Conversational Agent Implementation

SI-II - MP2

Beatriz Costa
Department of Electronics, Telecommunications, and Informatics.
University of Aveiro
N.Mec: 109657

Luís Silva
Department of Electronics, Telecommunications, and Informatics.
University of Aveiro
N.Mec: 88888

Pedro Carvalho
Department of Electronics, Telecommunications, and Informatics.
University of Aveiro
N.Mec: 84670

Tiago Pinho
Department of Electronics, Telecommunications, and Informatics.
University of Aveiro
N.Mec: 92938

***Abstract*—One of the most advanced technologies nowadays is the chatbot, that is a program that uses artificial intelligence to carry out conversations with an user. This project presents the contribution of different authors in works related to Chatbots, as well as a creation and an implementation of a chatbot from scratch using specific methods.**

Keywords: agents; chatbot; intelligent systems; machine learning; natural language processing

I. Introduction

A. Contextualization

With the evolution and triggering of AI our way of engaging in our day activities changed. A **chatbot** is an artificial intelligence program that simulates human conversation through text or voice interactions (HCI model - Human Computer Interaction model) [1]. According to the Cambridge Dictionary website the word chatbot means ”a computer program designed to have a conversation with a human being, especially over the internet”.[2]

Its concept can be traced back to 1950 when Alan Turing wondered if a computer program could talk to a group of people without them realizing that the interlocutor was not human [1]. Some defining moments that led chatbots to be what they are today were:

- 1966 - The first chatbot, ELIZA, was created. ELIZA returned the user's sentences in the interrogative form [1].
- 1972 - Appearance of PERRY. Simulated a patient with schizophrenia and it defined his responses based on a system of assumptions and ”emotional responses” [1].
- 1988 - AI is firstly used in chatbots when creating Jabberwacky [1].
- 1995 - Creation of ALICE (Artificial Linguistic Internet Computer Entity), the first online chatbot [1].
- 2001 - Development of SmarterChild which help people with their daily life tasks and retrieve information from databases about a lot of topics [1].
- 2010 - The famous Siri was created which was the pioneer of personal assistants chatbots [1].

Conversational agents, chatbots, have several applications, namely in the fields of education, business and e-commerce, health and entertainment. Productivity is the most important motivation for chatbot users, so, they are more commonly seen in the business field since they reduce service costs and are able to handle many customers simultaneously. At the end of 2016, thirty four thousand of chatbots were spread across many fields, some of those mentioned before [1].

Every passing year, chatbots are gradually becoming more fully aware of their interlocutor's feelings, so with the advent of chatbots, many people are having trouble distinguishing if who they're talking to on the Internet is a bot or, in fact, human. With this in mind, The Turing Test was created to see if robots could pass as human or not.[3]

B. Motivation and Objectives

There are two main types of chatbots. One that is task-oriented, which as the name implies, they generate automatic conversational responses to user inquiries. These chatbots are highly specific and structured. The other type of chatbots are data-driven and predictive chatbots, also referred to as virtual assistants. They apply predictive intelligence and monitor data and intents of the user, also being able to initiate conversations by themselves.

For this reasons, in this document we aimed to create two chatbots, one for each type, and then merge them together to create a even better global chatbot. Each chatbot has to present the following characteristics:

- Uses natural language processing (Portuguese and/or English) for some common sentences types.
- The chatbot presents the ability to accumulate information/knowledge provided by interlocutors (i.e. learn from interaction) and produce answers to questions.
- For grammatically incorrect sentences, or sentences not supported by the system, the chatbot reacts in a ”seemingly intelligent” way.

II. Related Work

As refered in section I, chatbots have been increasingly investigated and implemented for numerous applications in the most diverse branches of industry, such as education [4], healthcare [5] and even business [6].

There are two types of general approaches to chatbots, with them being rule-based chatbots, and Artificial Intelligence

(AI) based chatbots [7]. Within AI-based chatbots, there are two possible distinctions: Information-Retrieval chatbots and Generative Chatbots.

