Survey on Chatbot Design Techniques in Speech Conversation Systems

Sameera A. Abdul-Kader
School of Computer Science and Electronic Engineering/University of Essex Colchester/ UK
Diyala University/ Diyala/ Iraq

Dr. John Woods
School of Computer Science and Electronic Engineering/University of Essex Colchester/ UK

***Abstract*—Human-Computer Speech is gaining momentum as a technique of computer interaction. There has been a recent upsurge in speech based search engines and assistants such as Siri, Google Chrome and Cortana. Natural Language Processing (NLP) techniques such as NLTK for Python can be applied to analyse speech, and intelligent responses can be found by designing an engine to provide appropriate human like responses. This type of programme is called a Chatbot, which is the focus of this study. This paper presents a survey on the techniques used to design Chatbots and a comparison is made between different design techniques from nine carefully selected papers according to the main methods adopted. These papers are representative of the significant improvements in Chatbots in the last decade. The paper discusses the similarities and differences in the techniques and examines in particular the Loebner prize-winning Chatbots.**

Keywords—AIML; Chatbot; Loebner Prize; NLP; NLTK; SQL; Turing Test

I. Introduction

Speech is one of the most powerful forms of communication between humans; hence, it is the researchers' ambition in the human computer interaction research field to improve speech interaction between the human and the computer in order to simulate human-human speech interaction. Speech interaction with modern networked computing devices has received increasing interest in the past few years with contributions from Google, Android and IOS. Because they are more natural than graphic-based interfaces, spoken dialogue systems are beginning to form the primary interaction method with a machine [1]. Therefore, speech interaction will play a significant role in humanising machines in the near future [2].

Much research work has focussed on improving recognition rates of the human voice and the technology is now approaching viability for speech based human computer interaction. Speech Interaction splits into more than one area including: speech recognition, speech parsing, NLP (Natural Language Processing), keyword identification, Chabot design/personality, artificial intelligence etc. Chatbot is a computer program that have the ability to hold a conversation with human using Natural Language Speech.

In this paper, a survey of Chatbot design techniques in speech conversation between the human and the computer is presented. Nine studies that made identifiable contributions in Chatbot design in the last ten years are selected and then, reviewed. The different techniques used for Chatbots in the selected works are compared with those used in Loebner-Prize Chatbots. The findings are discussed and conclusions are drawn at the end.

II. Background

A. Human-Computer Speech interaction

Speech recognition is one of the most natural and sought after techniques in computer and networked device interaction has only recently become possible (last two decades) with the advent of fast computing.

Speech is a sophisticated signal and happens at different levels: "semantic, linguistic, articulatory, and acoustic" [3]. Speech is considered as the most natural among the aspects of human communication, owing to copious information implicitly existing beyond the meaning of the spoken words. One of the speech information extraction stages is converting speech to text via Automatic Speech Recognition (ASR) and mining speech information [4]; then, the resulting text can be treated to extract the meaning of the words.

Speech recognition is widely accepted as the future of interaction with computers and mobile applications; there is no need to use traditional input devices such as the mouse, keyboard or touch sensitive screen and is especially useful for users who do not have the ability to use these traditional devices [5]. It can help disabled people with paralysis, for example, to interact with modern devices easily by voice only without moving their hands.

B. Natural Language Toolkit (NLTK)

In order to deal with and manipulate the text resulting from speech recognition and speech to text conversion, specific toolkits are needed to organise the text into sentences then split them into words, to facilitate semantic and meaning extraction. One of these toolkits is the widely used NLTK which is a free plugin for Python.

The Natural Language ToolKit (NLTK) is a set of modules, tutorials and exercises which are open source and cover Natural Language Processing symbolically and statistically. NLTK was developed at the University of Pennsylvania in 2001allowing computational linguistics with three educational applications in mind: projects, assignments and demonstrations [6] [7]. It can be found within the Python Libraries for Graph manipulation GPL open license. NLTK is used to split words in a string of text and separate the text into parts of speech by tagging word labels according to their positions and functions in the sentence. The resulting tagged words are then processed to extract the meaning and produce a response as speech or action as required. Different grammar rules are used to categorise the tagged words in the text into groups or phrases relating to their neighbours and positions. This type of grouping is called chunking into phrases, such as noun phrases and verb phrases.

C. *Chatbot strategies*

To give suitable answers to keywords or phrases extracted from speech and to keep conversation continuous, there is a need to build a dialogue system (programme) called a Chatbot (Chatter-Bot). Chatbots can assist in human computer interaction and they have the ability to examine and influence the behaviour of the user [8] by asking questions and responding to the user's questions. The Chatbot is a computer programme that mimics intelligent conversation. The input to this programme is natural language text, and the application should give an answer that is the best intelligent response to the input sentence. This process is repeated as the conversation continues [9] and the response is either text or speech.

