Learning from Interaction: An Intelligent Networked-based Human-bot and Bot-bot Chatbot System

Jordan J. Bird, Anikó Ekárt and Diego R. Faria

Aston Lab for Intelligent Collectives Engineering (ALICE)
School of Engineering and Applied Science
Aston University, Birmingham, B4 7ET, UK.
{birdj1, a.ekart, d.faria}@aston.ac.uk

Abstract. In this paper we propose an approach to a chatbot software that is able to learn from interaction via text messaging between human-bot and bot-bot. The bot listens to a user and decides whether or not it knows how to reply to the message accurately based on current knowledge, otherwise it will set about to learn a meaningful response to the message through pattern matching based on its previous experience. Similar methods are used to detect offensive messages, and are proved to be effective at overcoming the issues that other chatbots have experienced in the open domain. A philosophy of giving preference to too much censorship rather than too little is employed given the failure of Microsoft Tay. In this work, a layered approach is devised to conduct each process, and leave the architecture open to improvement with more advanced methods in the future. Preliminary results show an improvement over time in which the bot learns more responses. A novel approach of message simplification is added to the bot's architecture, the results suggest that the algorithm has a substantial improvement on the bot's conversational performance at a factor of three.

Keywords: Artificial Intelligence, Natural Language Processing, Chatbot

1 Introduction

Both Artificial Intelligence (AI) researchers and industry are increasingly recognising the importance of chatbot systems. Chatbots are embedded in everyday life, performing various roles serving as assistants for end-users. Examples include consumer interaction geared towards problem-solving and addressing customer requests in a variety of industries [1] and even performing as a virtual home-assistant in home-automation duties within a modern household [2].

In this work, a Chatbot system learns through experience to respond to certain messages based on previous interactions, and gains more knowledge over time based on its interaction with users.

The main contributions of this work are three-fold:

1. A modular web-based chatbot system useful for different applications and with the potential in industry is presented.
2. Natural Language Processing (NLP) and Sentiment Analysis are applied for offensive vocabulary detection to prevent the current issues faced by chatbots in the open domain [12].
3. A novel approach is introduced for a message simplification layer in which colloquialisms and identical synonyms are mitigated by simplifying them into marking flags, and building responses in real-time.

This paper will first explore the related works (Section 2), highlighting milestones as well as open issues currently existing in the field. The proposed approach (Section 3) will then be detailed, followed by a preliminary set of results (Section 4). The paper then concludes with a discussion of results, future work and impact (Section 5).

2 Related Work

Sentiment analysis (a.k.a. opinion mining) identifies people's opinions, sentiments, emotions, appraisals and attitudes towards entities such as services and products, organizations and/or individuals, events and their attributes. Chatbot systems are tools that employ natural language processing and sentiment analysis for multiple purposes, examples include human-machine communication applied to business, health, and education. The objective measurement of the users experience and emotional states is a key factor to understand the level of satisfaction or mood given a certain context. Tackling this problem is very challenging, since the measurements of satisfaction levels or internal states might be too ambiguous, but new emerging technologies combined with computational intelligence can provide a new perspective for approaching the problem in a more effective manner. There are many channels to detect and perform sentiment analysis such as affection through facial expressions [17][18], voice [19], and also through text communication via natural language processing [4]. Body gesture can also be an interesting alternative for user behaviour or emotional expression understanding [20]. This work falls into the category of text communication between two agents (human-bot or bot-bot) using natural language processing and sentiment analysis via an intelligent web-based chatbot system. Related works are presented below to show the progress of chatbot systems in different domains, their challenges and open issues in this field.

Weizenbaum's ELIZA [22] system is often considered as the first ever chatbot system, which quickly became famous in the 1960s due to people engaging in conversations with it beyond the expectations of its creator.

Shawar and Atwell's exploration of using dialogue corpora to generate chatbots [3] found that a non-bot human experience (such as the script of two human beings conversing) was more than capable of forming the basis for a chatbot. This therefore suggests that rather than starting with no knowledge whatsoever, a dataset of conversations

could be provided to give the chatbot a useful head start in learning from interaction. A response format would need to be developed to organise this data into.

