International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 02 | Feb 2020 www.irjet.net p-ISSN: 2395-0072

A Review on Chatbot Design and Implementation Techniques

Ramakrishna Kumar[1], Maha Mahmoud Ali[2]

[1] *Deputy Head of Department of Electrical and Communication Engineering, College of Engineering, National University of Science and Technology, Muscat, Oman*
[2] *Student, College of Engineering, National University of Science and Technology, Muscat, Oman*

Abstract - *In recent times, the design and implementation of chatbots have received great attention from developers and researchers. Chatbots are Artificial Intelligence (AI) based conversational systems which are able to process human language through various techniques including Natural Language Processing (NLP) and Neural Network (NN). The main goal of this review is to summarize some of the most efficient implementation techniques that have been carried out in previous years. This paper is not only analyzing critically the previous works on chatbots but also suggests a proposed methodology in order to develop a state-of-the-art chatbot application that can be personalized easily according to customer needs. The proposed chatbot can be implemented using a couple of tools such as DialogFlow, TensorFlow, Android Studio and Firebase. The proposed Chatbot will be implemented using a couple of tools such as DialogFlow, TensorFlow, Android Studio, and followed by Machine Learning (ML) and Deep Learning (DL) techniques including Neural Machine Translation (NMT) and Deep Reinforcement Learning (RL) models.*

Key Words: Chatbot, Artificial Intelligence (AI), Natural Language Processing (NLP), Neural Network (NN), Machine Learning (ML), Deep Learning (DL)

1. INTRODUCTION

Artificial Intelligence (AI) is the science of making intelligent machines that are able to learn rules for using information, in order to reach approximate conclusions. It is categorized into two types:

- **Weak AI**, which includes systems that are designed and trained for a specific task, like Google Assistant.
- **Strong AI**, which includes systems that are intelligent enough to figure out a solution without human intervention, this type of AI is able to generalize human cognitive abilities, hence they are familiar with any type of tasks. [13]

Nowadays, there are many types of technologies incorporated with AI, such as automation, Machine Learning (ML), Natural Language Processing (NLP), machine vision, expert systems and robotics. Moreover, AI has played a great role in many life aspects including healthcare, education, business, finance, manufacturing and law. [13]

In fact, AI is a broad term that encompasses many subfields including Machine Learning (ML) and Deep Learning (DL)". Accordingly, ML is a subset of AI, and it includes the further advanced models and techniques that allow the machines to analyze the data and find rules to be followed, in order to develop AI applications. Thereupon, DL is included in the majority of AI applications as it is the newer field of ML that takes advantage of multi-layered artificial neural networks. The main purpose of using DL in AI applications is to achieve higher accuracy in some tasks such as speech recognition, object detection and language translation. In addition, DL is mostly used because of its ability to translate, extract or learn features automatically from huge data sets. The figure (Fig1.3) below shows the main difference between ML and DL. [6]

Chatbots are intelligent conversational systems that are able to process human language. A Chatbot can process the user input using the NLP tool, and then associate the input with intent, in order to produce an output. [16] There are two types of Chatbots, which are:

- **Rule-based Chatbots**: They are programmed to reply to specific questions that are predefined at the beginning. In this type of Chatbots, users are restricted to limited input options.
- **AI Chatbots**: They are programmed to interact with users as a real human, and they have the ability to keep track of context and word dictionary. In addition, this type of Chatbots requires many logic implementations. Moreover, they can be classified into three different categories, which are deep learning Chatbots, end-to-end systems and sequence-to-sequence models. [8]

Finally, the use of technology is expanding widely in everyday life and changing the way of providing services in many sectors. Consequently, Chatbots can be used in the education sector as a virtual assistant for students to clarify their doubts and make their life easier. [15]

2. REVIEW OF CHATBOT DESIGN AND IMPLEMENTATION

A number of selected studies that are achieved in the past five years are reviewed and explained below, in order to enhance the development of chatbots. The aim, methodologies, strengths and results of the papers are clearly mentioned and analyzed. Followed by other essential parameters including the limitations to be overcome, as well as the scope of further investigation to be considered.

Neural machine translation (NMT) is a technique for machine translation, which uses neural network models for learning a statistical model for machine translation. NMT model is based on Sequence-to-Sequence **(Seq2Seq) model** with encoder-decoder architecture. [2]

Bahdanau et al. (2015) [1] have carried out a research that aimed to develop **a Neural Machine Translation (NMT)** (English-to-French) by building a single neural network that is jointly learning to align and translate in order to maximize the translation performance. This NMT can be trained directly on the source text as well as the target text, whereas the previous approaches, such as the statistical machine translation and basic encoder-decoder approach, required careful setting and checking of each module in the pipeline of translation.[1] However there are various machine translations belong to a family of encoder-decoders, this machine translation differs from them because **it encodes the input sentence into a sequence of vectors** and then selects a subset of these vectors during decoding translation, whereas the input sentence is encoded into a single fixed-length vector in the other previous machine translation. The proposed model in this paper identifies a linguistically reasonable soft alignment between the input sentence and the corresponding output sentence. [1]

Bahdanau et al. (2015) [1] have introduced an enhancement to the basic encoder-decoder model where the input sentence is encoded into a sequence of vectors and then a subset of these vectors is chosen adaptively during the decoding translation. In other words, **the Sequence-to-Sequence (Seq2Seq) model of the NMT consists of two Recurrent Neural Networks (RNNs)** "As shown in Fig1", which are:

- **Encoder**: encodes the source sentence into a sequence of vectors.
- **Decoder**: defining a probability over the translation and decodes the target sentence.

