

PROJET EN SYSTÈME EXPERT:

Chatbot intelligent

**pour aider à prendre des décisions au niveau
éducatif**

Réalisé par :

- Ziad OUKHOYA
- Saad MBARKI
- Rachid IZEM
- Mouslim OUQASO

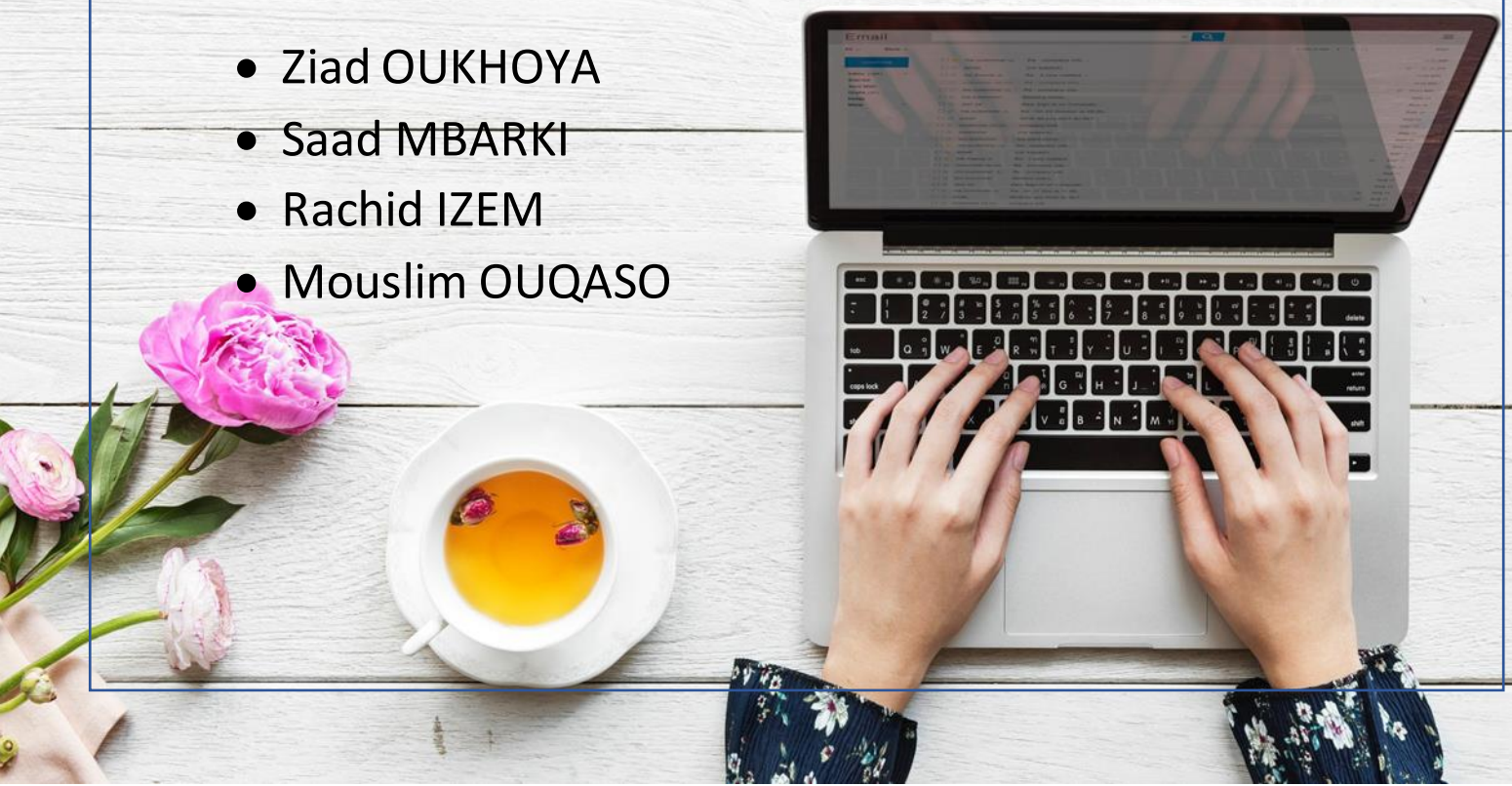


TABLE OF CONTENTS

Introduction	3
Analyse de L'EXISTANT :	4
Analyse du BESOIN :	4
conception du solution :	4
Réalisation du chatbot :	5
Conclusion.....	10



INTRODUCTION

Un chatbot aussi nommé dialogueur ou agent conversationnel, est un agent logiciel qui dialogue avec un utilisateur par le biais d'un service de conversations automatisées.

