation and compute a score for each sentiment label. This is the output, as said before.

We can use aswell tools as a help for our algorithms and neural networks in sentiment analysis such as **"Tweet sentiment application"**¹ developed by Dr. Christopher Healey that allows you to search for tweets based on keywords and shows us information about tweets containing those keywords that can be very useful such as Sentiment, Topics, Heatmap, Tag Cloud, Timeline, Map, Affinity, Narrative and tweets.

The most useful for us is the Sentiment information that show us a graphic representation of the sentiments shown in the tweets: *"Sentiment. Each tweet is shown as a circle positioned by sentiment, an estimate of the emotion contained in the tweet's text. Unpleasant tweets are drawn as blue circles on the left, and pleasant tweets as green circles on the right. Sedate tweets are drawn as darker circles on the bottom, and active tweets as brighter circles on the top. Hover your mouse over a tweet or click on it to see its text"* [Hea].

¹https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/



Figure 6: As we can see, most of the tweets associated to the keyword "fireworks" are positive. This can be useful when parsing some texts containing some words we don't know how to process.

3.2 Assistant-chatbots

Now we are going to research about Assistant chatbots. As we can read in [IB13], a chatbot with natural language algorithms and localised data could be used as a virtual personal assistant. However, this will require the program to be able to create metadata by forming links between data in given by the user and provide contextual outputs during the interaction that the user finds interesting and useful. The difference between the way that each chatbot examines the personal data allow us to categorize the apparently similar programs depending on where the data remains:

- *Information service providers:* They are systems that carry out requests but also provide support for the organisation behind the system. These programs don't have a localised library for their natural language processing. That means the program is not focusing on each individual user, although the data generated by the analysis of the user activity is saved and controlled by the Company. An example of this type of chatbot is Siri from Apple.
- *Content providers.* They are systems that provide the information to be used by other systems. They explore data from existing databases and create its own metadata. They interact with content providers and become into a large data libraries.
- *Mediation service:* They are systems that function as an intermediate between the user and other services. Mediation service could be really useful because it acts as a direct connection between the services and possible multiple users.

- *Third party service:* Contains services that can be applied to a system for its own improvement. This system doesn't provide services for the users, it gives support for the systems that the user is using. When this system belongs to the user, the third party system can't interact. Finally, the system could act as an intermediary for the support systems.
- *User owned services:* They work for the user personally, allowing private information kept between the user and the system. The support system does not only have local data and libraries, it also develops metadata and contextual data as a result of use analysis and history of interaction with user. All of contextually developed content in the data library is owned and under the control of the end user.

4 Comparison between chatbots and humans

How can we measure if our chatbot is "human" enough? Or how can we see the differences of our chatbot with respect to a human?

In 1950, Alan Turing proposed the Imitation Game as a replacement for the question, Can machines think? He predicted that by the year 2000 technological progress would produce computing machines with a capacity of log bits, and that with such machinery, a computer program would be able to fool the average questioner for 5 minutes about 70% of the time [Tur09]

In chatbots area, we usually measure the "human intelligence" level of chatbots using the Turing test.

