ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

UNIVERSITÉ DU QUÉBEC

RAPPORT TECHNIQUE
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
DANS LE CADRE DU PROJET DE FIN D'ÉTUDES
DU BACCALAURÉAT EN GÉNIE ÉLECTRIQUE

CONCEPTION DE PROCESSUS D'ANALYSE TEXTUELLE PERMETTANT DE DÉTERMINER LE NIVEAU DE VALIDITÉ D'ARTICLES

PAR
JAIDI BADR
JONATHAN BOUDREAU
NICOLAS CLERMONT

MONTRÉAL, LE 12 AVRIL 2021

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

MISE EN CONTEXTE DE LA PROBLÉMATIQUE ET DES OBJECTIFS VISÉS

Au cours des dernières années, internet a connu un essor de popularité via les réseaux sociaux. Avec l'avantage de contacts sociaux et d'échanges d'informations instantanés, viennent aussi ses désavantages. En effet, les fausses nouvelles sont maintenant omniprésentes sur toutes les plateformes. Ceci constitue un réel problème puisque de plus en plus de gens n'effectuent pas leurs recherches en lien avec ces informations et les tiennent pour acquis.

Ce projet a comme objectif la conception d'une intelligence artificielle qui effectuera la lecture de texte et retournera un niveau de validité associé. Pour ce projet, nous avions comme contrainte d'utiliser la base de données fournie par le client.

Dans un premier temps, nous allons définir la méthodologie utilisée pour ce projet. Nous allons par la suite détailler chaque aspect technique du projet : l'architecture, les plans, le code, etc.. Dans un troisième temps, nous allons analyser les performances obtenues et en discuter. Nous allons conclure ce projet avec des recommandations et dans un dernier temps, nous donnerons une réflexion sur les aspects de développement durable.

CHAPITRE 1

REVUE DE LITTÉRATURE

Dans le cadre de ce projet, nous avons effectué la conception des algorithmes après une revue de littérature. Nous avons choisi de baser nos choix sur l'algorithme *FastText* de Facebook et sur une grille de décision selon Google.

1.1 Prétraitement

Tout d'abord, avant d'envoyer les articles dans les modèles, la première étape est d'effectuer un nettoyage des données afin de maximiser l'efficacité des algorithmes. Cette étape est le prétraitement des données et elle a pour but de préparer les données pour l'étape de l'entraînement et de test. Son utilisation permet aussi d'améliorer le temps d'entraînement et de test des différents modèles.

Plusieurs différentes fonctions de nettoyage existent déjà dans l'analyse textuelle. Une des fonctions qu'on retrouve le plus fréquemment est la suppression de différents types de données. Les données subjectives à être supprimé dépendent de l'auteur qui trouve qu'un type de donnée est considéré comme du bruit dans l'entraînement de ses modèles. Dans notre cas, nous avons supprimé la ponctuation et les mots vides des articles puisqu'on considère qu'ils n'ont pas de valeurs dans nos algorithmes.

Le deuxième type de fonction de nettoyage est la modification de données afin de normaliser les articles en un gabarit pour tous les articles. Par exemple, dans notre cas, nous avons modifié les caractères majuscules en minuscules. De plus, afin de regrouper les données qui se ressemblent, par exemple, des verbes qui sont conjugués différemment, mais, qu'ils ont une terminaison différente. On peut modifier la terminaison des mots afin de récupérer seulement la partie importante du mot tout en gardant son sens. Différents types de coupures existent tels que la lemmatisation et l'enracinement (*stemming*). La dernière fonction qui est fréquemment utilisée se nomme *tokenization* et a pour but de segmenter les données en jeton. C'est-à-dire d'enlever la structure des phrases en attribuant une position à chacune des données dans un tableau.

1.2 FastText

Comme décrit dans le papier *Bag of Tricks for Efficient Text Classification*[1], la classification de texte est une tâche importante qui regroupe diverse application comme la recherche sur internet, la recherche d'information, la classification de document, etc. Les classificateurs linéaires sont souvent considérés comme une bonne base puisqu'ils sont simples et offrent une excellente performance.

FastText utilise le *bag of word* et le N-gram afin de capturer plus d'informations. Ils sont transformés en vecteurs et utilisés comme entrée dans un réseau de neurones. La régression logistique multinomiale est utilisée. Sur une large base de données, cette méthode offre une vitesse beaucoup plus rapide.

1.3 Grille de décision de Google

Puisque Google est un géant dans ce domaine, nous avons choisi comme deuxième technique d'utiliser cette grille de décision[2].

Avant de pouvoir utiliser cette grille, nous devons avoir déjà rassemblé notre base de données et bien en comprendre les caractéristiques. Ensuite, la première étape consiste à calculer la moyenne de nombre de mots par article. Comme montré dans le rapport précédent, nous avions un ratio moyen de 375 mots par article. Puisque notre moyenne est en dessous de 1500, selon cette grille, nous devons diviser les échantillons en mots n-gram pour ensuite les convertir en vecteurs. Nous devons ensuite classer les vecteurs en importance. Ils seront ensuite utilisés dans un perceptron multicouche.

CHAPITRE 2

MÉTHODOLOGIE DE TRAVAIL

Dans le cadre de ce projet de fin d'études, nous avons suivi une méthodologie de travail rigoureuse. La première étape consistait en la revue de littérature. Effectivement, suite à nos diverses recherches, nous sommes tombées sur les résultats obtenus par 2 géants dans le domaine, c'est-à-dire Facebook et Google. Nous avons basé nos choix de modèles selon FastText de Facebook et une grille de décision selon Google. Ceci nous a amenés à choisir la méthode de régression logistique ainsi que celle du perceptron multicouche.

Pour la réalisation du modèle de régression logistique selon FastText, nous avons utilisé la librairie de classification offerte par Facebook, en suivant les instructions de celles-ci. Il est possible de téléverser le code et de l'installer pour utilisation avec Python.

Pour la réalisation du modèle de perceptron multicouche selon la grille de décision de Google, nous avons utilisé la librairie recommandée par Google, TensorFlow[3]. Cette librairie permet de créer des perceptrons multicouches et de changer les paramètres de celui-ci avec un grand niveau de contrôle et une facilité relative.

Lors de la phase de réalisation, nous avons suivi une méthodologie de travail que nous allons développer ici. Le logiciel de version décentralisé Git a été utilisé afin de gérer l'évolution du contenu au moyen d'une arborescence. Ceci a pu faciliter le travail collaboratif. Nous avions un outil qui a suivi chaque changement apporté au projet et nous a permis de retourner en arrière, quand il y a eu des problèmes. Les changements apportés par chaque personne ont donc pu être fusionnés en une seule source.

