

---

## A new deep learning-based chatbot system for the Customer service companies

Tahar Mehenni<sup>a,\*</sup>, Nadjemeddine Boudjellal<sup>b</sup>

<sup>a</sup>University Mohamed Boudiaf of M'sila, Computer Science Department, 28000, M'sila, Algeria

<sup>b</sup>University Mohamed Boudiaf of M'sila, Computer Science Department, 28000, M'sila, Algeria

*Submitted 16/01/2019, accepted 17/03/2019*

---

### Abstract

People want to communicate with technology in the same manner they communicate with other human beings, and the communication between brands and their clients has never been so intense as it is nowadays. With the rapid development of technology, the customer experience is changing dramatically. Customers want more autonomy and self-service options, preferring to make a purchase or get information without interacting with the human representative of the brand. Therefore, the use of chatbots in customer service can be a solution to the crucial issue of improving customer-brand communication. Companies are using this technology to create better engagement with their clients with the help of messaging platforms, to offer a regular chat function, in-message purchasing, and many other advanced functions. In our work, we have explored two different chatbot systems, the first bot is an open domain deep learning chatbot that has been trained on our personal computer, and the second one is a customer service chatbot that has been designed and trained in Google cloud platform.

Keywords: Deep learning; chatbot; customer service; Dialogflow; sequence-to-sequence; TensorFlow

---

### 1. Introduction

Artificial intelligence has been one of the goals of computer science since its inception. Alan Turing presented his Turing Test in 1950, which is designed to figure out the ability of a machine to interact just like a human agent would. Since then, conversational agents have been trying to pass that test, and in 2013 a super computer managed to convince the panel of judges from the Royal Society that it was a 13-year-old-boy,

---

\* Corresponding author. Tel.: +0 213 554 388 241.

E-mail address: tmehenni@gmail.com / tahar.mehenni@univ-msila.dz.

marking the first time in the competition history for an AI to pass the Turing test. That conversational agent, or chatbot, was specifically tailored for the competition, and nowadays, hundreds of similar bots are available online for people to chat with. The potential of chatbots goes beyond simple conversations, they can be used as personal assistants to schedule meetings, check the weather or even suggest an outfit for the day. They can also be used by businesses to offer customer support services.

Traditionally, customer support happened over the phone, then with the advent of internet technology, email was also utilized, but in both mediums, an actual human being is responsible for answering the client's request. The problem with that is no matter how many people are working on customer support, it is never enough, especially for major brands and small startups. Customers now want more self-service options, as waiting times for support tickets are too long even for small inquiries. In order to improve accessibility, companies are redesigning the experience from human-to-human interactions into self-service models, implementing chatbots, that serve as automated customer support agents available 24/7, in messaging platforms to integrate closer communication services the customers are using.

Chatbots have seen resurgence in research topics in recent years due to the progress made in deep learning techniques, which can be attributed to the massive increase in computational power, the availability of huge datasets, and the development of new and better deep learning models. These breakthroughs solved the main problems with deep learning as it required a lot of data to train the models, and the new super-fast compute units allowed for relatively fast iterations on those models, which allowed researches to produce new more efficient models, like Sequence-to-Sequence. Thus, chatbots and personal assistants becoming usable in the real world are recently widely used and actively developed by companies and amateurs alike.

The remaining of this paper is organized as follows. Section 2 presents the related work in chatbot domain. In the section 3, a description of the proposed chatbot system is given. Experimental results and discussion are presented in section 4 and finally, the paper is concluded in section 5.

## **2. Related Work**

Communication between brands and their clients has never been so intense as it is nowadays. With the rapid development of technology, the customer experience is changing dramatically. Customers want more autonomy and self-service options, preferring to make a purchase or get information without interacting with the human representative of the brand. In order to fit the expectations of their customers, companies are reshaping the experience from human-to-human interactions into the advanced self-service experience. Therefore, the use of chatbots in customer service can be a solution to the crucial issue of improving customer-brand communication. Companies are using this technology to create a better engagement with their clients with the help of messaging platforms to offer a regular chat function, in-message purchases, and many other advanced functions.

