UFR STGI
BELFORT-MONTBÉLIARD

VEEMbot : The Intelligent Chatbot

TUTORED PROJECT REPORT
Master 2 Internet of Things

Enguerrand BOITEL
Elio ELHOYECK
Mohamad MAKHLOUF
Vincent RAMOUSSET

Tutor : Raphaël COUTURIER

2020 - 2021

Table of contents

Introduction

Nowadays, artificial intelligence (AI) is present everywhere. It can be present on a website to help people to find a product or a precise link for example, but it can also appear in video games such as open world games in a city, where there is a lot of people controlled by AI. They are also implemented into cars to drive themselves. Artificial intelligence can help companies too if there is a problem on a software or it can be used to entertain people (talk with a chatbot to have fun or to get information).

Increasingly, libraries and softwares help people to process text (get a verb and a noun in a sentence, correct words, get keywords, ...). In this context, we have developed VEEMbot, an intelligent "chatbot" (we will see after that it is not really a chatbot strictly speaking) to be able to ask open or closed ended questions. This software will answer questions by using the internet and will also have some other functionalities.

The global objective of this project is to develop a tool in Python using libraries already developed. We will adapt these libraries to our program (they allow us to gain time : it can be very hard to create a complete model for artificial intelligence in less than 4 months). To do so, the software, developed in Python, needs to be quick and its responses accurate and consistent.

Thanks to these information, our problematic is the following :

How could VEEMbot be an improvement for future customers ?

First of all, we will present more precisely what we need to do in the project we worked on since the beginning of October. Then, we will present how we set up this project and the organization we have created. And finally, we will present our work in detail.

1 Presentation of the project

VEEMbot is presented as an intelligent chatbot, but in fact this is not an accurate representation of what we are doing. This is more an intelligent tool where there will not be a continuous chat between the user and the tool. It is rather a user asking a specific question and the tool responding.

This tool will be made from scratch using Python. We will not be taking a bot that is finished and modify it rather than we will be taking ideas and building our own tool in order to create the functions in the way we want.

1.1 Objectives

Thanks to our meetings with Mr. COUTURIER and ourselves, we identified three main objectives :

- Answer a given question depending on its structure,
- Propose several basic functionalities,
- Answer in multiple languages.

Our main goal, the purpose of a smart assistant, is to answer a given question. This question can be open-ended (eg : *"Where is Montbéliard ?"*, with an infinity of possible answers) or closed-ended (eg : *"Is Montbéliard in France ?"*, with just two possibilities).

On the one hand, closed-ended questions are much more difficult that just finding a "random" answer between two possibilities (Yes or No) : it forces the algorithm to analyze all the text (source) and the question asked.

On the other hand, open-ended questions consist on finding the most suitable answer depending on the general idea of the question. Understanding the best way to find in which category the question is, was one of the biggest goal of our project.

Moreover, the assistant needs to answer all types of requests : it could be a simple calculation or retrieving the current time for example.

Finally, our assistant could be used in different languages (mostly in French and English) so we had to analyze the question to get the language used and, thus, answer correctly the question.

1.2 State of the art

If you are a person that has a slight knowledge about technology, you would see one thing in common between all big technological companies, that's developing their own interactive smart assistant. We can start by introducing advantages and disadvantages of Google assistant and Apple Siri just to name few, these are two of the most important assistants created today.

Google assistant has speech recognition, mobile integration and is at the same time a chat assistant. Google Assistant is available in more than 90 countries and in over 30 languages and is used by more than 500 million users monthly.

Apple's Siri is one of the most famous AI personal assistants and has the most popular AI apps. It uses natural language user interface (UI) and voice queries to function. It can perform functions such as:

- Make calls and send text messages.
- Answer questions and offer recommendations.
- Dictate location and read weather reports.

In order to be able to use Google assistant, it is not necessary to have an android (you can use it on Apple's products). However, in order to use Apple's Siri you need to own an Apple product. Both of these assistants took years in developing, and are still limited and have limited functionalities. For example ask Google assistant *"what can you do ?"* and it will list all the ideas about what it can help with.

Other smaller companies have already implemented their own assistant for example: Boursorama smart assistant and SoBot Societe Générale's virtual assistant. Both of them only respond to specific questions related to the bank, hence they are also limited.

There are also two other kind of assistant that exists in supermarkets. The first one is an assistant that can help in finding specific products (Chips are in sector B).

Figure 1: Super Market Assistant

The second one is a virtual assistant which is only used to scan products in order to process the prices and names and add them to your list. This project is currently being developed by Amazon and is the only one in the world that is today well implemented on a shopping cart.

After some researches we found that:

- There is no French question answering bot that responds to yes or no questions in French.
- Yes or No dataset for closed questions does not exist in french on the internet yet.

