A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks

Mohammad Nuruzzaman
School of Business
University of New South Wales
Canberra ACT, Australia
dr.zaman1981@gmail.com

Omar Khadeer Hussain
School of Business
University of New South Wales
Canberra ACT, Australia
o.hussain@adfa.edu.au

Abstract*— Nowadays it is the era of intelligent machine. With the advancement of artificial intelligent, machine learning and deep learning, machines have started to impersonate as human. Conversational software agents activated by natural language processing is known as chatbot, are an excellent example of such machine. This paper presents a survey on existing chatbots and techniques applied into it. It discusses the similarities, differences and limitations of the existing chatbots. We compared 11 most popular chatbot application systems along with functionalities and technical specifications. Research showed that nearly 75% of customers have experienced poor customer service and generation of meaningful, long and informative responses remains a challenging task. In the past, methods for developing chatbots have relied on hand-written rules and templates. With the rise of deep learning these models were quickly replaced by end-to-end neural networks. More specifically, Deep Neural Networks is a powerful generative-based model to solve the conversational response generation problems. This paper conducted an in-depth survey of recent literature, examining over 70 publications related to chatbots published in the last 5 years. Based on literature review, this study made a comparison from selected papers according to method adopted. This paper also presented why current chatbot models fails to take into account when generating responses and how this affects the quality conversation.

***Keywords*— Chatbot, Neural Network, Deep Learning, Natural Language Processing, Dialogue System.**

I. INTRODUCTION

Customer satisfaction with a company's services is often seen as the key to success and long-term competitiveness for a company. The insurance industry such as credit card insurance, is getting a lot of attention as customer satisfaction. Credit card insurance is a competitive market so a strong marketing strategy is vital [1]. Its inclusions are confusing and complex, in a world dominated by cashless payments, consumers are using credit cards at a growing rate. Most credit cards offer their consumers some form of embedded complimentary insurance product. Consumers are often not aware of these complementary products and it is difficult to understand the inclusions and benefits. For example, the majority of cards and accounts include complimentary travel insurance, however, customers are not aware of the detail around what this cover includes if the cover includes family or travelling companions, how the cover is activated and who to call when they need help or need to make a claim. In addition, insurance personnel required reference materials, policies and procedures. Getting all of this information they need is a challenge. Insurance personnel had to sift through long documents to find the answer. As a result, the only way to get help quickly was to pick up the phone and talk to underwriting or sales support – even for answers to FAQs or to basic "how-to" questions. This overloaded the call centres, resulting in long wait times as it takes a long time to process a single request. As a result, customer experience their interactions disappointed and dissatisfied which reduces the throughput and business performance drastically. Research showed that nearly 75% of customers have experienced poor customer service [2-4].

The technology platforms allow modelling the entire credit card insurance ecosystem with Artificial Intelligent (AI) to simulate scenarios of different economic, market and individual conditions. There is an increase in the demand for AI capabilities to interact with customers in benefits, insurance coverages and claims processes. Because it removes human factors and provides 24-hours service. This will advise the customer on the most appropriate course of action such as help customers to make clearer and easier to understand embedded benefits into a credit card, summarise level of coverage and insurance claims process. It will allow customers to utilize credit card coverages with the peace of mind, knowing they have independent experts looking after them. Furthermore, it can generate revenue and save costs for the credit card insurance industry.

In order to truly be effective and make business processes automated an alternate system is required. An advance dialogue system known as AI chatbot application system could automate the entire business processes. Thus, chatbot application system must have natural language processing (NLP), deep neural networks (DRN) so that it can understand what customers are looking for. In the case of the credit card insurance industry, chatbot can be used to answer basic questions, resolve insurance claims, sell products and make sure customers are properly covered by their insurance. AI chatbot can analyse data better than humans to more accurately predict each customer's risk, thereby providing customers with the right amount of insurance and companies with protection from risky customers.

II. BACKGROUND OF THE PROBLEM

The traditional chatbot's dialogue capability is too inflexible. It can answer to the user only if there is a pattern (lexical) matching between the user query and set of question-answer stored in its knowledge base [5]. The answers are given using a set of predefined responses. Traditional chatbots are lacking in the intuitive capability of human beings to see the meaning, relationships and possibilities beyond the reach of sense.

There are two main tasks in deep learning (DL). The first is to extract meaning from the input. The second is to generate an output from that, either a translation or a response in the case of a chatbot application. The major challenge in developing a good model is that creates an adequate sense of context and effectively related inputs to outputs. The sequence-to-sequence (seq2seq) model in deep recurrent neural networks (DRNN) with attention mechanism

978-1-5386-7992-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICEBE.2018.00019

provides an appropriate architecture to meet these challenges.

The purpose of this study is to explore the capability of the deep neural network to engage in human conversation, while at the same time sidestepping some of the limitations of statistical models and implementation mechanism.

