International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-2S2, January 2019

A Survey of Chatbot Design Techniques

R. Devakunchari, Rishabh Agarwal, Eshita Agarwal

Abstract - Chatbots gives us a fresh way to converse with computers. To get answers to our questions by a computer we either use a search engine, or fill out form, whereas a chatbot allows us to simply ask questions in the same manner that we would ask a human i.e., a chatbot is a program that mimics human conversation using Artificial Intelligence (AI). A chatbot is devised to be the ideal virtual assistant, helping one to complete different tasks such as answering questions, getting driving directions, turning up the thermostat in smart homes, to playing one's favourite tunes etc. Chatbots recently have gained a lot of popularity in the field of human-computer interaction. They are being used extensively in all sorts of applications like customer support, personal assistant, advising, sales, marketing etc. The technologies at the core of the rise of the chatbot are Machine Learning (ML) and Natural Language Processing (NLP). However, these chatbots lack one or more functionalities such as not being able to maintain a persona, unable to give personalized responses depending on the user or preventing faulty responses to unknown questions. The relevance of this paper is to review the various existing chatbot design techniques, discuss their strengths and evaluate them based on their uses.

Keywords: Chatbot, AIML, LSA, Patten Matching, ChatScript, Parsing, Language Tricks

I. INTRODUCTION

Human-computer interaction is the most upcoming area of research in the field of computer science and there are many ways to approach it, natural language (NL) being one of them. This paper focuses on chatbots, which are gaining fame due to the success of virtual assistants such as, Google Assistant, Jeannie, Alexa, S-Voice, Siri and others.

A chatbot's purpose is to conduct a conversation with humans, pretending to be another human being. The conversation message could be send through several mediums such as voice commands, test chats, graphical interfaces or graphical widgets. Chatbots can serve specific purposes based on the task they are made for such as searching the web, organizing files on the computer, setting up appointments, making flight reservations, answering general queries, purchasing products etc.

Manuscript published on 30 January 2019.
***** Correspondence Author (s)
R. Devakunchari, Assistant Professor, Department of CSE, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India. (e-mail: devakunchari.r@ktr.srmuniv.ac.in)
Rishabh Agarwal, UG student, Department of CSE, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India. (e-mail: rishabhagarwal_r@srmuniv.edu.in)
Eshita Agarwal, UG student, Department of CSE, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India. (e-mail: eshita_agarwal@srmuniv.edu.in)

The greatest difficulty that existing chatbots face is comprehending the questions and responses by the humans and maintaining a persona and context. Most of these bots still just work using pattern matching. They try to find a pre defined answer which correspond to the input. They neither keep record of the previous conversations nor do they generate personalized responses. This method does not provide us with a satisfactory conversation or lead a conversation with some particular purpose.

Because of the limitations of predefined responses, the developers and researchers of Chatbots went on to add new features and functionalities to the existing designs of chatbots. Some of these added features include using some ontologies, remembering information about the user from conversations and making small talk. These features have made chatbots much more successful and at the same time has introduced various different approaches, systems and solutions to the same problem.

The paper aims to create a survey of chatbot technologies and techniques and thus make it simpler for a developer to decide which technology to use for the creation of future chatbot systems

II. RELATED WORK

Bingquan Liu, Derek F. Wong and Min Zhang show user modelling for personalized response ranking in chatbots. Such responses have become very important in making the chatbot more humane [1].

Bayu Setiaji and Ferry Wahyu Wibowo have displayed the influence of using databases in chatbots for information retrieval and knowledge storage. The test results prove that using RDBMS for knowledge storage is very beneficial and long- lasting [2]. Bhavika Ranoliya, Nidhi Raghuwanshi and Sanjay Singh demonstrated the use of AIML tags. The illustration describes the functionality and description of each tag required to build chatbots based on AIML[3].

Muhammad Fairuz Abd Rauf and Zuraidy Adnan have very expertly explained the nature and use of chatbots and reviewed the various techniques required to develop them [4]. The work of Luka Bradeško and Dunja Mladenić in the survey of chatbot design techniques through the Loebner prize competition gives great insight into upcoming approaches in the development of modern chatbots [5].

III. REVIEW OF CHATBOT DESIGN TECHNIQUES

Based on several papers we have come to the inference that the design of Chatbots involves several techniques and approaches. In the following section we review some of the most popular techniques used by the developers.