In 2004, the effectiveness of chatbots deployed to teach students foreign languages was demonstrated [4]. The effectiveness of a robotic agent became clear when it was noted that the students were more open to admit fault to the bot rather than a person, showing that students were far less intimidated by a feeling of inadequacy during the learning process. This research is very promising, as it shows an application of a chatbot in education which would not only aid the education system, but further improve it.

Freudbot is an Artificial Intelligence Markup Language (AIML) based chatbot, an XML based approach to message pattern and response pairings. For experimental purposes it was designed to speak to students to aid with distance learning when staff were not available out of hours [5]. Heller et al. found that the experience was itself neutral, neither good nor bad, but explained that with more responses it would have been expected to be better. This suggests that more responses create a better experience for the user, and so learning from experience could constantly evolve a chatbot to become better with each and every interaction.

Tatai et al. ran tests of chatbots who preferred using different sentiments to see how a user would react and recorded how they found the experience [6]. Everyone, regardless of input sentiment, found a better experience when the chatbot response was positive. This suggests that a learning bot should respond to negative messages, but not risk reiterating them in the future, which would result in only positive responses, regardless of input. In terms of impact, this means the bot must analyse the sentiment of the messages in order to make use of this important work and as such allow the implementation of a chatbot that will not learn from negative messages.

The Loebner prize [7] is awarded to a chatbot which can fool a human into thinking they are conversing with another human. This is based on the “Imitation game” proposed by Turing in 1950 [21]. Shawar and Atwell [8] found that case-by-case training was required depending on localisation and audience. They also suggest that this problem could be overcome by having a more massive dataset to draw from. This suggests that larger datasets would improve the chatbot, but to contextualise it to an environment, the level of training required is dynamic and depends on its uses.

ALICE is an XML based chatbot [9] that makes use of pattern matching between user inputs as well as a hidden person to generate responses to a user’s message. The method of pattern matching to a stored set of messages was an effective enough method for the bot to have won the Loebner Prize three times [10] suggesting that the act of learning from interaction is an effective method of creating record-breaking chatbots, in terms of competitions that were won.

Tay, an AI chatbot that learns from previous interaction [11] caused major controversy due to it being targeted by internet trolls on Twitter [12]. The bot was exploited, and

after 16 hours began to send extremely offensive Tweets to users. This suggests that although the bot learnt effectively from experience, adequate protection was not put in place to prevent misuse. This implies two important influences - firstly, learning from interaction is reinforced as an effective technique. Secondly, the bot must have protection in place to prevent such an event from occuring.

The Google Cloud NLP API [13] is a library that can process a message and calculate sentiment and magnitude thereof. The API, once authorised, will receive a message as raw text, and then produce a response array containing pieces of sentiment information gathered from the input, therefore enabling us to be able to measure the sentiment of a message and deem whether or not it is positive or negative. This knowledge would allow for selective response elicitation based on positivity.

Despite the efforts spent on these extensive and high quality works in the area of scientific philosophy of chatbots, most, except for [4], are yet to produce systems capable of specifically long-term deployment into a real-world situation, rather concentrating on the learning theory behind their conception. This suggests that there is room for such systems in industry, making use of aforementioned techniques already well documented and published in respectable Machine Learning journals. To conclude, the level of current scientific knowledge is seemingly ahead of that employed in the real-world, and furthermore is not yet in a state to be used in real-world applications.

3 Proposed Approach

For user interaction with the bot, the user will input a message into an arbitrary front-end application. The bot makes use of an input/output structure of web-requests and therefore any application that accommodates this can be used. The system outputs its response in the form of a simple key-value JSON Array. To show the bot's potential, two front-end applications were developed (see Fig. 1), a browser-based application using asynchronous JavaScript to populate a chat window with results, and an automated Twitter bot that would perform the same process when replying to Tweets.