A common problem also found is that all of the previous assistants have the same problem responding to yes or no questions in others ways. For instance, Boursorama bank assistant replies with *"I am Sorry"*, SoBot gives you a list of answers to choose from that have nothing to do with the questions. The most advanced chatbots today are Alexa, Google assistant as well as Siri and they replies to yer or no questions by sending a link related to the main idea.

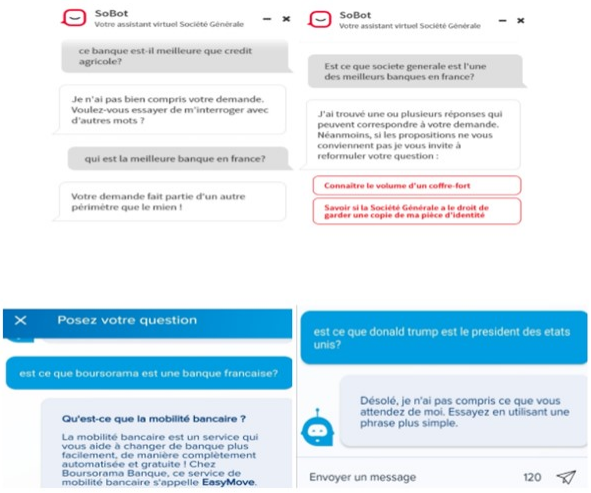


Figure 2: Examples of Chatbots responses.

1.3 Expected results

Our tutor Mr. COUTURIER informed us about a possible customer : Carrefour, and more generally supermarkets which would be interested in a project that could lead customers to the right shelf. These constraints forced us to create an operational and reliable software, but mostly an easy-to-use and versatile application. Thereby, we created a small video demonstrating the characteristics of our project, and proposed it to our tutor so that he could present it to the potential clients. The current lockdown that started in November has complicated matters.

Our final application will thus be a coherent assistant which responds correctly in French, English and possibly in other languages. It could be better if the assistant could understand the language of the question and responds according to that language. At the end of our project, we wish that the assistant could also answer other types of requests :

- *"What time is it ?"*
- *"How many does 12 times 3 minus 5 ?"*
- *"Good morning VEEMbot, how are you ?"*

1.4 Project target

We are currently focusing on companies which need a virtual assistant (artificial intelligence) to implement as a tool. It could be used by small companies that use and improve it but also huge companies that just take the idea to create a better one using their own resources. In fact, we can implement this solution in a terminal in a supermarket to help people to have new information and lead them to a specific product, about a question

they have, or just answer any type of requests.

Figure 3: Shopping cart

2 Organization of the project

2.1 Global organization

For this project, we have set up some files and internet websites to have a better project management.

We have created documents to lead us through the project (e.g. Project's summary - information) and diagrams (architecture diagram, software diagram and hardware diagram). Concerning the document view of the project, we have described what Mr. COUTURIER wants to have a better idea of the purpose. In addition, we tried to share all the interesting websites and documents that could help us to improve our project to all the members, so that each of us know what others are doing. This method also allows us to have a great traceability of what has been done, if for example at any moment we need to find an old source.

Moreover, in this document, we have expressed our objectives for this project (previously mentioned), our project target that is to say who could be interested in our project, the theoretical final project (*what is the result we attempt at the end of the project ?*) and the user case (description of a use of our work). With these elements in place, we have created the different diagrams to have a global view of all the work we need to do. A diagram can be more comprehensive than a file because it is more visual than a long text.

We have also created a Trello (Figure 4 and 19). Trello is an online project management tool that lets you know which tasks have been completed, which are in progress or which have been completed. This tool allows you to know precisely everything that has already been done and everything that remains to be done in the project.

Figure 4: Trello of the last week of the project

To communicate between all the members of the group, during September and October, we planned some meetings at UFR. However, during the lockdown, we had to use communication tools such as Facebook Messenger, WhatsApp or Discord to talk between each other. We have also used Discord to share our screen and Google applications to

complete documents and algorithms in the same time (Google Docs and Google Colab). To communicate with Mr. COUTURIER, we mailed him to obtain more information and to fix meetings if needed. We also used Google Meet for online meetings with our tutor.

Furthermore, in this special context, we work with groups of two people to be more efficient. Given that all the work is done with voice call, working with four people could become very complicated.

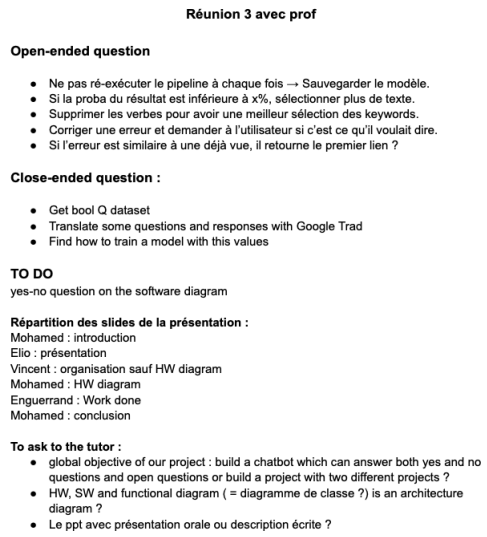
2.2 Agile Method

All of our meetings has been organized as follows :

- In the beginning of each meeting, we make the balance sheets of what has been done,
- During each meeting, between the members of the group, we define a maximal time slot in which we have to solve all of the problems,
- Finally, at the end of a meeting, we discuss together on what we will do for/during the next meeting and ask the tutor all the general or specific questions that are unknown (or implementation that could be done).