À cause du contexte actuel, le travail d'équipe en présentiel était impossible, nous avons donc utilisé un serveur dédié sur la plateforme Discord. Discord est un logiciel propriétaire gratuit de VoIP et de messagerie instantanée. Celui-ci nous a permis de communiquer verbalement et de partager votre écran, sans limites de temps.

Le langage de programmation pour ce projet a été la dernière version de Python à ce jour, soit la 3.9. La raison est simple, une sélection extraordinaire de librairies est disponible pour l'apprentissage machine. L'environnement de développement utilisé a été

Visual Studio Code. L'environnement est très simple à installer, gratuit et propose déjà une extension pour utiliser le langage Python.

Plusieurs librairies offertes ont été utilisées : Pandas, Numpy et Sklearn. Celles-ci ont facilité les tâches associées aux fichiers csv ainsi que la manipulation de données et calcul matriciel.

Nous avons utilisé le logiciel Microsoft Project pour garder à jour un diagramme de Gantt afin d'avoir un suivi de l'évolution du projet et des échéanciers.

CHAPITRE 3

DOCUMENTATION TECHNIQUE DU PROTOTYPEF

Dans le cadre de ce projet de fin d'études de conception d'une intelligence artificielle pour l'analyse textuelle, nous avons pu fournir un prototype viable et fonctionnel. Nous allons en décrire l'architecture, fournir le code implémenté, les tests nécessaires ainsi qu'une procédure d'utilisation.

3.1 Architecture

Le prototype comporte plusieurs sections. En effet, celui-ci comporte une section de traitement, qui prend en entrée les textes souhaités, ainsi que les modèles utilisés, qui s'entraînent avec les données et fournit un résultat.

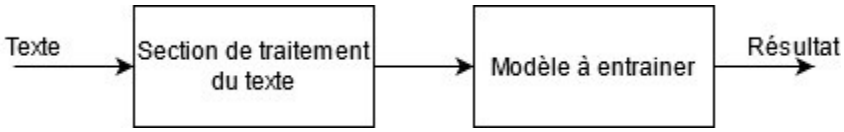


Figure 1 : Architecture générale du prototype

Cette architecture générale se décompose en plusieurs parties. Effectivement, la section de traitement reçoit les données en entrée et les organise en plusieurs fichiers individuels. Cette section prépare aussi le texte en supprimant les caractères inutiles du texte. Tels que la ponctuation, les caractères spéciaux, les mots de prépositions, etc. Le but du prétraitement est d'avoir un texte clair qui comporte seulement les mots nécessaires.

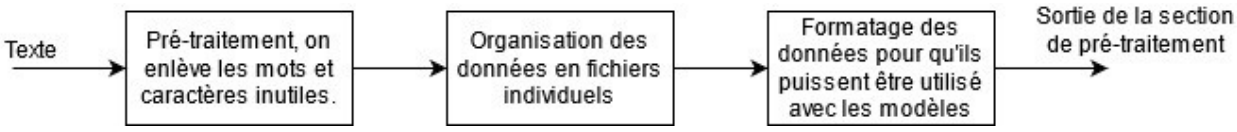


Figure 2 : Architecture fonctionnelle de la section de traitement

Une fois les données prétraitées, elles sont prêtes à être utilisées dans la prochaine section. Cette portion du prototype est celle qui reçoit en entrée les données filtrées et préparées pour ensuite entraîner les différents modèles utilisés. Nous entraînons ces modèles jusqu'à l'obtention du résultat souhaité fourni par les fonctions de test du modèle.

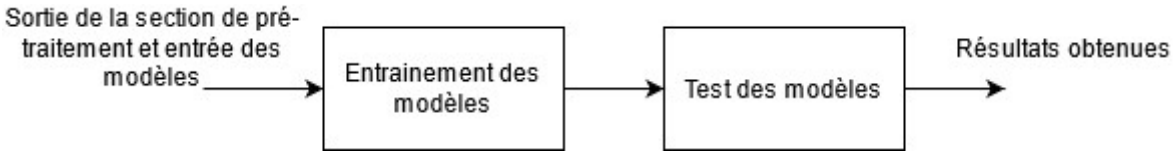


Figure 3 : Architecture fonctionnelle de la section des modèles, en groupe

En ce qui à trait aux modèles, puisque nous utilisons 2 modèles différents, aux fins de comparaison, nous effectuons l'entraînement des modèles de façon parallèle ainsi que les tests. Ceci, afin de n'utiliser qu'un seul modèle à la fois pour mieux contrôler les différents paramètres.

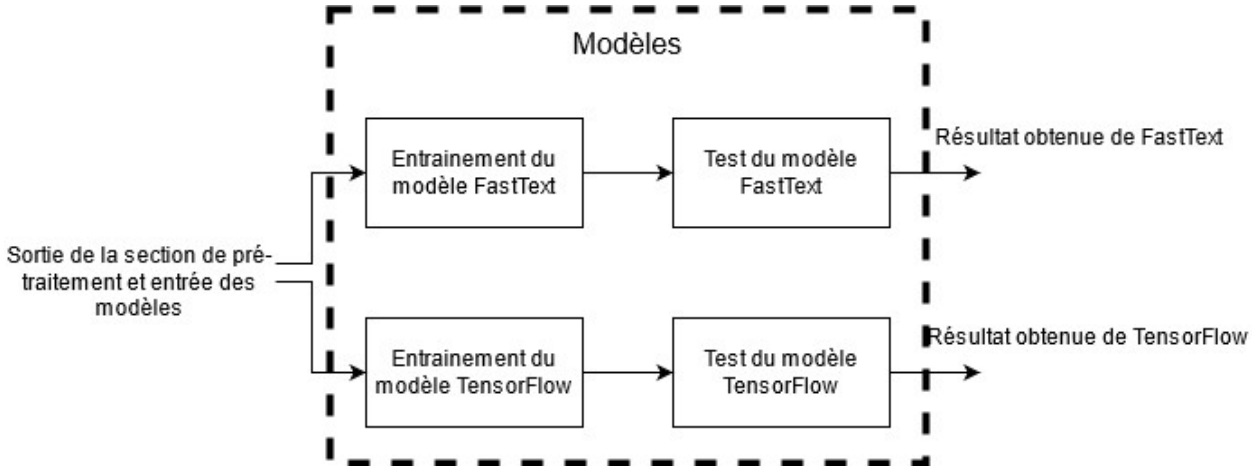


Figure 4 : Architecture fonctionnelle de la section des modèles, individuel

3.2 Code implémenté

Une classe principale se situe au milieu de notre projet, c'est-à-dire la classe Model. Celle-ci prend les données que l'on veut utiliser pour s'initialiser. Elle comporte des méthodes de prétraitement, de formatage, d'entraînement et de test. Ce fichier est utilisé pour les deux modèles, soit FastText et TensorFlow.

