INFO - H410 - TECHNIQUES OF ARTIFICIAL INTELLIGENCE :
PROJECT REPORT

ULB Virtual Assistant

Professor:

Bersini HUGUES

Sassine   NAIM
Matricule   440091

29 Mai 2020

Abstract

Digital Virtual Assistants are very common in today's world, and are mainly used to assist and to help us in every day tasks. They are already invading our websites, our phones and even our houses. Many of them are built and used through a ChatBot that will answer a user's questions and demands, for example, a mall could have a ChatBot to answer customers questions or help them find directions. The aim of this project is to develop a virtual assistant for the ULB, more specifically, a virtual bot that will answer typical questions asked by visitors, or students or even teachers.

Contents

1   Introduction

The rise of automation, along with increased computational power, novel application of statistical algorithms, and improved accessibility to data, have resulted in the birth of the personal digital assistant market, popularly represented by Apple's Siri, Microsoft's Cortana, Google's Google Assistant, and Amazon's Alexa.

As far as it goes, digital virtual assistants are evolving each and every day, with one ultimate goal ahead of them : understanding human commands, and correctly servicing their demanded needs. VA (virtual assistants) are indeed very helpful in today's world, very common and I would say, a new trend that everybody is following. Whether it's Siri that is giving you directions to your "rendezvous", Cortana that is reminding you to send an email to your boss, or the Brussels Airport Chatbot that is answering your typical questions and giving you flight information, virtual assistants are everywhere, helping us with anything, answering all of our daily typical questions and invading every possible market.

The aim of this project is to get a real-world experience with the usage of different modern Artificial Intelligence Techniques. That is why, I decided to build a Virtual Assistant for my own university, using Deep Learning (DL) and Natural Language Processing (NLP), two of the most commonly used and important AI techniques in today's digital world.

To start off this project report, I am first going to introduce the theoretical concepts I researched and used for this project, based on NLP, Neural Networks and DL. Then, I will be going into the design of my program, how I built it, and how I used NLP and DL to code a Virtual Assistant. I will end this report with the different results that I got.

2   Natural Language Processing

Let's start by defining what is Natural Language Processing and what do we use it for in today's world.

For us Humans, it is pretty easy to understand a conversation (of course in a language that we understand), it is pretty straightforward, to understand the meaning of a word, a sentence or a paragraph. When someone tells us something, we have the capability to understand what the other person is saying. We also have the capability to answer, and to exchange questions and answers. So for us Humans, it is easy to extract the meaning of a word or sentence, it is easy to distinguish between a question, an order, a suggestion, or even simpler, a verb or an adjective. But how can a machine do that? small spoiler, yes NLP is the answer

To make it short, Natural Language Processing is a field of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages. It is a discipline that focuses on the interaction between data science and human language.

For this project in particular, I used NLP to analyse the commands of a user. For example, if a user types in "Hello, how are you doing today?", I used NLP to process the given data, extracts the different words, analyse them and lemmatize them. I will be getting into more details in the sections below.

3   Neural Networks and Deep Learning

Artificial Neural Networks are a subset of Machine Learning. They constitute a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. They are mainly used to predict an output answer, based on a input data.

On the other hand, Deep Learning methods are also a subset of Machine Learning. Well, to be fair, they represent the "new and updated" version of Neural Networks. Deep learning constitutes the very cutting edge of artificial intelligence (AI). Instead of teaching computers to process and learn from data (which is how machine learning works), with deep learning, the computer trains itself to process and learn from data.

This is all possible thanks to layers of Neural Networks. Well a Neural Network that is made up of more than three layers – i.e. an input layer, an output layer and multiple hidden layers – is called a 'deep neural network', and this is what underpins deep learning. A deep learning system is self-teaching, learning as it goes by filtering information through multiple hidden layers, in a similar way to humans.

4   VAs and ChatBots in our World

Formally, a Virtual Digital Assistant is a software agent that can perform tasks or services based on the commands of a user. Here, we also use the word "ChatBot" to explicit the fact that a Virtual Assistant can also be used as a ChatBot, with a chatting interface.