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the back propagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional

nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shed light on sequential data such as text and speech (Krizhevsky et al., 2012, Lee et al., 2014).

Speech or text interaction between a human and a computer is gaining more and more popularity nowadays. People want to communicate with computers in the same manner they communicate with other human beings. One of the main tools used for analyzing speech and providing human-like answers is Natural Language Processing (NLP). In order to provide suitable responses based on phrases or keywords taken from questions as well as to keep the communication continuous, like any other language processing program, chatbot architectures fall into two classes: rule-based systems and corpus-based systems (Krizhevsky et al., 2012). Rule-based systems include the early influential chatbots. Corpus-based systems mine large datasets of human-human conversations, which can be done by using information retrieval (IR-based systems simply copy a human's response from a previous conversation) or by using a machine translation paradigm such as neural network sequence-to-sequence systems, to learn to map from a user utterance to a system response.

The normal feed forward Neural Network will not be able to have an input with a varied length. This was the need for many applications such as text processing where the input of the network will vary from each sentence. Therefore, if we want to keep track of each word placement in the sentence to detect dependencies between cohesive words being input to the network at different times. Recurrent neural networks (RNN) solve this problem by implementing a recurrent connection from a neuron to itself, so the neuron will be able to have its previous state as an input (Graves et al., 2006). A visualization of a single recurrent cell can be seen in Figure 1.

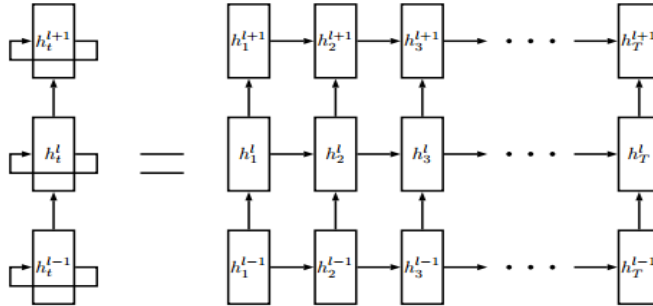


Fig. 1. A visualization of a single recurrent cell at time  $t$  to the left, and the same cell unrolled in time to the right. Note that the cell affects both the cells in the next layer and the cells in the same layer and next time step

For a standard recurrent neural network,  $h_t^l$  is updated according to equation 1, where layer  $l-1$  and  $l$  contains  $m$  and  $n$  neurons respectively.

$$h_t^l = \sigma(T_{m,n}(h_t^{l-1}) + T_{n,n}(h_{t-1}^l)) \quad (1)$$

Where the first part of the function contains the input from the previous layer  $l-1$  at time step  $t$ , and the second part contains the input from the current layer  $l$  at the previous time step  $t-1$  (Jargen and Graves, 2009, Graves, 2008).

There are many approaches and models having been made in dialogue systems, even our model, but using those models in a customer service dialogue system is new, since dialogue systems are doing heavy tasks, but

they still not trusted to make a final decision, that is why this area is active in research, also there is no proper evaluation method, that can evaluate the model, every evaluation should be made by user testing experience.

Vinyals and Quoc, 2015 published a paper describing their proposed model, they developed the neural conversation model (NCM), using sequence-to-sequence model. They had developed a chatbot only by training from one dialogue corpus, they also used Cleverbot (Carpenter, 2018) to evaluate their model by comparing the answers from their bot with the answers from Cleverbot. Using this evaluation scheme, they showed that their NCM performed better than Cleverbot.

Another dialogue modeling approach was proposed by Ameixa et al., 2014. Authors implemented a dialogue system to answer questions in one specific domain. Their method consists of splitting the dialogue corpus into utterance and response pairs. When they receive a question, they make a search in the dataset utterances and try to get the most similar one and simply output its corresponding response. They evaluated their model by giving the user the possibility to rate the bot answers if they were helpful or no.