Building a Chatbot needs highly professional programming skills and experienced developers to achieve even a basic level of realism. There is a complicated development platform behind any Chatbot which will only be as good as its knowledge base which maps a user’s words into the most appropriate response. The bot developer usually builds the knowledge base as well. However, there are some platforms which provide a learning environment. Writing a perfect Chatbot is very difficult because it needs a very large database and must give reasonable answers to all interactions. There are a number of approaches to create a knowledge base for a Chatbot and include writing by hand and learning from a corpus. Learning here means saving new phrases and then using them later to give appropriate answers for similar phrases [10].

Designing a Chatbot software package requires the identification of the constituent parts. A Chatbot can be divided into three parts: Responder, Classifier and Graphmaster (as shown in Figure. 1) [11], which are described as follows:

1) Responder: it is the part that plays the interfacing role between the bot’s main routines and the user. The tasks of the responder are: transferring the data from the user to the Classifier and controlling the input and output.

2) Classifier: it is the part between the Responder and the Graphmaster. This layer’s functions are: filtering and normalising the input, segmenting the input entered by the user into logical components, transferring the normalised sentence into the Graphmaster, processing the output from the Graphmaster, and handling the instructions of the database syntax (e.g. AIML).

3) Graphmaster: is the part for pattern matching that does the following tasks: organising the brain’s contents, storage and holding the pattern matching algorithms.

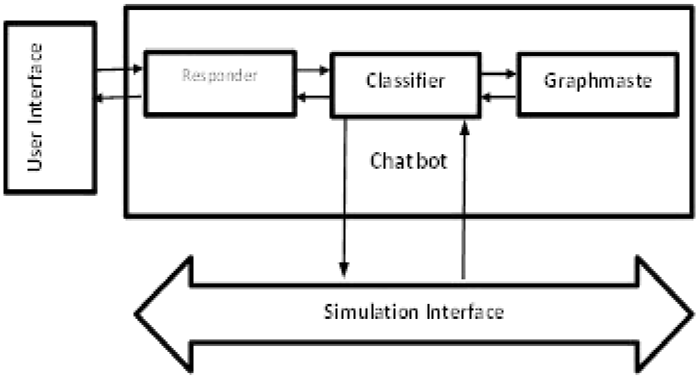

Fig. 1. Components of Chatbot [11]

D. *Chatbot Fundamental Design Techniques and approaches*

To design any Chatbot, the designer must be familiar with a number of techniques:

*1) **Parsing**: this technique includes analysing the input text and manipulating it by using a number of NLP functions; for example, trees in Python NLTK.*

*2) **Pattern matching**: it is the technique that is used in most Chatbots and it is quite common in question-answer systems depending on matching types, such as natural language enquiries, simple statements, or semantic meaning of enquiries [12].*

*3) **AIML**: it is one of the core techniques that are used in common Chatbot design. More details about this technique and the language used are explained in section 2.5 below.*

*4) **Chat Script**: is the technique that helps when no matches occur in AIML. It concentrates on the best syntax to build a sensible default answer. It gives a set of functionalities such as variable concepts, facts, and logical and/or.*

*5) **SQL and relational database**: is a technique used recently in Chatbot design in order to make the Chatbot remember previous conversations. More details and explanation are provided in section 2.6 below.*

*6) **Markov Chain**: is used in Chatbots to build responses that are more applicable probabilistically and, consequently, are more correct. The idea of Markov Chains is that there is a fixed probability of occurrences for each letter or word in the same textual data set [13].*

*7) **Language tricks**: these are sentences, phrases, or even paragraphs available in Chatbots in order to add variety to the knowledge base and make it more convincing. 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 these language tricks is used to satisfy a specific purpose and to provide alternative answers to questions [13].

*8) **Ontologies**: they are also named semantic networks and are a set of concepts that are interconnected relationally and hierarchically. The aim of using ontologies in a Chatbot is to compute the relation between these concepts, such as synonyms, hyponyms and other relations which are natural language concept names. The interconnection between these concepts can be represented in a graph enabling the computer to search by using particular rules for reasoning [13].*

E. Loebner Prize and Turing Test

a) Turing Test

In the field of Artificial Intelligence, Turing was the first to pose the question, "Can a machine think?" [14], where thinking is defined as the ability held by humans. According to this question and this definition, Turing suggests the "imitation game" as a method to directly avoid the question and to specify a measurement of achievement for researchers in Artificial Intelligence [15] if the machine appears to be human. The imitation game can be played between three people: (A) which is a man, (B) which is a woman, and (C) which is the interrogator and can be either a man or a woman. The aim of the interrogator here is to determine who the woman is and who the man is (A and B). The interrogator knows the two as labels X and Y and has to decide at the end of the game either "X is B and Y is A" or "X is A and Y is B". The interrogator also has the right to direct questions to A and B. Turing then questions what will happen if A is replaced with a machine; can the interrogator differentiate between the two? The original question "Can machines think?" can then be replaced by this question [14]. In this imitation game, the Chatbot represents the machine and it tries to mislead the interrogator to think that it is the human or the designers try to programme it to do so [16].