For the code part, we write a meeting report to list the problems we need to resolve. Then we can modify the algorithms, test them and finally implement them in the main program of the project. To finish the modifications, we comment the affected part with our judgments.

Concerning our small meetings (when just one or two students work on a specific subject), at the end of each, we inform the rest of the group about all the new parts we have done, the problems we have solved and the ones that appear (Figure 5 and 20).



Réunion 3 avec prof

Open-ended question

- Ne pas ré-exécuter le pipeline à chaque fois → Sauvegarder le modèle.
- Si la proba du résultat est inférieure à x%, sélectionner plus de texte.
- Supprimer les verbes pour avoir une meilleur sélection des keywords.
- Corriger une erreur et demander à l'utilisateur si c'est ce qu'il voulait dire.
- Si l'erreur est similaire à une déjà vue, il retourne le premier lien ?

Close-ended question :

- Get bool Q dataset
- Translate some questions and responses with Google Trad
- Find how to train a model with this values

TO DO
yes-no question on the software diagram

Répartition des slides de la présentation :
Mohamed : introduction
Elio : présentation
Vincent : organisation sauf HW diagram
Mohamed : HW diagram
Enguerrand : Work done
Mohamed : conclusion

To ask to the tutor :

- global objective of our project : build a chatbot which can answer both yes and no questions and open questions or build a project with two different projects ?
- HW, SW and functional diagram (= diagramme de classe ?) is an architecture diagram ?
- Le ppt avec présentation orale ou description écrite ?

Figure 5: Report of a meeting

2.3 Architecture

This architecture diagram describes how a chatbot is working. We also described who are the different actors and what are the roles of these persons in our project.

VEEMbot is composed of three different parts : the presentation layer, the application layer and the data layer, which are all presented on the Figure 6.

2.3.1 Presentation layer

The presentation layer concerns just the user. This layer is used by the user to ask a question but it is also used to display the response.

2.3.2 Application layer

The application layer concerns the artificial intelligence (AI). This layer shows us how it extracts the topic of the sentence and how it can understand the question by using text processing. This layer also contains an internet connection to get the text to answer the question (e.g. : *"What is a monkey ?"* : the AI makes a request using Wikipedia API to get a summary of the most important word of the question asked : monkey.). Then, when we have extracted these data, the AI can choose a consistent response about the question asked by the user.

2.3.3 Data layer

As we are using AI to answer questions, we have to train it before using. This part concerns the last layer of the Figure 6, the data layer. The AI is based on a model. This model is trained (usually in a second application) by using data which we want to learn such as text, JSON, images, ... To use this model, we just need to load the model previously trained and ask for it in the main application.

2.3.4 Actors

The different actors are the following :

- Mr. COUTURIER (our tutor),
- Administration part,
- Communication part,
- VEEMbot developers.

Mr. COUTURIER is the manager of the project. He helped us during the meetings, by answering our mails to lead us to get the best results possible and test our program.

The administration part consists on fixing the issues which can happen, work on upgrades and help the user to use the program. This actor was not that important because of the lockdown (we could not plan any meeting with the potential customers).

The communication part involves exchanging with our customers and if the work is conclusive, we can get internships to improve our work.

Finally, the members of the VEEMbot project are mainly the developers of the application.

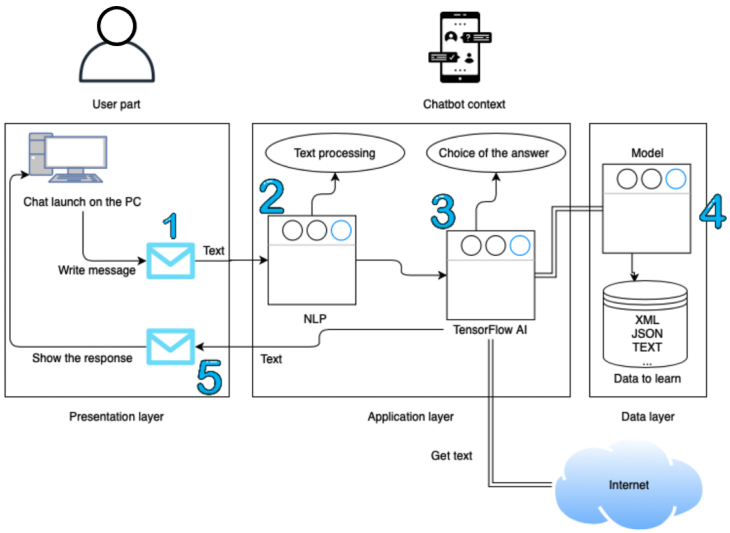


Figure 6: Architecture diagram

2.4 Hardware configuration

This hardware diagram represents a computer (Figure 7). As our program will be executed on a basic PC, this diagram includes the characteristics of a simple computer and we added some elements to fit to our algorithm such as the dedicated GPU.