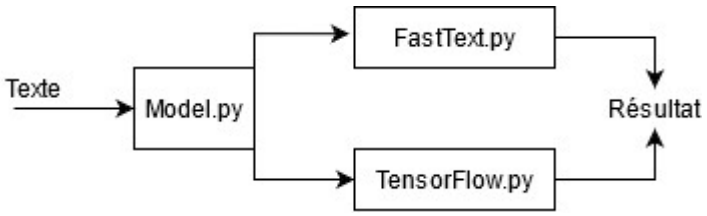


Figure 5 : Architecture générale des fichiers

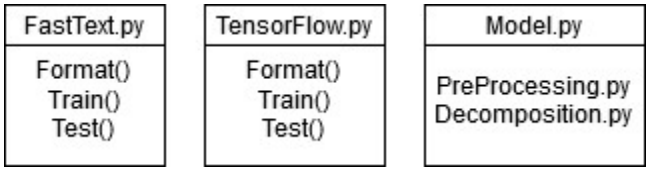


Figure 6 : Architecture fonctionnelle des fichiers

Le code est donc organisé comme on peut le voir dans les figures 5 et 6. Les actions générales feront partie de la classe Model.py, qui s'occupera de faire un formatage du code grâce à la fonction *Preprocess*(). Ensuite, les classes Fasttext.py et Tensorflow.py vont hériter de la classe Model.py. Ceux-ci auront les deux des fonctions *format*(), qui permettent de formater le texte sorti par *preprocess*() dans un format entrainable. Ensuite dans les mêmes classes, il y aura la fonction *train*() qui fera un entraînement avec le texte généré par *format*(). Et pour finir, il y aura la fonction *test*() qui testera les modèles respectifs de FastText et de TensorFlow. Nous allons décrire chaque fonctionnement général des fonctions et ensuite parler plus spécifiquement selon les modèles.

3.2.1 Fonction PreProcess()

La fonction *preprocess*() est similaire pour FastText et TensorFlow et est héritée à partir de la classe Model.py. La fonction sert à filtrer le texte et à appliquer des opérations qui permettent de réduire les éléments indésirables du texte. Les éléments à enlever sont: la ponctuation, les *stop word*, c'est-à-dire les mots qui n'apportent pas grand-chose au texte, par exemple les prépositions. Ensuite, l'opération de lemmatisation est appliquée. Cette opération permet de réduire les mots à des formes plus simples en conservant la racine, exemple, formaient devient forme. Cette fonction divise ensuite les données. Un cinquième des données ont été utilisé pour les tests et le restant était encore divisé en cinq parties, un cinquième pour la validation croisée et le restant pour l'entraînement.

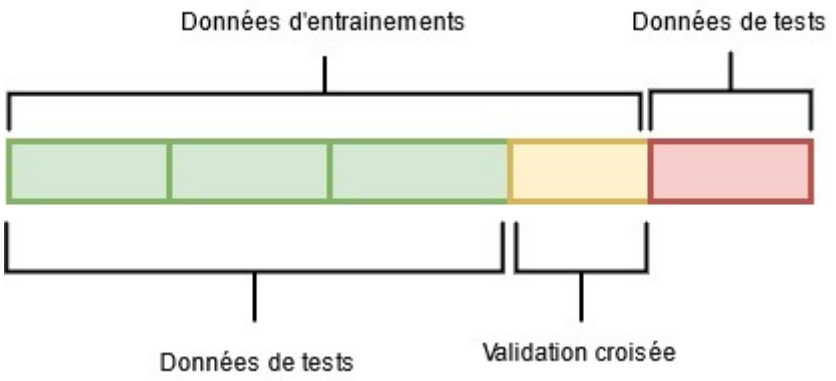


Figure 7 : Division des données

Ensuite, la méthode de division *K-fold* est utilisée pour séparer les données en 5 sets. Les modèles vont apprendre séparément sur chaque et le score final sera la moyenne obtenue des résultats de chaque set.

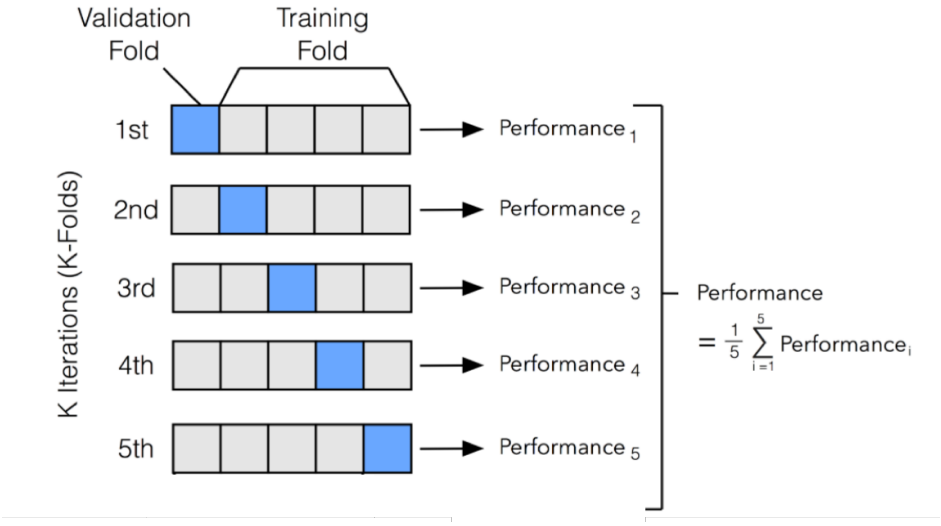


Figure 8 : Méthode K-Folds[4]

3.2.2 Fonction Format()

La fonction *format*() va mettre les données selon le bon format pour l'entraînement en fonction du modèle. Pour FastText, on veut des données dans le format suivant.

```
__label__true obama excellent health still use nicotine gum doctorwashington
__label__true trump national security adviser vow tackle north korea nuclear
__label__true factbox republican obamacare plan would repeal medicaid expansi
__label__true turkish police detain 25 suspect islamic state militant istanbu
__label__true china former top graftbuster warn plot seize powerbeijing reute
__label__true israel peace talk palestinian government reliant hamasjerusalem
__label__true trump national security aide flynn resign russian contactswashi
__label__true trump agree lawmaker immigration tweet white housewashington re
__label__true kirkuk declare curfew iraqi kurdish independence referendumkirk
__label__true rwanda charge critic president incite insurrectionkigali reuter
__label__true cyprus president seek second fiveyear term jan 18 votenicosia r
__label__true defiant u prosecutor fire trump administrationwashington reuter
__label__true pressure trump price resign health secretary private plane upro
__label__true exclusive trump target illegal immigrant give reprieve deportat
__label__true senator call panel investigate russian hackingwashington reuter
__label__true ugandan special force accuse eject mp parliamentkampala reuters
```

Figure 9 : Exemple de données pour FastText

Donc, on veut un article par ligne qui commence avec __label__ suivi du nom de la classe, soit *true* ou *fake* qui sont ensuite suivis du texte. Les données formatées de FastText sont enregistrées dans ./data/FastText/. Tensorflow quant à lui veut le format suivant.