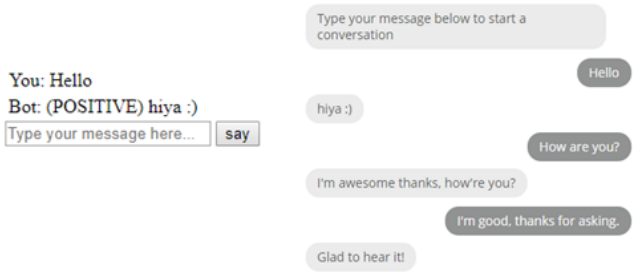

Fig. 1. HTML and UI bot interfaces.

System architecture. In terms of the bot itself, a black box system is developed (see Fig. 2) seen in steps 1 to 4.1.1, the message from the user is processed and results are generated accordingly. Firstly, to avoid the the aforementioned Microsoft Tay disaster, a pattern-matching algorithm is executed, comparing the input message to a library of all of the known offensive words in the English Language [16], together with a list of Political terms. If the bot were to detect any of these terms, a response will not be generated.

Secondly, the Google Cloud Sentiment Analysis API is requested to analyse the sentiment of the input message. This allows the bot to detect whether the message is considered positive, negative, or neutral (without emotion). Furthermore, following the work performed into user experience with emotional chatbots [6], the bot will only store the unknown message if, and only if, it is not considered of negative sentiment. This in effect will prevent the bot from having the ability to produce a negative response to a message regardless of the input sentiment. In addition, sentiment analysis is also performed (See Fig.2) during step 2 in real-time after a response has been generated, to mitigate a slight change in said message's sentiment after the simplification and flagging is reversed, effectively preventing any incorrect measurements or representations.

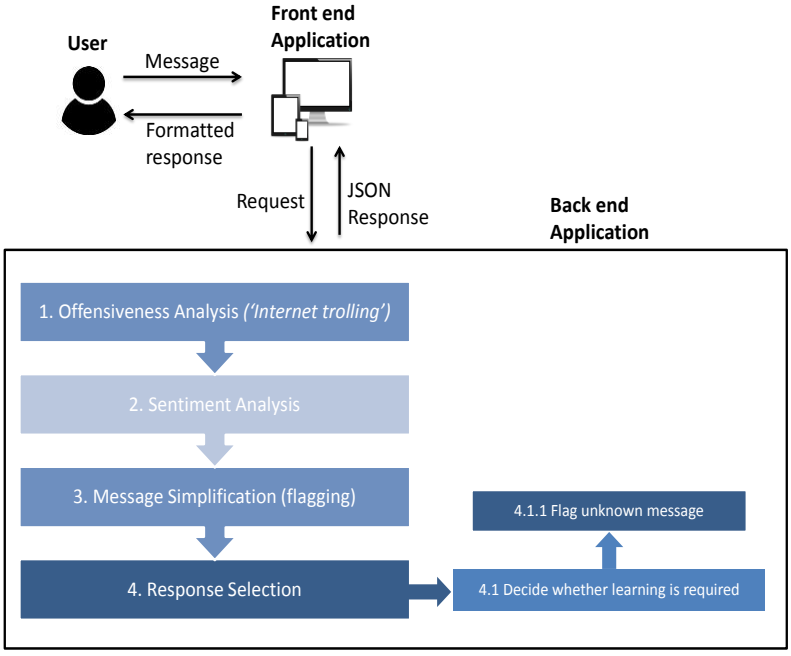

Fig. 2. Proposed architecture overview and flow of system process.

LSTM Overview. The Google Sentiment Analysis tool [13] is trained using Long-Short-Term-Memory (LSTM), where multiple recurrent neural networks (RNN) predict an output based on their input and their current state [14]. The general idea is as follows.

Firstly, a logical forget gate at the current timestep *ft* will decide which information to discard and delete: *Wf* represents the learning-weighting matrix, *h* represents the output vector of the unit at provided timestep *t-1*, *xt* being the input vector at defined timestep *t*, and finally *bf* is a bias vector applied to the process.