Firstly, we represented the basic components of a PC, which have no influence on the performance of the computer, as for example the monitor (to show the answer of the AI), the keyboard (to ask a question), the mouse or the storage (access time and writing time reduced for SSD than HDD). The power supply needs to be powerful to support the CPU and GPU performances.

Secondly, the components which have an influence on the performance of the computer are the following : CPU and GPU. The GPU needs to be powerful because of the training of the model. So if the GPU is more powerful, it will take less time, and conversely. The goal is to spend as less time as possible on the training part. The CPU

is also important, but less than the GPU. The CPU will be used to execute the programs (main and training) and the basic tasks in our algorithm (launch GPU functions, compute some elements but not too heavy, declare some variables, ...). It will be used in our main program to talk with the assistant too.

Finally, the last part is the database and the Internet. The Internet is used for online requests to find the summary of the keyword. Concerning the database, it is used to train our model. It will store the different data in different formats such as JSON, XML, text, ...

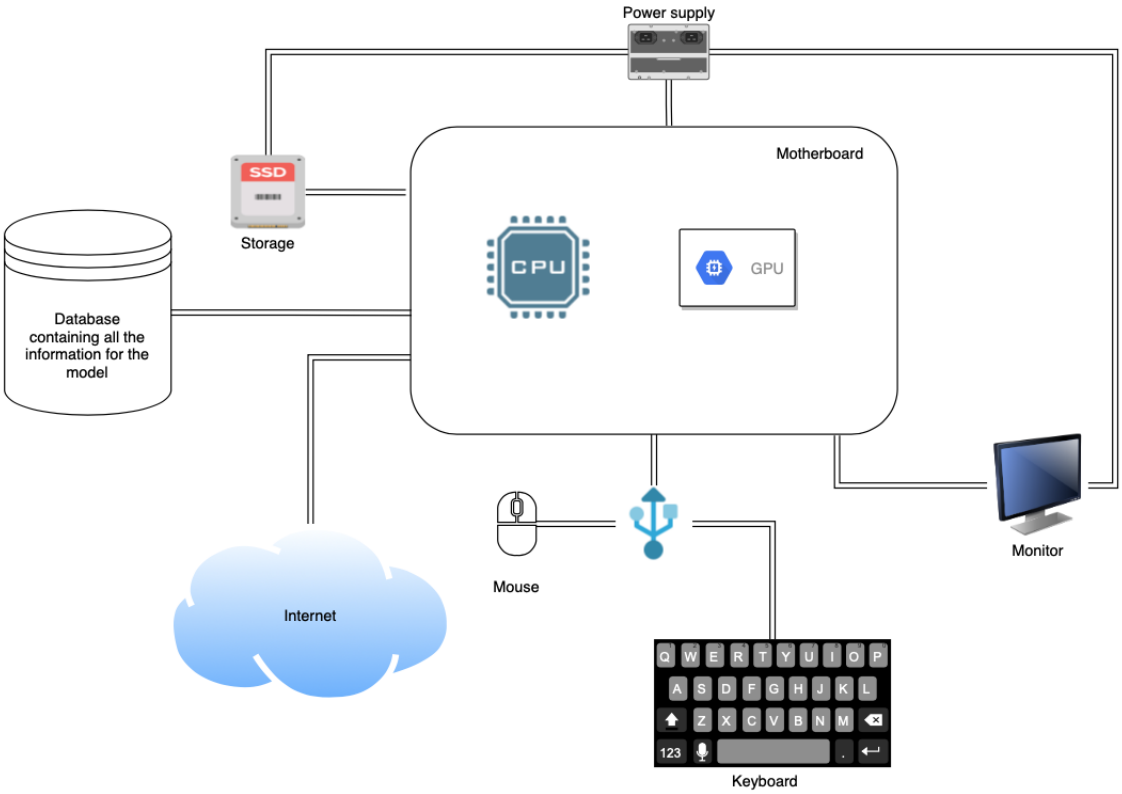


Figure 7: Hardware diagram

3 Detailed presentation

3.1 Software organization

The software diagram (Figure 18) describes how our algorithm works at each step.