A recent dialogue-based evaluation systems was proposed by Higashinaka et al., 2015. They developed an evaluation model based on what they called system breakdowns. The system breakdowns are the level in the conversation where the dialogue system made a bad mistake that can stop the conversation.

### 3. Description of the proposed chatbot

The concept of a sequence to sequence (*seq2seq*) model was originally proposed by Gehre et al., 2014 and Quoc et al., 2014. RNNs are very powerful machine learning models that have achieved very good results for problems like speech recognition and visual object recognition. However, there was another problem related to the sequence of words. As an example, let us have the following sentences: *the food was good, not bad at all* and, *the food was bad, not good at all*. We have the same words and length but the sequence matter because it could change the whole meaning of the sentence. For this reason, we have a new powerful technique that uses RNNs, called *sequence-to-sequence (seq2seq)*.

To create a sequence-to-sequence model we first have to create our recurrent neural network with  $L$  layers and then divid it into an encoder and a decoder. The encoder's task is to encapsulate the information of the input text into a fixed representation. The decoder's task is to take that representation and generate a variable length text that best responds to it (Adit, 2018). The basic model was extended to use multi-layer cells (specifically Long short-term memories (LSTMs)) (Bengio, 2014).

The input is a tokenized sentence with length  $T$ , which is converted into a string of one-hot vectors  $(x_1, x_2, \dots, x_T)$ . At each time step, a word  $x_t$  is embedded into a vector  $x'_t$ . The embedded word  $x'_t$  is then fed as the input to the encoder, which consists of a multi-layered recurrent neural network with LSTM-cells.

When all  $T$  inputs has been fed to the encoder, the network will be used for decoding. At each time step  $t > T$ , the network will output a word  $w_t$ . The network will be run until a special end-of-sentence symbol is produced. The input to the decoder  $w_t$  at time  $t > T$  consists of the last word  $w_{t-1}$  generated by the network, where the first decoder input is a special GO symbol. The predicted word  $w_t$  at decoding time  $t$  is calculated in equation 2, where  $V(i_w)$  is the  $i$ th word in the vocabulary and  $O_t$  is the output of probability. An example run of a sequence-to-sequence network can be seen in Figure 2. The string "*How are you?*" is fed one word at a time to the network, and it will generate words until it generates a special EOS token (Koo et al., 2014).

$$\begin{aligned} i_w &= \operatorname{argmax}(O_t) \\ w_t &= V(i_w) \end{aligned} \quad (2)$$

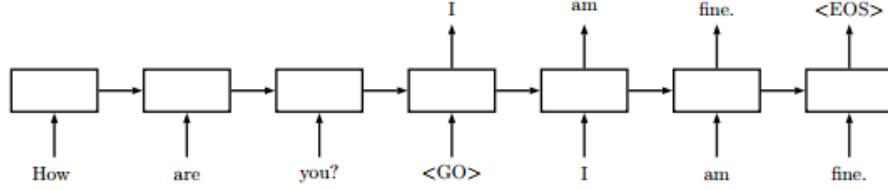


Fig. 2. A sequence-to-sequence model which encodes the sentence "How are you?" and produces during decoding the sentence "I am fine. EOS". When decoding, the previously generated output is used as input for the next time step, except for the first word, where GO is used as input. The decoder stops when an EOS is generated.

In our work, we have explored two different chatbot systems, because we faced several problems during research and design, the first bot is an open domain deep learning chatbot that has been trained on our personal computer, and the second one is a customer service chatbot that we designed and trained in Google cloud platform.

We created a first model and trained it with an open domain dataset. The training phase took about 3 weeks of our personal computer crunching numbers non-stop, and the results were encouraging.

In the second chatbot, we use Google cloud platform, where models can be implemented and trained on its powerful cloud. It helps by getting around most of the common challenges (for example: datasets, preprocessing, training time, etc), as well as providing the option of adding a personality to the bot by giving it the ability to respond to personal questions such as the bot's name and other predefined information. Also with the power of Google machine server the bot will be trained perfectly to be able to generate a sensible response, it also allows the bot to connect to other services via webhooks which are cloud functions that can take series of parameters and do some work like for example connect to a database to save or query some data and then return a response. The service also allows for easy integration with the most used social media and chat platforms and even smart home devices like Amazon's Alexa or Google's own Google Home.