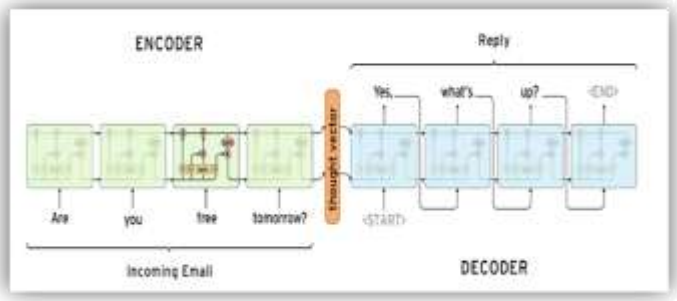

Fig. 1: The Sequence-to-Sequence Model.

For performance optimization, Bahdanau et al. **applied a Neural Attention Mechanism** in the decoder that will assist the decoder to decide parts of the source sentence to pay attention. In addition, this mechanism will relieve the encoder from the necessity to encode all information in the source sentence into a fixed-length vector.

By using Bahdanau et al. approach [1], the proposed model will be able to cope better with long sentences compared with the previous approaches that used a single fixed-length vector. The proposed NMT model by Bahdanau et al., called RNNsearch, is based on a **bidirectional Recurrent Neural Network (BiRNN)** which consists of a forward and a backward RNN. BiRNN is preferred than the usual RNN because the annotation of each word in BiRNN includes the summaries of both the following and preceding words, where the annotation of each word in RNN contains the summaries of only the preceding words. Equally important, **two types of models are trained to generate translations, which are:**

- **RNN Encoder-Decoder (RNNencdec) with Long Short-Term Memory (LSTM),** to build a novel architecture that learns to translate and align jointly.
- **RNNsearch** (the proposed model by Bahdanau et al.).

As a result, s high translation performance system is achieved on the task of English- French translation comparable to previous basic encoder-decoder approach and existing state-of-the-art phrase-based. In addition, the proposed model finds a linguistically plausible soft alignment between the input sentence and the corresponding output sentence. According to the quantitative results, the proposed model **(RNNsearch)** surpass the conventional RNNencdec and achieved as high as the phrase-based translation system. Furthermore, RNNsearch-50 showed a great performance even with sentences of length 50 or more, which means that the

proposed model RNNsearch even, surpass the performance of RNNencdec-50. Moreover, according to the qualitative results, RNNsearch surpasses the performance of RNNencdec model at translating long sentences and dealing with target and source phrases of different lengths. [1]

The Bahdanau et al. approach [1] **has many strengthens** including providing qualitative and quantitative results in the paper, providing the model architecture and the training procedure as appendixes, the ability to cope with long sentences, using BiRNN model to enhance the annotation of each word by enabling it to contain the summaries of both the following and preceding words.

On the other hand, the approach has some limitations since the proposed NMT model is not able to handle unknown and rare words since only sentences consisting of known words are considered during translation. In addition, NMT tends to produce a short-sighted output that ignores the directionality of future conversations. [1] The two main weakness of this NMT that disappointed the authors are:

- The generic responses (e.g. Ok, I do not know).
- The inconsistent responses (e.g. asking the same question twice and have different answers).

This NMT model can be enhanced by using architectures such as a hybrid of an RNN or a de-convolutional neural network in order to improve the performance of the RNNencdec model. Furthermore, the problem of rare word translation can be addressed by using the NMT, which can be trained on the data that is augmented by the output of a word alignment algorithm and enable the system to emit a pointer, for each of out-of-vocabulary (OOV) word, to its matching word in the source sentence and then translate the OOV words using dictionary in a post-processing step. [1]

Research has been carried out by **Li et al.** [9]**,** which is built on top of a bunch of existing ideas for building neural conversational agents including Bahdanau et al. approach [1] to control against generic and inconsistent responses problem, which is faced in Bahdanau et al. NMT approach. Li et al. model is a Seq2Seq model + attention, but with the **Maximum-likelihood estimation (MLE loss)** objective function. It is first trained with the usual MLE loss and then fine-tuned with **policy gradients** to be optimized for specific conversational properties. The proposed model simulates **two virtual agents** that can be rewarded with policy gradient methods to get a good sequence. In order to improve Bahdanau et al. NMT approach, Li et al. introduced a new model, called **Neural Reinforcement Learning (RL)**, which allows developers to set long term rewards. In order to realize these, with seq2seq as a backbone, two virtual agents are working to maximize the possible reward while searching for possible replies.