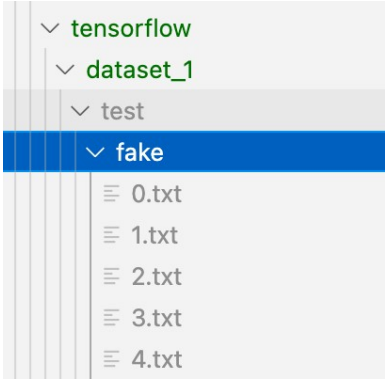


Figure 10 : Format de données pour Tensorflow

C'est-à-dire une architecture de fichier réparti selon le type des données, ensuite selon leur classe et pour terminer, elles sont divisées en fichiers textes. La fonction format permet donc de s'assurer que chaque librairie d'entraînement a les données dont elle a besoin pour l'entraînement. Les données formatées de Tensorflow sont enregistrées dans ../data/tensorflow/.

3.2.3 Fonction Train()

La fonction *train*() va entraîner les données générées grâce à *format*() et générer des modèles.

Pour FastText, on peut ajuster les paramètres suivants: le nombre d'époques d'entraînement et le taux d'apprentissage. Les modèles générés seront sauvegardés dans ../models/FastText/.

Pour TensorFlow, on peut ajuster les paramètres suivants: le nombre d'époques d'entraînement, le nombre de couches d'entraînement, le nombre de neurones à l'entrée et le nombre de neurones par couche cachée. Les modèles générés seront sauvegardés dans ../models/tensorflow/.

3.2.4 Fonction Test()

La fonction *test*() teste les modèles générés par la fonction *train*() et imprime la moyenne des résultats ce qui donne le score de chacun des modèles. Pour résumer, voilà un schéma décrivant le rôle respectif de chacune des fonctions.

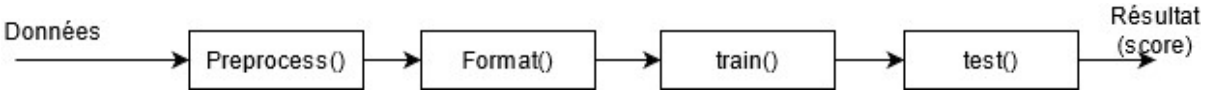


Figure 11 : Ordre d'utilisation des fonctions

Ainsi, après exécution du code, le répertoire du projet aura la structure suivante :

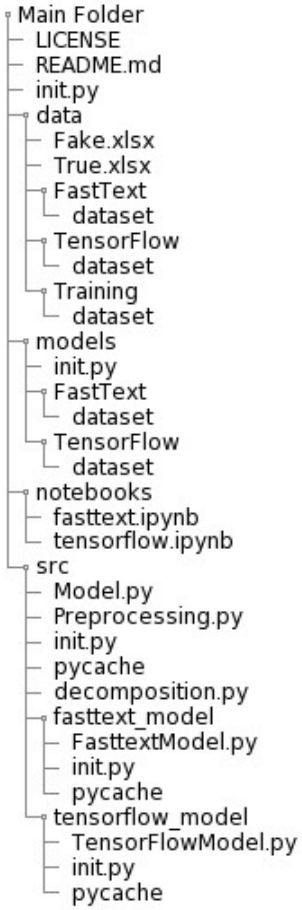


Figure 12 : Architecture des fichiers du projet

3.3 Tests nécessaires

Nous avons effectué plusieurs tests afin de vérifier la validité du projet. En effet, le tableau suivant décrit les paramètres utilisés pour chaque modèle ainsi que les paramètres de prétraitement.

Tableau 1 : Comparaison des scores pour les différents tests

	Fastext			TensoFlow		
Test	Précision	Paramètres	Précision	Exactitude	Paramètres	Preprocessing
1	65,58%	-Epochs : 100 -Learning-rate: 0.5	-	57,03%	-Epochs : 40 -Nb couches: 16 -Nb neurones: 10000 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words
2	65,40%	-Epochs : 100 -Learning-rate: 0.75	-	57,30%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 10000 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words
3	66,23%	-Epochs : 100 -Learning-rate: 0.25	-	56,56%	-Epochs : 40 -Nb couches: 8 -Nb neurones: 10000 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words
4	66,38%	-Epochs : 100 -Learning-rate: 0.125	-	52,75%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 20000 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words
5	66,41%	-Epochs : 1000 -Learning-rate: 0.125	-	64,08%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 5000 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words
6	66,49%	-Epochs : 100 -Learning-rate: 0.075	-	80,90%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 2500 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words

7	67,29%	-Epochs : 100 -Learning-rate: 0.0075	-	93,16%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 1250 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words
8	-	-	-	96,16%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 625 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words
9	-	-	-	95,81%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 300 -Vecteur: 250	-Ponctuation -Lemmatisation -Stop words
10	-	-	-	93,48%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 625 -Vecteur: 500	-Ponctuation -Lemmatisation -Stop words
11	-	-	99,28%	96,96%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 625 -Vecteur: 125	-Ponctuation -Lemmatisation -Stop words
12	80,87%	-Epochs : 100 -Learning-rate: 0.0075	98,62%	98,01%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 625 -Vecteur: 125	-Ponctuation -Lemmatisation
13	86,52%	-Epochs : 100 -Learning-rate: 0.0075	99,22%	97,96%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 625 -Vecteur: 125	-Ponctuation -Stop words
14	93,67%	-Epochs : 100 -Learning-rate: 0.0075	98,91%	98,69%	-Epochs : 40 -Nb couches: 32 -Nb neurones: 625 -Vecteur: 125	-Ponctuation

3.4 Procédure d'utilisation

Afin de bien utiliser le projet, plusieurs étapes sont nécessaires. En effet, nous allons les énumérer dans l'ordre ci-dessous. Avec l'aide de Jupyter, vous n'avez qu'à ouvrir les projets et utiliser ce dont vous avez besoin.