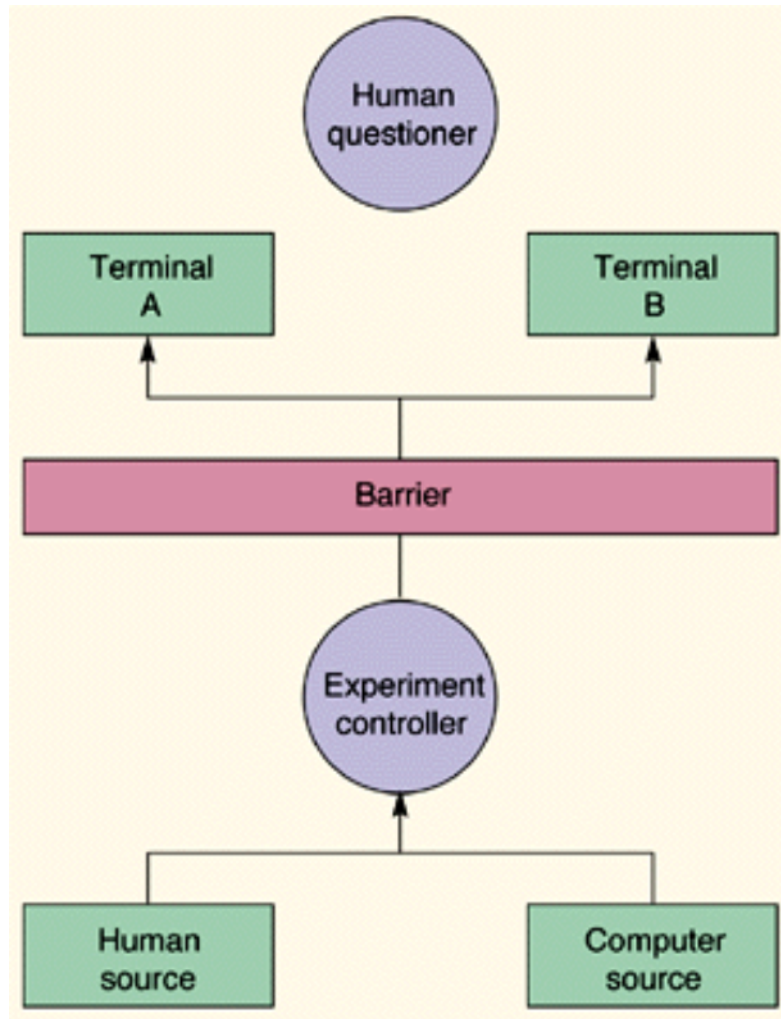


Figure 7: Graphic illustration of the turing test.[tur]

As we can see, it has the following elements:

- An human questioner or tester, who ask questions and chats using 2 terminals.
- Two differents sources to chat with: An human source and a computer source that we want to evaluate, int this case a chatbot.
- An expermint controller checks if everything is going ok.
- A barrier to isolate the tester from the sources.

After that, we ask the human questioner which source he thinks it's the human source and which one the computer source. Finally, based on the answers of the testers, we can measure whether the intelligence of our chatbot is enough or not to pass the test.

For example, *"Cleverbot passed the 2011 Turing Test at the Techniche TechnoManagement Festival held by the Indian Institute of Technology Guwahati. Of the 1334 volunteers who participated in four-minute typed conversations with either Cleverbot or real humans, 59% rated Cleverbot as human while 63% rated the real humans as human (Aron, 2011), suggesting that Cleverbot is one of the most advanced and human-like conversational agents currently available to the public."*[HFF15]

4.1 Loebner prizes

In [Exe] we can read that "The Loebner Prize, named after its founder and philanthropist Hugh Loebner, is an annual world-wide contest to test the state-of-the-art in artificial intelligence (AI)."

In 1991, Dr. Hugh Loebner, the National Science Foundation, and the Sloan Foundation started the Loebner Prize Competition: an annual contest between computer programs to identify the most human programs, and eventually to award \$100,000 to the program that first passes an unrestricted Turing test [Eps92]. This competition has been criticized as a parlor game, rewarding tricks rather than furthering the field of Artificial Intelligence [Shi94].

For example, those are the last year Loebner prizes:

- 2016: Mitsuko created by Steve Worswick
- 2015: Rose created by Bruce Wilcox
- 2014: Rose
- 2013: Mitsuko
- 2012: Chip Vivant! created by Mohan Embar
- 2011: Rose
- 2010: Alice bot created by Richard Wallace

5 Conclusion

We, in the first place, have learned in this paper about the history of chatbots, a quite recent area of study in natural language processing through artificial intelligence. We have seen in depth the complete functioning of the chatbot that created Joseph Weizenbaum (E.L.I.Z.A.), and as a result of this have emerged many more chatbots today. We want to emphasize the importance of developing the idea of chatbots, because it is something that is practically used in all companies that have customer service. It is also evident that like most of this type of tools, you can get tools that are simply entertainment, for example cleverbot.