The proposed RL model consists of two agents, assuming p is the statement generated from the first agent, and q is the one from the second agent (As shown in Fig. 2). These Agents will talk to each other in turn, so the whole conversation can be expressed as a sequence of sentences generated from two agents as p1, q1, p2, q2, ... Moreover, It can be viewed that the action taken by agents along with the policy defined by seq2seq model has generated a sequence of sentences. In (Li et al.) model, seq2seq parameters are optimized in order to maximize future rewards using the policy search. [9]

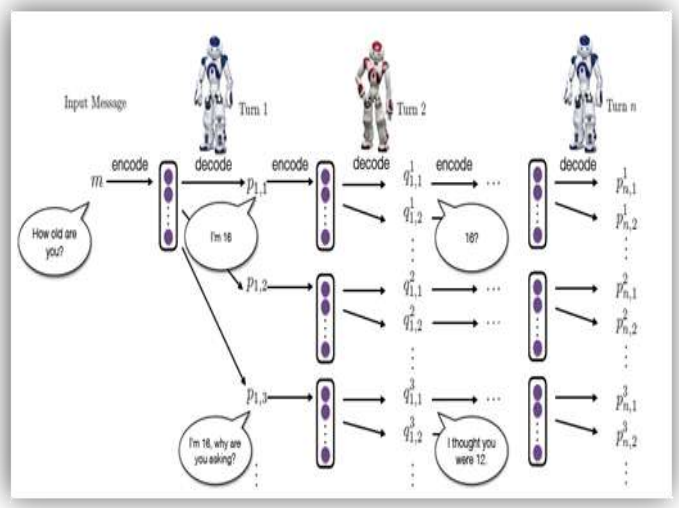

Fig. 2: Dialogue simulation between the two agents.

The components of (Li et al.) sequential decision problem are Action, State, Policy and Reward. Specifically, **three types of rewards** are defined, which are:

- **Ease of Answering**: The probability of responding to a query with a list of the handpicked dull response. Lower probability results in a higher reward.
- **Information flow**: Consecutive responses from the same agent should have different information. Smaller cosine similarity results in a better flow.
- **Semantic coherence (Semantic Integrity)**: mutual information between source and target. For questions, the response must make sense with respect to the query. [9]

The usual supervised objective function is used to pre-train the proposed model (RL Model), by taking the last two utterances as a source. After that, two stages of policy

gradient learning are generated, the first stage is based on mutual information reward only, and the second one is a combination of the three. The policy network (Seq2Seq model) generates a probability distribution for each response when given a state (preceding utterance). In order to estimate the gradient in each iteration, the network is frozen to generate multiple responses from the model. Then, the rewards of each response are then averaged and gradients are calculated using MLE function from the first L tokens that appear in the response. Then finally, the remaining T-L tokens are calculated using policy gradients. At that time, L gradually approaches zero. As a result, the evaluation of the result will be based on the duration (length) of the conversation, the number of unique unigrams and bigrams that appeared (The Diversity), and the human evaluation. [9]

As a result, high performance is achieved by the RL proposed model of (Li et al.) in comparison with (Bahdanau et al.) [1] NMT model. As a result, more interactive responses than other previous baselines are generated The RL based model. The definition of Li et al. good conversation is to consider the future (forward-looking), interactive and coherent. Moreover, The RL model achieved a great performance in handling a conversation over to the user by having a tendency to end the response with another question. Finally, according to the qualitative analysis conducted by (Li et al.), it is obvious that the RL model is able to generate more sustained and interactive conversations that foster a more sustained conversation. This paper can be considered as the preliminary step towards introducing a neural conversational model based on the long-term success of dialogues.

The (Li et al.) proposed model [9] has many strengthens like avoiding generic responses and the ease of responding compared to previous approaches. In addition, using policy gradients to fine-tune a pre-trained network, and then use these gradients to encourage the two virtual agents to interact with each other and explore the complete space of responses.

On the other hand, the model has some limitations including the hardness in the evaluation of conversational agents since metrics such as perplexity and BLEU do not necessarily reward desirable conversational properties. Hence, they are intentionally avoided. [9]

A new **Answer Sentence Selection (ASS)** approach was introduced by **Yu et al.** [18]**,** which is developed based on by applying distributional sentence models, in order to match questions with answers via considering the encoding semantic. Whenever a question is given to ASS, it chooses the correct sentence from a set of candidate sentences. By using this approach, the problem of previous models such as feature-based semantic models is addressed, which was the struggle of adapting to new domains. Unlike previous approaches, the proposed model by (Yu et al.) does not need extensive human-annotated external resources or feature engineering, which makes it more flexible and simpler.