The goal of VEEMbot is to answer any type of question. This objective involves that we have to take into account all of the following possibilities :

- A simple sentence (e.g. : *"I'm happy today"*)
- A open-ended question (e.g. : *"Where is Paris ?"*),
- A closed-ended question (e.g. : *"Is Paris in France ?"*),
- A simple calculation (e.g. : *"2 plus 3"* or *"2 * 3 - 4"*),
- A question depending on the time (e.g. : *"What time is it ?"*),

Moreover, during the test of VEEMbot, we faced multiple problems : what if the user made a mistake in the question, if he wants to speak to VEEMbot and get a vocal answer, if he wants more information about the answer given or if he wants to communicate with the intelligent chatbot in a foreign language ?

Step by step, we have improved our algorithm so that the user can use it more easily, and find an answer to his request in a natural way.

3.1.1 Find answer

The first algorithm we tried to develop is answering an open question.

First of all, we started by using NLP (Natural Language Processing) to answer open-ended questions. Of course, we have improved the original algorithm given by our tutor to fit our needs : we had to find a text where the algorithm could answer the question. To do so, we have implemented the Wikipedia-API that find a text related to a given word (See Figure 8). This API takes in parameter the language we want and the word or group of words we are interested in, and retrieves the relevant Wikipedia page.

```
5. Summary : A chatbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing
Most chatbots are accessed on-line via website popups or through virtual assistants. They can be classified into usage categories that include:
```

Figure 8: Summary for the keyword : "chatbot"

In the same time, the second group, composed of Elio and Mohamad, was focused on answering closed-ended questions. Even though the solution could appear as more simple than answers of open-ended questions (there is just two possibilities), answering this kind of questions is much more difficult : in fact, the algorithm has to take into account all of the words of the question, and all of the data, to get a more accurate response. Furthermore, the trainable dataset (BoolQ dataset) was not available in French, so we had two main possibilities :

- Translate the question in English, find the answer and translate it again into French,
- Translate the whole BoolQ dataset into French, train in to answer any French question (it only answers closed-ended questions).

We implemented the two options and finally found that using the French dataset translated get better and more accurate results.

Another way to use VEEMbot is to ask him a simple calculation. We have noticed that the browser Google can answer some calculations, but just when using symbols $*$, $-$, $/$, $+$, and some words such as *times*, *minus*, *divided by* and *plus*. This works in English, and in any other language you have set. However, if you set the language to English, it will not answer any result when the calculation is in another language. For example : if the language is set to English and we are looking for the result of *"2 moins 4"*, Google will find the right language of our request, French but will not display the calculator such as if we have asked him an English calculation.

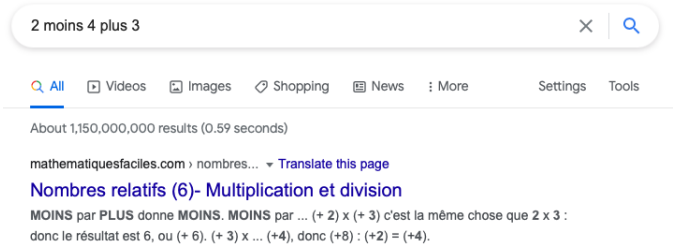


Figure 9: Example of Google : no calculation detected

Furthermore, we implemented this functionality in VEEMbot so that our application understand every calculation entered in both French and English.

```
Question : 2 minus 4 plus 3

1. Language : fr

It is 1
```

Figure 10: Example of VEEMbot : calculation detected

3.1.2 Sentence analysis

We used the package *"TextBlob"* to detect the language used in the request, which is more accurate than *"langdetect"*, so that the answer will be given in the same language.

Secondly, we had to find the most important word in the sentence given by the user, to then find the relevant summary on the Internet. Using *"rake"* and *"NLTK (Natural Language ToolKit)"* was very useful, even though some results were not correct due to some little words : *"the"*, *"a"*, *"I"*, ... We thus implemented a new function that removes all of these useless words in the request of the user (named stopwords).

Moreover, after testing the answer of both types of question, we faced a new problem: the keyword returns sometimes both verbs and the word that interests us. We had thus to analyze the sentence, to find the verb and finally remove it. The analysis done by *NLTK* seems like the following (Picture 11)

```
[('London', 'NNP'), (',', ','), ('is', 'VBZ'), ('it', 'PRP'), ('in', 'IN'), ('the', 'DT'), ('UK', 'NNP'), ('?', '.')]
```

Figure 11: Sentence analyzed using NLTK

We finally just needed to tell the algorithm to understand to check if there is any verb in the keyword found previously, and, if yes, remove it.

Simultaneously, as both opened and closed question answering gave good results, we tried to train an AI that could understand if a given question is open or closed ended. However, we did not find any good algorithm on the internet, and all of our tries have been in vain.

We thus selected an easier way to choose the type of question : with the sentence analysis used before, we developed an algorithm that understand exactly where are verbs, modal words and subjects. This worked better than expected.

3.1.3 Get data

To find the data needed to answer each type of question, we had to use the internet and more precisely Wikipedia, that allows us to get any information we are interested in. However, Wikipedia pages are all related to their own keyword, which is single (no plural, no past tense for a verb ...). Thus, we tried firstly to get the most important word of the question asked. This step worked well, except if the keyword used to find the Wikipedia summary does not exist with the spelling given by the user in his request. In this special case, we were forced to propose to the user all the possible words related to the keyword found (Figure 12).