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t] + b_f\right) \tag{1}$$

Secondly, the cell must decide on which information to store. The variable *i* represents the input data being received by the cell, and does so through a logical input gate. C_t being the vector of the new values generated by the process.

$$i_t = \sigma(W_i . [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C_t} = \tanh(W_c . [h_{t-1}, x_t] + b_c) \tag{3}$$

Thus, the cell's parameters are therefore updated using the calculated variables (1-3) in a convolutional operation:

$$\mathrm{C}_t = \mathrm{f}_t * \mathrm{C}_{t-1} + \mathrm{i}_t * \tilde{C_t} \tag{4}$$

An output is consequently generated where O_t represents the cell's output gate at the current timestep, *t*. The internal (hidden) state of the cell is subsequently updated to match its new value:

$$\mathrm{o}_t = \sigma(\mathrm{W}_o[h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

This LSTM paradigm is used extensively in modern Machine Learning applications due to its ongoing record breaking effectiveness, one of which was most notably Microsoft's speech recognition system being at a genuinely human level of complexity [15]. A user's message was passed at the second step in the process (see fig.2) to the API and its responses were used accordingly. The Google Sentiment Analysis toolkit made use of the LSTM by deriving the sentiment score on a scale of -1.0 to 1.0 (most negative, through neutral, to most positive) as well as its magnitude (the strength of the derived sentiment score regardless of value). These values are used in both input to adhere to the findings of Tatai et al. [6] in sentiment impact of user experience, as well as provided by the system to the front-end application to be used in a platform-applicable approach.

Proposed Simplification Step. A novel approach of message simplification is employed in (see Fig. 2) step 3. Messages that contain colloquialisms or otherwise identical synonyms that do not change the meaning of a message, although greatly impacting the structure, will be simplified by replacing said terms with a flag.

Possibilities of message s where $x \in X$ are the set of all known flag-phrase pairs are given as the product of all possibilities of said phrase within the message.

$$P(s) = \prod_{x=1}^{X} P(x)\,. \tag{7}$$

An iterative process takes into account the stored steps of flag-phrase relationships and replaces the phrases based on their flag parent. Possibilities, therefore, can be calculated via the product of the set of phrase siblings within flags existent in the message *s*.

The reason for the introduction of this simplification step is that the usage of this layer would expand the learning capabilities by overcoming the differentiation of spoken sentences in terms of societal colloquialisms and synonyms with identical meaning. This is done by denoting them with flags rather than retaining the original string data. These are arbitrarily stored sets of strings that have been created manually where an index simply defines either a flag or a phrase eg. {"[GREETING]", "Hello", "Hi"} where at index 0 a flag exists, in this case "[GREETING]". All values at index $i>0$ therefore, are children of said flag.

For example, if the bot were to know ten phrases for 'hello', and five for 'happy' the sentence 'Hello! I am happy!' would be simplified as '[GREETING]! I am [ADJ_HAPPY]!' which would in turn have 50 possible combinations. Without this method, fifty messages would have been learned individually, whereas with its implementation, only one exchange is needed to learn responses to all messages.

Fig. 3. Previous interaction comparison between lingual simplification algorithm active and inactive.

Fig. 3 details the usage of lingual simplification in terms of two messages that, other than a slight differentiation of sentiment (through wording), have identical meanings. Without the layer active, the two messages must be learnt from individually. With the lingual simplification active, the experiences gathered from one of the message are applied to the other due to their identical meaning.

Response Selection. (See Fig. 2) Step 4 is comprised of three sub-layers. Firstly, pattern matching is performed against the bot's stored dataset of message-response pairs. Thresholds are preliminarily defined as 60% and 90% (see Fig. 4), the former defining an accurate response that needs to be learnt from, and the latter defining an accurate response that does not require further learning. Results below either values will flag the message for further learning, whereas a result below 60% will have the bot change conversation due to the response being too inaccurate to give.

Learning is only performed on the non-negative messages due to the findings of user chatbot experience in terms of response sentiment [6].

