Catégorisez automatiquement des questions

Hui-Guan TAI

Résumé

Le traitement naturel du langage, ou Natural Language Processing (NLP) en anglais, est un ensemble de traitements très utilisé dans l'exploration et l'analyse de texte, notamment pour permettre aux ordinateurs de lire, déchiffrer et donner du sens au langage humain.
Cependant, le choix d'un modèle est complexe. Dans ce rapport, nous présentons une méthodologie pour préparer le texte (constitué à partir de questions du site StackOverflow), l'évaluer et le modéliser. Cette méthodologie donnera lieu à un modèle qui sera utilisé pour un moteur de suggestion de tag.

Table des matières

INTRODUCTION **2**

METHODOLOGIE **2**

JEU DE DONNEES 2
PROFIL DES DONNEES 2
PREPARATION 4
Principe *4*
Implémentation *4*
BAG OF WORDS 5
Principe *5*
Implémentation *5*
TF-IDF 6
Principe *6*
Implémentation *6*
ANALYSE EN COMPOSANTES PRINCIPALES 11
LABELLISATION 11
MODELISATION (SUPERVISEE) 12
Bayes Naif *12*
Random Forest *12*
Réseau de neurones *12*

RESULTATS **14**

CONCLUSIONS **14**

RÉFÉRENCES **14**

Introduction

Stackoverflow est le site de référence pour toutes les questions réponses qui concernent la programmation informatique. Ce portail réunit une large communauté de développeurs de tout niveau et de toutes nationalités sur un grand éventail de thématiques.
Pour arriver à se retrouver dans ce volume conséquent de données, le tag (étiquette) permet d'indexer et de (re)trouver facilement et rapidement une réponse à une question donnée.

L'enjeu de cette étude est de trouver une méthode efficiente qui permette d'associer le bon tag correspondant à une question posée. Cette méthode servira de base à l'algorithme d'une API de suggestion de tag.

Pour ce faire, nous nous appuierons sur des techniques de traitement (automatique) du langage naturel. Ce domaine d'étude qui a vu le jour dans les années 50, a connu un nouvel essor depuis 2018 avec l'accélération du développement de l'intelligence artificielle.

Méthodologie

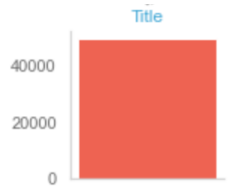
Jeu de données

Pour permettre à ses utilisateurs de récupérer un jeu de questions/réponses, le site stackoverflow met à disposition un portail de requetage sql :
https://data.stackexchange.com/stackoverflow/query/new
Pour les besoins de l'étude (suggestion de tag en fonction de questions) et constituer un jeu de donnée représentatif, nous ne garderons que les questions (correspondant au critère postetypeid=1).
Pour garantir une expérience utilisateur confortable pour tout le monde, le site limite la durée d'une requête. Nous limiterons donc le volume des questions à 50000 questions datant de 2010 à 2020.
Nous utiliserons la requête SQL suivante à utiliser telle quelle dans la fenêtre de requêtes :
"select * from posts WHERE Id < 50000 and PostTypeId=1"

Profil des données

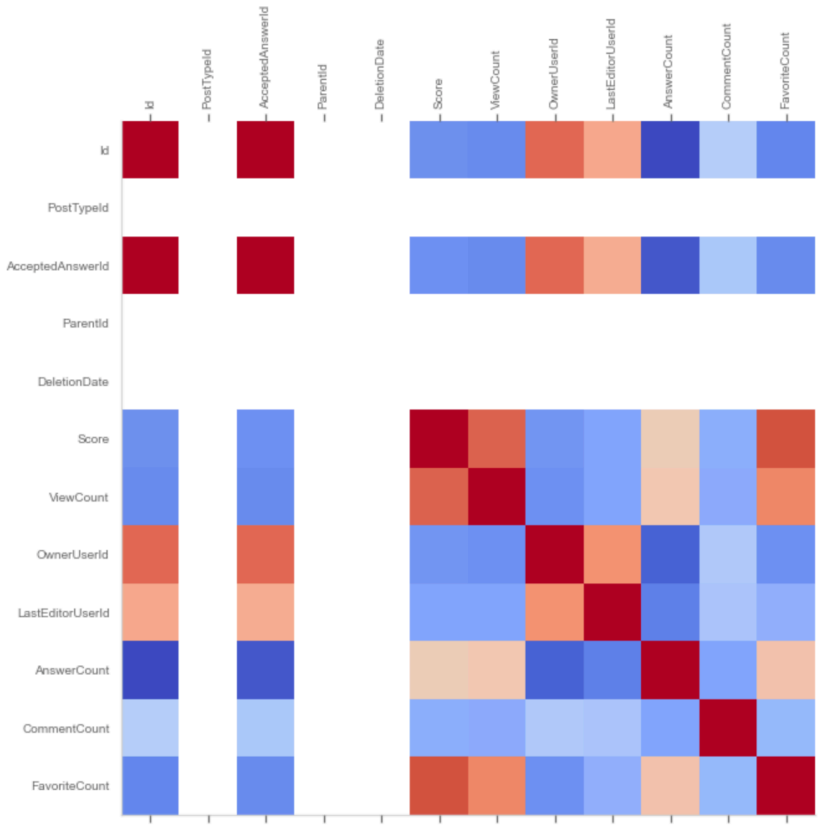
Une première analyse sur les données catégorielle met en évidence plus de 90% de valeur manquantes sur les noms des utilisateurs qui ont posté, ainsi que sur les dates (clôtures, suppression).