A ChatBot is typically used in today's industry to answer customers questions and provide guidance the users. For example, on the Brussels Airport Website, you could start a discussion with their ChatBot, asking about the flight number, the flight info ...
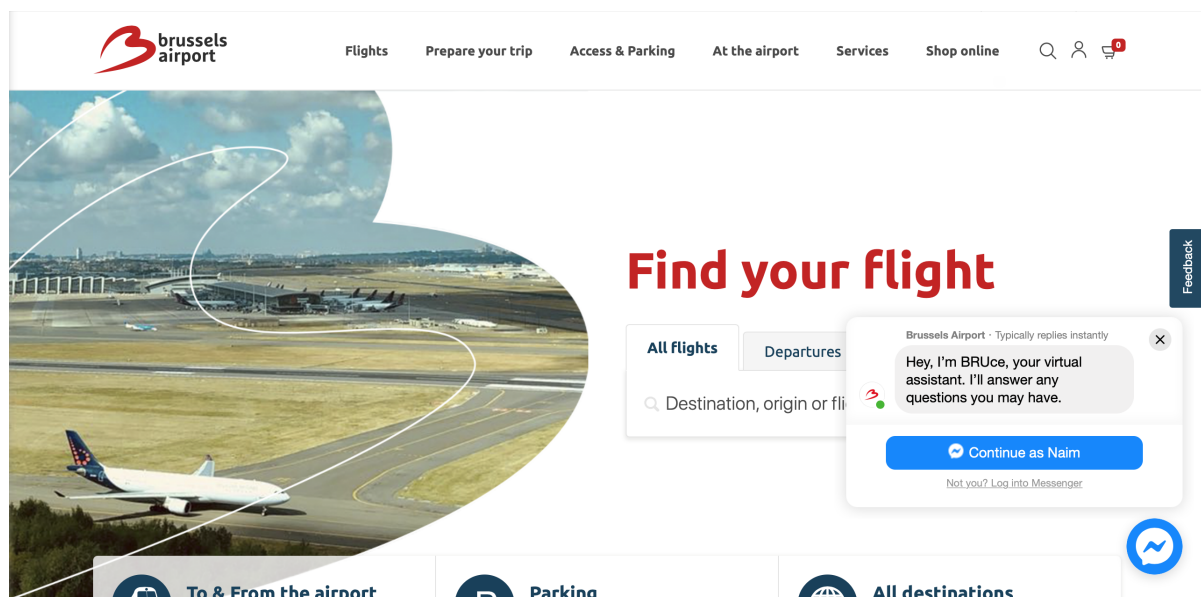

Figure 1: Brussels Airport Website/ChatBot

In my case, I decided to build a virtual assistant, but that is dedicated to questions that are directed to the university. Such a VA could easily be used on the web page of the university, or even in classes, or on campus. It could be used as a bot that performs tasks, and is mainly used by new comers, where the VA could advice them, show them where to eat, what to do, where to study ...

So my goal for this project was simple, to build a program that will take a user's command, or question, analyse and understand it, then will answer with the right appropriated answer, where the questions would majorly be about the ULB.

5 Designing and Building the ULB ChatBot

After introducing all the theoretical concepts, in this section, I will try and connect all the previous ideas together, to explain how I built my VA.

NB : the way to communicate with my ChatBot here is by typing the commands, I also implemented a version where the commands and the answers could be dictated, so a voice assistant, but since the project presentation is going to happen by video conference, I preferred to use the typed version, because it will be much clearer during the presentation, but a future vocal version of the VA could easily be implemented, I already have the majority of the code for that

I used the Python Libraries *nltk* for Natural Language Processing, and the *Keras* for the Deep Learning Part.

Here is the basic execution of the program : the VA starts by greeting the user with a welcoming message : "Hey, my name is SAL! How can I help you today?" (Oh yes, I forgot to mention, my ChatBot's name is SAL).Then, the user types in the command, typically it is a sentence that could be a question, or an order, for example : "Hi! This is my first day as a student at the ULB and I am a bit lost, can you give me a map of the campus?" The program here, needs first of all to analyse the sentence given using NLP, and then will run this sentence through a Deep Learning Network to get the correct answer, in this case, return a map of the campus! Let's get into more details on how this works

5.1 Building and Training the Model

For the machine to be able to understand the different commands of the user, I needed to train the deep learning neural network. But first of all, train it with what?
Here are the different steps that I took, to build the data set, build the neural network, and train it :

- I spent a lot of time at the beginning to categorise the different information, someone would want from a university ChatBot, so i went through the ULB websites and created a list of different answer topics to classify the multiple answers my VA could

7

give. Those answers can be divided into two major categories, and multiple sub categories. But for the two major categories : we can have typical already signed-in user type of question like : "Can i get my schedule for the week?" or "Show me my grades", and for the second big category, its all types of questions none-related to a specific student, like "where can i eat on campus?" "Who founded the ULB?" etc ...