Testing will be performed by having three individuals having a twenty-message conversation with the bot (10 user messages and 10 bot responses), and will be automated by recreating them on the system with and without the simplification algorithm active. Conversational dataset will persist throughout the three conversations. This method of testing is followed so two identical conversations can be compared on two identical chatbot states.

Learning Methodology. The learning process of the bot is proposed as follows. If a bot were not to understand a message input, it would change conversation. The conversation would be changed to a message that the bot has previously not understood (from another user) and the user's response would be stored as a potential candidate for a message-response pair, as well as their measured sentiment (see Fig. 4). This calculation is simply based on a pattern similarity between the two messages (the input, and the closest previously seen message) which is logically relevant as the simplification layer has been executed, meaning the simplified flags have been taken into account in said pattern match (See Fig 3.)

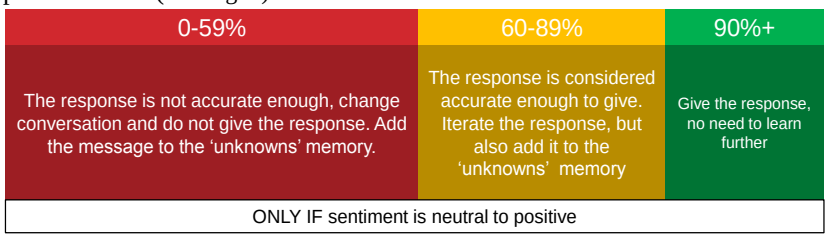

Fig. 4. Bot reactions to detected percentage similarity.

4 Preliminary Results

Testing was performed by observing three individuals conversing with the bot for a period of 20 messages; conversations were repeated on identical datasets both with the simplification layer turned off, and on. The environment selected was a web-based interface that asynchronously requested responses from the bot server. A small general conversational dataset of 200 message-response pairs was deployed to give the bot a

starting knowledge. The message dataset was produced by having simple, general conversations with the bot.

The 200 messages were pre-processed via the layer for the simplification testing which began at 478 combinations – "known responses" referring to both the stored responses and their combinations due to a combinational message being treated as a message (see proposed approach). Thus, the illusion of a larger dataset was produced.

Table 1 details the bot's Known Response Increase (KRI) over the course of three conversations. Without the novel method of message simplification, a total of 27 new responses were learnt and on average, a response was deemed accurate enough 58.3% of the time. On the other hand, with the simplification layer activated, the exact same conversations resulted in 82 new responses learnt and on average created an accurate message response 68.3% of the time. The calculation of percentage success is simply the ratio of the number of messages the bot replied to the total number of messages, ie. contrasting those that were replied to as opposed to the user messages that were not understood. This is seen further in figures 5 and 6, in which success is indicated as binary result of 0/100 (could not reply/did reply).

Table 1. Conversation success with and without Message Simplification.

Conversation	KRI	Success [%]	KRI	Success [%]
	No simplification		With simplification	
1	8	65	21	80
2	8	60	28	70
3	11	50	33	55
Overall	27	58.3 Avg.	82	68.3 (Avg.)

Figures 5 and 6 give a graphical representation of the three sequential conversations over time, where X is the number of the message of the total 1-60 messages. Success (%) shows the percentage success in terms of accuracy of the response where 0 was deemed inaccurate and 100% was a reply from the bot. Known responses shows the learning process over time through the increasing number of known responses the bot has to input messages.