Le chatbot utilisé à l'origine des bibliothèques de questions et réponses, mais les progrès de l'intelligence artificielle lui permettent de plus en plus "d'analyser" et "comprendre" les messages par le biais des technologies de traitement du langage naturel (NLP) et d'être doté de capacités d'apprentissage liées au machine learning.

Donc, on va créer un chatbot pour aider les étudiants à prendre une décision claire pour continuer leur étude en se basant sur leur réponse à des questions précises.

ANALYSE DE L'EXISTANT :

De nos jours, il existe une masse d'étudiants déprimés au cycle supérieur, ceci peut conduire à abandonner leur étude ou recommencer leur cycle des le début en perdant des années à cause d'un mauvais choix du domaine ou du filière d'étude.

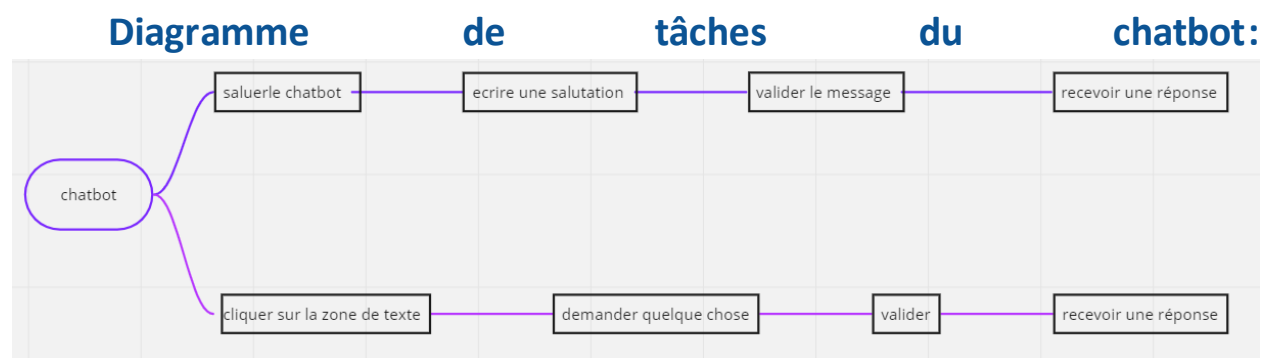
Ce problème est causé par la diversité des domaines et les filières de chaque domaine avec beaucoup d'information sur chaque filiere, ce qui rendre la tâche du choix du filiere ou du domaine plus difficile. D'où, l'étudiant ne pourrait pas prendre la bonne décision

ANALYSE DU BESOIN :

En se basant sur l'analyse de l'existant, on peut définir les besoins des étudiants dans ce qui suit :