III. LITERATURE REVIEW

*A. **Definition of a chatbot***

A chatbot is a conversational software system that is designed to emulate communication capabilities of a human being that interacts automatically with a user. It represents a new, modern form of customer assistance powered by artificial intelligence via a chat interface.

Chatbots are based on AI techniques that understand natural language, identify meaning, emotion, and design for meaningful responses. For example, it makes it easy for customers to get responses to their queries in a convenient way without spending their time waiting in phone queues or send repeated emails. Chatbots can reduce the number of customer calls, average handling time and cost of customer care. However, it is not easy to achieve these functionalities as it requires various complex interactions between systems. Note that the word ‘AI chatbot application system’ or ‘AI chatbot’ is used in this study as a synonym for a conversational agent or advanced dialogue system.

*B. **Taxonomy of Chatbot***

The recent interest in chatbots can be attributed to two key developments [6]. Firstly, messaging service growth has spread rapidly over the past few years. It incorporates features such as payments, ordering and booking, which would require a separate application or website. So rather than downloading a series of separate applications, users can perform tasks such as buy goods, book restaurant and ask questions all through their favourite messaging apps. Example of some of the popular apps are Facebook Messenger, WhatsApp, WeChat and Line. Secondly, advanced AI techniques in combination with machine learning and deep learning techniques have made considerable progress to improve the quality of understanding and decision making on cheap processing power. It can handle the vast amount of data and process it to get results that exceed human performance.

Chatbot applications can be grouped into four different categories, namely service, commercial, entertainment and advisory chatbot [7]. Service chatbots are designed to provide facilities to customers. For example, logistics firm to respond to questions about deliveries and provide copies of dispatch documents through instant messaging channel rather than emails or phone calls. Commercial chatbots are designed to streamline purchases for customers. For example, a pizza company can take delivery orders or notify promotions via messaging interface. Entertainment chatbots are designed to keep customers engaged with sports, favourite band, movies or other events. It offers the option of placing bets, detail on upcoming events and ticket deals. Advisory chatbots are designed to provide suggestions, give recommendations on service, offer maintenance or repair goods. This type of chatbot can contact people, offer support and advice tips when it is needed.

According to [8], chatbot application can be classified into two groups such as task-oriented and non-task-oriented. Task-oriented chatbots aim to assist the customers to complete certain tasks and have short conversations. For example, Siri, Google Now, Alexa dialogue agents can give travel directions, find restaurants and help to make phone calls or texts. On the other hand, Non-task-oriented chatbots focus on conversing with customers to answer questions and entertainment.

In this paper, we divided chatbot applications into four groups such as goal-based, knowledge-based, service-based and response generated-based as shown in Fig.1.

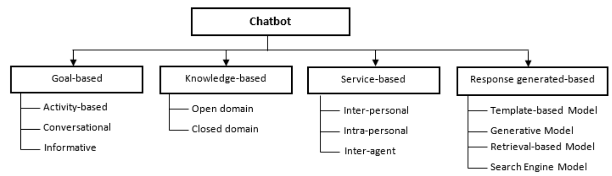

Fig 1 **Taxonomy of Chatbot Application**

i. Goal-based Chatbot

Goal-based chatbots are classified based on the primary goal aim to achieve. They are designed for particular task and setup to have short conversations to get information from the user to complete the task. For example, a company deploys chatbot on their websites to help the customer to answer their question or address problems.

ii. Knowledge-based Chatbot

Knowledge-based chatbots are classified based on the knowledge they access from the underlying data sources or the amount of data they are trained on. The two main data sources are open-domain and closed-domain. Open-domain data sources answer depends on general topics and respond appropriately. Example of open-domain are Allen AI Science and Quiz Bowl. Closed-domain data sources focus on a particular knowledge domain. All information required for answering the question is provided in the dataset itself such as Daily Mail, MCTest and bAbI.

iii. Service-based Chatbot

Service-based chatbots are classified based on facilities provides to the customer. It could be personal or commercial purpose. For example, logistics company could provide copies of dispatch documents through chatbot rather than phone calls or customer can make a meal order from MacDonald.

iv. Response Generated-based Chatbot

Response Generated-based chatbots are classified based on what action they perform in response generation. The response models take input and output in natural language text. The dialogue manager is responsible for combining response models together. To generate a response, dialogue manager follows three steps. First, it uses all response models to generate a set of responses. Second, returns a response based on priority. Third, if no priority response, the response is selected by the model selection policy.

The focus of this research is on response generated-based chatbot. In this category, there are various response models that are based on four categories namely—Template-based

Model, Generative Model, Retrieval-based Model and Search Engine Model as shown in Fig 2.

