



Année 2020-2022

Certification Développeur en Intelligence Artificielle

*Projet chatbot pour Simplon
(projet E2)*

Sommaire

1.Contexte.....	P3
2.Déroulement du projet.....	P3
3.Nouvelle Modélisation.....	P7
4.Conclusion.....	P9
5.Bibliographie.....	P10

1. Contexte du projet

Notre Organisme de formation Simplon, où nous passons actuellement notre certification, est sollicité tous les jours pour des questions au sujet de diverses formations. Ce travail prend du temps car il faut répondre à chacune des personnes. Les questions sont sensiblement les mêmes, les réponses facilement trouvables sur le site web, le but étant donc de ne plus avoir à répondre à toutes ces questions et d'automatiser tout cela.

La création d'un chatbot est donc primordiale pour ne plus perdre de temps et éviter d'encombrer l'équipe pédagogique.

Le chatbot sera créé à partir d'un modèle déjà existant, le but étant d'améliorer ce modèle. Les personnes souhaitant poser des questions ne seront plus obligées d'attendre une réponse par mail mais pourront passer par le chatbot en direct qui sera sur le Facebook du site internet de la formation.

2. Déroulement du projet

Tout d'abord un cahier des charges a été fait avec l'équipe pédagogique pour bien identifier la demande, le temps que peut économiser l'équipe pédagogique, le budget pour faire ce projet, pour avoir l'état des lieux de l'ancien chatbot.

Ensuite il fallait une bonne organisation pour organiser une équipe de 11 personnes. Nous avons d'abord défini un chef d'équipe (de projet) puis nous avons créé un Trello pour répartir les tâches de chacun :

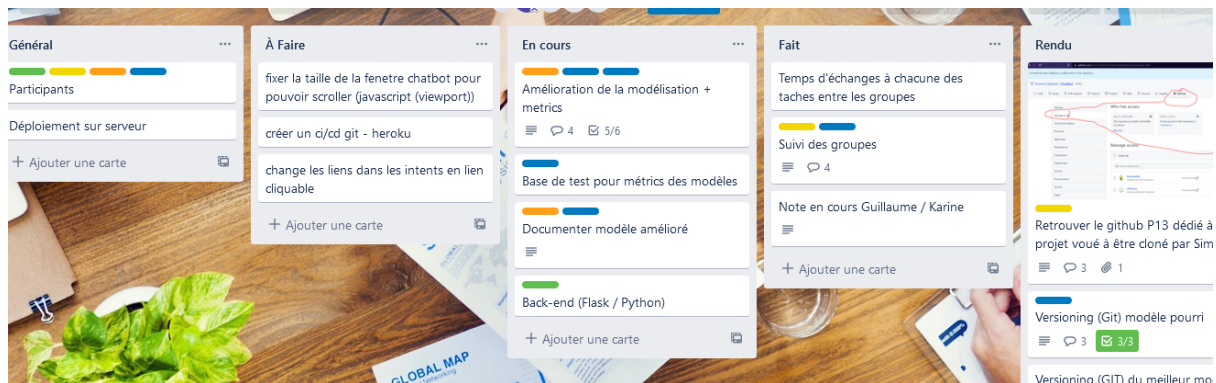


Figure 1 : Trello du groupe

Le chatbot de Simplon est un modèle qui à été abandonné et qui ne répond aux questions que par des liens qui renvoient sur le site internet Simplon. Et ce modèle n'est pas capable d'envoyer les intents au bon endroit.

Nous avons donc ensuite fait une réunion avec les 11 apprenants, pour définir toutes les tâches et se les partager.