3.4.1 Étape 1 : préparer l'environnement

Afin de bien utiliser le projet, il faut installer les librairies et logiciels nécessaires (se référer à la section méthodologie de travail).

3.4.2 Étape 2 : obtenir la localisation du répertoire actuel

```
In [2]: import os, sys

In [3]: cwd = os.getcwd()
        cwd

Out[3]: '/Users/badr/Documents/ETS/Hiv2021/PFE/VS/git/fake-news-detection/notebooks'

In [4]: main_repository = os.path.join(cwd, os.pardir)

In [5]: sys.path.append(os.path.join(main_repository, 'src'))
```

Figure 13 : Obtention de la localisation du répertoire

3.4.3 Étape 3 : Créer l'objet

Créer l'objet selon le modèle choisi.

```
In [5]: from fasttext_model.FasttextModel import FasttextModel

In [6]: Fasttext = FasttextModel(main_repository, 5)
```

Figure 14 : Création de l'objet avec FastText

```
In [6]: from tensorflow_model.TensorflowModel import TensorflowModel

In [7]: Tensorflow = TensorflowModel(main_repository, 5)
```

Figure 15 : Création de l'objet avec TensorFlow

3.4.4 Étape 4 : préparer les données

Préparer les données selon le modèle choisi.

```
In [7]: Fasttext.preprocess()

  0%|          | 0/5 [00:00<?, ?it/s]/Users/badr/Documents/ETS/Hiv2021/PFE/V
S/git/fake-news-detection/notebooks/../src/Preprocessing.py:68: VisibleDepre
cationWarning: Creating an ndarray from ragged nested sequences (which is a
list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shape
s) is deprecated. If you meant to do this, you must specify 'dtype=object' w
hen creating the ndarray.
  return np.array([nltk.word_tokenize(corpus[i]) for i in range(len(corpus))
])
100%|██████████| 5/5 [07:21<00:00, 88.22s/it]

In [8]: Fasttext.format()

100%|██████████| 5/5 [00:26<00:00,  5.33s/it]
```

Figure 16 : Préparation des données avec FastText

```
In [ ]: Tensorflow.preprocess()

In [7]: Tensorflow.format()

100%|██████████| 5/5 [01:48<00:00, 21.67s/it]
```

Figure 17 : Préparation des données avec TensorFlow

3.4.5 Étape 5 : entraîner les modèles

Entraîner selon le modèle choisi.

```
In [9]: Fasttext.train(learning_rate=0.0075, epochs=100)

100%|██████████| 5/5 [06:29<00:00, 77.93s/it]
```

Figure 18 : Entraînement du modèle avec FastTest

```
In [21]: Tensorflow.train(embedding_dim = 32, max_features = 625, sequence_length =

  0%|          | 0/5 [00:00<?, ?it/s]
Found 28746 files belonging to 2 classes.
Found 7188 files belonging to 2 classes.
Epoch 1/40
899/899 [==============================] - 16s 17ms/step - loss: 0.5649 - bi
nary_accuracy: 0.7421 - val_loss: 0.4790 - val_binary_accuracy: 0.7038
```

Figure 19 : Entraînement du modèle avec TensorFlow

3.4.6 Étape 6 : tester les modèles

Tester selon le modèle choisi.

```
In [10]: Fasttext.test()

  0%|          | 0/5 [00:00<?, ?it/s]Warning : `load_model` does not return
WordVectorModel or SupervisedModel any more, but a `FastText` object which i
s very similar.
 20%|██        | 1/5 [00:01<00:05,  1.35s/it]Warning : `load_model` does not
return WordVectorModel or SupervisedModel any more, but a `FastText` object
which is very similar.
 40%|████      | 2/5 [00:02<00:03,  1.24s/it]Warning : `load_model` does not
return WordVectorModel or SupervisedModel any more, but a `FastText` object
which is very similar.
 60%|██████    | 3/5 [00:03<00:02,  1.19s/it]Warning : `load_model` does no
t return WordVectorModel or SupervisedModel any more, but a `FastText` objec
t which is very similar.
 80%|████████  | 4/5 [00:04<00:01,  1.14s/it]Warning : `load_model` does no
t return WordVectorModel or SupervisedModel any more, but a `FastText` objec
t which is very similar.
100%|██████████| 5/5 [00:05<00:00,  1.15s/it]

In [11]: Fasttext.precision

Out[11]: 0.9366859369745028
```

Figure 20 : Test du modèle FastText

```
In [8]: Tensorflow.test()

100%|██████████| 5/5 [16:21<00:00, 201.36s/it]100%|██████████| 5/5 [16:21<0
0:00, 196.37s/it]
  0%|          | 0/5 [00:00<?, ?it/s]
Found 8985 files belonging to 2 classes.
281/281 [==============================] - 8s 26ms/step - loss: 0.1145 - acc
uracy: 0.9677 - precision: 0.9950
 20%|██        | 1/5 [00:09<00:38,  9.73s/it]
Found 8985 files belonging to 2 classes.
281/281 [==============================] - 7s 24ms/step - loss: 0.1007 - acc
uracy: 0.9693 - precision_1: 0.9929
 40%|████      | 2/5 [00:18<00:27,  9.31s/it]
Found 8983 files belonging to 2 classes.
281/281 [==============================] - 7s 24ms/step - loss: 0.0933 - acc
uracy: 0.9724 - precision_2: 0.9932
 60%|██████    | 3/5 [00:27<00:18,  9.23s/it]
Found 8983 files belonging to 2 classes.
281/281 [==============================] - 8s 24ms/step - loss: 0.0922 - acc
uracy: 0.9699 - precision_3: 0.9890
 80%|████████  | 4/5 [00:36<00:09,  9.18s/it]
WARNING:tensorflow:5 out of the last 5 calls to <function recreate_function.
<locals>.restored_function_body at 0x7ff3215e4f70> triggered tf.function ret
racing. Tracing is expensive and the excessive number of tracings could be d
ue to (1) creating @tf.function repeatedly in a loop, (2) passing tensors wi
th different shapes, (3) passing Python objects instead of tensors. For (1),
please define your @tf.function outside of the loop. For (2), @tf.function h
as experimental_relax_shapes=True option that relaxes argument shapes that c
an avoid unnecessary retracing. For (3), please refer to https://www.tensorf
low.org/guide/function#controlling_retracing and https://www.tensorflow.org/
api_docs/python/tf/function for  more details.
Found 8983 files belonging to 2 classes.
281/281 [==============================] - 8s 24ms/step - loss: 0.0933 - acc
uracy: 0.9699 - precision_4: 0.9941
100%|██████████| 5/5 [00:46<00:00,  9.35s/it]

In [9]: Tensorflow.precision

Out[9]: 0.9928376197814941

In [10]: Tensorflow.accuracy

Out[10]: 0.9698569774627686
```