Generally, in the question-answering process, answers are retrieved by converting the question into a database of queries and then apply this query subsequently to the current knowledge base. Likewise, this ASS model (approach) projects Questions and Answers (QA) into vectors and then learn a semantic matching function between QA pairs to combine this function with a simple subsequently. In other words, it chooses a sentence that included the information needed for answering the given question from a group of candidates that derived by using an information extraction system. **ASS uses two models** to project sentences into vector space representation, which are **Bag-of-words model** and **Biagram model**. Moreover, the proposed model uses only two non-distributional features, word matching weighted by IDF values and question-answer pair word matching, which helps to make the approach simpler than previous ones. By using the proposed model, ASS is able to capture complex semantics of a sentence compared with the previous approaches, since the Biagram model is based on a **Convolutional Neural Network (CNN).** CNN-based models have been shown its effectiveness in some applications such as twitter sentiment prediction, semantic role labelling, and semantic parsing. Figure 3 illustrates the architecture of the CNN-based sentence model in one dimension. The bigrams used by Yu et al. in this approach are with one convolutional layer, and Fig. 4 shows the architecture of the CNN-based sentence model in one dimension. Using this composition model was a great choice because it made the proposed model sensitive to word order. Equally important, this composition model enabled the proposed approach to learn the internal syntactic structure of sentences; therefore, it is capable of capturing long-range dependencies. A number of experiments were conducted on the ASS dataset, which is created **from the TREC QA track**, and the results show the effectiveness of the proposed model that matching the state-of-the-art results. [18]

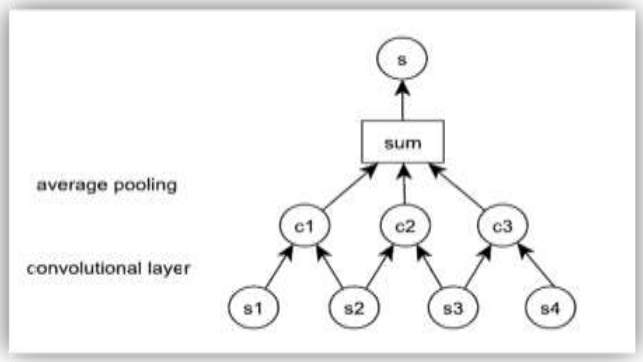

Fig. 3: The architecture of a one-dimensional convolutional neural network.

As a result, the ASS proposed model achieved good performance and matched the state of the art performance on the task of selecting the answer sentence. Moreover, it emphasizes that **a Neural Network-based sentence model** can be applied to the task of **Answer Sentence Selection**. In Addition, this model works effectively on the dataset, which is created from the TREC QA track. The proposed model is trained on both **TRAN-ALL and TRAIN datasets** and compared with three baseline models. The result of the comparison shows valuable evidence of the effectiveness of the model since the proposed model surpassed all the baseline models. Moreover, the ASS model can be applied across any language directly, since it does not depend on any external resources. [18]

The (Yu et al.) The approach has many strengthens since it has the ability to select answers, for a given question, among a group of candidates that are not encountered during training, unlike the previous approaches that were not able to do the answer sentence selection process, unless the candidates are encountered during the training. In addition, the model is equipped effectively to deal with proper nouns and cardinal numbers, since it can count the number of co-occurring words in the question-answer pairs. Equally important, the proposed model is simple compared with previous approaches that have a heavy reliance on resources. (Yu et al.) The approach can be used with any language and does not require any hand-coded resources. This ASS model is sometimes preferred among other models because it is sensitivity to word order and the ability to capture information form n-grams. [12]

On the other hand, **the ASS proposed model has some limitations** including is the disability to learn word embeddings from the dataset of the selected answer. In the same way, the existing words in dictionaries could not be able to cover all the words required for the dataset. In addition, the aim of the paper is not clearly mentioned. [18]

In order to improve this ASS model, an investigation of more complex models, like the convolutional network-based sentence model with **several feature maps** and **higher-order n-grams**, can be done. In addition, a **recursive neural network-based model** can be applied to enable the proposed model to work with textual entailment and phrase detection. [18]

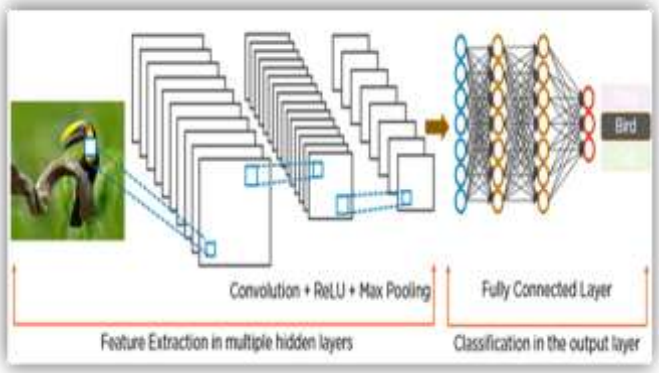

Fig .4: The architecture of CNN.

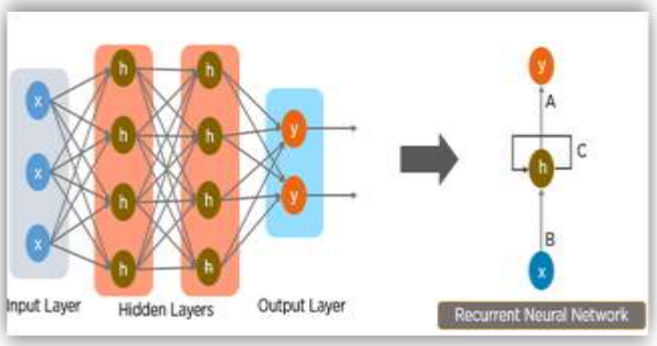

Fig. 5: The architecture of RNN.