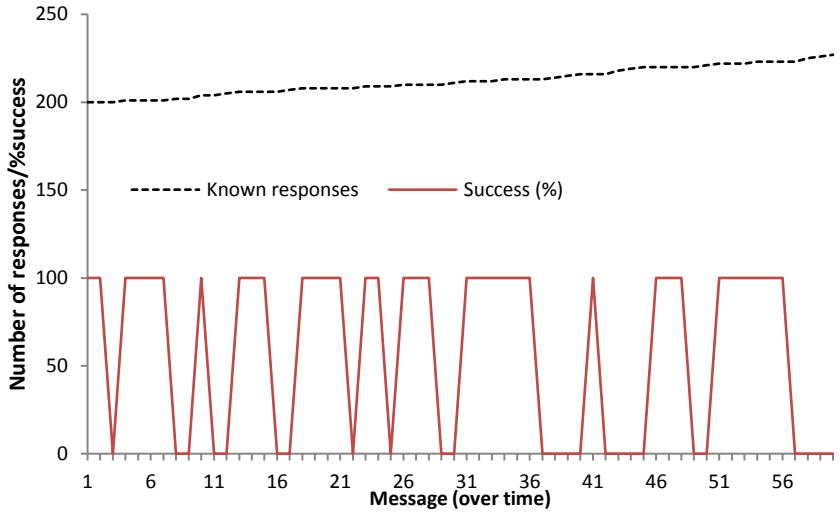

Fig. 5. Performance through three iterative twenty-message conversations (no simplification).

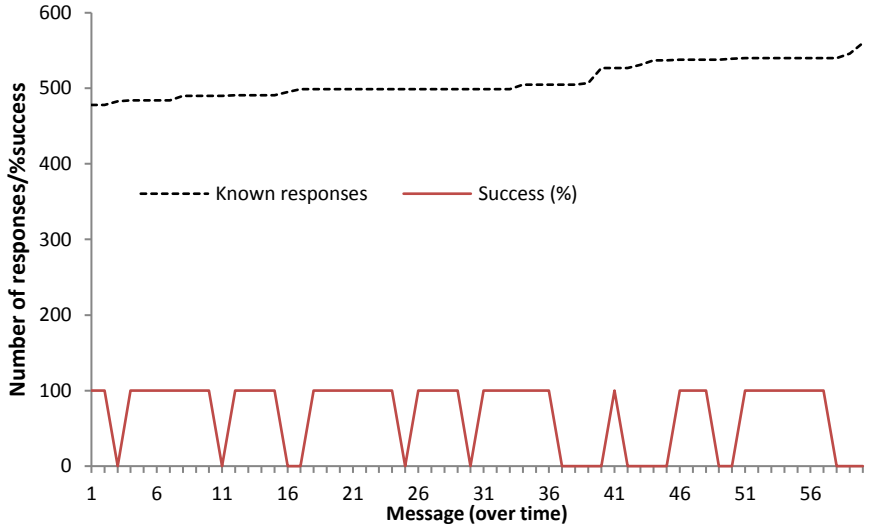

Fig. 6. Performance through three iterative ten-message conversations (with simplification).

5 Conclusion

The learning ability is clearly indicated by the preliminary results. The number of known message-response pairs grows over time, with experience. Message success does not seem to improve, but further extensive research is needed to explore the relationships between known responses and conversational success. The experiment only

covered sixty messages (three twenty message pairs), but many more would be needed to explore said relationship.

Furthermore, conversational re-creation during testing shows the massive improvement when the message simplification layer is employed to mitigate colloquialisms, and even though this system tends to get fewer opportunities to learn (due to its higher success rate), it will make use of these opportunities far better than the system without will ever do.

Future Work and Impact. The decrease of message success gives the illogical impression of the bot performing worse, as it accumulates more knowledge. This is likely due to the user's differing conversational subjects, due to the fact that more knowledge quite literally correlates to more responses in this system. More extensive conversations with the bot must be obtained to gain a more accurate figure of success over time. Pattern matching learning thresholds (60%, 90%) during the learning process were set arbitrarily at levels that made logical sense, as they performed as expected during the testing stage (see Table 1). Further improvement to an accuracy-based learning system in terms of string patterns can be expected by setting more effective threshold values.

Jia found that chatbots had the potential to not only aid in the education system, but also effectively improve it [4]. The further introduction of selective learning from experts, and through this, the formation of an expert system could ultimately lead to a conversational aid in required situations. For example, a teaching assistant may answer many uniquely-phrased questions with logically identical, or close-to, answers. An expert chatbot born from the knowledge of said teaching assistants could, for example, lead to a more efficient University Laboratory paradigm in which students are aided by both human and machine.

Following this vein of thought, a system deployed *post-learning* from many experts in the field of psychotherapy could introduce the usage of knowledge re-application in situations such as mental health counseling. This is a strong social impact that, as of yet, has not been achieved.