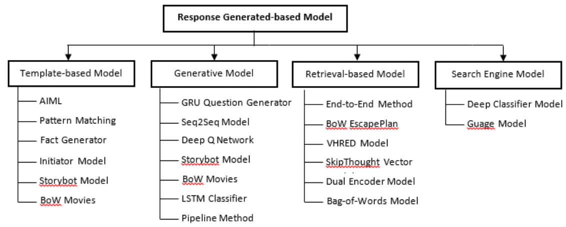

Fig. 2 **Classification of Generated-based Models**

C. Existing Chatbots

i. Elizabot

Elizabot is one of the earliest well-known chatbots in its long history. It was developed at MIT Lab in 1966 [9] and was intended to demonstrate natural language conversation between humans and machines to provide Rogerian psychotherapy. As Rogerian psychotherapy primarily encourages the patient to talk more rather than engaging in a discussion. Elizabot responses are personal questions that are meant to engage the patient to continue the conversation. It uses rule-based techniques and a script to respond to patient's questions with keyword matching from a set of templates and context identification. The model detects the appropriate template and selects the corresponding responses. If there are multiple templates, a template is selected randomly. which the model runs it through a set of reflections to better format the string for a response. Elizabot was able to convince some people and aid patient's treatment suffering from psychological issues. Nonetheless, Eliza could not provide anything comparable to therapy with a human therapist.

The drawback of Elizabot is to keep a conversation going. Besides, Eliza is incapable of learning new patterns of speech or words, discover context through interaction and logical reasoning capabilities [10].

ii. Alicebot

Artificial Linguistic Internet Computer Entity also referred to as ALICE. It was inspired by [9] and developed by [11]. Alicebot is based on the updated version of Eliza's pattern or architecture. Nevertheless, Alicebot is still purely based on pattern matching and depth-first search technique to user's input. It is a form of XML dialect that encodes rules for questions and answers. It uses a set of artificial intelligence markup language (AIML) templates to produce responses given to the dialogue history and user utterance [12]. At first, AIML receives the user sentence as input and stored in known as a category. Each category consists of a response template and set of conditions that give meaning to the template know as context. Then the model prepossesses it and matched against nodes of the decision tree. When user input is matched, the chatbot will response or execute an action. The AIML templates repeat the user's input utterance using recursive techniques and it is not always meaningful responses. Therefore, string-based rules are required to determine if the response creates a correct or meaningful.

The drawback of Alicebot is modelling of personality to define the chatbot behaviour such as traits, attitudes, mood, emotions and physical states [13]. The botmaster must integrate personality elements within the AIML. However, this is not a straightforward task. Alicebot is also incapable of generating appropriate responses, no reasoning capabilities and unable to generate human-like responses (Turing test). It requires a large number of categories to create a robust bot and may lead to unfeasible, difficult to maintain or time-consuming application. Alicebot does not have intelligence features like NLU, sentiment analysis and grammatical analysis to structure a sentence. In addition, if the same input repeats during the conversation, Alicebot gives same answers at most of the time.

iii. Elizabeth bot

It is an adaptation of Eliza program developed by [14]. However, it has been enhanced and generalized to increase both flexibility and its potential adaptability in selection, substitution, and phrase storage mechanism. Elizabeth bot uses four steps to generate a response. First is a command line script in a text file, where each line is started with a script command notation, no keyword message. These notations are single characters, one for each rule-type such as- 'W' for welcome message, 'Q' for quitting message, 'N' for no match etc. Each script command has an index code that is generated automatically. It can also be indexed using a user special code. Second is input transformation rules and map input to another form to be compatible with the defined keywords. Third is output transformation rules and changes personal pronouns to be appropriate as a response. Fourth is first keyword patterns to be matched. It tries to give a different answer by using different selection responses for the same question [15]. The nature of some rules in Elizabeth bot may cause iteration, which is solved by applying the rule only once.

The drawback of Elizabeth bot is, it does not provide a way to partition or splitting the user input sentence and then combine their results. According to Elizabeth bot's structure, it will be difficult to do the splitting. Furthermore, a lot of complicated appears related to writing some rules in uppercase and others in lowercase which may cause a lot of errors and give unsuitable answers. However, Elizabeth bot has the ability to give the derivation structure for a sentence using the grammatical analysis, keywords extraction and pattern matching.

iv. Mitsuku

Mitsuku is a most widely used standalone human-like chatbot created by [16] using AIML. It was designed for general typed conversation based on rules written in AIML [17] and an integration in a bot network such as twitter, telegram, firebase, twilio to serve as a personality layer. Mitsuku bot uses NLP using heuristic patterns and hosted at Pandorabot. Bot modules abstract a lot of the work that goes into creating a robust chatbot system. In order to integrate its module, need to include some AIML categories to route inputs from users. Whenever bot fails to find a better match for an input, it will automatically redirect to the default category. Mitsuku can hold a long conversation, learns from the conversation, remembers personal details about the user (age, location, gender, etc.). Its feature includes the ability to reason with specific objects. For example, if someone says "Can you eat a house?" Mitsuku will look up the properties for "house" and find the value of "made_from" is set to "brick" and reply "No" as a house is not edible. Mitsuku is a multi-lingual bot and uses supervised machine learning. As it learns something new, the data is sent to the human manager for verification. Only verified data can be further incorporated and used by the app. However, Mitsuku is not effective without a large amount of training data, fail to provide dialogue management components.