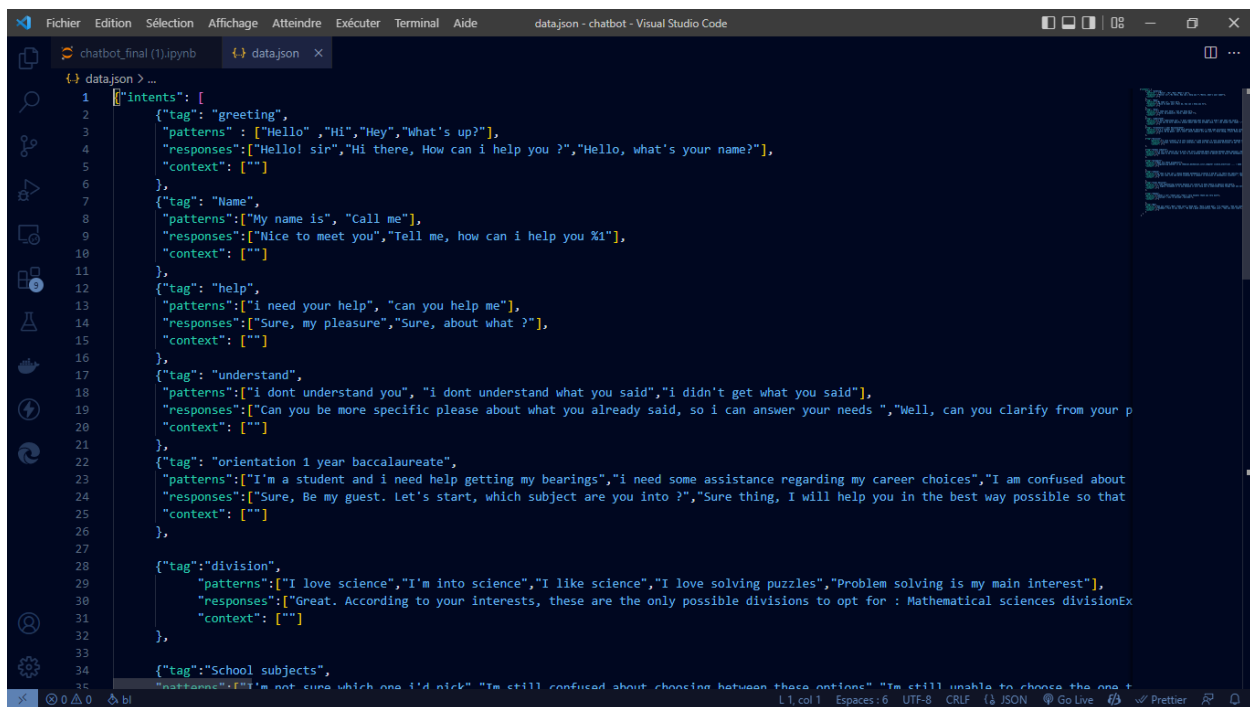
- Des recommandations claires
- Gagner du temps
- Offrir un accompagnement personnalisé.

CONCEPTION DU SOLUTION :



Base de connaissance du chatbot

Pour fonctionner de manière optimale, un Chatbot a besoin d'une technologie dite de back-end. La plupart du temps, une base de connaissance est idéale pour faire office de back-end. Le Chatbot permet ainsi de faire la liaison entre l'utilisateur et la base de connaissance. Les bases de connaissances sont en effet difficiles à appréhender pour un utilisateur non-informaticien. Réaliser un Chatbot relié à une base de connaissance permet donc de rendre accessible l'expertise d'un domaine à tous.



```
1  {"intents": [
2    {
3      "tag": "greeting",
4      "patterns": ["Hello", "Hi", "Hey", "What's up?"],
5      "responses": ["Hello! sir", "Hi there, How can i help you ?", "Hello, what's your name?"],
6      "context": [""]
7    },
8    {
9      "tag": "Name",
10     "patterns": ["My name is", "Call me"],
11     "responses": ["Nice to meet you", "Tell me, how can i help you %1"],
12     "context": [""]
13   },
14   {
15     "tag": "help",
16     "patterns": ["i need your help", "can you help me"],
17     "responses": ["Sure, my pleasure", "Sure, about what ?"],
18     "context": [""]
19   },
20   {
21     "tag": "understand",
22     "patterns": ["i dont understand you", "i dont understand what you said", "i didn't get what you said"],
23     "responses": ["Can you be more specific please about what you already said, so i can answer your needs ", "Well, can you clarify from your p"],
24     "context": [""]
25   },
26   {
27     "tag": "orientation 1 year baccalaureate",
28     "patterns": ["I'm a student and i need help getting my bearings", "i need some assistance regarding my career choices", "I am confused about"],
29     "responses": ["Sure, Be my guest. Let's start, which subject are you into ?", "Sure thing, I will help you in the best way possible so that"],
30     "context": [""]
31   },
32   {
33     "tag": "division",
34     "patterns": ["I love science", "I'm into science", "I like science", "I love solving puzzles", "Problem solving is my main interest"],
35     "responses": ["Great. According to your interests, these are the only possible divisions to opt for : Mathematical sciences divisionEx"],
36     "context": [""]
37   },
38   {
39     "tag": "School subjects",
40     "patterns": ["I'm not sure which one i'd pick", "I'm still confused about choosing between these options", "I'm still unable to choose the one t"],
41     "context": [""]
42   }
43 ]}
```

RÉALISATION DU CHATBOT :

1. Langage de programmation utiliser

- **Python** : c'est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux

développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.

- **JSON** (JavaScript Objet Notation) est un langage léger d'échange de données textuelles. Pour les ordinateurs, ce format se génère et s'analyse facilement. Pour les humains, il est pratique à écrire et à lire grâce à une syntaxe simple et à une structure en arborescence. JSON permet de représenter des données structurées (comme XML par exemple).