Figure 21 : Test du modèle TensorFlow

3.4.7 Étape 7 : tester sur un article quelconque

Une fois les modèles entraînés et testés avec un score satisfaisant, on peut tester ceux-ci sur un vrai article pour avoir une idée de la précision de celui-ci sur un article quelconque. En voici un exemple. L'article suivant[5] a été pris au hasard et sera testé comme un exemple d'un article vrai. L'article suivant[6] a aussi été pris au hasard et sera testé comme un exemple d'un article faux. On commence par définir le l'article qu'on veut prédire dans la variable *text*.

```
In [13]: text

Out[13]: ' Members of Congress may use campaign funds to hire bodyguards, FEC rules (
         CNN) — Members of Congress can use campaign funds to hire bodyguards, federa
         l election regulators ruled Thursday -- nearly three months after the violen
         t January 6 siege on the US Capitol raised fresh concerns about lawmakers\'
         safety.  The 5-1 vote by the Federal Election Commission allows lawmakers to
         use donors\' money for "bona fide, legitimate, professional personal securit
         y" against threats that arise as part of their jobs.  The action came in res
         ponse to a request from officials with the National Republican Congressional
         Committee and the National Republican Senatorial Committee and falls in line
         with previous FEC actions that allow politicians to use campaign money to up
         grade security at their homes. But the commission spent hours of their onlin
         e meeting tussling over how to properly define security personnel after Demo
         cratic lawyers raised the specter of some lawmakers using donors\' money to
         pay right-wing militia members.  In a Wednesday letter to the commission, Ma
         rc Elias and other attorneys representing Democratic campaign committees urg
         ed regulators to craft the rules narrowly so that "campaign funds are not im
         properly used to fund groups organized to harass and intimidate political op
         ponents."  "In the past election cycle, some individuals who are now Members
         of Congress displayed troubling ties to extremist groups, including some sel
         f-proclaimed \'militias,\' such as the Proud Boys, the Oath Keepers, and the
         Three Percenters," the Democratic lawyers wrote. "In some cases, these group
         s purported to provide \'security\' at events attended by Congressional cand
         idates and Members of Congress."  One Democrat on the commission, Ellen Wein
         traub, said she was concerned about lawmakers operating at a far remove from
         their constituents and that untrained guards could improperly block the publ
         ic from engaging with elected officials.  "I never thought of us as a countr
         y where the leadership of the country had to be surrounded by armed guards a
         nd ... needed to keep the public at arm\'s length," Weintraub said.  But Jes
         sica Furst Johnson, a lawyer representing the Republican campaign committees
         , said lawmakers have pressing security concerns.  The threats they face, sh
         e retorted, do not involve "people who are showing up at homes in the middle
         of the night to have a nice conservation about legislation. We are talking a
         bout situations where members are, unfortunately, feeling threatened with th
         eir children in their homes in the middle of the night." '
```

Figure 22 : Définition de l'article dans la variable text

Ensuite on importe selon le modèle entraîné.

```
In [17]: import fasttext

In [22]: f_model_path = os.path.join(
             Fasttext.model_path, 'fasttext', 'model_1')

         model = fasttext.load_model(f_model_path)
```

Figure 23 : Importation du modèle FastText

```
In [73]: import tensorflow as tf

In [74]: tf_model_path = os.path.join(
                         Tensorflow.model_path, 'tensorflow', 'model_1')

         model = tf.keras.models.load_model(tf_model_path)
```

Figure 24 : Importation du modèle TensorFlow

On termine avec le teste de l'article

```
In [23]: model.predict(text)

Out[23]: (('__label__true',), array([0.86624551]))
```

Figure 25 : Test de l'article vrai selon FastText

```
In [26]: model.predict(text)

Out[26]: (('__label__fake',), array([0.99912077]))
```

Figure 26 : Test de l'article faux selon FastText

```
In [120…  model.predict([text])

Out[120…  array([[0.0977672]], dtype=float32)
```

Figure 27 : Test de l'article vrai selon TensorFlow

```
In [122…  model.predict([text])

Out[122…  array([[0.00042763]], dtype=float32)
```

Figure 28 : Test de l'article faux selon TensorFlow

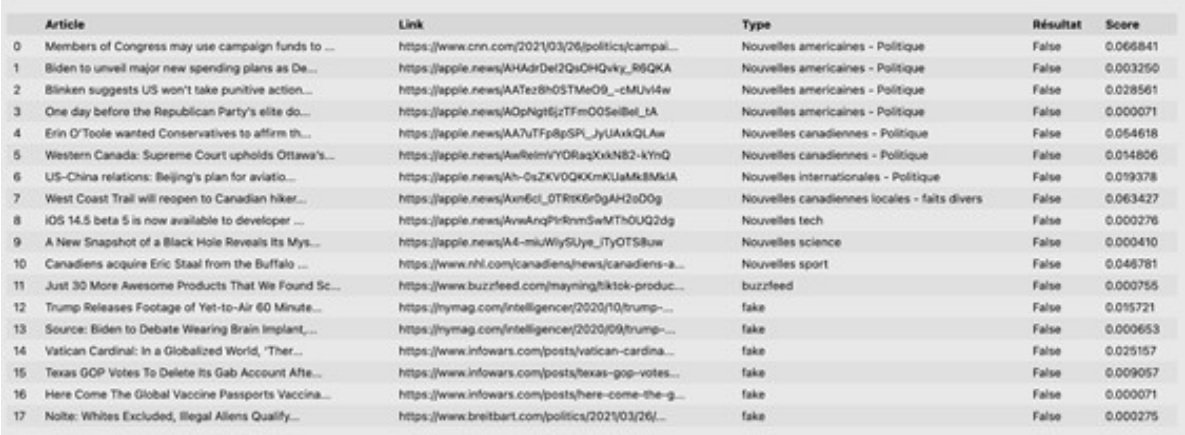
CHAPITRE 4

RÉSULTATS ET DISCUSSION

Suite à nos divers tests, nous recueillons comme résultats : 93.67% avec FastText et 99.28% avec TensorFlow. On peut voir qu'en testant avec les données qu'on a, le score du modèle entraîné avec Tensorflow est beaucoup plus précis, peut être même "trop" précis. Il ne faut pas oublier que les données que nous avons sont limitées à un groupe de thèmes précis, et se concentrent surtout sur les nouvelles Américaines, ce qui limite les possibilités d'apprentissage et par ce fait, les applications possibles. Pour mieux comparer les deux modèles, des articles ont été sélectionnés au hasard, voilà comment les modèles se comportent dans les deux cas (en ordre FastText, Tensorflow).