	missing_values	%_missing_values
Id	0	0.000
PostTypeId	0	0.000
AcceptedAnswerId	20235	40.470
ParentId	50000	100.000
CreationDate	0	0.000
DeletionDate	50000	100.000
Score	0	0.000
ViewCount	0	0.000
Body	0	0.000
OwnerUserId	953	1.906
OwnerDisplayName	47214	94.428
LastEditorUserId	22316	44.632
LastEditorDisplayName	48469	96.938
LastEditDate	21867	43.734
LastActivityDate	0	0.000
Title	0	0.000
Tags	0	0.000
AnswerCount	0	0.000
CommentCount	0	0.000
FavoriteCount	34828	69.656
ClosedDate	47733	95.466
CommunityOwnedDate	49823	99.646
ContentLicense	0	0.000



Nous excluons donc ces variables de l'étude.
Les distributions montrent que les variables catégorielles restantes body, title, creationdate sont très étalées et prennent des valeurs très hétérogènes

Quant aux variables quantitatives, la matrice de corrélation, indique une corrélation forte entre le Score et ViewCount (nombre de vues) ainsi qu'entre le owneruserid (id du commentateur) et le lasteditoruserid (id du dernier commentateur).



Ces variables constituent les métadonnées décrivant le contexte autour de la question (date d'edition, commentateurs, nombre de vues) mais pas le contenu, l'objet des questions.

Pour déterminer les tags les plus pertinents, pour qu'ils suggèrent un certain contenu, cela nécessite d'identifier les éléments distinctifs d'un contenu.
L'objet d'une question est présent, à priori, dans le titre et dans le corps de la question.

Ce qui nous amène à nous concentrer sur l'analyse textuelle des variables body (corps) et title (titre).

Pour réduire le temps de traitement, nous concentrerons volontairement l'étude sur 2000 questions. L'étude des données portera donc pour chacune de ces questions sur le body title et tag.

Préparation

Échantillon de départ des titres, corps et tags

Title	Body	Tags
comprehensive image processing example using cocoa api	can someone point out a comprehensive example on image processing using cocoa api? i am developing an application...	<cocoa><image-processing>
testing gps in iphone simulator problem	i found some various articles about testing the gps (corelocation) in the iphone simulator. \nit seems pretty str...	<iphone><gps><core-location>
create pdf inapp ios	<blockquote>\n possible duplicate: \n <a href="https://stackoverflow.com/questions/1360912/...	<ios><pdf><pdf-generation>
flex 4 stopping an embbeded swf	i have this flex 4.1 script: \n\n<pre><code><fx:script>\n <![cdata[\n [embed(source="res/sw...	<apache-flex><actionscript-3>
html form - firing the form submit using the enter key does not work when focus is on a select list	i'm stuck on the above problem. i have a simple form as follows, with a text input, a select list and a submit bu...	<html><forms><submit>

Principe

Pour mettre en évidence les termes importants d'une question, nous filtrons les mots, caractères superflus, et les homogénéisons (transformation en minuscule)

Tags

Les tags contiennent des balises (<>), qui n'apportent peu de sens supplémentaire. Nous les retirons également.