Pattern Matching

Early chatbots used pattern matching to produce responses to queries. This technique deployed matching pattern to generate appropriate response from users' questions, which depends on different types of matching such as simple statements or the meaning of enquires. Pattern matching is often referred to as "brute force" as the developer of the system needs to describe every pattern and its response. The first chatbot created by Weizenbaum in 1966, ELIZA [6], used the pattern matching algorithm to generate an proper answers to the user input. For instance, ELIZA would process the user's input "I am very happy." by looking up in a pre-defined dictionary for the corresponding keywords. After matching a keyword ELIZA applies an associated input-response rule. According to the applied rule, ELIZA transforms the phrase "I am" into "You are". The response generation algorithm then adds the phrase "I am delighted to hear" prior to "you are" to generate the response "I am delighted to hear you are happy." Figure 1 shows a conversation between a user and ELIZA. In this figure, we can see ELIZA using pattern matching to respond to the user.

Pattern matching is the most commonly used technique in chatbots. In every existing chatbot system, a variants of the pattern matching algorithm exist. The variants can vary in their complexity, but the underlying idea of matching keywords remains the same. Pattern matching techniques were adopted by many bots [7] including SHRDLU [8] Speech Chatbot [9], PC Therapist III [10], Chatterbot in "TinyMUD", TIPS, FRED, HEX, Albert One.

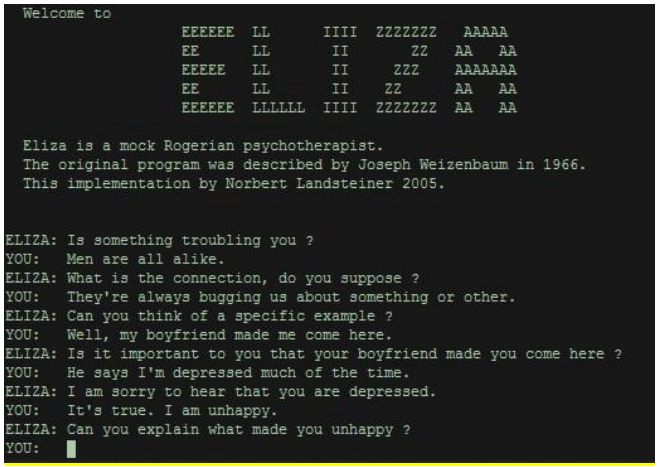

Fig. 1. ELIZA Chatbot

AIML

To create a Chatbot, there is a need for an easily understandable and universal language which is also flexible. AIML (Artificial Intelligence Markup Language) is one of the widely used approaches that satisfies the requirements. It is a derivative of XML. To make the job of modelling easier, in relation to conversation and a "stimulus-response" process is the motive of the AIML language is. It is a standard structure for the input-response patterns. AIML is based on the technology developed for A.L.I.C.E. (the Artificial Linguistic Internet Computer Entity). It represents the knowledge put into Chatbots. AIML's syntax is based on XML. AIML's syntax is XML based and is composed mostly of input rules (categories) with appropriate output. The input rule/ pattern should include the entire input and is case insensitive. A wildcard (*) can also be used which binds it to one or more words. An AIML script consists of several "categories", which are defined by the tag <category>. Each category consists of only one <pattern> tag, which defines a possible user input, and at least one <template> tag, which specifies the chatbot's response for the user's input. For example:

```
<category>
<pattern>How are you</pattern>
<template>I am fine!</template>
</category>
<category>
<pattern>Who is * </pattern>
<template> He is my brother </template>
</category>
```

AIML can recursively call itself. This is its real power. It can call itself with the input as parameters using the <srai> tag and the contents of * using <star/>.

```
<aiml version = "1.0.1" encoding = "UTF-8"?>
<category>
<pattern> WHO IS ALBERT EINSTEIN </pattern>
<template>Albert Einstein was a German
physicist.</template>
</category>
<category>
<pattern> WHO IS Isaac NEWTON </pattern>
<template>Isaac Newton was a English physicist and
mathematician.</template>
</category>
<category>
<pattern>DO YOU KNOW WHO * IS</pattern>
<template>
<srai>WHO IS <star/></srai>
</template>
</category>
</aiml>
```

In recent years, AIML has been one of the most used technologies. AIML is a technology based on pattern matching [12]. In order to understand the AIML tags, an AIML interpreter is needed. The AIML specification i.e. either 1.0 or 2.0 decides which is implemented. Developing chatbots based on AIML do not require skills in a specific programming language. Hence, this approach facilitates the easy development of chatbots. Thus, many chatbots have been created using AIML technology such as HmoristBot, Ella, MathGame and Mitsuku [13].