	Article	Link	Type	Résultat	Score
0	Members of Congress may use campaign funds to ...	https://www.cnn.com/2021/03/26/politics/campai...	Nouvelles americaines - Politique	[__label__true]	[0.8655688]
1	Biden to unveil major new spending plans as De...	https://apple.news/AHAdrDel2QsOHQvky_R6QKA	Nouvelles americaines - Politique	[__label__true]	[0.92760366]
2	Blinken suggests US won't take punitive action...	https://apple.news/AATezBh0STMeO9_-cMUvI4w	Nouvelles americaines - Politique	[__label__true]	[0.5219696]
3	One day before the Republican Party's elite do...	https://apple.news/AOpNgt6jzTFmO0SeiBeI_tA	Nouvelles americaines - Politique	[__label__true]	[0.9043543]
4	Erin O'Toole wanted Conservatives to affirm th...	https://apple.news/AA7uTFp8pSPi_JyUAxkQLAw	Nouvelles canadiennes - Politique	[__label__fake]	[0.6802929]
5	Western Canada: Supreme Court upholds Ottawa's...	https://apple.news/AwRelmVYORaqXxkN82-kYnQ	Nouvelles canadiennes - Politique	[__label__true]	[0.5412521]
6	US-China relations: Beijing's plan for aviatio...	https://apple.news/Ah-0sZKV0QKKmKUaMk8MkIA	Nouvelles internationales - Politique	[__label__true]	[0.9857506]
7	West Coast Trail will reopen to Canadian hiker...	https://apple.news/Axn6cl_0TRtK6r0gAH2oD0g	Nouvelles canadiennes locales - faits divers	[__label__fake]	[0.5413083]
8	iOS 14.5 beta 5 is now available to developer ...	https://apple.news/AvwAnqPirRnmSwMTh0UQ2dg	Nouvelles tech	[__label__fake]	[0.99955374]
9	A New Snapshot of a Black Hole Reveals Its Mys...	https://apple.news/A4-miuWiySiUye_iTyOTS8uw	Nouvelles science	[__label__fake]	[0.9970895]
10	Canadiens acquire Eric Staal from the Buffalo ...	https://www.nhl.com/canadiens/news/canadiens-a...	Nouvelles sport	[__label__fake]	[0.54190737]
11	Just 30 More Awesome Products That We Found Sc...	https://www.buzzfeed.com/mayning/tiktok-produc...	buzzfeed	[__label__fake]	[0.990149]
12	Trump Releases Footage of Yet-to-Air 60 Minute...	https://nymag.com/intelligencer/2020/10/trump-...	fake	[__label__fake]	[0.9408776]
13	Source: Biden to Debate Wearing Brain Implant,...	https://nymag.com/intelligencer/2020/09/trump-...	fake	[__label__fake]	[0.96281147]
14	Vatican Cardinal: In a Globalized World, 'Ther...	https://www.infowars.com/posts/vatican-cardina...	fake	[__label__fake]	[0.696766]
15	Texas GOP Votes To Delete Its Gab Account Afte...	https://www.infowars.com/posts/texas-gop-votes...	fake	[__label__fake]	[0.9502674]
16	Here Come The Global Vaccine Passports Vaccina...	https://www.infowars.com/posts/here-come-the-g...	fake	[__label__true]	[0.7410048]
17	Nolte: Whites Excluded, Illegal Aliens Qualify...	https://www.breitbart.com/politics/2021/03/26/...	fake	[__label__fake]	[0.99911696]

Figure 29 : Tests effectués avec FastText



	Article	Link	Type	Résultat	Score
0	Members of Congress may use campaign funds to ...	https://www.cnn.com/2021/03/26/politics/campai...	Nouvelles americaines - Politique	False	0.066841
1	Biden to unveil major new spending plans as De...	https://apple.news/AHAdrDel2QsOHQvky_R6QKA	Nouvelles americaines - Politique	False	0.003250
2	Blinken suggests US won't take punitive action...	https://apple.news/AATez8h0STMeO9_-cMUvI4w	Nouvelles americaines - Politique	False	0.028561
3	One day before the Republican Party's elite do...	https://apple.news/AOpNgt6jzTFmO0SeiBel_tA	Nouvelles americaines - Politique	False	0.000071
4	Erin O'Toole wanted Conservatives to affirm th...	https://apple.news/AA7uTFp8pSPi_JyUAxkQLAw	Nouvelles canadiennes - Politique	False	0.054618
5	Western Canada: Supreme Court upholds Ottawa's...	https://apple.news/AwReImVYDRaqXxkN82-kYnQ	Nouvelles canadiennes - Politique	False	0.014806
6	US-China relations: Beijing's plan for aviatio...	https://apple.news/Ah-0sZKV0QKKmKUaMk8MkIA	Nouvelles internationales - Politique	False	0.019378
7	West Coast Trail will reopen to Canadian hiker...	https://apple.news/Axn6cl_0TRtKBr0gAH2oDOg	Nouvelles canadiennes locales - faits divers	False	0.063427
8	iOS 14.5 beta 5 is now available to developer ...	https://apple.news/AvwAnqPlrRnmSwMTh0UQ2dg	Nouvelles tech	False	0.000276
9	A New Snapshot of a Black Hole Reveals Its Mys...	https://apple.news/A4-miuWlySUye_iTyOTS8uw	Nouvelles science	False	0.000410
10	Canadiens acquire Eric Staal from the Buffalo ...	https://www.nhl.com/canadiens/news/canadiens-a...	Nouvelles sport	False	0.046781
11	Just 30 More Awesome Products That We Found Sc...	https://www.buzzfeed.com/mayning/tiktok-produc...	buzzfeed	False	0.000755
12	Trump Releases Footage of Yet-to-Air 60 Minute...	https://nymag.com/intelligencer/2020/10/trump-...	fake	False	0.015721
13	Source: Biden to Debate Wearing Brain Implant,...	https://nymag.com/intelligencer/2020/09/trump-...	fake	False	0.000653
14	Vatican Cardinal: In a Globalized World, 'Ther...	https://www.infowars.com/posts/vatican-cardina...	fake	False	0.025157
15	Texas GOP Votes To Delete Its Gab Account Afte...	https://www.infowars.com/posts/texas-gop-votes...	fake	False	0.009057
16	Here Come The Global Vaccine Passports Vaccina...	https://www.infowars.com/posts/here-come-the-g...	fake	False	0.000071
17	Nolte: Whites Excluded, Illegal Aliens Qualify...	https://www.breitbart.com/politics/2021/03/26/...	fake	False	0.000275