b) Loebner Prize

In 1990 an agreement was held between Hugh Loebner and The Cambridge Centre for Behavioural Studies to establish a competition based on implementing the Turing Test. A Gold Medal and $100,000 have been offered by Hugh Loebner as a Grand Prize for the first computer that makes responses which cannot be distinguished from humans'. A bronze medal and an annual prize of $2000 are still pledged in every annual contest for the computer which seems to be more human in relation to the other competitors, regardless of how good it is absolutely [15]. It is the first known competition that represents a Turing test formal instantiation [13]. The competition has been run from 1991 annually with slight changes made to the original conditions over the years. The important thing in this competition is to design a Chatbot that has the ability to drive a conversation. During the chat session, the interrogator tries to guess whether they are talking to a programme or a human. After a ten-minute conversation between the judge and a Chatbot on one side and the judge and a confederate independently on the other side, the judge has to nominate which one was the human. The scale of non-human to human is from 1 to 4 and the judge must evaluate the Chatbot in this range [16]. According to this judgement, the more human Chatbot is the winner.

No Chatbot has ever achieved the golden medal and passed the test to win the Loebner Prize. However, some Chatbots have scored as highly as 3 out of the 12 judges believing they were human. There is a winning bot every year and there is a list of Chatbots called Loebner Prized Chatbots. This list commences from 1991 to the current date.

c) Prized Chatbots and Their Design Techniques

Although no Chatbot has won the Loebner Prize yet, there is a winning Chatbot each year and the standard of entry continues to improve with time. Table 1 shows the prized Chatbots as the name of the programmer, the programme name, the year they won, and the techniques used to design and programme them.

F. AIML

To build a Chatbot, a flexible, easy to understand and universal language is needed. AIML, a derivative of XML is one of the widely used approaches that satisfies the requirements. AIML represents the knowledge put into Chatbots and is based on the software technology developed for A.L.I.C.E. (the Artificial Linguistic Internet Computer Entity). It has the ability to characterise the type of data object (AIML objects) and describe partial conductance of the programmes that it processes. These objects consist of two units: topics and categories; the data contained in these categories is either parsed or unparsed [19].

The purpose of the AIML language is to simplify the job of conversational modelling, in relation to a "stimulus-response" process. It is also a mark-up language based on XML and depends on tags which are the identifiers that make snippets of codes to send commands into the Chatbot. The data object class is defined in AIML as an AIML object, and the responsibility of these objects is modelling conversational patterns. This means that each AIML object is the language tag that associates with a language command. The general structure of AIML objects is put forward by [20]:

```
<command> List of parameters </command>
```

The most important object among the AIML objects is category, pattern, and template. The task of the category tag is defining the knowledge unit of the conversation. The pattern tag identifies the input from the user and the task of template tag is to respond to the specific user input [20]; these are the most frequent tags and the bases to design AIML Chatbots with an intelligent response to natural language speech conversations. The structure of category, pattern, and template object is shown below:

```
<category>
    <pattern> User Input</pattern>
          <template>
              Corresponding Response to input
          </template>
</category>
```

TABLE I. LOEBNER PRIZED CHATBOTS' DESIGN TECHNIQUES AND APPROACHES [13]

Year	Programme Name	Winner Designer Name	Design Technique
1991	PC Therapist	Joseph Weintraub	Canned and non-sequitur responses in addition to pattern matching after parsing, and word vocabulary that make it remember sentences.
1992	PC Therapist	Joseph Weintraub	
1993	PC Therapist	Joseph Weintraub	
1994	TIPS	Thomas Whalen	A personal history model database like the system with pattern matching.
1995	PC Therapist	Joseph Weintraub	The same as in 1991.
1996	HeX	Jason Hutchens	Has got a trick sentences database, Markov Chain models, pattern matching, and a model of personal history.
1997	Converse	David Levy	A database for facts, pattern matching, proactivity, WordNet synonyms, a statistical parser, ontology, a list of proper names, and a modular of weighted modules.
1998	Albert One	Robby Garner	Hierarchical structure of previous Chatbots, such as Fred, Eliza, pattern matching and proactivity.
1999	Albert One	Robby Garner	
2000	A.L.I.C.E	Richard Wallace	Advance pattern matching, AIML.
2001	A.L.I.C.E	Richard Wallace	
2002	Ella	Kevin Copple	Language tricks, phrase normalisation, pattern matching, WordNet, and expanding abbreviation.
2003	Jabberwock	Juergen Pirner	Markov Chains, simple pattern matching, context free grammar (CFG), and parser.
2004	A.L.I.C.E	Richard Wallace	The same as in 2000.
2005	George (Jabberwacky)	Rollo Carpenter	No scripts or pattern matching, a huge database of responses of people, and they are based on the Chatbot Jabberwacky.
2006	Joan (Jabberwacky)	Rollo Carpenter	
2007	UltraHAL	Robert Medeksza	Scripts of pattern matching and VB code combination.
2008	Elbot	Fred Roberts	Commercial Natural Language Interaction system.
2009	Do-Much-More	David Levy	Intelligent Toys Commercial Property.
2010	Suzette	Bruce Wilcox	AIML based chat script with database of variables, triples and concepts.
2011	Rosette	Bruce Wilcox	
2012	Chip Vivant	Mohan Embar	Responses using unformatted chat script and AI, and ontology.
2013	Mitsuku	Steve Worswick	Based on rules written in AIML [17].
2014	Rose	Bruce Wilcox	It contains a comprehensive natural language engine to recognise the meaning of the input sentence accurately. A chat script is also included in the design [18].