Rule-based chatbots were the considered first approach of implementation of this specific programs. They look through pattern matches in the users' speech and if the input cannot fit into a rule, it gives an inaccurate answer. On the other hand, AI-based chatbots are based on Machine Learning algorithms, allowing them to learn from existing datasets from the most various backgrounds.

In terms of Information Retrieval based models, the algorithm retrieves the information needed based on the user's input given a dataset of textual information [8]. They include a pre-defined set of possible answers so the chatbot is able to process the query and pick one of the answers available in its dataset [9]. For this reason, this type of models are less suitable to develop a personality, which is an important trait for the conversational type of chatbots. Despite this, there has been lots of progress and evolution in developing new algorithms to solve this issue.

In this section of the document we present the associated works of implementations and applications related to conversational and action-based chatbots, both being included in the category of Information Retrieval types of AI-based chatbots. In sections IV and V we will focus on presenting and describing our approaches to this type of architectures.

[10] proposed a deep architecture to model the complicated matching relations between two objects for matching tasks in natural language. This co-occurrences of words are able to define a context and improve the conversational aspect of a chatbot.

Other important improvement was the one proposed by [9], that aimed on obtaining more contextual information in order to improve the quality and the correctness of the output. The process is achieved by a Deep Neural Network that ranks and then merges the set of question and answer matched with the last input and also the pair of question and answer matching with previous conversation turns. In this way, the algorithm can build a contextual information from the user's previous queries, retrieving a better answer within the knowledge base that is stored.

The introduction of Transformers [11] was one of the most important innovations in terms of Deep Learning language models because of the dispensation of recurrence and convolutions in its entirety, since it is based solely on attention mechanisms. This model replaces the RNN models like long short-term memory (LSTM) and permits training on larger datasets than was before achievable. With this evolution, appears some pretrained systems like BERT (Bidirectional Encoder Representations from transformers) [12]. This system is the base for the implemented action-based chatbot, which will be discussed further on.

III. Natural Language Understanding (NLU)

Natural Language Understanding is a subset of Natural Language Processing (NLP) and is a field that focuses on determining what is the meaning or purpose of the user's speech. NLU enables the system to retrieve from an unstructured text what we are going to refer to as Intent. Intents codify the perceived meaning of a user's interaction with the system.
NLU has been growing in popularity due to the machine's ability to perform language-based analysis around the clock and in a consistent manner.

IV. Conversational Chatbot

This Chatbot will focus on the conversational part of Chatbots, therefore will be called the Conversational ChatBot. It will handle anything not functional and isn't capable of handling orders. It focus on answering closer to what a human would, since it also learns from the user, in case it fails.

A. Dataset

We can dare to say that the dataset is one of the most important parts of the conversational Chatbot, if not the most. It allows the Chatbot to train on natural language and understand strings of words and the meaning of it.

The dataset used to train this part of the Chatbot, came from *Kaggle* and it's called **Chatbots: Intent Recognition Dataset**. We chose this dataset because it has a big variation of intents and density of examples of inputs and responses. We also added more data to make the training of the Chatbot more complete and removed unused parts of the data.

To make the chatbot able to learn through interaction, the dataset will evolve and grow with input from the user. Anytime the chatbot answers wrongly, the user can correct him by choosing an intent or creating a new one as well as a suitable answer.

B. Training

The Chatbot is trained by a Deep Supervised Learning model, where the labels are the intents and the data is the examples of inputs on the dataset.

1) Pre-processing: This segment focus on making the data suitable to the model that will be used.

First we export the data from the dataset, using only the *intent* and *text* since these are what matter for the Chatbot to learn and predict. The *text* represents the training data and the *intent* represents the target labels

After that, we use the function *LabelEncoder* from *scikit-learn* library to change the labels into a more understandable format to the model.

To end this segment, we will tokenize the training data. Since we are working in natural language, we need to tokenize the data to filter and organize it in isolated words in lists.