The NMT (Bahdanau et al., 2015) approach [1], which is based on Seq2Seq model that uses **RNN** with bi-directional LSTM cells, and the **ASS (Yu et al., 2014)** [18] approach that uses **CNN-based sentence model**, are both based on **a Deep Learning Neural Network (DNN).** CNN is a feed-forward NN that is mainly used to classify objects, recognize images and use minimal amounts of preprocessing "See Fig. 4", whereas RNN is working on the principle of saving each output of a layer and then feed this back into the input while training it in a recurring loop "See Fig. 5". The table below illustrate the difference between RNN and CNN in details.

To conclude, with respect to **Question Answering System** field, RNN is mainly preferred when the system requires a text generation with respect to context, and CNN is preferred in time series data or capturing information form n-grams. CNN is suitable for data types that have a spatial relationship, like if there is an order relationship between words in a document. [3]

Table. 1: A comparison between RNN and CNN.

RNN	CNN
The conversation is a sequence of words (Can handle arbitrary input and output lengths)	The conversation is a fixed size (Cannot handle sequential data)
Considers the previously received inputs along with the current input. The LSTM cells used in RNN model allow RNN to memorize previous inputs.	Considers only the current input, and it cannot remember the previous input.
Uses time-series information, hence it is the best suitable model for systems that take the conversation context in its consideration.	Uses connectivity pattern between its neurons, hence the neurons are arranged in such a way that enables CNN to respond to overlapping regions tiling the visual field.
Used to create a combination of subcomponents (e. g. text generation, language translation)	Used to break a component (e. g. image) into subcomponents (e. g. object in an image)
It is ideal for text and speech generation.	It is ideal for images, videos processing and ranking candidate sentences.

An open domain Chatbot engine was developed by **Qiu et al. (2017)** [11]**,** based on the attentive Seq2Seq model introduced by **(Bahdanau et al.)** [1]**.** The aim of this paper is to enable bots to answer customer questions in the E-commerce industry in order to offer a better user experience. The proposed system by Qiu et al. integrates the joint results of **Information Retrieval (IR)** and **Seq2Seq based generation models**, and then uses a **Rerank model** to optimize the joint results. This system differs from other previous works by using the IR model to rerank the union of retrieved and generated answers. Other systems often fail to handle long-tail questions and generate inconsistent or meaningless answers. [11]

The hybrid approach "As shown in Fig. 6" introduced by Qiu et al. consists of **four main components**: a QA knowledge Base, an IR model, a Generation model, and a Rerank model. The Generation and Rerank models are both built on the same attentive Seq2Seq model, and the QA knowledge Base is constructed from the chat log for six months of the authors' online customer service centre. Firstly, the IR model is used to retrieve a set of QA pairs (candidate answers) from the **Knowledgebase** by using BM25 Algorithm to get the most likely pairs and then take the paired answer of the most similar one as the answer. Secondly, the candidate answers are sorted (reranked) using the attentive Seq2Seq Rerank model. If the top candidate has a higher score than a specific skill, it will be selected as the answer; otherwise, the Generation based model will offer the answer. In the Generation model, there are **three important implementations** employed using **Tensorflow** [17] library including **Bucketing and Padding** to handle different lengths of questions and answers, Softmax over sampled words to speed up the training process, and **Beam search** decoder to make the Generation more reasonable. [11]

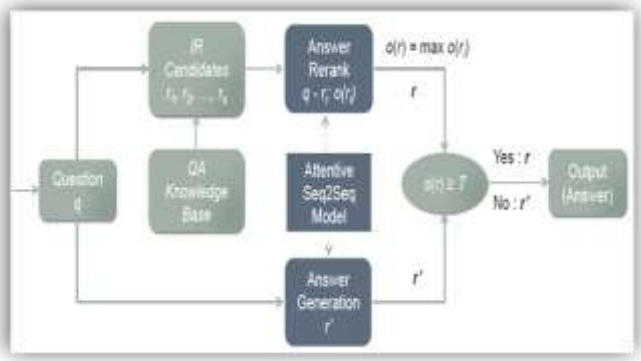

Fig. 6: Overview of the hybrid proposed approach.

As a result, the proposed hybrid approach outperforms both IR and Generation based models. Moreover, the Chat engine achieved a better performance in comparison with other public Chatbots. In addition, the results of a series of evaluations, which were conducted to assess the effectiveness of the proposed approach, shows outstanding performance when compared with both IR and Generation models. Finally, the authors launched **The AliMe Chat** for a real-world industrial application, and better results than other Chatbots were observed.

The AliMe Chat has special strengthens including the ability to handle long questions. This has been achieved by using the combination of the three models as discussed above. **On the other hand**, the system has some limitations in guaranteeing consistency and rationality.