The bot that we implemented is an SNTF (Société Nationale des Transports Ferroviaires) chatbot, that helps SNTF clients figure out trains schedules, book tickets, report problems, and receive answers to some of the most common questions.

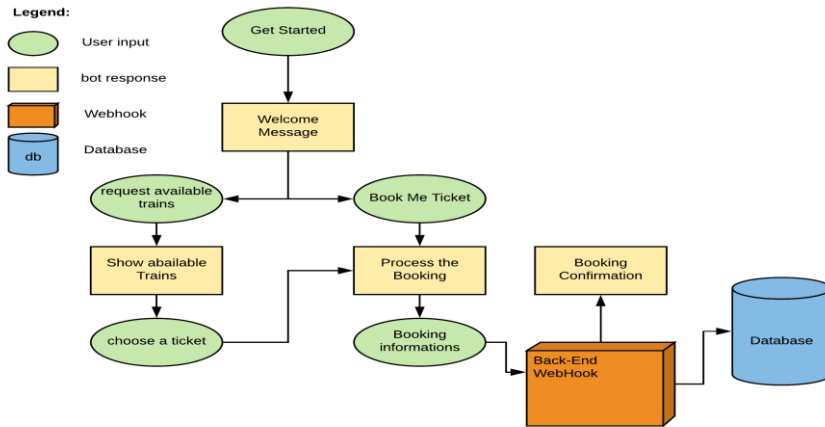


Fig. 3. Conversational Flow Diagram

Figure 3 represents a conversation between a client and our bot. It illustrates the user's inputs, the bot's responses and the different calls to external sources, allowing us to have an overview over the whole conversation and the parts of our system.

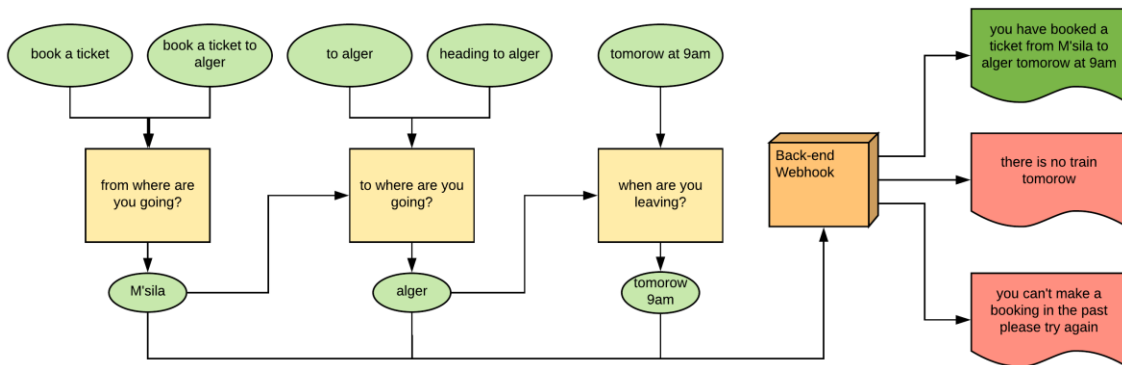


Fig. 4. Diagram representing only the Booking flow

In Figure 4 we can see an example of a booking conversation flow and how it is processed, if a required piece of information is missing the bot will prompt the user to enter it.

Also, we can see in Figure 5 that the bot has a Decision engine which will decide if it should use the neural network, or if the request is just a simple question, like a greeting to begin the conversation, or it could be a question about the bot's personality, and like we mentioned before, the personality is one of the common challenges, so the common question about the bot personality must not be generated from the neural network because it will have a different answer every single time.