Matching of words or phrase patterns for Chatbots with keywords needs to be as accurate as possible. The pattern matching for language 'query' for AIML is simpler than for example SQL. However, this does not mean that AIML is a simple question and answer database. It depends on more than one matching category because it uses a recursive tag like <srai> [19]. It is important to give a variety of responses from the knowledge base to achieve the highest number of possible matches.

G. SQL

A Relational Data Base (RDB) is one of the techniques recently used to build Chatbot knowledge bases. The technique has been used to build a database for a Chatbot, i.e. to enable the Chatbot to remember previous conversations and to make the conversation more continuous and meaningful. The most familiar RDB language is SQL (Structured Query Language), which can be used for this purpose.

SQL or MYSQL has gained a high recognition in RDB because it is the high-level language for nonprocedural data. Query blocks nesting to arbitrary depths is one of the most interesting features of it, and the SQL query can be divided into five basic kinds of nesting. Algorithms are developed to change queries that include these basic nesting types into "semantically equivalent queries". Semantically equivalent series are adjustable to achieve effective processing via existing query processing subsystems. SQL as a data language is implemented in ZETA; also as a calculus-based and block-structured language, it is implemented in System R, ORACLE, as well as SEQUEL[21]. Some researchers, as seen in the next sections, have recently used SQL to generate a database that saves the conversation history in order to make a search for any word or phrase match easier. This technique gives continuity and accuracy to the dialogue because it enables the dialogue system to retrieve some previous information history.

III. Speech Analysis and Response

Speech analysis can be divided into three stages: (i) voice recognition and conversion to text, (ii) text processing, and (iii) response and action taking. These stages are explained as follows:

Firstly, speaker independent speech passes through a microphone to a digital signal processing package built in the computer to convert it into a stream of pulses that contain speech information. Specific instructions can be used to read input speech then to convert it into text. This stage provides speech text for processing in the next stage. The diagram which illustrates this stage is shown in Fig. 2.

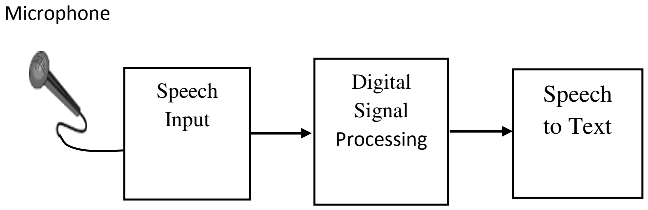

Fig. 2. The stage of speech recognition and converting to text

Secondly, the resulting text is split into separate words for tagging with parts-of-speech labels according to their positions and neighbours in the sentence. Different types of grammar can be used in this stage to chunk the individual tagged words in order to form phrases. Keywords can be extracted from these phrases by eliminating unwanted words in chinking operations. These keywords can be checked and corrected if they are not right. The phases of the text processing stage are shown in Fig. 3.

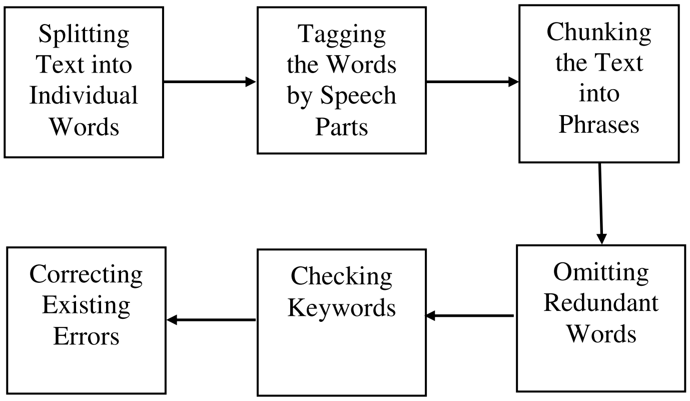
Splitting Text into Individual Words
Tagging the Words by Speech Parts
Chunking the Text into Phrases
Omitting Redundant Words
Checking Keywords
Correcting Existing Errors

Fig. 3. The Stage of Text Processing

Finally, a Chatbot can be built to give the desired intelligent response to a natural language speech conversation. The input to this Chatbot is keywords released from the speech text processing; the output is the programmed response, which will be, for example, an application running or any other text or speech response. Fig. 4 shows a brief diagram of the third stage.