- I then created a data set. For each and every answer category, I wrote a list of questions that could lead to that answer. For example, for the category : "Courses", typical questions would be "can I get my list of courses for this semester?" or "what are the different courses that I have?"

- Using NLP I pre-processed the data, cleaned it and made it as usable as possible. More specifically, I used the *nltk* functions : *word_tokenize* to tokenize each and every question, and the *lemmatize* function to lemmatize the different words. Lemmatization in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form.

- By tokenizing and lemmatizing, I created a bag of words for my questions, which only contains base words in attempt to represent related words. So this bag of base words, contains the different patterns that were taken from the different questions.

- To represent the data numerically, I then create a bag of words array. For each and every sentence taken as input we give an array of dimension the length of the bag of base words generated in the previous step. So each element of the array represents a certain pattern in the bag of base words. The element of the array takes value 1 if a word in the input sentence matches the pattern, and 0 if it none of the words match the pattern.

- The Training Data is finally complete by assigning to the previously built bag of words array, the corresponding answer number.

Now that the Training Data is generated and completed using NLP, through the *nltk* library, let's talk about the neural network :

- I used the Keras Library to Build a 4 layer Neural Network

- The input layer has a number of neurons equal to the size of the bag of base words, so equal the size of the array

- The first hidden layer contains 128 Neurons, the second 68 and the third is the output layer, containing a number of neurons equal to the number of answers categories

- To train the model I made multiple data passes through the network. So each and every array of the training data, representing a certain sentence is given as input to the Neural Network. The data will pass through it, predict an output. The output will be a certain answer category. I use the basic stochastic gradient descent, softmax, and relu methods (rectified linear unit) as optimisation, normalisation and activation functions.

5.2 Testing the Model

Now that the model is built, it is pretty straightforward to see how I tested it, but I will re-explain it here :

- The program takes the sentence as input

- It cleans it up, and tokenizes the different words

- It lemmatizes the words, so we can get the base of them

- Then it creates an array for the sentence, each element of the array, like in the previous case, represents a word in the bag of base words built during the training.

- The array element takes value 1 if a word in the input sentence matches the corresponding element of the bag of base words, 0 if not.

- We then give the array as input to the Neural Network, it passes it through the different layers, and gives us a number as output. This number represents a certain answer category

6 Results and Conculsion

After this long and heavy part on every theoretical part this project covered, let's get into the results.

So first, we have a complete program, that given a sentence as an input, which should typically correspond to a question related to the university, will make this input go through Natural Processing Methods to analyse it, and will then give it to a Neural Network to predict the correct answer for this specific question.

In terms of In Sample Error for the training, I got a precision of 0.985, a value that I am satisfied with. The Virtual Assistant answers correctly to majorly all the tests that I ran on.

For the different answers, I coded multiple ways to provide the answer to the user. For example, if a user wants to get his university grades, the VA will typically load his MONULB page where his grades are printed out, giving him direct access to all his grades, in a single command.

In terms of GUI, the Chatting Interface is not totally complete! The program fully runs in terminal, but the Interface is not yet aesthetically pleasing, that is why I won't post any screenshots of the program here, and wait for the presentation to show them!

In conclusion, I can proudly say that this project has brought me more experience and knowledge, than any kind of project I went through during the course of my studies. In fact, thanks to the liberty that the Professor gave us, to develop our own ideas, to go as far as we want to go, to develop and build what we wanted and how we wanted, I learned and acquired a large set of skills and knowledge, in AI, as well as many different fields that relate to coding and designing an Artificial Intelligence based program. Indeed, to be able to understand the principles and the underlying structures of NLP and Deep Learning, I had to follow my Professor's lectures and conduct a lot of personal researches, specially because Virtual Assistants are a very specific type of AI.

In the end, even if my program is not the most powerful VA or ChatBot ever made, I am satisfied that I learned a lot along the road, and I believe that is why these projects are conceived for.

References

[1] The hype behind InferSent - Towards Data Science.

[2] Keras Tutorial: The Ultimate Beginner's Guide to Deep Learning in Python.

[3] Keyword Extraction: A Guide to Finding Keywords in Text.

[4] PyOhio | Presentation: Natural Language Processing in Python.

[5] Python Programming Tutorials.

[6] Python Programming Tutorials.

[7] Text Classification: a comprehensive guide to classifying text with machine learning.

[8] Text Mining: The Beginner's Guide.

[9] Your Guide to Natural Language Processing (NLP) - Towards Data Science.