Body, title

La langue (anglaise ou française) emploie dans ses phrases des mots pour articuler, ponctuer ou désigner (articles, ponctuation, coordination, etc…). Ces mots sont facultatifs pour comprendre la signification d'une phrase. Ce sont les stopwords que nous retirons

Les mots a priori différents peuvent être des déclinaisons d'une racine commune qui désigne la même chose (par ex :réunion, réunir, réuni). Pour simplifier l'analyse et ne garder que la racine commune, nous utilisons la technique de lemmatisation qui consiste à parcourir un dictionnaire de déclinaisons pour retrouver et garder la racine.

Implémentation

Nous utilisons la librairie nltk dédié au traitement naturel du language à la fois pour supprimer les stopwords, et pour lemmatiser le corpus.

Pour tokeniser le texte, nous appelons l'expression régulière pour identifier tout mots de plus de 2 caractères.

On remarque que la lemmatisation de base n'est pas tout à fait correcte:

Parce que "are" n'est pas converti en "to be" et "hanging" n'est pas converti en "hang" comme attendu.

Cela peut être corrigé si nous précisons le type de mot (verbe, adjectif, nom, adverbe) que c'est. Pour cela nous fournissons la balise "part of speech" correcte (balise POS) comme deuxième argument de lemmatize()

	Title	Body	Tags	FilteredBody	FilteredTitle	BodyTokens	TitleTokens	BodyLemmas	TitleLemmas
45583	wordpress > create category as child of "uncategorized" (catid=1)	i've got a script in my functions.php file that checks for the existence of several categories that are required by my theme, and if they are not present, it creates them... `if(!get_cat_id('my-category')){wp_create_category('my-category');}` i need to modify this script in two ways... first, i need to create the category so that its parent category is the "uncategorized" category (or id 1). second, if the category already exists, but its parent is not id 1, i need to make it so.	wordpress categories	got script functions php file checks existence several required theme present creates need modify script two ways first need create parent un id second alre exists parent id need make	wordpress create category child uncategorized catid 1	['got', 'script', 'functions', 'php', 'file', 'checks', 'existence', 'several', 'required', 'theme', 'present', 'creates', 'need', 'modify', 'script', 'two', 'ways', 'first', 'need', 'create', 'parent', 'un', 'id', 'second', 'alre', 'exists', 'parent', 'id', 'need', 'make']	['wordpress', 'create', 'category', 'child', 'uncategorized', 'catid']	['get', 'script', 'function', 'php', 'file', 'check', 'existence', 'several', 'require', 'theme', 'present', 'creates', 'need', 'modify', 'script', 'two', 'way', 'first', 'need', 'create', 'parent', 'un', 'id', 'second', 'alre', 'exists', 'parent', 'id', 'need', 'make']	['wordpress', 'create', 'category', 'child', 'uncategorized', 'catid']
23834	how to reshape matlab matrices for this example?	i have a 40x16 matrix or 8 5x16 one below the other i.e. aligned vertically. i want to get a 5x128 matrix from that such that i align the 8 5x16 matrices horizontally. is there an efficient/quicker (rather than the hardcoded for loops) way to do this? i want the individual 5x16 matrices intact.	matlab matrix reshape	matrix x one e aligned verti want get matrix align x matrices horizontally efficient quicker rather har loops way want individual matrices int	reshape matlab matrices example	['matrix', 'one', 'aligned', 'verti', 'want', 'get', 'matrix', 'align', 'matrices', 'horizontally', 'efficient', 'quicker', 'rather', 'har', 'loops', 'way', 'want', 'individual', 'matrices', 'int']	['reshape', 'matlab', 'matrices', 'example']	['matrix', 'one', 'align', 'verti', 'want', 'get', 'matrix', 'align', 'matrix', 'horizontally', 'efficient', 'quicker', 'rather', 'har', 'loop', 'way', 'want', 'individual', 'matrix', 'int']	['reshape', 'matlab', 'matrix', 'example']

A l'issue de cette préparation, nous pouvons tenter de comparer et d'évaluer les questions.

Bag Of Words

Nous utilisons pour cela le modèle BOW (sac de mots).

Principe

L'algorithme Bag of Words (BoW – sac de mots) compte combien de fois un mot apparaît dans un document. Ces nombres de mots nous permettent de comparer des documents et d'évaluer leurs similitudes pour des applications telles que la recherche, la classification de documents et la modélisation de sujets.