2) Model: The model will be implemented through *Keras*. *Keras* will simplify the design of the model through already existing layers. The main focus will be on using a Embedding Layer.

The Embedding Layer allows the model to transform the words in vectors of values with a defined size, each word as a set of values defined in the correspondent vector.This way Embedding Layer works like a lookup table. The words are the keys in this table, while the vectors are the values.

The rest of the model is more general and has the following structure:

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 20, 16)            16000
_________________________________________________________________
global_average_pooling1d (Gl (None, 16)                0
_________________________________________________________________
dense (Dense)                (None, 16)                272
_________________________________________________________________
dense_1 (Dense)              (None, 16)                272
_________________________________________________________________
dense_2 (Dense)              (None, 24)                408
=================================================================
Total params: 16,952
Trainable params: 16,952
Non-trainable params: 0
```

Figure 1: Conversational Chatbot Model

C. Interacting with the Chatbot

Since our Chatbot is trained, now we can use it for what it was trained for. The user may input whatever he wishes. If the user presents their name in any similar form as ”My name is <NAME>”, the Chatbot will remember it and will be able to call the user by the name anytime it’s asked or it makes sense to. Another available functionality is asking the time. Further than that, the Chatbot is focused on trying to have a normal conversation with the user.

1) Intent recognition and Feedback: The Chatbot will try to predict the intent of the input done by the user using the trained model. It pre-processes the input text to be on the same format as how the model was trained. The predicted intent will have to be decoded so the Chatbot can directly consult the dataset.

After getting a prediction, the Chatbot will randomly choose a response correspondent to the predicted intent.

2) Learning by user interaction: Anytime the Chatbot guesses wrong, the user can correct him by writing ”wrong” in chat. This will enter in stage where the user is able to choose what was the intent of what they said, and in case the intent does not exist in the dataset, it can be added and an answer will be asked from the user. Any change will be directly saved onto the dataset, so next time it’s trained, the Chatbot may have a chance to answer accordingly.

3) Limitations: The limited Dataset is a downside for the Chatbot, meaning that if the Chatbot interacted with several users and collected more intents, text and responses, having the Chatbot grow through usage.

Another limitation is the fact that we have to manually input the corrections so the Chatbot can learn. Although it is a downside, it comes from compromise to be sure that the Chatbot is learning exactly how it is being teached.

V. Action Chatbot

A. Dataset

The dataset used for this section containing close to 14,000 entries, each entry being composed of two columns, text and label respectively. With this dataset there are initially 7 Intents that are initially defined PlayMusic, AddToPlaylist, RateBook, GetWeather, BookRestaurant, SearchCreativeWork, SearchScreeningEvent although in our implementation we will allow the user to expand on the dataset and identifiable classes.

B. Bidirectional Encoder Representations from Transformers

This method will make use of BERT (Bidirectional Encoder Representations from Transformers) for the Intent recognition process. Bert hinges on transformers, a deep learning model, that, contrary to other models like the one explored in MP1 - Long-Short Term Memory(LSTM) - that read the input sequentially, processes the entire input at once and so can retrieve contextual information of a word from all of its surroundings.

1) Transformers: Transformers apply a method of self-attention where inputs interact with each other in an effort to understand which one should get the most attention, for any given input three vectors (query, key and value) are derived from it and used to calculate the output. To get the output for a given input we first calculate the attention scores, for N inputs there will be N attention scores as these are calculated by the multiplication of the current input’s query by its own and each of the other inputs’ key vectors. Then the value derived from each input will be multiplied by the the attention scores resulting from their respective keys and the sum of all N resulting vectors is the output for the given input.

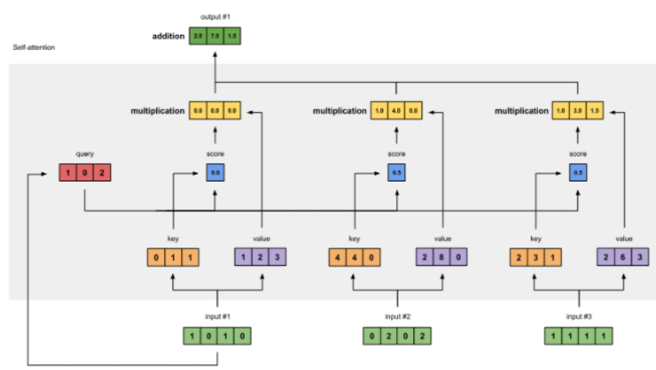

Figure 2: Self-attention mechanism representation

When training BERT two strategies are employed:

2) Masked Language Modelling - MLM: Masked Language Modelling consists of feeding BERT with a sequence and expecting the ouput to match the input, the catch is that roughly 15% of the words in the input sequence will be masked and so the model will have to predict the values of the masked words based on the context provided by the remaining words in the sequence and with this the weights are adjusted and BERT becomes more sensitive to context.

3) Next Sequence Prediction - NSP: Next Sequence Prediction is the practice of presenting two sentences, A and B, to the model and asking it if sentence A precedes sentence B. This teaches BERT to understand longer-term dependencies between sentences.

4) Fine Tuning: In our implementation we are starting off from a pre-trained BERT model so the previously described processes have already been performed but we need to suit it to our use case and so we are going to train it using the information contained in the dataset (original and any user added text+label rows that may have been created by this point). With the original dataset content we get over 100 million trainable parameters and when the training process is done the model is now able to recognize the intents.