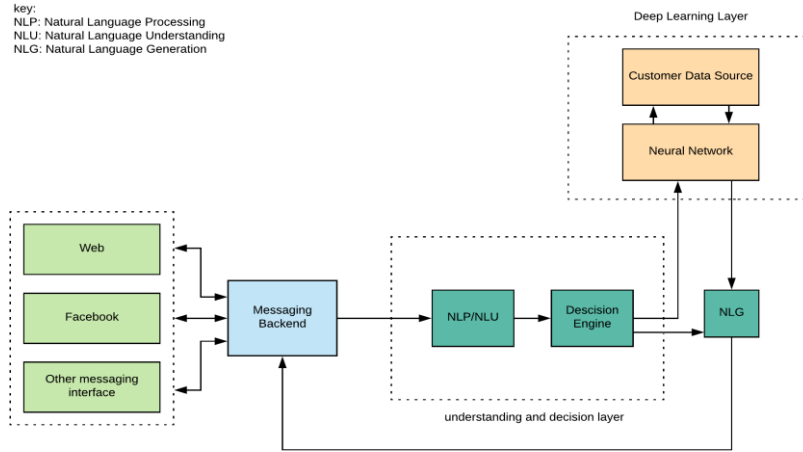


Fig.5. Architecture Diagram for Chatbots

#### 4. Experimental Results and discussion

Experimentation was performed using Python language and the TensorFlow tool. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. TensorFlow is the newest open source library written in Python for numerical computation. It has immediately a great success in the Machine Learning community and in less than one year it also had a lot of support and development by Google itself, more over by many community projects, developed in any area of Deep Learning.

Datasets used along the different phases of experimentation are: Cornell Movie--Dialogs Corpus and Reddit conversation dataset.

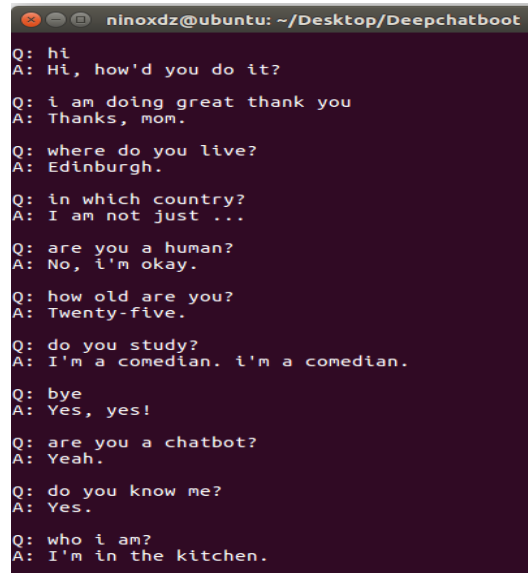
The Cornell Corpus is a large set of imagined conversations, starting from movie scripts crawled from various sites. Metadata for conversation analysis and duplicate-script detection involved mostly automatic matching of movie scripts with the IMDB movie database; clean-up resulted in 617 unique titles tagged with genre, release year, cast lists, and IMDB information. This corpus contains a metadata-rich collection of fictional conversations extracted from raw movie scripts:

- 220,579 conversational exchanges between 10,292 pairs of movie characters.
- involves 9,035 characters from 617 movies.
- in total 304,713 utterances.

Reddit conversation dataset contains 1.7 billion comments total, with data about the author, subreddit, position in the comment tree, and comment score for each post, this dataset is over one terabyte uncompressed, so it would be best for larger research projects, each month comment can be downloaded separately.

#### 4.1. Full deep learning chatbot

After 3 weeks of non-stop training using this huge dataset on the personal computer, the resulting chatbot was not able to produce good responses, it did not have a personality, it made some grammatical mistakes, all of that was expected due to the dataset we used and the amount of time we were able to run the training. The chatbot did however show some results. We managed to train a model that always generates complete and correct sentences, though they do not make sense in the context of the conversation (see Figure 6).