2. Outil de développement

- **Visual Studio Code** est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git.

3. Bibliothèques utiliser :

- **TFLearn** est une bibliothèque d'apprentissage en profondeur modulaire et transparente construite sur Tensorflow qui fournit une API de plus haut niveau à TensorFlow afin de faciliter et d'accélérer les expérimentations, tout en restant totalement transparente et compatible avec elle.
- **TensorFlow** est une bibliothèque open source, permettant d'exécuter des applications de machine learning et de deep learning. Cet outil dédié à l'apprentissage automatique a été développé par Google, et est fortement utilisé dans le domaine de l'intelligence artificielle (IA).

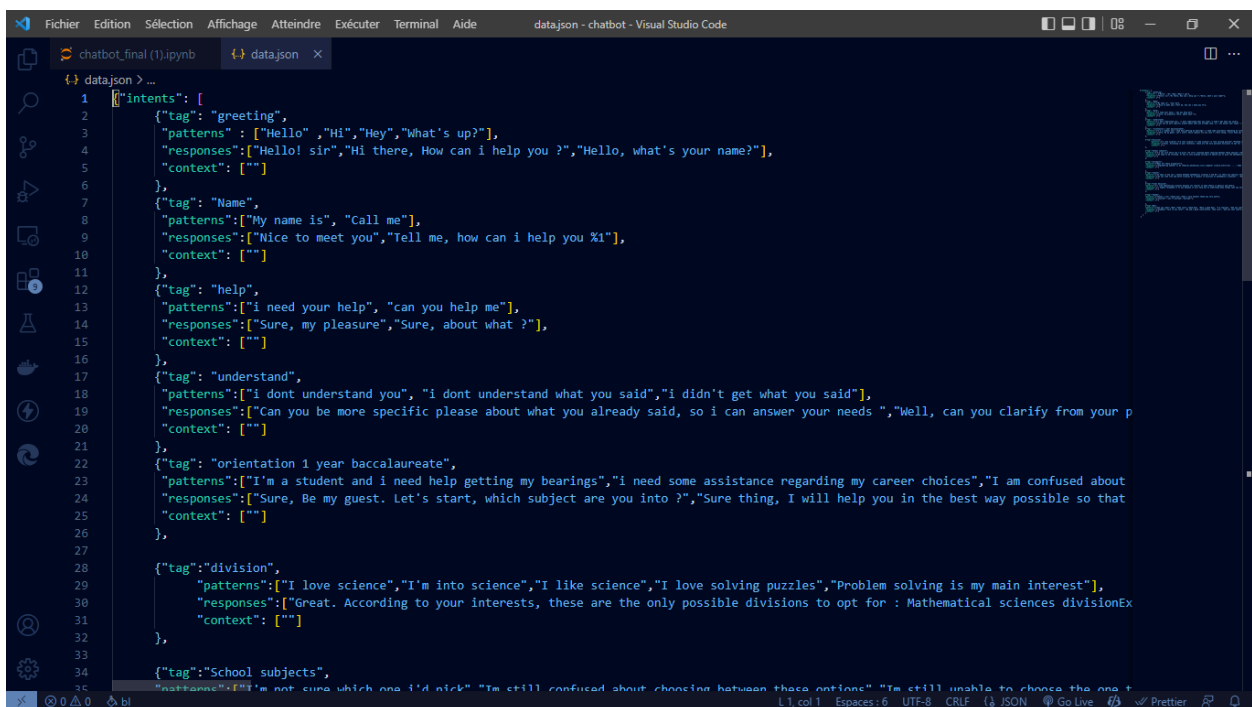
- **NLTK**, ou Natural Language Toolkit, est une suite de bibliothèques logicielles et de programmes. Elle est conçue pour le traitement naturel symbolique et statistique du langage anglais en langage Python.
- **NumPy** est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

4.Étape de réalisation :

Importer le fichier de données json :

```
import json
with open('data.json') as json_data:
    intents = json.load(json_data)
```