A la suite de cela, nous avons une meilleur visibilité du Reste à Faire :

- Refaire les intents
- Améliorer le modèle
- Faire du scrapping pour récupérer les données en temps réel

Nous avons commencé par faire un Modèle Conceptuel des Données (MCD) puis un Modèle Logique des Données (MLD) et enfin un Modèle Physique des Données (MPD) :

Classes d'Entités:

SIMPLON (Key, Lieu, Debut, Localisation, Duree, datelimit, methode, Paiement)

Figure 2 : MLD

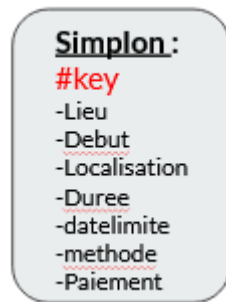


Figure 3 : MCD

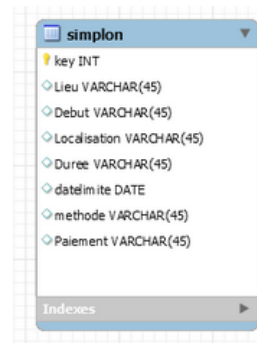


Figure 4 : MPD

Nous avons fait une maquette du front pour voir si l'aspect plaisait à notre référente pour le chatbot : Anaëlle Garcia.



Figure 5 : Maquette du chatbot

Quand elle nous a donné son accord, on est donc passé à la partie Front-end, Back-end. En parallèle de cela, nous avons aussi recréé les intents ainsi que créé un nouveau modèle plus performant.

Voici le front end que l'on a déployé :



Figure 6 : Front-end du chatbot

Et voici le fichier json :

```
{
  "intents": [
    {
      "tag": "contact",
      "patterns": ["Je peux téléphoner ?", "numéro téléphone", "appeler", "appeller", "appel", "apel", "apelle", "comment avoir quelqu'un"],
      "responses": ["Nous privilégions les échanges par mail, n'hésitez pas à nous joindre via le formulaire de contact : https://simplon.co"],
      "context": [""]
    },
    {
      "tag": "Mode",
      "patterns": ["Distanciel ou présentiel ?", "Les formations sont elles disponibles en distanciel ?", "Les formations sont elles dis"],
      "responses": ["Les formations peuvent être en distanciel ou présentiel selon les cas. Cette information est disponible dans l'inti"],
      "context": [""]
    },
    {
      "tag": "Prix",
      "patterns": ["Quel est le prix des formation ?", "Prix ? ", "Combien coûte vos formations ?", "Quel charge pour vos formations ?", "C"],
      "responses": ["Les formations dispensées par Simplon sont entièrement gratuites pour les apprenant.e.s car elles sont financées pa"],
      "context": [""]
    },
    {
      "tag": "Salutations",
      "patterns": ["Bonjour", "Hello", "Hi", "Salut", "Greeting", "slt", "bjr", "bonsoir"],
      "responses": ["Bonjour, comment puis-je vous aider ?"],
      "context": [""]
    },
    {
      "tag": "Formation",
      "patterns": ["formation", "Qu'est ce que vous proposez comme formation ?", "Quelle type vous faites ?", "numerique", "informatique"],
      "responses": ["Pour toute question concernant les formations à venir, merci de consulter la page des formations : https://auvergne"],
      "context": [""]
    },
    {
      "tag": "Identite",
      "patterns": ["Qu'est-ce-que simplon ?", "Simplon ?", "Puis-je avoir des informations sur Simplon ?", "c'est quoi simplon ?", "Qui à"],
      "responses": ["Vous trouverez toutes les informations sur Simplon ici : https://simplon.co/a-propos.html"],
      "context": [""]
    },
    {
      "tag": "none",
      "patterns": [""],
      "responses": ["Je n'ai pas compris votre question, voulez-vous des informations sur simplon ? Nos formations ?"],
      "context": [""]
    },
    {
      "tag": "Date",
      "patterns": ["Date", "période", "janvier", "fevrier", "mars", "avril", "mai", "juin", "juillet", "aout", "septembre", "octobre", "novembre", ""],
      "responses": ["nik ta mere ?"],
      "context": [""]
    }
  ]
}
```

Figure 7 : Fichier json

Ensuite il fallait réfléchir à un moyen de l'implémenter sur la page Facebook de Simplon. Nous avons donc pensé à un déploiement et à un lien cliquable qui mènerait au chatbot. Puis nous l'avons déployé sur Heroku via Flask.

Une fois que l'on s'était assuré que le modèle trouvait les intents liés aux questions posées, il a fallu donc remplacer au sein des réponses du chatbot les intents trouvés par le model par des requêtes SQL pour que l'information demandée soit récupérée au bon endroit.