The sentimental analysis in chatbots is another area that is not very developed due to the recent emergence of convolutional neural networks, which represent the main tool in the development of programs that are able to interact with a human person as if they had feelings. Despite this we have seen how there are many tools available to users, trained with huge sets of data that allow a good approximation for such analysis.

Some of the advantages of the implementation or addition of analysis of feelings to our chatbot have directly to do with the analysis of feelings for any company, for example, the analysis of data in social networks to improve marketing campaigns, create chatbots that empathize better with the user or client, etc.

Finally, we also want to give importance to the Loebner awards, since rewarding the implementation of programs that simulate human intelligence, where chatbots are included, is a way to motivate the development and research in this field.

References

- [CO12] Collette Curry and James D O'Shea. The implementation of a story telling chatbot. *Advances in Smart Systems Research*, 1(1):45, 2012.
- [Cre06] CreativeVirtual.com. *Creative Virtual*, 2004-2006.
- [Eps92] Robert Epstein. The quest for the thinking computer. *AI magazine*, 13(2):81, 1992.
- [Exe] University Exeter.
- [GBB11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520, 2011.
- [Hea] Ramaswamy Healey. "tweet sentiment analysis".
- [HFF15] Jennifer Hill, W. Randolph Ford, and Ingrid G. Farreras. Real conversations with artificial intelligence: A comparison between humanhuman online conversations and humanchatbot conversations. *Computers in Human Behavior*, 49:245 – 250, 2015.
- [HZY07] Jizhou Huang, Ming Zhou, and Dan Yang. Extracting chatbot knowledge from online discussion forums. In *IJCAI*, volume 7, pages 423–428, 2007.
- [IB13] Peter Imrie and Peter Bednar. Virtual personal assistant. In *ItAIS 2013*. AIS Electronic Library (AISeL), 2013.
- [Kol14] Janet Kolodner. *Case-based reasoning*. Morgan Kaufmann, 2014.
- [lu08] Bing lu. Opinion mining. 2008.
- [Mol13] Iñigo Benito Molinero. El absurdo en arte e inteligencia artificial. 2013.
- [Nao96] Moni Naor. Verification of a human in the loop or identification via the turing test. *Unpublished draft from http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human_abs.html*, 1996.
- [PL08] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(12):1–135, 2008.
- [pyta] <https://www.clips.uantwerpen.be/pages/pattern-en>.
- [pytb] <https://www.clips.uantwerpen.be/pages/pattern-en#sentiment>.
- [SA02] Bayan Abu Shawar and Eric Atwell. *A comparison between ALICE and Elizabeth chatbot systems*. University of Leeds, School of Computing research report 2002.19, 2002.

- [SA07] Bayan Abu Shawar and Eric Atwell. Chatbots: are they really useful? In *LDV Forum*, volume 22, pages 29–49, 2007.
- [Shi94] Stuart M Shieber. Lessons from a restricted turing test. *arXiv preprint cmp-lg/9404002*, 1994.
- [Thi17] Small Sure Thing. *Implementing The Famous ELIZA Chatbot In Python*. <https://www.smallsurething.com/implementing-the-famous-eliza-chatbot-in-python>, 2017.
- [tur] <https://annamaeduardo.files.wordpress.com/2014/01/turing-test.gif>.
- [Tur09] Alan M Turing. Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65. Springer, 2009.
- [Vei66] Joseph Weizenbaum. *ELIZA - a computer program for the study of Natural Language Communication between man and machine*. 1966.
- [VL15] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.
- [Wal03] Richard Wallace. The elements of aiml style. *Alice AI Foundation*, 2003.