v. Cleverbot

Cleverbot is one of the most popular entertainment chatbots that implement rule-based AI techniques to communicate with human [18]. It is developed by [19] to collect a large amount of data based on conversational exchanges with people online through crowdsourcing. Unlike other chatterbots, Cleverbot's responses are not pre-programmed. Instead, it simulates natural conversation by learning from user input and relying on feedback to interact. When user input a sentence, Cleverbot finds all keywords or phrase matching the input. After searching through its saved conversations, it responds to the input by finding how a user responded to that input when it was asked. Cleverbot is unique in that it ''learns'' what users have said to it in previously saved conversations and uses this knowledge to determine how to respond to new conversations [20]. To enhance the realism of the conversation, the bot has its own human avatar that shows emotions. The underlying technology in Cleverbots not only processes verbal and textual interactions but also facial expressions and movements to create a more authentic conversation. The drawback of Cleverbot is unpredictable responses and bad tendency to suddenly change the subject and respond without context. It is also unable to continue a long conversation, not accurate in language translation and may not suitable for children.

vi. Chatfuel

Chatfuel provides a drag and drop user-friendly interface for making a rule-based chatbot. It was developed by [21]. It is an artificial intelligence module train the bot to map input sentences to output. It allows response prompts and integration with services such as social media, third-party, CRM. With analytics capabilities, users can collect and view valuable information on chatbot performance and subscriptions quickly and effectively. Users can dictate the conversational rules via the Chatfuel dashboard to ensure the chatbot understands and answers user requests efficiently. It also allows a json integration for accommodating custom logic into the bot. the most attractive point of service is simple to build a rule-based bot which is suitable for small businesses. The drawback of Chatfuel is- it is quite inflexible in terms of conversation flows and does not support knowledge-based and multi-language. Additionally, NLP is limited, funky to setup and poor documentation. However, it is capable of grabbing the user's intent.

vii. ChatScript

ChatScript is a scripting-based commercial chatbot developed by [22]. It uses pattern matching techniques similar to AIML. It is a combination NLP engine and dialogue management system. It included some control scripts. This is merely another ordinary topic of rules that invokes API functions of the engine. A rule consists of a type, label, pattern and output. Rules are bundled into collections called topics such as keywords that allows the engine to automatically search the topic for relevant rules based on user input. Unlike AIML, which finds the best pattern match for an input, ChatScript first finds the best topic match, then executes a rule contained in that topic. ChatScript is well suited for stand-alone application such as information kiosks, help desk. Though it has an excellent documentation, it is difficult to implement. The drawback of CharScript is it is difficult to learn and there are no hosting services. It is also difficult to embed in a web page [20].

viii. IBM Watson

Watson is rule-based AI chatbot developed by IBM's DeepQA project [23]. It is designed for information retrieval and question-answering system that incorporates natural language processing and hierarchical machine-learning method. Watson uses a broad range of mechanisms to identify and assign feature values such as names, dates, geographic locations or other entities to generated response. The machine learning system then learns how to combine the values of these features into a final score for each response. Based on that score, it ranks all possible answers and selects one as its top answer. Watson incorporates a variety of technologies including Hadoop, Apache Unstructured Information Management Architecture (UIMA) framework to examines the phrase structure and the grammar of the question to better gauge what's being asked.

Applications for the Watson's underlying cognitive computing technology are almost endless. Because it can process text mining and complex analytics on huge volumes of unstructured data and handle enormous quantities of data. As the application gains experience with more input, it can find enough patterns to make accurate predictions. Besides the advantages of Watson, it has some major drawback such as it does not process structure data directly, no relational databases, higher maintenance cost, targeting towards bigger organizations and take longer time and effort to teach Watson in order to use its full potential.

ix. Microsoft LUIS

Language Understanding Information Service (LUIS) is a domain-specific AI engine developed by Microsoft [24]. It makes natural language and information processing using intents and prebuilt domain entities model. LUIS performs NLP against Big Data to find intents from a sentence. It is designed to identify valuable information in conversations, interprets user goals (intents) and extracts information (entities). Active learning is also used to continuously improve the quality of the natural language models. A model starts with a list of general user intentions such as "Book Flight" or "Contact Help Desk." Once the intentions are identified, user supply example phrases called utterances for the intents. Then label the utterances with any specific details user wants LUIS to pull out of the utterance. After the model is designed, trained, and published, it is ready to receive and process utterance. LUIS receives the utterance as an HTTP request and responds with extracted user intentions.