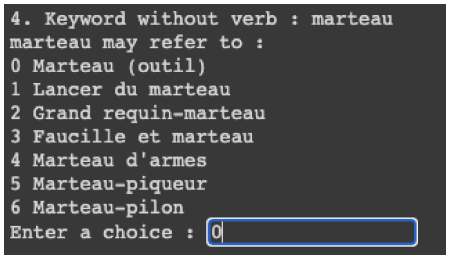


Figure 12: The user selects the best topic of his question

As all the data is found using Wikipedia, the algorithm cannot answer time-depending questions. Indeed, we created a function that analyzes the input and understand if the user wants to get the current time, and finally retrieves Figure 13.

Figure 13: Question about time

Moreover, to improve as far as possible VEEMbot, we tried to catch another possible case : if the user does not want any answer or information, but if he is just interesting in chatting with his assistant. To that end, we reused our TPs done with Mr. SALOMON during our Deep Learning courses. In fact, one of the algorithm done during this course consists in analyzing a sentence to understand if its feeling is positive or negative. We thus turned this algorithm into a code that returns a sentence related to the input given by the user (Figure 14):

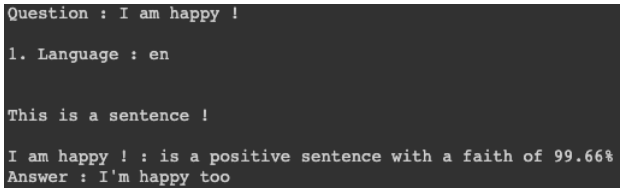


Figure 14: VEEMbot analyzes sentences

3.1.4 Analyze score

After each answer found by VEEMbot, we get a score that is the reliability of the response given. This step thus concerns only the answer of questions (closed and open-ended) and the analysis of the sentence.

Concerning the questions, we have implemented a function that found another summary or text on Wikipedia if the score is less than 70 %. However, this step increased a lot the amount of time needed to get the answer, which is crucial in that kind of assistant, so we decided to remove it (in addition, the score is sometimes not reliable : the answer is correct and the score less than 50 %).

Regarding the sentence analysis, we implemented another function that get the general feeling of a sentence given by the user and retrieve the following :

- If the user's feeling is positive (Figure 14, above),
- And if not (Figure 15) :

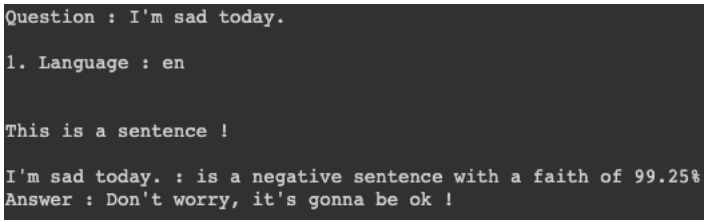


Figure 15: VEEMbot analyzes negative sentences

3.1.5 Communicate with the user

Displaying the answer on the screen of the user was not that difficult : we are using Colab for the moment, even if we tried to create a User Interface (it was difficult to implement an UI on this website, and we preferred to create a complete chatbot that focusing on this part).

Another crucial functionality for an interactive chatbot consists in the communication with the user. In fact, there are two ways to share information:

- Using the keyboard : the user write his request and VEEMbot displays the answers on the screen,
- Orally : the user tells the assistant his request, and VEEMbot tells his answer too.

As we were using COLAB to run our algorithm, listening to the user was a quite difficult implementation because of the need to allow a microphone access onto the user browser. We used two packages to do so : *"TextBlob"* and *"SpeechRecognition"*. This functionality is displayed as follows (Figure 16):

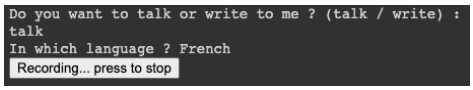


Figure 16: VEEMbot listens to the user

...and concerning the answer (Figure 17):

Figure 17: VEEMbot answers orally

3.2 Balance Sheets

This project was worth it due to two main learning.

3.2.1 Technical skills

During our work, we spent a long time developing in Python on Colab. We firstly tried to download all the packages needed on our personal computers, but this took a lot of GPU and ROM space and depends on each version of Python (e.g. : a same algorithm was not working due to different Python and packages versions). We thus decided to use Colab, so that each of us get the same results, free to our computer power and versions.

Given that VEEMbot takes more than a thousand lines (we created it from scratch), we have improved widely our Python skills. For example, our first main problem was to distinguish open-ended questions from closed-ended questions. There was no question detection type on Python that works well in multiple languages and for any sentence organization (*"Is London in the UK ?"* $\rightarrow$ *"London, is it in the UK ?"*) so we created our own code.