```
Layer (type)                 Output Shape              Param #
=================================================================
input_ids (InputLayer)       [(None, 38)]              0

bert (BertModelLayer)        (None, 38, 768)           108890112

lambda (Lambda)              (None, 768)               0

dropout (Dropout)            (None, 768)               0

dense (Dense)                (None, 768)               590592

dropout_1 (Dropout)          (None, 768)               0

dense_1 (Dense)              (None, 7)                 5383
=================================================================
Total params: 109,486,087
Trainable params: 109,486,087
Non-trainable params: 0
```

Figure 3: BERT model parameters

	precision	recall	f1-score	support
PlayMusic	0.94	0.98	0.96	86
AddToPlaylist	1.00	0.99	1.00	124
RateBook	1.00	1.00	1.00	80
SearchScreeningEvent	1.00	0.91	0.95	107
BookRestaurant	0.98	1.00	0.99	92
GetWeather	1.00	0.98	0.99	104
SearchCreativeWork	0.90	0.96	0.93	107
accuracy			0.97	700
macro avg	0.97	0.97	0.97	700
weighted avg	0.97	0.97	0.97	700

Figure 4: BERT model statistics

C. Implementation

Now that BERT is able to recognize the intent of a given interaction we can start feeding it user input but not before performing the proper pre-processing tasks.

1) Data Pre-processing: The pre-processing tasks consist of tokenizing the input, adding special tokens that mark the beginning of the input [CLS] and the end of each sentence [SEP] and padding the sequences so that the size is uniform. The resulting data can now be passed on to BERT whose output will be the corresponding intent.

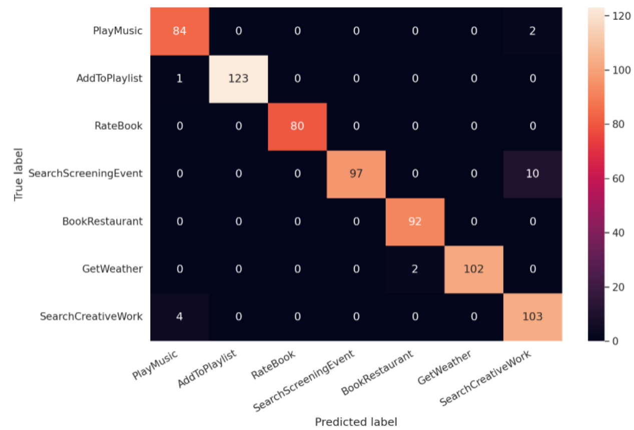

Figure 5: BERT model confusion matrix

2) Response and learning from interaction: Having access to the intent the last step in the process is to react accordingly. Having a clear distinction between intents is fundamental in defining the key elements of a request so that we can fulfill it. We retrieve the necessary information from the input using Spacy and its Named Entity Recognition (NER) capabilities, if the information is not available or was not correctly identified the program will specifically query the user for the missing bits of information. Taking as an example the BookRestaurant intent, we have identified 5 key elements for a successful completion of the request name of the restaurant, date of the reservation, time of the reservation, number of people. If the input given by the user is ”Make a reservation for tomorrow” the only element present is the date and so the user will be prompted for the remaining elements.
Although we were able to achieve very satisfactory scores when identifying the intent of an input the system is not without its flaws and so we present the user with the opportunity to, when an input's intent has been mislabeled, correct this by adding the mislabeled text to the database under the correct label with the option of using one of the existing labels or creating a new one. The changes will only come into effect after BERT is trained with the updated database.