Powerful developer tools are combined with customizable pre-built apps and entity dictionaries, such as Calendar, Music, and Devices, so the user can build and deploy a solution more quickly. Dictionaries are mined from the collective knowledge of the web and supply billions of entries, helping the model to correctly identify valuable information from user conversations. The major drawback of LUIS required Azure subscriptions. However, LUIS integrates seamlessly with the Azure Bot Service, making it easy to create a sophisticated bot.

x. Google Dialogflow

Dialogflow known as Api.ai and it was developed by Google [25] and part of Google Cloud Platform. It lets app developers provide their users to interact with interfaces through voice and text exchanges powered by machine learning and natural language processing technologies. This lets them focus on other integral parts of app creation rather than on delineating in-depth grammar rules. Dialogflow recognizes the intent and context of what user says. Then match user input to specific intents and uses entities to extract relevant data from them. And finally, allow the conversational interface to provide responses. The drawback of Dialogflow is no handheld device version, not interactive user interface and poor documentation.

xi. Amazon Lex

Amazon Lex is an AWS service for building conversational interfaces into applications using voice and text. It was developed by Amazon [26]. It provides deep learning functionality and flexibility of natural language understanding (NLU) and automatic speech recognition (ASR) to build highly engaging user experiences with lifelike, conversational interactions. Amazon Lex integrates with AWS Lambda that user can easily trigger functions for execution of back-end business logic for data retrieval and updates. The drawback of Amazon Lex is not multilingual, currently, support only English. Unlike Watson, Lex has a critical process to follow for web integration. Besides that, preparation of dataset is complicated, the utterances and entities mapping are somewhat critical.

IV. RELATED WORKS

Information extraction and user intention identification are central research topics in natural language processing (NLP). There have been several models introduced by researchers in past years. Recent development in deep learning, deep neural network models have shown interest and promise for building self-learning chatbots. However, there have been several related attempts to address the seq2seq model problems with deep learning approaches such as recurrent neural networks (RNN), deep neural networks (DNN) and convolutional neural networks (CNN).

Wu et. al., (2017) studied the problem of response selection for long conversation in retrieval-based chatbots. The task required matching a response candidate with a conversation context, whose challenges include how to recognize important parts of the context, and how to model the relationships among utterances in the context. Existing matching methods may lose important information in contexts. The authors interpret them with a unified framework in which contexts are transformed to fixed-length vectors without any interaction with response before matching. The authors proposed a new matching framework called sequential matching framework (SMF) that can sufficiently carry the important information in contexts to matching and model the relationships among utterances at the same time. At the first step, SMF interacts with a response and transforms the pair to matching vector. The matching vectors are then accumulated following the order of the utterances in the context with a recurrent neural network (RNN). Finally, the context-response matching is calculated with the hidden states of the RNN. The study used a sequential convolutional network and sequential attention network. In the evaluation, the study conducted an experiment on two public datasets to test their performance. Experimental results showed that both models can significantly outperform the state-of-art matching methods. The authors demonstrated the models are interpretable with visualizations that provide insights on how to capture and leverage the important information in contexts for matching.

Stroh and Mathur [28] applied RNN to solve question answering task. Authors created an example in Tensorflow GRU using seq2seq model with GloVe word vectors proposed in [29]. They separated representations for the query and each sentence of an essay. At first, RNN encoder process the essay and mark ‘Q’ as question start symbol. Then added ‘GO’ symbol to instruct the network to start decoding. The decoder produces an answer sequence followed by ‘STOP’ symbol. This indicates that processing has ended. The study trained the RNN using cross-entropy error on the decoder output with bAbI English dataset. The dataset is a set of 20 QA tasks with 1000 training examples. Each task consists of several context-question-answers. It prepared and released by Facebook. Each task aimed to test a unique aspect of reasoning and towards testing a specific capability of QA learning model. The result shows that this approach often underperforming relative to other approaches such as dynamic memory networks, end-to-end networks. However, it tended to do well on tasks with yes/no questions. The Authors suggested that the model might be improved if it is replaced with an attention mechanism that treats sentences independently. Additionally, sentence selection schema can be replaced by a more intelligent sentence selection module with learnable weights.

One study [10] introduced an attention mechanism that allows DNN to focus on different parts of their input. This idea was successfully applied to neural machine translation (NMT) by [30]. Followed by [31], the authors form an opinion that the use of a fixed-length context vector is problematic for translating long sentences. Their approach was multilayered Long Short-Term Memory (LSTM) with a limited vocabulary. They used one LSTM to map the input sequences to a vector of fixed dimension and then another deep LSTM to decode the target sequences from the vector. When they concatenate an input sentence with a target sentence, each word is far from its corresponding word. The average distance between corresponding words in input and target sentence is unchanged. During training, the authors were able to reverse the order the words in the source sentences, but not the target sentences. By doing so, they introduced many short-term dependencies that made the learning problem much simpler. The simple trick of reversing the words in the source sentences is the key contribution of their work. The result of their work obtained a BLEU score of 34.81 on the WMT’14 dataset to rerank 1000 hypotheses. It was involved in four deep LSTM layers using a beam-search decoder and 1000 parameters at each LSTM layer. The result suggested that the approach would likely do well on other seq2seq problems. Finally, they demonstrated that a simple, straightforward and unoptimized approach could outperform a mature LSTM system.