Figure 30 : Tests effectués avec TensorFlow

En effet, en regardant les résultats d'articles choisis aléatoirement, on peut voir que subjectivement, FastText performe clairement mieux et nous donne quelque chose de plus proche de la réalité. TensorFlow, quant à lui, n'est pas proche des classes définies, il reste proche de 0 (faux), mais les scores relatifs entre les articles sont justes (Les articles vrais se situent plus proche de la catégorie vraie que les articles faux). On peut donc conclure que Tensorflow apprend trop bien le type de données qui lui sont offertes et devient donc un expert qui ne se trompe presque jamais quand il est limité à ce type de données là. Il serait donc préférable d'utiliser celui-ci pour entraîner un modèle pour évaluer des articles dans un contexte précis. Fasttext, quant à lui, est mieux équipé pour évaluer des articles de façon générale et donc, pour des applications simples où on désire classifier beaucoup de données avec des résultats rapides, celui-ci serait un excellent choix. Celui-ci se limite par contre par le fait qu'il ne donne pas beaucoup de contrôle à l'utilisateur et donc l'utilisation doit être conforme aux attentes de celui-ci et des situations spécifiques ne sont pas recommandées pour celui-ci.

CHAPITRE 5

CONCLUSION ET RECOMMANDATIONS

En conclusion, depuis l'apparition des réseaux sociaux et l'augmentation de leur popularité, nous avons connu une augmentation phénoménale d'informations circulant sur internet. En effet, les fausses informations sont maintenant très présentes dans notre quotidien. Ceci constitue un réel problème sur lequel nous avons décidé de travailler. Effectivement, nous avons pu concevoir dans le cadre du projet de fin d'études une intelligence artificielle qui fait l'analyse textuelle des articles donnés et qui nous retourne un niveau de validité. Nous avons commencé par décrire la revue de littérature que nous avons faite. Nous avons ensuite décrit la méthodologie employée dans le cadre du projet. La documentation technique du prototype a été décrite. Nous avons terminé par une analyse des performances, nous avons présenté les résultats.

Suite à ce travail, nous avons plusieurs recommandations futures afin d'améliorer le projet. En effet, une base de données avec des types d'articles plus variée aurait été utile puisque l'intelligence n'était limitée qu'a un seul type soit la politique américaine. Ceci lui aurait permis d'apprendre une plus grande quantité d'informations.

Ensuite, nous recommandons d'utiliser FastText pour avoir des bons résultats rapidement. Nous recommandons de ne pas l'utiliser quand nous voulons avoir un contrôle sur les différents paramètres.

Par la suite, nous recommandons d'utiliser TensorFlow pour avoir des bons résultats, un contrôle désiré, des champs spécifiques, mais il faut avoir de très bonnes données. Nous recommandons de ne pas l'utiliser quand le temps est un facteur important puisqu'il prend beaucoup de temps à entraîner.

CHAPITRE 6

RÉFLEXION SUR LES NOTIONS DE DÉVELOPPEMENT DURABLE

Le fléau des fausses nouvelles est un problème que devra résoudre la société au cours du 21 siècle. Elle impacte différents aspects tels que la société, l'environnement et l'économie. Le but de la fausse information est de tromper le lecteur en influençant son opinion sur le sujet de l'article. En effet, lorsque quelqu'un lit une fausse nouvelle, la personne peut se faire une idée erronée vis-à-vis d'autres personnes ou sur une situation qui est à l'actualité du jour. Dernièrement, on a vu l'impact de la fausse information sur les élections présidentielles des États-Unis qui a mené à l'assaut meurtrier du Capitole. Plusieurs informations de fraude électorale ont été publiées et vues par des millions de personnes ce qui a créé une vague de manifestation violente contre le résultat des élections.

En ce qui à trait à la conception de notre programme, nous n'affectons pas les aspects du développement durable. L'utilisation du programme permet d'aider les gens à vérifier les sources qui permettront de prendre une décision réfléchie vis-à-vis les faits vérifiés. La décision des gens peut impacter l'univers de l'économie, de la société et de l'environnement. Par exemple, une personne pourrait arrêter d'acheter du Nutella en raison des études qui ont prouvé que l'huile de palme est dangereuse pour l'environnement. Ici, cette décision a un impact sur l'environnement et sur l'économie. Dans le futur, l'augmentation des champs de compétence de l'algorithme aura un impact important sur l'aspect social en lien avec les fausses nouvelles. L'impact de l'utilisation du programme pour ensuite s'en départir rajouterait des doutes sur la crédibilité d'article sur internet.

BIBLIOGRAPHIE

[1] Armand Joulin, Edouard Grave, Piotr Bojanowski et Tomas Mikolov. 2016. « Bag of Tricks for Efficient Text Classification ». Facebook AI Research, En ligne. <https://arxiv.org/pdf/1607.01759.pdf>. Consulté le 25 Janvier 2021.

[2] Google. « Machine Learning Guides ». In Google developpers. En ligne <Step 2.5: Choose a Model | ML Universal Guides>. Consulté le 25 Janvier 2021.

[3] TensorFlow. En ligne <https://www.tensorflow.org/?hl=fr>. Consulté le 12 Mars 2021

[4] GitHub. « K-Fold Cross Validation ». In GitHub IO. En ligne <http://ethen8181.github.io/machine-learning/model_selection/model_selection.html>. Consulté le 12 Mars 2021.

[5] Fredereka Schouten. 2021 « Members of Congress ay use campaign funds to hire bodyguards, FEC rules ». In CNN politics. En ligne <https://www.cnn.com/2021/03/26/politics/campaign-funds-security-congress-lawmakers-fec-ruling/index.html>. Consulté le 22 Mars 2021.

[6] Julie Mcmahon. 2021 « Nolte : Whites excluded, illegal aliens qualify for Oakland's 500$ month payouts ». In Newsakmi. En ligne < https://newsakmi.com/news/usnews/nolte-whites-excluded-illegal-aliens-qualify-for-oaklands-500-month-payouts/>. Consulté le 22 Mars 2021.