Moreover, we had also to understood some AI codes even though we just trained only once an AI algorithm (for closed-ended questions). Indeed, we understood how *NLP*, *NLTK* and *Transformers (Pipeline)* work to use them the best way possible and more efficiently. We discovered how to choose the right model, depending on different parameters (question type and language) and we worked on data analysis (the better the conditions given to the model, the better the results).

3.2.2 Soft skills

Due to the current lockdown that began on November 2020, we discovered a new way to work on projects. In fact, all of the meetings and works took place online. This organization was a little bit difficult to put in place in the beginning of the project, but we tried to work regularly and to inform our tutor Mr. COUTURIER as much as possible.

Moreover, it forced us to create a real team work.

We did not forget our final goal : create an intelligent vocal assistant that is reliable and works perfectly regardless of the user's knowledge.

Conclusion

During this special year due to a lockdown that lasted most of the semester, we created a virtual assistant which seems to be operational and easy to use. We tried to implement as much as possible all of the functionalities that allows the user to ask VEEMbot any type of questions, from a simple calculation to a more complex closed or open ended question. Even though VEEMbot is answering any request of the user, there is still some improvements to implement so that it would be fully functional.

The current lockdown forced us to word remotely, without any social interaction and possibility to see each other, but created a necessary team work. We also had to put in place a better organization than before by doing more meetings to know who is doing what and to communicate in a more efficient way by using some communication tools.

Moreover, the development of such assistant involves the understanding of all the libraries and imports that allows VEEMbot to understand and answer a given request. Of course, we shared all our knowledge and work to each member of the group. We created small groups of two students in order to minimize the time to develop functionalities and to create a real team work.

We are convinced that VEEMbot could be useful for supermarkets, if a user interface and some small improvements (e.g. : time needed to load the model, detection of keywords) could be done. A simple improvement would for instance be the addition of the supermarket database where all the products are referenced so that the user could ask a request about both products and general questions.

List of figures

References

[1] Understand how NLP Tranfsormers Pipeline works, “TowadsDataScience” : https://towardsdatascience.com/what-are-transformers-and-how-can-you-use-them-f7ccd546071a

[2] Question Answering Transformers, “Huggingface”: https://huggingface.co/transformers/usage.html

[3] Find Stop Words in a text, “PyPI rake-nltk“ : https://pypi.org/project/rake-nltk/

[4] Retrieve text from Wikipedia, “PyPI wikipedia-api” : https://pypi.org/project/Wikipedia-API/

[5] BoolQ Dataset, “GitHub Google Research Datasets” : https://github.com/google-research-datasets/boolean-questions

[6] Translation of the question (first library), “PyPI langdetect” : https://pypi.org/project/langdetect/

[7] Translation of the question (final library), “Translating Strings in Python with TextBlob” : https://stackabuse.com/translating-strings-in-python-with-textblob/

[8] NLTK, “Natural Language Toolkit” : https://www.nltk.org

[9] Get the keyword and remove stopwords, “PyPI stopwords“ : https://pypi.org/project/stopwords/

[10] Search for more information, “PyPI googlesearch-python“ : https://pypi.org/project/googlesearch-python/

[11] Speech to text and text to speech, “PyPI SpeechRecognition“ : https://pypi.org/project/SpeechRecognition/