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Parsing

Parsing is the method used to examine text or a string of symbol either by using natural language or computer language. It is a technique which picks up the input text and transforms it into a set of lexicons or words (lexical parsing) to determine its grammatical arrangement. Additionally, the lexical structure is checked to see if it forms an allowable expression (syntactical parsing). In technical terms, parsing is a technique that is used to analyse a set of strings into its grammatical elements that could contain semantic or other information. This technique used NLP functions such as trees in Python NLTK.

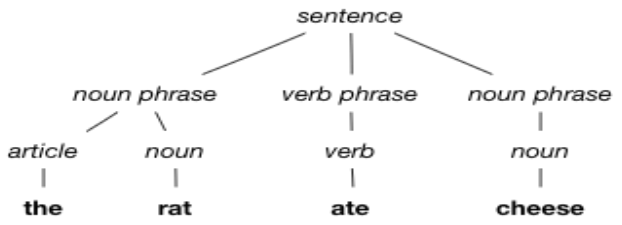

Fig. 2. Parse Tree

The earlier parsers were very simple. They used to look for familiar keywords in approved order. For instance, “please take the water” and “can you get the gold” would be both parsed into “take water”. Multiple input sentences can be covered with a limited set of patterns can using this technique. Complete grammatical parsing of the natural language sentences are done by complicated and advanced parsers. Figure 2 depicts a generic parse tree. Parsing has been used by many Chatbots including PARRY [14], SHRDLU, Speech Chatbot, PC Therapist III, Jabberwock, LARRI, RITEL, Senior Companion, Justin and Justina, MetaTutor, My Science Tutor, Amilon, Speech Dialog System (dbot) and Pharmabot [15].

ChatScript

Chatscript is another chatbot development language based on pattern matching. When no matches occur in AIML, it comes to use and it aims to be a successor of the AIML language. ChatScript focuses on giving better syntax to create a rational default response, which makes it easier to maintain. Concepts, continuations, logical and/or, variables, fact triples and functions are some of the additional functionalities introduced by ChatScript. With these functionalities it tries to make up for the need of ontologies inside the script. Instance of a script defining a concept of vegetable and one pattern can be seen below:

Topic: ~food(~vegies vegetable food eat)
t: What is your favourite food?
a: (~vegies) I like vegetables too.
a: (~rock) I would rather listen to rock music than eating it.

Words starting with ~ are concept sets. ChatScript use of concept sets heavily. Concept sets are lists of words sharing meaning. ~vegies would be the list of all the known vegetables. The simple pattern (~vegies) is informed if any vegetable is mentioned right after the chatbot asks for favourite food. Standard if-else, loops, user-defined functions and calls, and variable assignment and access are supported by ChatScript.

Chatbots developed using Chatscripts include Suzette, Rosette, Albert and a conversational agent of Bogatu and colleagues.

SQL and Relational Database

A Relational Data Base (RDB) is one of the techniques used in recent times to create Chatbot knowledge bases. The Chatbot remembers previous conversations and makes the conversation smoother and meaningful. It is used to enhance the capability of chatbot’s keyword and pattern matching by providing augment ways of storing data as well as improving the process performance.

SQL (Structured Query Language) is the most famous RDB language. SQL or MYSQL has gained popularity in RDB because it is the high-level language for non-procedural data. Nested query blocks of varying depths is its most powerful feature. Algorithms have been developed to alter these nested query types into simpler "semantically equivalent queries". SQL as a data language is implemented in ZETA and as a calculus-based and block- structured language, it is implemented in System R, ORACLE and SEQUEL. Recently, chatbot developers have started using SQL to generate a database to save conversations with a user in order to facilitate search for any word or phrase. This technique gives continuity, context and accuracy to the conversation because it enables the chatbot system to retrieve some previous information. Figure 3 shows how a user's conversation history is stored in a database.