3) Limitations: This method has a few limitations. First is that the creation of new intents requires manual posterior addition of new code to process lsaid intent. Secondly, new intents need to be populated before BERT can reliably recognize them, this will take time.

VI. Merge and Results

For a complete user experience we then proceeded to combine the two methods allowing the user to, within a single system, have access to both the action and conversational chabots capabilities presented as one cohesive unit.
Upon initialization the first state the user is presented with is the conversational but at any moment the user can alternate between the two functionalities by typing ”conversation” (to

access the conversational chatbot) and ”action” (to access the action chatbot). In any one of the agents typing ”wrong” will trigger the process for database updating and typing ”bye” will end the interaction.

As extra work and because the group found it interesting we employed a speech recognition solution that allows the user to use their voice to communicate with Clementina.

VII. Conclusion

In conclusion, as expected, since we have a small dataset, the results, for both approaches, sometimes are not very accurate, however with the constant improvement and increase of the size of the dataset better results can be anticipated.

References

[1] E. Adamopoulou and L. Moussiades, “Chatbots: History, technology, and applications,” *Machine Learning with Applications*, vol. 2, p. 100006, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827020300062

[2] “Chatbot.” [Online]. Available: https://dictionary.cambridge.org/pt/dicionario/ingles/chatbot

[3] “What is a chatbot?” [Online]. Available: https://www.oracle.com/middleeast/chatbots/what-is-a-chatbot/

[4] R. Winkler and M. Söllner, “Unleashing the potential of chatbots in education: A state-of-the-art analysis,” in *Academy of Management Annual Meeting (AOM)*, 2018.

[5] A. N. Vaidyam, H. Wisniewski, J. D. Halamka, M. S. Kashavan, and J. B. Torous, “Chatbots and conversational agents in mental health: a review of the psychiatric landscape,” *The Canadian Journal of Psychiatry*, vol. 64, no. 7, pp. 456–464, 2019.

[6] N. Thomas, “An e-business chatbot using aiml and lsa,” in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016, pp. 2740–2742.

[7] E. Adamopoulou and L. Moussiades, “An overview of chatbot technology,” in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2020, pp. 373–383.

[8] T. Makhalova, D. Ilvovsky, and B. Galitsky, *Information Retrieval Chatbots Based on Conceptual Models*, 06 2019, pp. 230–238.

[9] R. Yan, Y. Song, and H. Wu, “Learning to respond with deep neural networks for retrieval-based human-computer conversation system,” ser. SIGIR ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 55–64. [Online]. Available: https://doi.org/10.1145/2911451.2911542

[10] Z. Lu and H. Li, “A deep architecture for matching short texts,” *Advances in neural information processing systems*, vol. 26, 2013.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762.pdf

[12] J. Devlin, M.-W. Chang, K. Lee, and K. N. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018. [Online]. Available: https://arxiv.org/abs/1810.04805