Pour ce faire lors de la réponse, nous envoyons une requête SQL à la place de l'intent demandé :

```
for i in list_of_intents:
    if(i['tag'] == tag):
        if i['tag'] == 'Formation':
            result = ("Les formations disponible à Simplon dans la région du Rhône sont : " +
        else:
            result = random.choice(i['responses'])
        break
return result
```

Figure 8 : Exemple de code qui remplace l'intent par une requête

Nous avons également rajouté des techniques de Natural language Processing (NLP) pour que le modèle soit plus précis. Nous avons utilisé :

- la lemmatisation : La **lemmatisation** désigne un traitement lexical apporté à un texte en vue de son classement dans un index ou de son analyse. Ce traitement consiste à appliquer aux occurrences des lexèmes sujets à flexion (en français, verbes, substantifs, adjectifs) un codage renvoyant à leur entrée lexicale commune (« forme canonique » enregistrée dans les dictionnaires de la langue, le plus couramment), que l'on désigne sous le terme de *lemme*. ex : étoiles devient étoile, luisent devient luire
- La suppression des stopwords : lors du traitement des données on enlève les mots communs. ex : le, la, on, etc...
- Du bag of word : c'est-à-dire qu'on va vérifier chaque élément de la phrase pour savoir si le mot est présent dans la phrase.

Exemple de phrase:

"Bonjour comment allez vous ?"

Les mots sont respectivement aux index 0,4,86,88.

La phrase ressemble donc à ceci :

[1,0,0,0,1,00,1,0,1]

- Spellchecker : Librairie permettant de corriger les fautes d'orthographe d'un mot, ex : bonjour devient bonjour.
- Similarity : trouve les mots de vecteur similaire en fonction d'un dictionnaire.

3.Nouvelle Modélisation

Pour le nouveau modèle utilisé nous avons configuré les paramètres comme suit :

- Donnée d'entrée : Nos vecteurs bag of word de taille 87 (ajouter les neurones code)
- Sortie : Nos vecteurs de classes
- Optimizer : Stochastic Gradient Descent
- Loss Function : Categorical Cross Entropy
- Type de modèle :

Le modèle est un simple ANN séquentiel composé de couches Denses (Fully connected), de Drop Out et d'une couche softmax en sortie.

Les couches de Drop Out permettent de parer à l'Overfitting.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	11264
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 7)	455

```

=====
Total params: 19,975
Trainable params: 19,975
Non-trainable params: 0
=====

```

Figure 9 : Summary du nouveau modèle

Pour récapituler voici un schéma de ce qui à été fait par rapport à l'ancien modèle.

✕ : Amélioration apportée □ : Donnée / variable □ : Processus

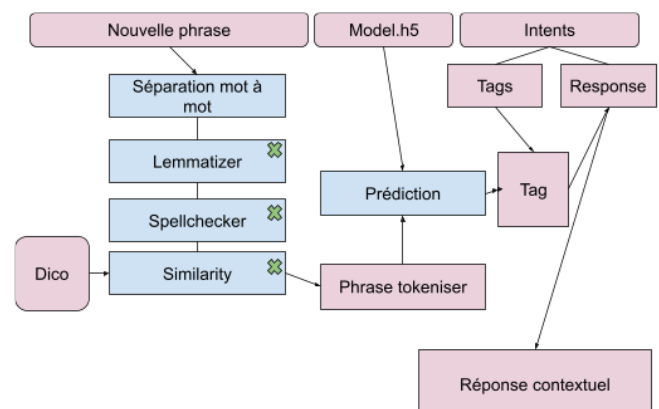
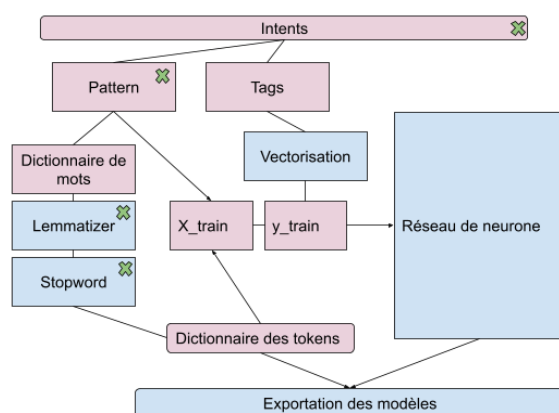


Figure 10 : Schéma-1 du nouveau Chatbot

Figure 11 : Schéma-2 du nouveau Chatbot

Avec les améliorations, nous pouvons remarquer que le modèle prédit mieux les intents. En effet, nous sommes passés d'une probabilité de 0.78 à 0.99.