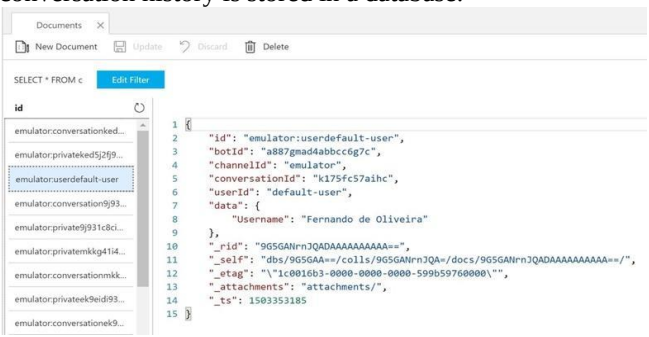

Fig. 3. Database storing user’s conversation history

Language Tricks

This technique uses sentence, phrases or paragraphs in Chatbots to diversify the knowledge base so that it becomes more convincing to its human users. The types of language tricks are:

- Canned responses.
- Typing errors and simulating key strokes.
- Model of personal history.
- Non Sequitur (not a logical conclusion)

Each of the above language tricks is used to serve a particular purpose and provide alternative and differently structured answers to user questions. These language tricks are used to pass off as human and fool the users.

Canned Responses: These are “canned” i.e., precooked or predefined (hard coded) responses to questions. If a chatbot uses only patterns to answer the questions, then these responses can be called canned responses. This would result in extremely large number of patterns making them unmanageable.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Hence canned responses are only used by chatbots to answer questions of the user that are not anticipated and fed into the knowledge base of the chatbot.

Typing Errors and Simulating Keystrokes: In this technique, the chatbots "pretend" that they are replying by typing the answers word by word. Some of the bots even fake backspacing and spelling mistakes in order to simulate a human being, trying to make the user experience even more real.

Model of Personal History: In order to make the chatbot more convincing and human like, the developers usually give a personal touch to it by adding a story (imaginary or otherwise) to its responses. This includes childhood memories, information about parents, religious views, interests, hobbies, likes, dislikes etc. This enriches the "social background" of the chatbot and the user becomes convinced that he/she is talking to a person who remembers about them.

Non Sequitur: Non sequitur (Latin) is a statement or conclusion that does not logically follow from the previous statement ie, such a response breaks the logical flow of a conversation. For example, "My refrigerator is not working. I better go out for a movie." Such statements with no logical conclusion are planted in the chtabots to enhance small talks. This feature is usually added to streamline the user's internet experience and make it funny and spontaneous.

You: hello
NonSequitur Bot: sick
You: how are you
NonSequitur Bot: and
You: what do you know
NonSequitur Bot: chop off the t and just die
You: why so mean
NonSequitur Bot: 0 Quotes
NonSequitur Bot: a new social security card washed up
NonSequitur Bot: if i say anything nonsensical

HeX [8], CONVERSE [5], PC Therapist III [5], and TIPS [5] are examples of conversational agents that use language tricks.

Latent Semantic Analysis

Latent Semantic Analysis (LSA) is an approach in natural language processing (NLP). It is used in distributional semantics. It produces a set of concepts by inspecting the relationship between a set of documents. The concepts produced identify the documents and the terms used in them. In LSA it is assumed that words with similar meanings occur in similar chunks of text or in similar context (the distributional hypothesis). A matrix is constructed from a large piece of text containing word counts per paragraph. To reduce the number of rows, mathematical technique called singular value decomposition (SVD) is used. However, the resemblance among the columns is preserved. Words are then compared by calculating the dot product between the normalizations of the two vectors formed by any two rows. Dot product values close to one represent very similar words, whereas values close to zero represent very dissimilar words. This technique gives more intelligence to the chatbot. For example, according to LSA sport and football has semantic similarity in the vector space.

The LSA block comes to use when AIML fails to give an answer to a particular question asked by the user. It is nearly impossible for a developer to anticipate and feed all possible questions that can be asked, to the chatbot. This is the main challenge faced by many chatbots. All the unanswered questions from AIML will be spontaneously routed to the LSA block. Figure 4 shows how AIML and LSA blocks are accessed and how they handle a query.