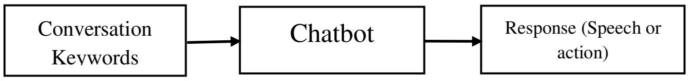

Fig. 4. The Stage of Response and Action Taking

a) Main Parameters

Conversation techniques between a human and a computer can be either chatting by typing text or speech dialogue using the voice. The processing of the information in both techniques is the same after converting speech to text in the case of speech dialogue. A diagram showing the main steps of analysis and processing required to perform human computer conversation is shown in Fig. 5.

The main parameters which affect human computer interaction quality in conversational systems design are: (i) the techniques used to analyse the text using different grammar sets to produce keywords, (ii) pattern matching techniques used inside the Chatbot and depend on a variety of data base access techniques and (iii) the type of response according to the specific application. The focus in this survey is mainly on Chatbot design techniques and a comparison is made between them in terms of the software used, the contribution to the research field in new techniques, and the breadth and depth of the knowledge base used.

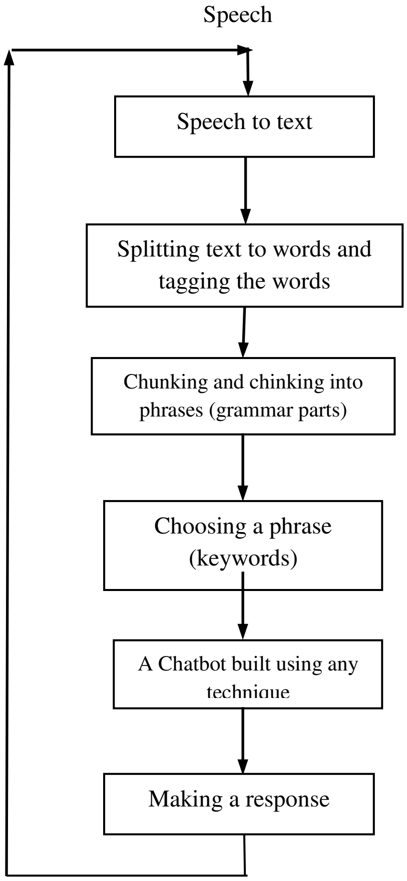
Speech
Speech to text
Splitting text to words and tagging the words
Chunking and chinking into phrases (grammar parts)
Choosing a phrase (keywords)
A Chatbot built using any technique
Making a response

Fig. 5. The main steps of analysis and processing to perform human computer conversation

IV. A Review on Recent Chatbot Design Work

A considerable body of work is associated with Chatbots and they have recently become a promising technique for human-computer interaction. Dialogue systems have been built to meet a variety of applications and can be applied in a number of fields. A number of selected studies between 2003 and 2013 are reviewed and explained below.

- Although creating a new type of Chatbot is a contribution to the field there are a limited number of options available to the software designer. The authors in [10] created knowledge bases for Chatbots by combining the attributes of two other Chatbots. The authors processed the knowledge bases using three filters to eliminate overlapping, identify personal questions, and reject unwanted words or topics. The corpus is built from a combination of an ALICE foundation type Chatbot, which is a *QA form,* and another, such as CLEVERBOT or JABBERWACKY, which are good for handling conversational chatter. The authors processed the Chatbot to either *dialog* or *QA pair* format according to gathered interaction ordering. Then, according to the processed interaction, they produced a Chat corpus with around 7800 pairs of interactions in total. The purpose of their study was to improve Chatbot design techniques.
- Chatbots tend to evolve from one contribution to the next with extensions added by subsequent researchers, adding new features to the software. The author in [22] looked at how to extend *serious types* of games by adding dialogue using simple Chatbots. In fact, it is a serious and positive step in conversation insertion into the games world. The existing *serious game* EMERGO has been used as a case study of the work. The author describes the Chatbot-EMERGO, which is designed to train students or trainees in a medical treatment environment [22]. The purpose of the study is to enhance speech interaction between the training programme and the trainees or students.
- A new Chatbot can be designed to solve health problems or any other application in a wide variety of fields. In [23] the authors presented the Chatbot ViDi (Virtual Dietician) that interacts with diabetic patients as a virtual adviser. The authors proposed a special design for the Chatbot ViDi to make it remember the conversational paths taken during the question and answer session. The path splits into three levels of 9 questions each and it can be obtained by analysing the parameter Vpath which determines the path taken by the patient. The natural language that is used to interface with the user is the Malaysian local language.
- An extension has been made to the chat bot ViDi when the authors in [24] proposed the entire redesign of the ViDi Chatbot by employing the advantages of a relational database. They also added an extension and prerequisite algorithm to update ViDi into a web-based Chatbot. The authors used web programming languages such as PHP, HTML and XHR to implement the coding of the Chatbot in addition to Asynchronous Javascript + XML (AJAX). Again Malaysian is used. The extension of ViDi designed in [23] makes it available to users on the internet through a web browser.
- Pattern matching techniques can also be applied in the Chatbot design world, and can lead to increased accuracy of retrieval. The authors in [25] proposed a new technique for keyword matching using ViDi, ([23] and updated in [24]) as a test environment. The proposed technique is called One Match or All Match Categories (OMAMC). OMAMC is used to test the generation of possible keywords associated with one sample sentence. Then, the results are compared to other keywords generated by another previous Chatbot around the same sample sentence. It is found that OMAMC improves on keyword matching compared to previous techniques. This new approach is likely to be found in future instantiations of Chatbots.
- Educational systems are another application of Chatbots. The objective is to answer students' questions or to test for an examination by asking questions and assessing the answers. In [26] the authors concentrates on an improvement to the Chatbot CHARLIE (CHAtteR Learning Interface Entity). The platform is an INtelligent Educational System (INES) with an AIML Chatbot incorporated inside. The performance and contribution of CHARLIE are documented in his paper and CHARLIE is able to establish a general conversation with students; it can show the material of the courses they study and it is prepared to ask questions associated with the material learned. Educational applications of dialogue systems are particularly useful and are highly interactive. They can be improved and updated easily since they are used in an academic environment.
- The application of Chatbots to Disability care requires the design of packages and systems in order to empower disabled people with new technologies. The authors in [5] suggested a question-answer educational system for disabled people, considering natural language speech and isolated word conversation. The system has been designed using an AIML knowledge base with limited vocabulary including voice recognition or "groups of phonemes and words". The AIML question-answer system is implemented to give answers to queries, and then training data of 2000 words is used to test it. 200 words of the data were used in the test and 156 of them were recognised; therefore, the system accuracy was 78%. The aim of the study was to insert it in English language tutorial software easy access by disabled people. People with blindness and hand paralysis can benefit from adding this kind of feature into E-learning systems.
- Introducing new matching models represents true innovation within Chatbots. In [27] the author proposed a new model that produces a new sentence from two existing sentences. The study proposes employing a Genetic Algorithm (GA) to build a new