In the seq2seq approach, the decoder has to keep track of the output and generated content can be transformed from the relevant parts in the source. However, standard seq2seq model struggles with generating long responses since it has to keep track of everything. And decoder has fixed-length hidden state vector which leads to incoherent or even contradictory outputs. To combat this, one of most recent study [32] has introduced chatbot's response generation problem through a practical approach, referred as seq2seq with glimpse model and stochastic beam-search decoding technique. Authors' defined input sequence is the conversation history and the output sequence is the response. At the first, glimpse model included on the encoder side and then trained dataset on fixed-length segments of the target-side. It allowed scaling up the training to larger datasets without running into any memory issues. Second, to generate long, coherent and diverse responses using MAP-decoding of the beam-search technique. It is to break up the reranking over shorter segments and rerank segment-by-segment. Thereby, injecting diversity earlier during the decoding process. Finally, they integrated target-side-attention mechanism into the decoder so it can keep track of what has generated. Together, their study found that these two methods lead to increase capacity in the hidden state for modelling semantics required during generation of longer responses. In the end, authors have trained on a combined dataset of over 2.3B conversation messages from the web. In human evaluation studies showed that their method generated longer responses with a higher proportion rated as acceptable and excellent.

According to Microsoft and Standford University research (2016), showed how to apply DL to model future reward in chatbot dialogue. despite the success of Seq2Seq models in dialogue generation, two problems arise. First, Seq2Seq models are trained by predicting the next dialogue turn in a given conversational context. The authors used the maximum-likelihood estimation (MLE) objective function. However, it is not clear how well MLE the real-world goal of chatbot development, teaching a machine to converse with human, providing informative feedback, interesting and diverse. Second, the dialogue system becomes stuck in an infinite loop of repetitive responses. This is due to MLE-based Seq2Seq model's inability to account for repetition. The authors suggested that a conversation framework model needed that has the ability to integrate developer-defined rewards which mimic the true goal of chatbot development. The model used the encoder-decoder architecture as its backbone, simulated conversation between two virtual agents to explore the space of possible actions while learning to maximize expected reward. The parameters of an encoder-decoder RNN define a policy over an infinite action space consisting of all possible utterances. At the end, the authors evaluated the model on diversity, length as well as human judges. The result demonstrated that the proposed framework model generates more interactive responses and manages to foster a more sustained conversation in dialogue simulation.

In [9], the authors proposed an attention-based seq2seq mechanism. They enhanced the top hidden vector at the decoder side with a weighted average of the encoder hidden vectors. The weights can be calculated through a generalized matrix where the weight matrix is part of the parameters to be learnt. By adding attention to the deep seq2seq model, the authors were able to better align inputs and outputs. Another study [33] proposed DeepProbe which built on attention-based seq2seq RNN for query understanding, ad recommendation and user interaction. The authors first applied the seq2seq model in DeepProbe to understand and rewrite user questions into one of standard query form, which is submitted to an ordinary recommendation system. The rewrite is submitted to the recommendation system to retrieve a set of candidate answers. Secondly, they evaluated seq2seq model to score and pick better question-answer. The result showed that with attention mechanism and LSTM the model could rewrite questions with better quality measured by BLUE score. Finally, to make active user interactions, they built a prototype of a chatbot, which can ask questions that maximize information gain, allowing for a more efficient user intention identification process. At the end of their study, they evaluated by BLEU and human judge evaluation. Both demonstrate significant improvements compared with current state-of-the-art systems. However, it is ongoing research and required more improvement and experiment. This study would like to take this opportunity and enhance their work with a helpful experiment which can measure better efficiency for a user to acquire the information they need.

Another study (Guo, H. 2015) introduced a novel schema for seq2seq learning with a Deep Q-Network (DQN) which decodes the output sequence iteratively. In each iteration, an encoder-decoder LSTM employed to automatically generate informative features to represent the internal states and formulate a list for DQN. The list contains words probabilities at each time step and DQN learns to make decision on which action will be selected from the list to modify the current sequence.

For evaluation, a straight forward and effective method for decoding search as suggested by (Sutskever, Vinyals, Le, 2014) to deploy beam search algorithm. At each timestep, the decoder extends each partial sentence in the beam search with every possible word in the vocabulary. Finally, the model was trained to decode 10,000 natural sentences. Their experiments indicated that when compared to a beam search LSTM decoder, the proposed method performed competitively well. When decoding sentences from the training set, it significantly outperformed, in terms of BLUE score obtained. It was calculated based on the closeness between the target sentence and the decoded output sentence after DQN takes an action. The authors computed the similarity of this sentence pair using the popular score metric in statistical translation.