References

1. Kuligowska, K. (2015). Commercial chatbot: performance evaluation, usability metrics and quality standards of embodied conversational agents.
2. Alexa, Amazon. "Amazon" (2014).
3. Shawar, Bayan Abu, and Eric Atwell. (2003). "Machine Learning from dialogue corpora to generate chatbots." Expert Update journal 6.3: 25-29.
4. Jia, Jiyou. "The study of the application of a web-based chatbot system on the teaching of foreign languages." (2004). Society for Information Technology & Teacher Education International Conference. Association for the Advancement of Computing in Education (AACE). (pp. 1201-1207)

5. Heller, B., Proctor, M., Mah, D., Jewell, L., & Cheung, B. (2005). Freudbot: An investigation of chatbot technology in distance education. In EdMedia: World Conference on Educational Media and Technology (pp. 3913-3918). Association for the Advancement of Computing in Education (AACE).
6. Tatai, G., Csordás, A., Kiss, Á., Szaló, A., & Laufer, L. (2003). Happy chatbot, happy user. In Intelligent Virtual Agents (pp. 5-12). Springer Berlin/Heidelberg.
7. Mauldin, M. L. (1994). Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition. In AAAI (Vol. 94, pp. 16-21).
8. Shawar, B. A., & Atwell, E. (2007). Different measurements metrics to evaluate a chatbot system. In Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies (pp. 89-96). Association for Computational Linguistics.
9. Wallace, R. (2001). Artificial linguistic internet computer entity (ALICE). Retrieved from https://www.chatbots.org/chatbot/a.l.i.c.e/. Last accessed 25/5/2018
10. The Exeter Blog. (2014). The Loebner Prize, a Turing Test competition at Bletchley Park. Retrieved from https://blogs.exeter.ac.uk/exeterblog/blog/2014/12/08/the-loebner-prize-a-turing-test-competition-at-bletchley-park/
11. Microsoft (March). Tay AI. Retrieved from https://twitter.com/tayandyou. Last accessed 25/5/2018.
12. Wakefield, J. (2016). BBC News. Microsoft chatbot is taught to swear on Twitter. Retrieved April 12, 2018, from http://www.bbc.co.uk/news/technology-35890188.
13. "Google Cloud Products" (n.d.). Retrieved March 28, 2018 from https://cloud.google.com. Last accessed 25/5/2018
14. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
15. Haridy, Rich (2017). "Microsoft's speech recognition system is now as good as a human". newatlas.com. Last accessed April 6, 2018 from https://newatlas.com/microsoft-speech-recognition-equals-humans/50999/.
16. AllSlang. (n.d.). Swear Word List, Dictionary, Filter, and API. Retrieved March 11, 2018, from https://www.noswearing.com/
17. D. R. Faria, M. Vieira, F. C. C. Faria, C. Premebida (2017). Affective Facial Expressions Recognition for Human-Robot Interaction. IEEE International Symposium on Robot and Human Interactive Communication, 805-810.
18. 18. D. R. Faria, M. Vieira, F. C. C. Faria (2017). Towards the Development of Affective Facial Expression Recognition for Human-Robot Interaction. International Conference on Pervasive Technologies Related to Assistive Environments, 300-304.
19. D. Bertero, F. Siddique, C. Wu, Y. Wan, R. Chan, P. Fung (2016). Real-Time Speech Emotion and Sentiment Recognition for Interactive Dialogue Systems. Conference on Empirical Methods in Natural Language Processing, 1042–1047.
20. M. Vieira, D. R. Faria, U.Nunes (2015). Real-time Application for Monitoring Human Daily Activities and Risk Situations in Robot-assisted Living. 2nd Iberian Robotics Conference, 449-461.
21. A. M. Turing (1950) Computing Machinery and Intelligence. Mind 49: 433-460.
22. J. Weizenbaum (1976). Computer Power and Human Reason: From Judgment to Calculation. New York: W.H. Freeman and Company.