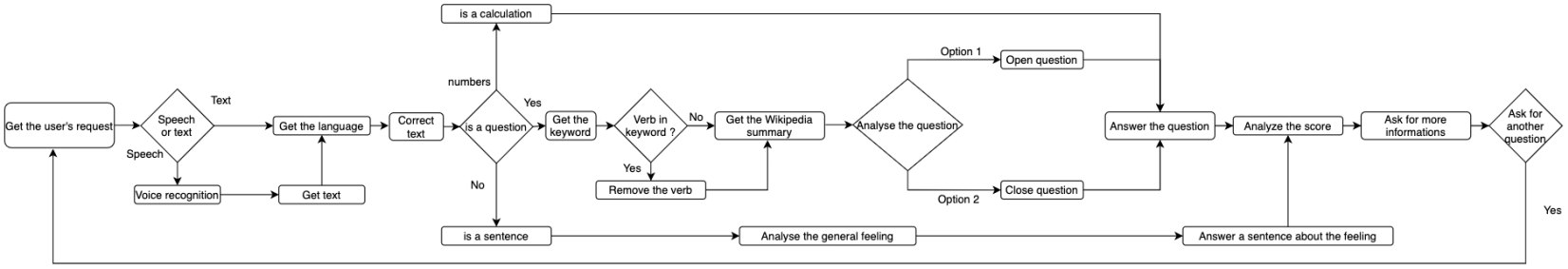


Figure 18: Software diagram

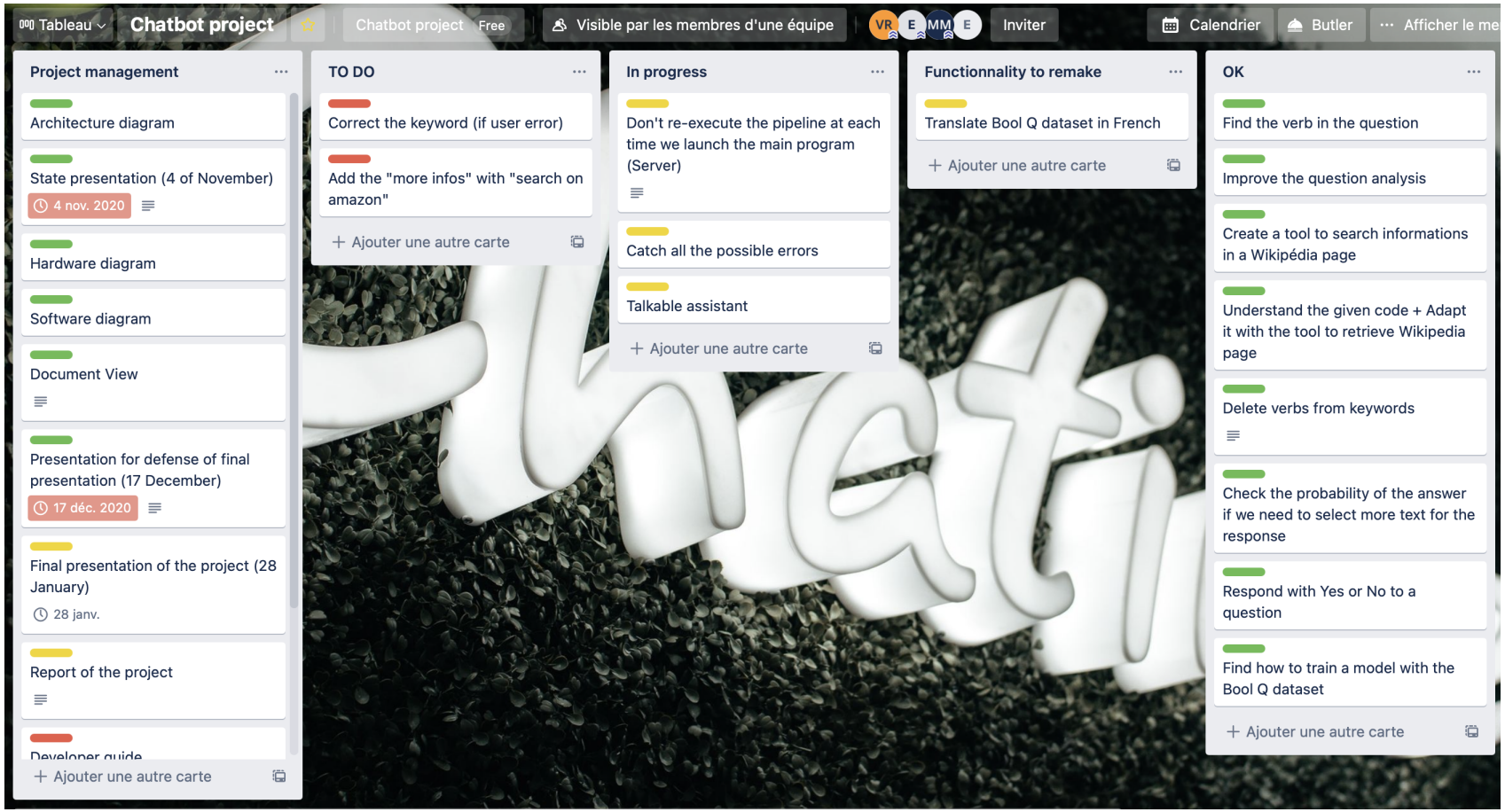


Figure 19: Trello at the end of the project

Réunion 3 avec prof

Open-ended question

- Ne pas ré-exécuter le pipeline à chaque fois → Sauvegarder le modèle.
- Si la proba du résultat est inférieure à x%, sélectionner plus de texte.
- Supprimer les verbes pour avoir une meilleur sélection des keywords.
- Corriger une erreur et demander à l'utilisateur si c'est ce qu'il voulait dire.
- Si l'erreur est similaire à une déjà vue, il retourne le premier lien ?

Close-ended question :

- Get bool Q dataset
- Translate some questions and responses with Google Trad
- Find how to train a model with this values

TO DO

yes-no question on the software diagram

Répartition des slides de la présentation :

Mohamed : introduction
Elio : présentation
Vincent : organisation sauf HW diagram
Mohamed : HW diagram
Enguerrand : Work done
Mohamed : conclusion

To ask to the tutor :

- global objective of our project : build a chatbot which can answer both yes and no questions and open questions or build a project with two different projects ?
- HW, SW and functional diagram (= diagramme de classe ?) is an architecture diagram ?
- Le ppt avec présentation orale ou description écrite ?

Figure 20: Report of a meeting