This system can be enhanced by using other context-aware techniques like Context-Sensitive Model (Sordoni et al., 2015) and Neural Conversation Model (Sutskever et al.,

2014). In addition, the system can be empowered with characters and emotion to provide a better user experience.

One of the most effective Chatbots that are developed in the past three years is **SuperAgent Chatbot**, a customer service Chatbot for E-commerce websites, which is developed by **Cui et al.** [4]**.** SuperAgent is an add-on extension engine that provides the customers with the best answer among a huge existing data sources within a web page. This Chatbot is a great way to supplement customer service offerings since a Chatbot is more economical and indefatigable than the traditional customer service. Nowadays, customer service's support staff spend a lot of time in answering customers' questions, which can be cost-effectively answered by machines. The exhaustion felt by support staff, the wasted time in answering questions, and the difficulty in supporting 7x24 services contributed to the aggravation of this problem, hence Cui et al. developed this Chatbot. [4]

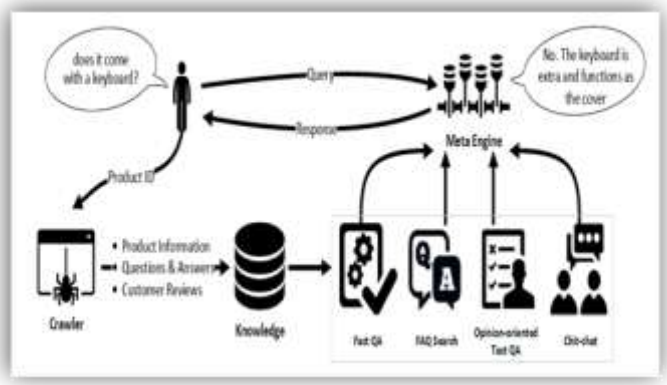

Fig. 7: The system overview of SuperAgent Chatbot.

The collected data comes from large-scale data available in e-commerce websites that are appropriate to feed into the chat engine. The chat engine is decomposed into **five sub-engines** "See Fig. 7":

1. **Fact Questions & Answers (QA**) engine for **Product Information (PI):**
   It is designed to answer questions regarding the facts of the products. This has been done using the **DSSM model**, a deep learning-based matching framework. It matches the input question with every attribute name, and then it selects the most matched attribute name that passes a predefined threshold. After that, this attribute is used to generate the response sentence based on a set of predefined templates. [4]

2. **FAQ search engine for QA:**
   It is designed to answer frequently asked questions by finding the most similar question in QA pairs from given a set of QA pairs and a customer's question, then it returns the corresponding answer as the reply. This has been achieved by training a Regression Forest model and using some features like DSSM Model, GloVe, and word mover's distance. [4]

3. **Opinion-oriented text QA for Customer Reviews (CR):**
   It is designed to answer questions based on the customers' review. This has been done using the following approaches:
   - **A hybrid approach** [11] for extracting the aspects from review sentences.
   - **A sentiment classifier** (Tang et al., 2014) for determining the polarity of the sentence regarding the mentioned aspect.
   - **A Lucene toolkit** for indexing the aspects and polarity together with keywords and retrieving the top twenty candidate sentences, and a Regression model for ranking all candidate sentences based on a set of features designed at different levels of granularity. These features include a Translation model (Bahdanau et al., 2015) [1] and two CNN-based models. [18]

4. **Chit Chat Engine for replying to greetings:**
   It is designed to reply to queries that cannot be answered by the previous three engines and to make the conversation as smooth as possible. This model is based on the NMT model (Bahdanau et al., 2015) [1], which is trained on twitter conversation data. Usually, chit chat engines tend to be topic deviated, but Cui et al. avoided this by using a smart reply approach for e-mail reply suggestions (Kannan et al., 2016) to predefine a set of permitted responses. [4]

5. **Meta Engine** for merge and prioritize the results from the different engines. [4]

As a result, when users visit a product page, SuperAgent crawls the information of HTML and scrape PI, QA and CR data from the webpage, and then process the sub engines in parallel. If a high confidence answer is found from the first three engines, SuperAgent returns this answer as a response. Otherwise, the reply will be generated from the Chit Chat engine. [4] In addition, the accuracy of the FAQ search task for the proposed model [4] surpasses the accuracy of the previous models including the 1st place team by a large margin; hence, the effectiveness of the proposed model is confirmed on the task of the FAQ search task. Moreover, Cui et al. achieved state-of-the-art results in opinion mining task and outperformed previous relevant approaches. In addition, the Chit Chat conversation model achieved a similar perplexity of Kannan et al.'s result and the Cui et al. engine's output proved to be very topic-coherent.

The Cui et al. (SuperAgent) Chatbot has many strengthens, which are:

- It can easily leverage large-scale data, as well as crowd-sourcing styles and publicly available e-commerce data.
- It has a set of state-of-the-art **Natural Language Processing (NLP)** and machine learning techniques, which are used in the Chabot's sub-engines.
- It does not need to deploy web crawlers for the websites since it is associated with each product webpage as an add-on extension. As a result, it directly improves the customer's online.