sentence depending on the sentences that are retrieved from an available database. The proposal is presented in order to adapt the GA to a natural language structure.

- The proposal in [27] was implemented when the authors in [9] presented their new approach to Chatbot design. The approach combines indexing and query matching methods with pattern matching and applies Information Retrieval (IR) techniques to produce a new sentence from existing ones. In their study, the existing sentences became the initial population of the GA, then the swap and crossover operators were applied to produce the new sentence as a new generation of the GA. Experimental evaluation for the Chatbot before and after applying the sentence combination approach were presented. The purpose of the approach was to improve the diversity of the Chatbot response. The two main contributions of the study are i) converting two sentences into one and ii) Applying information retrieval techniques to Chatbots.

As seen in the above review, conversational techniques can be applied to a variety of different applications involving the interaction between people and computers. Efforts to insert conversation into these different systems is shown to be useful with all studies concluding that adding a Chatbot to a system or software improves the interaction with the system.

V. Selected Factors Influencing Chatbot Design

Commonalities and differences in Chatbot designs have been highlighted with the Influential factors included in the survey. A summary of these factors can be seen in table 2.

TABLE II. A Summary of the Selected Factors Influencing Chabot Design

Study	**Factors Influencing Chatbot Design**								
	Voice	*Text*	*Creating new Chatbot*	*Using available Chatbots*	*AIML usage*	*SQL usage (Relational Database)*	*Matching technique*	*Corpus (knowledge base)*	*Application*
Pereira et al [10]	Yes	Yes	NO	Yes	Yes	NO	Edger Chatbot matching technique (combination of TfIdf algorithms with natural language normalization)	Edgar Chatbot	Chatbot design.
Rosmalen [22]	NO	Yes	NO	Yes	Yes	Yes	QA matching form	AIML	Medical education
Lokman et al [23]	NO	Yes	Yes	NO	Yes	Yes	QA matching form	VP bot	Health assistance
Lokman et al [24]	NO	Yes	NO	Yes	NO	Yes	Prerequisite Matching	ViDi Chatbot	Health assistance
Lokman et al [25]	NO	Yes	NO	Yes	NO	Yes	One-Match All-Match Category (OMAMC)	ViDi Chatbot	Health assistance
Mikic et al [26]	NO	Yes	NO	Yes	Yes	NO	AIML category pattern matching	AIML	Educational systems
Bhargava et al [5]	Yes	NO	Yes	NO	Yes	NO	AIML category pattern matching	AIML	E-learning
Vrajitoru [27]	NO	Yes	Yes	NO	NO	NO	Genetic Algorithms (GA)	Manual pattern and data chosen	Any
Ratkiewicz [9]	NO	Yes	Yes	NO	NO	NO	Genetic Algorithms (GA)	Manual pattern and data chosen	Any

VI. Summary of Significant Improvements in the Analysed Studies

Each of the selected studies made improvements in Chatbot design. A summary of contributions made is shown in table 3.

VII. Discussion

The examination of factors which influence Chatbot design shows that there are commonalities and differences between the highlighted studies.