A screenshot of a terminal window with a dark background. The title bar at the top reads 'ninoxdz@ubuntu: ~/Desktop/Deepchatboot'. The terminal displays a series of question-and-answer pairs between a user (Q) and a chatbot (A). The chatbot's responses are often nonsensical or out of context, such as 'I am not just ...' when asked about a country, and 'I'm in the kitchen.' when asked who it is.

```
ninoxdz@ubuntu: ~/Desktop/Deepchatboot
Q: hi
A: Hi, how'd you do it?
Q: i am doing great thank you
A: Thanks, mom.
Q: where do you live?
A: Edinburgh.
Q: in which country?
A: I am not just ...
Q: are you a human?
A: No, i'm okay.
Q: how old are you?
A: Twenty-five.
Q: do you study?
A: I'm a comedian. i'm a comedian.
Q: bye
A: Yes, yes!
Q: are you a chatbot?
A: Yeah.
Q: do you know me?
A: Yes.
Q: who i am?
A: I'm in the kitchen.
```

Fig. 6. Deep Learning Chatbot Conversation

The results we achieved with this approach were not completely satisfactory but still a valuable experience, as we learned first-hand about the difficulties in this field, and we now have several ideas on how to tackle the problems in future works. One of the things that we can use to directly improve our bot, could be using a more specialized dataset, acquiring such a dataset will be a challenge in itself, one way would be to start a community project to build a test dataset for a specific customer service scenario, so that researches can use it to test new methods. Another problem we faced is processing power; this problem is solved by acquiring better hardware or gaining access to a super computer or a computer powerful enough to cut down on training times.

#### 4.2. Dialogflow chatbot

Dialogflow is a platform owned by Google that allows to create a natural language interface by providing actionable data based on the input given. The platform includes speech recognition, deep learning, natural language understanding, machine learning as well as text to-speech capabilities. The platform works on the basis of intents and entities recognized from the user's utterances rather than on a predefined flow pattern branching only based on the response of the user.



Dialogflow includes machine learning capabilities to further improve the detection of the intentions from the user utterances. Intents include the following sections: user says, action, response and contexts. Contexts can be used to pass information from previous conversations or external sources. For the intent to be triggered, all the contexts defined for the intent must be active. It is possible to prioritize the intents in case several intents are identified, define fallback and follow-up intents, and define text responses. Rich responses can be used in case of using one of the following one-click integrations that supports rich responses: Facebook Messenger, Slack, or Telegram.

The entities can be copied or moved to another agent easily through the User Interface. Exporting the entities in JSON or CSV format and uploading them back is also supported.

With Dialogflow we had a head start with the platform's already trained neural network that can handle conversations with users. It provides automatic extraction of the essential information from the user's input and allows a good level of customizability which is important from both technical and business standpoints. It also allows giving the bot a personality that corresponds with the company's vision.

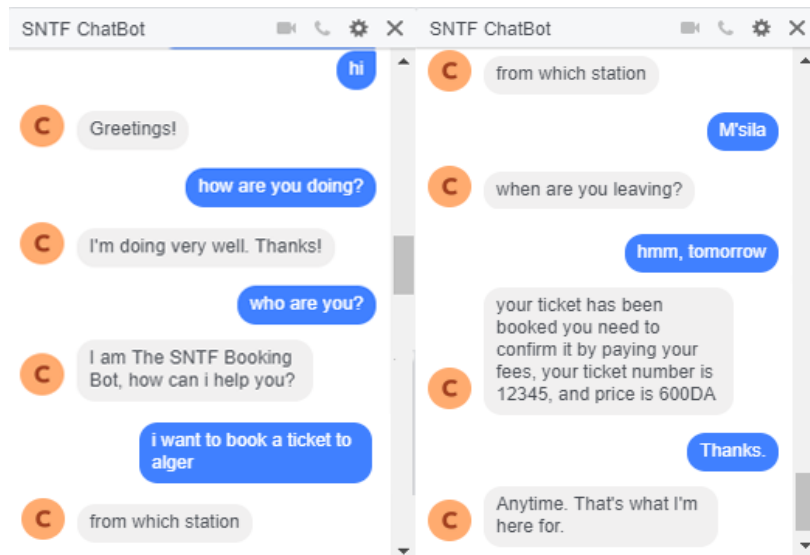


Fig. 7. SNTF bot Conversation on Facebook messenger

As it can be seen in figure 7, the chatbot has the ability to handle normal questions that has nothing to do with the task, from greeting messages to simple questions about its personality. It also detects the user's intent, detects the parameters and can even prompt the user to provide missing ones. Sometimes an external call to webservice or a database is needed, and the bots sends a POST request to a backend webhook that does any work outside the Dialogflow platform.