IV. RESULTS AND DISCUSSIONS

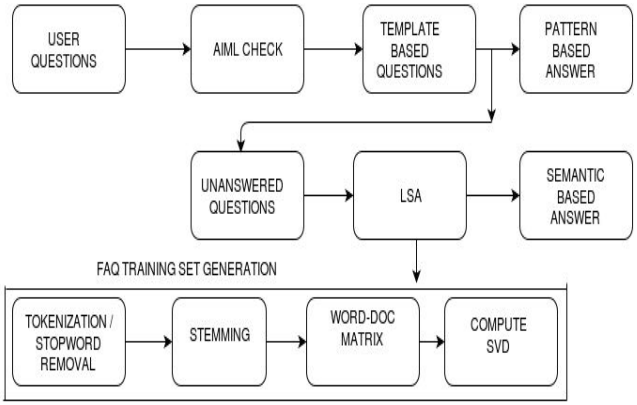

Fig. 4. Flow diagram for chatbot using AIML and LSA

V. CONCLUSION

Chatbots (or conversational agents) are gaining a lot of popularity for web services like academia, customer care, marketing, entertainment, etc. These agents not only save time but also save manpower. These conversational agents are not completely independent. This human-computer interaction will be more effective if the system queries the missing data provided by the user to provide a satisfactory answer. Several techniques used to create a chatbot are discussed and analysed in this paper. This paper not only behaves as a review of all the chatbot creation techniques but also as a repository of these techniques and their use cases so developers can understand and apply the techniques relevant to their chatbot.

REFERENCES

1. Liu, B., Xu, Z., Sun, C., Wang, B., Wang, X., Wong, D. F., ... & Wang, B. "Content Oriented User Modeling for Personalized Response Ranking in Chatbots", IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), Vol. 26, Issue 1, pp. 122-133, 2018.
2. Setiaji, B., & Wibowo, F. W. "Chatbot using a knowledge in database". In proceedings of 7th International Conference on Intelligent Systems, Modelling and Simulation, IEEE, Bangkok, pp. 72-77, 25-27th Jan 2016.
3. Ranoliya, B. R., Raghuwanshi, N., & Singh, S. "Chatbot for university related FAQs". In proceedings of Advances in Computing, Communications and Informatics (ICACCI), International Conference on (pp. 1525-1530). IEEE, 2017.
4. Ahmad, N. A., Che, M. H., Zainal, A., Rauf, M. F. A., & Adnan, Z. "Review of Chatbots Design Techniques". In International Journal of Computer Applications, Vol. 181, Issue 8, pp. 7-10, 2018.
5. Bradeško, L., & Mladenić, D. "A survey of chatbot systems through a loebner prize competition". In proceedings of Slovenian Language Technologies Society Eighth Conference of Language Technologies pp. 34-37, 2012.
6. Weizenbaum, J. "ELIZA—a computer program for the study of natural language communication between man and machine". Communications of the ACM, Vol. 9, Issue 1, 36-45, 1966.

7. Hutchens, J. L. “How to pass the Turing test by cheating”. School of Electrical, Electronic and Computer Engineering research report. Perth: University of Western Australia, 1997.
8. Winograd, T. “Understanding natural language”. Cognitive psychology, Vol. 3 Issue 1, pp. 1-191, 1972.
9. Seneff, S., Hirschman, L., & Zue, V. W. “Interactive problem solving and dialogue in the ATIS domain”. In Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991.
10. Weintraub, J. “History of the PC Therapist”. Online at http://www.loebner.net/Prizef/weintraub-bio.html, 1986.
11. Wallace, R. “The elements of AIML style”. Alice AI Foundation, 2003.
12. Marietto, M. D. G. B., de Aguiar, R. V., Barbosa, G. D. O., Botelho, W. T., Pimentel, E., França, R. D. S., & da Silva, V. L. “Artificial intelligence markup language: a brief tutorial”. arXiv preprint arXiv:1307.3091, 2013.
13. Abdul-Kader, S. A., & Woods, J. C. “Survey on chatbot design techniques in speech conversation systems”. International Journal of Advanced Computer Science and Applications, Vol. 6, Issue 7, 2015.
14. Colby K. M., “Artificial Paranoia: A computer program for the study of natural language communication between man and machine”, Communications of the ACM, Vol. 9, pp. 36-45, 1975.
15. Comendador, B. E. V., Francisco, B. M. B., Medenilla, J. S., & Mae, S. “Pharmabot: a pediatric generic medicine consultant chatbot”. Journal of Automation and Control Engineering Vol, 3, Issue 2, pp. 137-140, 2015.