Chatbots are very useful in handling multiple customers and supporting 24/7 service. **On the other hand, the SuperAgent Chatbot has some limitations** since it could not replace humans in performing complex assignments, at least not yet.

The SuperAgent Chatbot [4] can be further enhanced by integrating a customer's query intent detection module, in such a way sub-engines can be better leveraged. Likewise, context modelling could be further investigated to support multi-turn queries.

Gregori [7] has reviewed a number of modern Chatbot platforms, NLP tools, and their application and design. Gregori's aim is to develop a Chatbot for Online Masters of Computer Science **(OMSCS)** program in The Georgia Institute of Technology. Different aspects of using Chatbots are discussed in the paper including customer service and education. In education, Gregori has mentioned some examples of Chatbots such as ANTswers, a librarian Chatbot at University of California based on A.L.I.C.E open-source framework, AdmitHub "Pounce", a custom virtual assistant for Georgia State University (GSU) admissions that has a knowledge base of about thousand FAQ. Pounce Chatbot has achieved great results and proved to be very successful to handle student queries. [7]

Generally, a **Chatbot consists of four main components**:

1. **Front-end**, which uses NLP and AI to determine the user's intent and provides a reply to the user's intent. To sum up, it is responsible for communicating with the user.
2. **Back-end**, which is responsible for creating the knowledge base and consuming the domain's corpus.
3. **Knowledgebase**, which represents the knowledge of the Chatbot in a format that is consumable by the front-end. In addition, the domain corpus is classified and tagged in the knowledge base.
4. **Corpus**, which represents the domain. It can be structured or unstructured. The OMSCS's corpus is based on FAQ's. [7]

There are various Chatbot platforms "as shown in table. 2", which are used to develop Chatbots. In this paper, Gregori focused on **four platforms (NLU tools), Wit.ai, API.ai** "AKA DialogFlow"**, LUIS, and Amazon Lex,** and compared the test results for all of them, in order to come up with the best platform among them to be used for developing the proposed Chatbot. Each tool of them was trained with the same utterances and then passed with the same test questions. For testing**, (Pythonanywhere.com)** is used, which is a test tool's execution engine. Table2.3 shows the test results of each platform (tool). The integers in the table indicate the confidence percentage of an intent. As can be seen, Amazon Lex platform does not provide a confidence score; this is because (Pythonanywhere.com) does not have the Amazon Lex server URL whitelisted. Hence, The Amazon Lex editor 'Test Bot' tool is used instead of (Pythonanywhere.com) in testing Amazon Lex platform. [7]

Table. 2: Test results of the platforms.

Test Question	WIT	LUIS	API	LEX
How many hours are required to graduate?	84	21	68	NA
How many classes are required to graduate?	56	16	40	NA
When can I apply?	99	100	100	NA
Where can I apply?	99	100	100	NA
What is the application process?		6		NA
What is the admission process?		5		NA
How does the admission process work?		7		NA
Is the GRE required?	99	100	100	NA
What is the minimum GRE score required?	99	100	84	NA
What are foundation courses?	93	100	74	NA
How many foundation courses are required?	55	57	41	NA
What are the admission criteria?	67	99	49	NA
where do I find the admission criteria?		87	35	NA

The result shows that Wit.ai provided an incorrect response to the question, "What are the OMSCS admission requirements?" because the question is categorized as "value" as opposed to "education". It has one more limitation in matching the input text to an intent since its response does not include an intent field when it could not match the input text to an intent. For **LUIS** tool, the result shows that it

has also provided an incorrect categorization for the same question because of the same failure in categorization. Then when comes to **Amazon Lex**, it also provided an incorrect answer, because it did the same mistake in intent categorization. Finally, when **API.ai** is tested, the result was satisfactory compared with the other three tools. It returned the correct intent "admission" with a high confidence score; this is because API.ai supports assigning responses to intents. In addition, API.ai can provide a default to be used to the domain questions. Under those circumstances, API.ai was chosen as the NLU platform for developing the proposed Chatbot. [7]

3. PROPOSED METHODOLOGY

The following techniques can be combined together and used, in order to develop a state-of-the-art Chatbot application:

- A Neural Machine Translation (NMT) model, followed a Bidirectional Recurrent Neural Network (BIRNN), and enhanced by a Neural Attention Mechanism, using Tensorflow.
- DialogFlow (API.ai) software tool for handling the natural language processing, Intent classification and response generation. In addition, it will be used to integrate the Chatbot with external APIs, which are a Facebook messenger and WhatsApp.
- A Neural Reinforcement Learning (RL) model, to avoid generic responses and allow the application to handle longer conversations, using Tensorflow. [17]
- A firebase real-time database, to store the data that will be fed into the bot as well as the student details.
- Android Studio, to integrate DialogFlow and firebase through it, and develop an application that can be installed freely on any android device.