Contenus du fichier data.json :



```
datajson > ...
1 [{"intents": [
2   {"tag": "greeting",
3     "patterns": ["Hello", "Hi", "Hey", "What's up?"],
4     "responses": ["Hello! sir", "Hi there, How can i help you ?", "Hello, what's your name?"],
5     "context": [""]}
6   },
7   {"tag": "Name",
8     "patterns": ["My name is", "Call me"],
9     "responses": ["Nice to meet you", "Tell me, how can i help you %1"],
10    "context": [""]}
11  },
12  {"tag": "help",
13    "patterns": ["I need your help", "can you help me"],
14    "responses": ["Sure, my pleasure", "Sure, about what ?"],
15    "context": [""]}
16  },
17  {"tag": "understand",
18    "patterns": ["I dont understand you", "I dont understand what you said", "I didn't get what you said"],
19    "responses": ["Can you be more specific please about what you already said, so i can answer your needs ", "Well, can you clarify from your p"],
20    "context": [""]}
21  },
22  {"tag": "orientation 1 year baccalaureate",
23    "patterns": ["I'm a student and i need help getting my bearings", "I need some assistance regarding my career choices", "I am confused about"],
24    "responses": ["Sure, Be my guest. Let's start, which subject are you into ?", "Sure thing, I will help you in the best way possible so that"],
25    "context": [""]}
26  },
27  {"tag": "division",
28    "patterns": ["I love science", "I'm into science", "I like science", "I love solving puzzles", "Problem solving is my main interest"],
29    "responses": ["Great. According to your interests, these are the only possible divisions to opt for : Mathematical sciences divisionEx"],
30    "context": [""]}
31  },
32  {"tag": "School subjects",
33    "patterns": ["I'm not sure which one i'd pick", "I'm still confused about choosing between these options", "I'm still unable to choose the one t"],
34    "context": [""]}
35  ]
36  ]
```

Créer des données d'entraînement et de test :

```
training = []
output = []
output_empty = [0] * len(classes)

for doc in documents:
    bag = []
    pattern_words = doc[0]
    pattern_words = [stemmer.stem(word.lower()) for word in pattern_words]
    for w in words:
        bag.append(1) if w in pattern_words else bag.append(0)

    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1

    training.append([bag, output_row])

random.shuffle(training)
training = np.array(training)

train_x = list(training[:,0])
train_y = list(training[:,1])
```

Création du model :

```
# | neural network
net = tflearn.input_data(shape=[None, len(train_x[0])])
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, 8)
net = tflearn.fully_connected(net, len(train_y[0]), activation='softmax')
net = tflearn.regression(net)

# model
model = tflearn.DNN(net, tensorboard_dir='tflearn_logs')
# Start training (apply gradient descent algorithm)
model.fit(train_x, train_y, n_epoch=1000, batch_size=8, show_metric=True)
model.save('model.tflearn')
```


Classification des réponses :

```
ERROR_THRESHOLD = 0.25
def classify(sentence):
    # generate probabilities from the model
    results = model.predict([bow(sentence, words)])[0]
    # filter out predictions below a threshold
    results = [[i,r] for i,r in enumerate(results) if r>ERROR_THRESHOLD]
    # sort by strength of probability
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append((classes[r[0]], r[1]))
    # return tuple of intent and probability
    return return_list

def response(sentence, userID='123', show_details=False):
    results = classify(sentence)
    # if we have a classification then find the matching intent tag
    if results:
        # loop as long as there are matches to process
        while results:
            for i in intents['intents']:
                # find a tag matching the first result
                if i['tag'] == results[0][0]:
                    # a random response from the intent
                    return print(random.choice(i['responses']))
            results.pop(0)
```

Exécution du chatbot :

```
i=0
while i==0:
    inputt=input("user: ")
    if inputt!="":
        print('Agent: ',end="")
        response(inputt)
    else:
        i=1
```

```
user: hi
Agent: Hello, what's your name?
user: call me saad
Agent: Nice to meet you
user: i need your help
Agent: Sure, about what ?
user: i need some assistance regarding my career choices
Agent: What are you into?
```

CONCLUSION

Maintenant, l'implémentation de notre robot est prête, mais, on a constaté qu'il est essentiel de définir les bons cas d'usages.

Dans ce projet, nous avons compris les chatbots et implémenté une version d'apprentissage en profondeur d'un chatbot en Python qui est précis. Les chatbots sont utilisés partout et toutes les entreprises sont impatientes d'implémenter des bots dans leur flux de travail.

La raison qui explique notre choix d'un chatbot intelligent c'est d'aider les étudiants à prendre des bonnes décisions et choisir le bon chemin d'étude pour ne pas perdre beaucoup de temps et avoir des employer qui aiment leur travaille.