## 5. Conclusion

In this paper, we set out to make a chatbot geared toward customer support to tackle the problem companies face in that aspect of dealing with their customers in order to offer a more flexible and efficient experience. After weeks of training the chatbot with the available hardware resources and datasets, as well as exploring some cloud services, we found that the results were encouraging. However, the chatbot was not able to answer domain-specific questions or show a kind of personality. For the second chatbot, we used the Dialogflow cloud platform to implement a SNT1F bot that is able to interact with clients, book tickets, answer FAQs, and provide schedule information. The second chatbot performs better and gives more right answers with a certain personality.

We believe that the deep learning approach is still far from being able to result in a customer service conversational agent due to the inability to control the bot response making it really hard to produce relevant and correct responses to the customers' inquiries. We also believe that the use of platforms like Dialogflow is going to increase due to the relatively short time and small resources it takes to develop a reliable chatbot compared to an in-house solution using deep learning or other approaches. The main aim of future works will be acquiring a proper customer service dataset, trying to explore new deep learning approaches, and using open-source libraries for all parts of our work. One recent and promising approach consists of using two deep neural networks, the first one is a conversation response generator like the one that we worked with, while the second one is a decision making neural network that takes the output of the first one, and decides whether to accept the response or send it back to generate another one.

## References

- Krizhevsky, A., Sutskever, I., G. Hinton, E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In "Proceedings of the 25th NIPS Conference" 12.
- Lee, L.J., Frey, B.J., Leung, M.K., Xiong, H.Y., 2014. Deep learning of the tissue regulated splicing code. *Journal of Bioinformatics*.
- Graves, A., Fernandez, S., Jergen, S., Faustino, G., 2006. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In "Proceedings of the 23rd ICM" 06.
- Jergen, S., Graves, A., 2009. Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks. *Advances in Neural Information Processing Systems* 21.
- Graves, A., 2008. Unconstrained On-line Handwriting Recognition with Recurrent Neural Networks. *Advances in Neural Information Processing Systems* 20.
- Gehre, A., Bougares, F., Schwenk, H., Bengio, Y., Cho, K., van Merriënboer, B., 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR*.
- Quoc, V., Le, I.S., Vinyals, O., 2014. A Sequence-to-Sequence Learning with Neural Networks. *CoRR* abs/1409.3215.
- Adit, D., 2018. How I Used Deep Learning To Train A Chatbot To Talk Like Me. <https://adeshpande3.github.io/How-I-Used-DeepLearning-to-Train-a-Chatbot-to-Talk-Like-Me> (accessed Apr. 24, 2018).
- Bengio, Y., Bahdanau, D., Cho, K., 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473.
- Koo, T., Petrov, S., Sutskever, I., Hinton, G.E., Vinyals, O., Kaiser, L., 2014. Grammar as a Foreign Language. *CoRR* abs/1412.7449.
- Vinyals, O., Quoc, V., 2015. A neural conversational model. *CoRR*, abs/1506.05869.
- Carpenter, R., 2018. A Clever Bot. <http://www.cleverbot.com>. (accessed March 14, 2018).
- Ameixa, D., Coheur, L., Fialho, P., Quaresma P., 2014. Luke, I am your father: Dealing with out-of-domain requests by using movies subtitles. *Intelligent Virtual Agents: 14th International Conference*, Boston, MA, USA, Timothy Bickmore, Stacy Marsella, and Candace Sidner editors.
- Higashinaka, R., Mizukami, M., Funakoshi, K., Araki, M., Tsukahara, H., Kobayashi, Y., 2015. Fatal or not? Finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. *Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, page 2243.