V. CHATBOT'S LIMITAITONS

There are several studies found that are trying to develop ideal application of a chatbot, that can have a natural conversation and indistinguishable from humans. But it is far from to achieve. From the summary of the literature, the following drawbacks exist for providing with efficient and effective chatbot conversation.

i. Fixed Rule-based: Previous chatbots are built with a fixed set of rules, template-based matching and very simple machine learning approach.

ii. Grammatical Errors: It does not recognize grammatical errors.

iii. Predefined or Closed-domain: Most of them are able to answer questions only on closed-domain or based on predefined in the database.

iv. Ambiguity: The context and meaning of the sentences are unclear or not appropriate meaning with the word.

v. Language Structure: Each language has a different sentence making structure. For example, structure of texts, punctuation and use of spaces differ between languages. Existing chatbots are not able to differentiate it.

vi. Semantics: Semantics is the meaning of sentences or words in the form of human natural language. Previous chatbots are not coped with natural language processing whether it is for creating a response or analysing questions.

vii. Sentiment Analysis: The existing chatbots are not able to detect the emotion of the subject human is talking about. The chatbot should be able to tell from the way the text or speech pattern is presented whether the human is angry, sad or happy.

viii. Recommender System: Existing chatbots do not ask questions, explain or advice on the user topic. They just collect information and provide responses from knowledge-base. The chatbot should able to write questions based on previous answers.

ix. Accuracy: Chatbots are programmed have a conversation like human to accomplish a task. However, existing chatbots have bad tendency to suddenly change the subject and generate unpredictable responses. Sometimes it responds without context. Thus, the accuracy does not achieve at satisfactory level.

x. Self-learning: The previous chatbots do not use supervised machine learning approaches. They are incapable of learning new patterns of speech or words, discover context through interaction and logical reasoning. Most of them are not able to train a classifier to map from sentences to intents and sequence model to slot filters.

xi. Support Third-party Integration: Existing chatbots do not support third-party Integration such as knowledge-based and is not multilingual. Most of them support English language only. It is also difficult to embed in a web page due to the critical process flow for web integration.

xii. Data Processing: They do not process structure data directly and there are no relational databases. Besides that, preparation of datasets are complicated, the utterances and entities mapping is critical.

xiii. User Interface: Existing chatbots do not have an interactive user interface and maintain poor documentation.

A newer chatbot is required with deep learning capabilities to overcome the above mention limitations. It will not only analyse human input but also generate appropriate responses. if chatbots are well “trained”, it can recognize the human natural languages and can react accordingly to any situation. However, the big disadvantage is that these natural responses require a great amount of learning time and data to be able to learn the vast amount of possible inputs. The training will prove if the AI chatbot able to handle the more challenging issues that are normally obstacles for simpler chatbots.

VI. COMPARISON

Chatbot	Technical Specification		Drawback
	Input/output	Technique	
Eliza [9]	Basic Pattern matching with templates to generate a response	Template-based	No logical reasoning capabilities, inappropriate responses
Alice [11]	Pattern matching to represent input and output	Recursive techniques	Grammatical analysis to structure sentences
Elizabeth [14]	Command line script as Input rules, and output transformation rules to generate responses.	Iterative	Does not split input and combine the result
Mitsuku [16]	AIML Category to route input from the user	NLP with heuristic patterns, supervised ML	Failed to provide dialogue components
Cleverbot [19]	Matches keywords for input and response based on previous chat	Rule-based	Unpredictable responses without context
Chatfuel [21]	Map input sentences to output	Rule-based	Inflexible conversation flows
Chat Script [22]	Pattern matching	Script-based	Difficult to learn and embed in a web page
Watson [23]	Identify feature values to generate responses based on the score	Rule-based NLP, UIMA	Does not process structure data. No relational databases
LUIS [24]	Identify valuable info. from user conversation	NLU with the prebuilt domain, Active learning	Required Azure subscription
Dialog flow [25]	Match input to specific intents and uses entities to extract	NLP, ML	No interactive UI and does not support handheld devices
Amazon Lex [26]	Matches keywords for input and response	NLU, AWS Lambda	Not multilingual, mapping utterances & entities are very difficult

VII. CONCLUSION

The insurance industry has long been bogged down by outdated practices. However, the combination of a new wave of thinking and newly developed artificial intelligence technology has the potential to completely change the customer experience to provide great service in a way that resonates with modern customers. This study presents sequential attention mechanism in deep recurrent neural networks, an architecture for the development of AI chatbot system with self-learning capabilities. The main aim is to fill in a gap in this research area and providing a flexible chat interface for question answering.

REFERENCES

[1] Morgan, B., How Artificial Intelligence will Impact the Insurance Industry in Forbes. July 25, 2017.

[2] Siddiqui, M. and T. Ghosh Sharma, Analyzing customer satisfaction with service quality in life insurance services. Vol. 18. 2010.

[3] MarketTools. Measuring and Improving Customer Satisfaction in the Insurance Industry. MarketTools 2018 Retrival on 4/5/2018]; Available from: http://www.customerthink.com/files2/MarketTools%20CustomerSat_Insurance%20Industry%20Solution%20Brief.pdf.

[4] Aksu, H., Customer Service: The New Proactive Marketing, in huffingtonpost.com, https://www.huffingtonpost.com/hulya-aksu/customer-service-the-new-_b_2827889.html. 2013.