Although the processing steps are the same for voice and text after the voice to text conversion, there are distinct differences in the use in conversational systems, particularly in terms of their applications. Text is used in most of the studies, except [5], due to simplicity, whereas voice is used in [5] and [10] for special needs applications e.g. for disabled people. The response in the case of disability applications should be a voice response. The commercial mobile applications (Chatbots) which have emerged recently, e.g. Cortana and Siri, accept speech as an input and give a voice response in addition to text.

New Chatbots have been created in [5], [9], [23], and [27], which add new techniques or use improved previous designs. Also new techniques, algorithms or extensions have been added to existing Chatbots in [10], [22], [24], [25], and [26] in order to improve their function or extend available software by adding chat interaction. For example, the Loebner Prized Chatbot ALICE (which won three times) was improved several times in later iterations, and Joan (Jabberwacky) was the updated form of George (Jabberwacky).

Knowledge bases are built using different techniques. For example, AIML, which is the technique first used with the ALICE Chatbot, is used to build the Chatbots in [5], [10], and [26], while SQL (or RDB) is used in [24] and [25]. Both AIML and SQL are used in [22] and [23]. Neither AIML or SQL are used in [9] and [27]. The use of SQL (no clear evidence of using it in Loebner Prized Chatbots) added a new technique to knowledge-bases, namely the Relational Data Base, which enables the Chatbot to remember previous conversations by accessing the history stored in the database designed using SQL. However, an AIML knowledgebase is still effective for Chatbot designs; for example, Mitsuku Chatbot won Loebner Prize in 2013 and it was based on AIML.

In order to design new Chatbots or extend previous ones, each study has used a corpus that is different from the other as illustrated in table 2. The corpus that is relied on to build a Chatbot affects the design because it affects the knowledge base of the Chatbot and then the accuracy of the response since the response is a knowledgebase reflection.

The application column in table 2 shows that each Chatbot has been designed to meet certain needs for conversation by holding a chat with a specific group of people in a specific organisation. The work in the future needs more focus on general purpose conversational systems by designing Chatbots with more comprehensive knowledge bases in order to cover general topics by using the latest techniques.

Table 3, which covers the contribution presented by each of the selected studies, displays how each has made an improvement to Chatbot design in spite of using different techniques, algorithms, or programmes.

TABLE III. A SUMMARY OF CONTRIBUTIONS FOR CHATBOT DESIGN IN ANALYSED STUDIES

Study	**Significant Improvements**
Pereira et al [10]	Producing a new corpus (knowledge base) that avoids overlapping, identifies personal questions, and rejects unwanted words or topics by combining available QA and dialogue formats.
Rosmalen [22]	Extending an existing serious game by adding a simple Chatbot to give the opportunity for trainees to be aware of work and activities on the first day of their employment.
Lokman et al [23]	Designing a new Chatbot (ViDi) that has the ability to remember previous conversation in order to work as a virtual adviser for diabetic patients.
Lokman et al [24]	Redesigning and extending the Chatbot ViDi by adding the prerequisite matching techniques in order to attain a conversational manner rather than a QA form and make it available to users on the internet via a web browser.
Lokman et al [25]	Proposing a new matching technique OMAMC in order to produce improved results by reducing matching time and increasing context flexibility.
Mikic et al [26]	Updating the Chatbot CHARLIE to incorporate it into the platform INtelligent Educational System (INES) in order to improve the conversation between students and educational systems.
Bhargava et al [5]	Designing a new AIML based Chatbot of natural language speech and limited word input and output so as to use it in an E-learning systems to enable disabled people to learn via speech.
Vrajitoru [27]	Proposing a new innovative pattern matching approach in a Chatbot. The authors adjusted Genetic Algorithms with natural language to generate a new sentence from existing ones in order to improve the diversity of response.
Ratkiewicz [9]	i) Implementing the model proposed in [27], i.e. employing GA in pattern matching to produce a new sentence from sentences retrieved from an existing database in order to increase the diversity of responses. ii). Applying information retrieval techniques to the Chatbot.

VIII. Conclusions

In this paper, the literature review has covered a number of selected papers that have focused specifically on Chatbot design techniques in the last decade. A survey of nine selected studies that affect Chatbot design has been presented, and the contribution of each study has been identified. In addition, a comparison has been made between Chatbot design techniques in the selected studies and then with the Loebner Prize winning Chatbot techniques. From the survey above, it can be said that the development and improvement of Chatbot design is not grow at a predictable rate due to the variety of methods and approaches used to design a Chatbot. The techniques of Chatbot design are still a matter for debate and no common approach has yet been identified. Researchers have so far worked in isolated environments with reluctance to divulge any improved techniques they have found, consequently, slowing down the improvements to Chatbots. Moreover, the Chatbots designed for dialogue systems in the selected studies are, in general, limited to particular applications. General-purpose Chatbots need improvements by designing more comprehensive knowledge bases.