In the beginning, DialogFlow will be used to handle the NLP, intent classification, training and text generation. Then, the text generation responses will be improved using TensorFlow software tool by integrating it with DialogFlow in fulfilment part. Then, a firebase real-time database will be used to create the required database. Further, Android studio will be used to develop the application and integrate it with the previously mentioned software tools. Moreover, DialogFlow will be used again to integrate the application with external APIs like Facebook Messenger and WhatsApp. Finally, the application could be accessed through two different ways, which are:

- An Android Application, for Android users.
- An external API like Facebook Messenger and WhatsApp, for iOS users.

4. CONCLUSION

In conclusion, this paper reviewed the design and implementation of Chatbot techniques. A number of previous relevant works are outlined properly, along with their strengthens and weaknesses. Furthermore, a proposed methodology for implementing a state-of-the-art chatbot has been suggested.

REFERENCES

[1] Bahdanau et al., 2015. Neural machine translation by jointly learning to align and translate. In ICLR 2015. San Diego, May 7-9, 2015. San Diego: Bahdanau. pp. 1-15.

[2] Brownlee J., 2017. A Gentle Introduction to Neural Machine Translation. [ONLINE]. Available from: https://machinelearningmastery.com/introduction-neural-machine-translation/. [Accessed: 7 May 2019].

[3] Brownlee J., 2018. When to Use MLP, CNN, and RNN Neural Networks. [ONLINE] Available from: https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/. [Accessed: 8 May 2019].

[4] Cui et al., 2017. The system overview of SuperAgent Chatbot. [ONLINE]. Available from https://www.aclweb.org/anthology/P17-4017. [Accessed: 9^th May 2019].

[5] Freeman L., 2016. Machine Learning - RNN vs CNN at a high level - Data Science Stack Exchange. [ONLINE]. Available from: https://datascience.stackexchange.com/questions/11619/rnn-vs-cnn-at-a-high-level. [Accessed: 8 May 2019].

[6] Gen O., 2019. Notes on Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL). [ONLINE]. Available from: https://towardsdatascience.com/notes-on-artificial-intelligence-ai-machine-learning-ml-and-deep-learning-dl-for-56e51a2071c2. [Accessed: 22 May 2019].

[7] Gregori, 2017. Evaluation of Modern Tools for an OMSCS Advisor Chatbot. In Summer 2017. Georgia, 2017. Georgia: Association for Computational Linguistics. pp. 1-7.

[8] Hubtype, 2018. Rule-Based vs AI Chatbots. [ONLINE]. Available from: https://www.hubtype.com/blog/rule-based-vs-ai-chatbots/. [Accessed: 22 May 2019].

[9] Li et al., 2016. Deep Reinforcement Learning for Dialogue Generation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Texas, November 1-5, 2016. Texas: Association for Computational Linguistics. pp. 1192-1202.

[10] Molnár & Szűts, 2018. The Role of Chatbots in Formal Education. In IEEE 16th International Symposium on Intelligent Systems and Informatics. Subotica, May 13-15, 2018. Subotica: IEEE. pp. 1-7.

[11] Qiu et al., 2017. AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, July 30 to August 4, 2017. Vancouver: Association for Computational Linguistics. pp. 1-6.

[12] Rebedea T., 2017. Intro to Deep Learning for Question Answering. [ONLINE]. Available from:

https://www.slideshare.net/TraianRebedea/intro-to-deep-learning-for-auestion-answering. [Accessed: 8^th May 2019].

[13] Rouse R., 2018. What is AI (artificial intelligence)? - Definition from WhatIs.com. [ONLINE]. Available from: https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence. [Accessed: 22 May 2019].

[14] Sharma A., 2018. What is the difference between CNN and RNN? - Quora. [ONLINE]. Available from: https://www.quora.com/What-is-the-difference-between-CNN-and-RNN. [Accessed: 8 May 2019].

[15] Singh R., 2018. AI and Chatbots in Education: What Does The Future Hold. [ONLINE]. Available from: https://chatbotsmagazine.com/ai-and-chatbots-in-education-what-does-the-futurehold-9772f5c13960. [Accessed: 22 May 2019].

[16] Techlabs M., 2017. What Are The Inner Workings of a Chatbot? – Chatbots Magazine. [ONLINE]. Available from: https://chatbotsmagazine.com/what-is-the-working-of-a-chatbot-e99e6996f51c. [Accessed: 22 May 2019].

[17] TensorFlow. 2019b. Tensorflow general architecture. [ONLINE]. Available from: https://www.tensorflow.org/guide/extend/architecture. [Accessed: 〖21〗^st May 2019].

[18] Yu et al., 2014. Deep Learning for Answer Sentence Selection. In NIPS Deep Learning and Representation Learning Workshop. Montreal, December 12, 2014. Montreal: Lei Yu. pp. 1-9..

BIOGRAPHIES

First Author – Ramakrishna Kumar, Deputy Head of Electrical and Communication Engineering Department, College of Engineering, National University of Science and Technology, Muscat, Oman

Second Author – Maha Mahmoud Ali, Student, College of Engineering, National University of Science and Technology, Muscat, Oman