[illegible]

Figure 12 : Ancienne probabilité

```
[30] predict_class("Coucou !",model = model)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.p
[{'intent': 'Salutations', 'probability': '0.99650973'}]
```

Figure 13 : Nouvelle probabilité

1	Mode	Le distanciel est il possible ?	Mode	Mode
2	Prix	Quel sont vos tarif	Prix	Prix
3	Salutations	Bonjour	Salutations	Salutations
4	Formation	Quel education proposé vous	Identite	Formation
5	Identite	Qui es tu ?	contact	contact
6	none	chien chat emma	none	none
7	contact	Quel est votre numéro de téléphone ?	contact	contact
8	Mode	Faut il être en présentiel ?	Mode	Mode
9	Prix	Combien coûte une formation ?	Prix	Prix
10	Salutations	Bonsoir, comment allez vous ?	Identite	Salutations
11	Formation	Quel cycle proposez vous ?	Identite	Identite
12	Identite	Simplon c'est quoi ?	Identite	Identite
13	none	J'aime manger du chien au curry	none	contact
14	contact	existe une adresse mail ?	contact	contact
15	Mode	Puis-je faire la formation en ligne ?	Prix	Prix
16	Prix	la formation est-elle gratuite ?	Prix	Prix
17	Salutations	Coucou !	none	Salutations
18	Formation	avez vous une formation disponible sur Lyon ?	Formation	Formation
19	Identite	Salut t'es qui ?	contact	contact
20	none	afafafaegerzhfd	none	none

Figure 14 : Comparaison de la détection des intents. Colonne 1 : Intent d'origine
Colonne 2 : Question posé, Colonne 3 : détection des intents de l'ancien modèle,
Colonne 4 : détection des intents du nouveau modèle

4.CONCLUSION

Finalement, ce sont tous les changements décrits ci-dessus qui ont fait que nous avons maintenant un chatbot qui comprend beaucoup mieux les questions et qui sait mieux répondre et orienter les futurs apprenants.

Nous le savons, Simpion était un organisme social et solidaire, nous pouvions avoir des questions de personnes étrangères ou n'écrivant pas bien le français. Notre chatbot prend maintenant en compte cette spécificité et peut s'adapter à la compréhension d'un français approximatif.

Autres exemples avec une faute d'orthographe:

Avant:

[illegible]

Figure 15 : Ancien modèle prédiction

Après:

```
] predict_class("Bonjour",model = model)

[{'intent': 'Salutations', 'probability': '0.99650973'}]
```

Figure 16 : Nouveau modèle prédiction

Pour rappel, la probabilité que notre modèle identifie le type de question est passé de 79% à 99% de chance, ce qui est une nette amélioration.

5.BIBLIOGRAPHIE

- (1)<https://www.delftstack.com/fr/howto/python/python-spell-checker/#correcteur-orthographique-avec-la-biblioth%C3%A8que-pyspellchecker-en-python>
- (2)<https://datascientest.com/spacy>
- (3)<https://www.analyticsvidhya.com/blog/2021/06/build-your-own-ai-chatbot-from-scratch/>
- (4)<https://towardsdatascience.com/a-simple-chatbot-in-python-with-deep-learning-3e8669997758>
- (5)<https://openclassrooms.com/fr/courses/4470541-analysez-vos-donnees-textuelles/4854971-nettoyez-et-normalisez-les-donnees>
- (6)<https://www.geeksforgeeks.org/python-word-similarity-using-spacy/>
- (7)<https://www.datacorner.fr/bag-of-words/>
- (8)https://fr.wikipedia.org/wiki/Mot_vide#:~:text=En%20recherche%20d'information%2C%20un,du%20%C2%BB%2C%20%C2%AB%20ce%20%C2%BB%E2%80%A6
- (9)<https://towardsdatascience.com/how-to-implement-seq2seq-lstm-model-in-keras-shortcutnlp-6f355f3e5639>