[5] Abbate, M.L., U. Thiel, and T. Kamps. Can proactive behavior turn chatterbots into conversational agents? in IEEE/WIC/ACM International Conference on Intelligent Agent Technology. 2005.

[6] Accenture, Accenture Interactive: Chatbots in Customers Service 2017.

[7] Barker, S., How chatbots help. MHD Supply Chain Solutions, 2017. 47(3): p. 30.

[8] Chen, H., et al., A Survey on Dialogue Systems: Recent Advances and New Frontiers. ACM SIGKDD Explorations Newsletter, 2017. 19(2): p. 25-35.

[9] Weizenbaum, J., ELIZA: a computer program for the study of natural language communication between man and machine. Commun. ACM, 1966. 9(1): p. 36-45.

[10] Epstein, J. and W.D. Klinkenberg, From Eliza to Internet: a brief history of computerized assessment. Computers in Human Behavior, 2001. 17(3): p. 295-314.

[11] Wallace, R.S., The Anatomy of A.L.I.C.E, in Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer, R. Epstein, G. Roberts, and G. Beber, Editors. 2009, Springer Netherlands: Dordrecht. p. 181-210.

[12] Jurafsky, D. and J.H. Martin, Speech and Language Processing (2nd Edition). 2017: Prentice-Hall, Inc. ch. 28, pp. 418-440.

[13] Lemaitre, C., C. A. Reyes, and J. Gonzalez. Advances in Artificial Intelligence - IBERAMIA 2004. in 9th Ibero-American Conference on AI, Puebla, November 22-26. 2004. México.

[14] Weizenbaum, J., A response to Donald Michie. International Journal of Man-Machine Studies, 1977. 9(4): p. 503-505.

[15] Shawar, B. and E. Atwell, A comparison between Alice and Elizabeth chatbot systems. 2002.

[16] Worswick, S. Mitsuku Chatbot : Mitsuku now available to talk on Kik messenger. 2010 Retrieval on 04/05/2018]; Available from: https://www.pandorabots.com/mitsuku/.

[17] Higashinaka, R., et al. Towards an open-domain conversational system fully based on natural language processing. in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. 2014.

[18] Vinyals, O. and Q. Le, A Neural Conversational Model. 2015.

[19] Carpenter, R. Cleverbot 1997 13 November 2011.

[20] Hill, J., W. Ford, and I. Farreras, Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. Vol. 49. 2015.

[21] Dumik, D. Chatfuel. 2015 23/04/2018]; Available from: https://everipedia.org/wiki/chatfuel/.

[22] 22. Wilcox, B. Chatbots fail to convince judges that they're human. 2011 23/04/2018].

[23] Nay, C., Knowing what it knows: selected nuances of Watson's strategy, in IBM Research News 2011, IBM.

[24] Microsoft. Microsoft Cognitive Services: LUIS. 2015 [cited 24/04/2018; Available from: https://www.luis.ai/home.

[25] Google. Dialogflow 2010 23/04/2018]; Available from: https://dialogflow.com/.

[26] Amazon Web Services, I. Amazon Lex – Build Conversation Bots. 2017 23/04/2018]; Available from: https://docs.aws.amazon.com/lex/latest/dg/what-is.html.

[27] Nimavat, K. and T. Champaneria, Chatbots: An overview. Types, Architecture, Tools and Future Possibilities. 2017.

[28] Stroh, E. and P. Mathur, Question answering using deep learning. 2016.

[29] Pennington, J., R. Socher, and C. Manning, Glove: Global Vectors for Word Representation. Vol. 14. 2014. 1532-1543.

[30] Bahdanau, D., K. Cho, and Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate. 2014.

[31] Sutskever, I., O. Vinyals, and Q.V. Le, Sequence to sequence learning with neural networks, in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. 2014, MIT Press: Montreal, Canada. p. 3104-3112.

[32] Shao, Y., et al., Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models. 2017. 2210-2219.

[33] Yin, Z., K.-h. Chang, and R. Zhang, DeepProbe: Information Directed Sequence Understanding and Chatbot Design via Recurrent Neural Networks, in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017, ACM: Halifax, NS, Canada. p. 2131-2139.

[34] Kalchbrenner, N. and P. Blunsom, Recurrent continuous translation models. Vol. 3. 2013. 1700-1709.

[35] Cho, K., et al., Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 2014.

[36] Li, X., et al., End-to-End Task-Completion Neural Dialogue Systems. 2017.

[37] Hakkani-Tür, D.Z., et al. Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM. in INTERSPEECH. 2016.

[38] Manning, C.D., et al. The Stanford CoreNLP Natural Language Processing Toolkit. in ACL. 2014.

[39] Li, X., et al., A User Simulator for Task-Completion Dialogues. CoRR, 2016. abs/1612.05688.

[40] Wang, H., et al., A dataset for research on short-text conversation. 2013. 935-945.

[41] Tran, V.-K. and L.-M. Nguyen. Semantic Refinement GRU-Based Neural Language Generation for Spoken Dialogue Systems. 2018. Singapore: Springer Singapore.