Although some commercial products have emerged recently in the market (e.g. Microsoft Cortana) as dialogue Chatbots, improvements need continuous research and lack a common solution.

Each researcher needs to robustly document any successful improvements to allow the human computer speech interaction to agree a common approach. This will always be at odds with commercial considerations.

References

[1] C. I. Nass, and S. Brave, Wired for speech: How voice activates and advances the human-computer relationship: MIT Press Cambridge, 2005.

[2] Y.-P. Yang, “An Innovative Distributed Speech Recognition Platform for Portable, Personalized and Humanized Wireless Devices,” Computational Linguistics and Chinese Language Processing, vol. 9, no. 2, pp. 77-94, 2004.

[3] J. P. Campbell Jr, “Speaker recognition: a tutorial,” Proceedings of the IEEE, vol. 85, no. 9, pp. 1437-1462, 1997.

[4] C.-H. Lee, "From knowledge-ignorant to knowledge-rich modeling: a new speech research paradigm for next generation automatic speech recognition", 2004.

[5] V. Bhargava, and N. Maheshwari, “An Intelligent Speech Recognition System for Education System,” 2009.

[6] E. Loper, and S. Bird, "NLTK: The natural language toolkit." pp. 63-70, 2002.

[7] S. Bird, "NLTK: the natural language toolkit." pp. 69-72, 2006.

[8] A. M. Galvao, F. A. Barros, A. M. Neves, and G. L. Ramalho, "Persona-aiml: An architecture developing chatterbots with personality." pp. 1266-1267, 2004.

[9] J. Ratkiewicz, “Evolutionary Sentence Combination for Chatterbots Dana Vrajitoru Computer and Information Sciences Indiana University South Bend, 1700 Mishawaka Ave,” 2004.

[10] M. J. Pereira, and L. Coheur, “Just. Chat-a platform for processing information to be used in chatbots,” 2013.

[11] D. J. Stoner, L. Ford, and M. Ricci, “Simulating Military Radio Communications Using Speech Recognition and Chat-Bot Technology,” 2003.

[12] K. Meffert, "Supporting design patterns with annotations." pp. 8 pp.-445, 2006.

[13] D. Mladenić, and L. Bradeško, “A survey of chabot system through a Loebner prize competition,” 2012.

[14] A. M. Turing, “Computing machinery and intelligence,” Mind, pp. 433-460, 1950.

[15] B. Kirkpatrick, and B. Klingner, “Turing’s Imitation Game: a discussion with the benefit of hind-sight,” Berkeley Computer Science course. See http://www. cs. berkeley. edu/~ christos/classics/ttest. pdf Accessed, vol. 1, pp. 13, 2009.

[16] P. Hingston, “A turing test for computer game bots,” Computational Intelligence and AI in Games, IEEE Transactions on, vol. 1, no. 3, pp. 169-186, 2009.

[17] R. Higashinaka, K. Imamura, T. Meguro, C. Miyazaki, N. Kobayashi, H. Sugiyama, T. Hirano, T. Makino, and Y. Matsuo, "Towards an open domain conversational system fully based on natural language processing", 2014.

[18] B. Wilcox, "Winning the Loebner's," http://www.gamasutra.com/blogs/BruceWilcox/20141020/228091/Winning_the_Loebners.php, 2014.

[19] R. Wallace, “The elements of AIML style,” Alice AI Foundation, 2003.

[20] M. d. G. B. Marietto, R. V. de Aguiar, G. d. O. Barbosa, W. T. Botelho, E. Pimentel, R. d. S. França, and V. L. da Silva, “Artificial Intelligence MArkup Language: A Brief Tutorial,” arXiv preprint arXiv:1307.3091, 2013.

[21] W. Kim, “On optimizing an SQL-like nested query,” ACM Transactions on Database Systems (TODS), vol. 7, no. 3, pp. 443-469, 1982.

[22] P. Van Rosmalen, J. Eikelboom, E. Bloemers, K. Van Winzum, and P. Spronck, “Towards a Game-Chatbot: Extending the Interaction in Serious Games,” 2012.

[23] A. S. Lokman, and J. M. Zain, "An architectural design of Virtual Dietitian (ViDi) for diabetic patients." pp. 408-411, 2009.

[24] A. S. Lokman, and J. M. Zain, “Extension and prerequisite: An algorithm to enable relations between responses in chatbot technology,” Journal of Computer Science, vol. 6, no. 10, pp. 1212, 2010.

[25] A. S. Lokman, and J. M. Zain, “One-Match and All-Match Categories for Keywords Matching in Chatbot,” American Journal of Applied Sciences, vol. 7, no. 10, pp. 1406, 2010.

[26] F. A. Mikic, J. C. Burguillo, M. Llamas, D. A. Rodríguez, and E. Rodríguez, "CHARLIE: An AIML-based Chatterbot which Works as an Interface among INES and Humans." pp. 1-6, 2009.

[27] D. Vrajitoru, "Evolutionary sentence building for chatterbots." pp. 315-321, 2003.