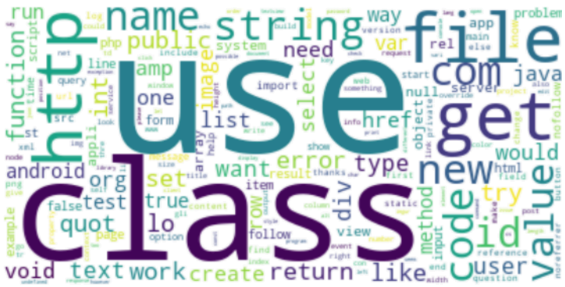
La méthode bag of words ne prends pas en compte la sémantique des mots. Par example, les mots maison et appartement sont souvent employés ensemble, cependant dans le modele bow, les vecteurs correspondants sont différents.

Le sac de mots ne prend pas en considération l'ordre des mots dans lesquels ils apparaissent dans un document, et seuls les mots individuels sont comptés.

Dans certains cas, l'ordre des mots peut être important.

Implémentation

La librairie Sci-kit learn propose une fonction de vectorisation par comptage CountVectorizer



dfSumBowTitlesHead

	count
use	202
file	122
get	99
error	85
value	73
work	68
array	65
string	64
android	61
data	59

dfSumBowBodiesHead

	count
use	1940
class	1472
file	1391
get	1385
http	1270
string	1229
new	1197
code	1145
id	1127
name	1120

Ici , les mots les plus fréquents dans le corps du corpus sont ‘use’, ‘class’, ‘file’, ‘get’ et dans les titres, ‘use’,’file’,’get’,’error’.
Chacun de ces mots, individuellement, est très générique, permet difficilement de déterminer clairement la thematique technique de la question.
De plus, cette méthode donne de l’importance aux mots les plus fréquents même s’ils ont peu d’importance (adjectifs, determinants, conjonction de coordination, etc…)

L’algorithme "Term Frequency-Inverse Document Frequency" attribue une valeur à un mot en fonction de son importance dans un document et proportionnellement à son importance dans tous les documents d’un corpus, ce qui élimine mathématiquement les mots naturels de la langue anglaise et sélectionne les mots les plus descriptifs du texte.

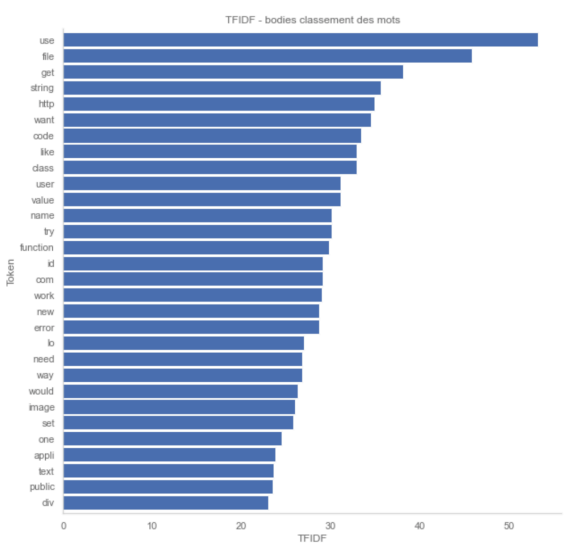
TF-IDF

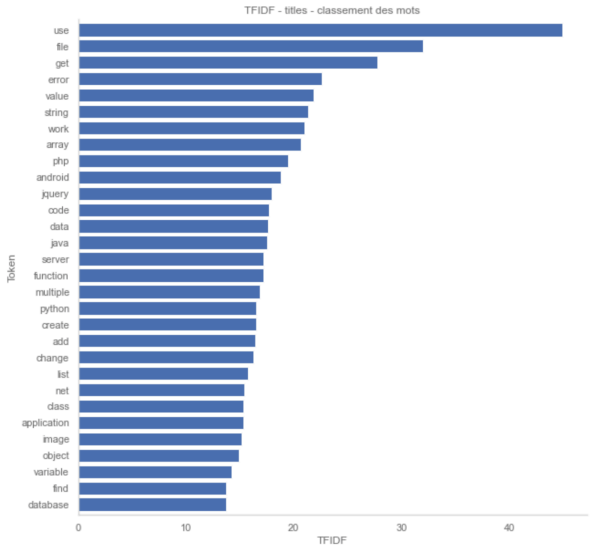
Principe
Le TFIDF est calculé en multipliant la fréquence du terme par l'inverse de la fréquence du document.

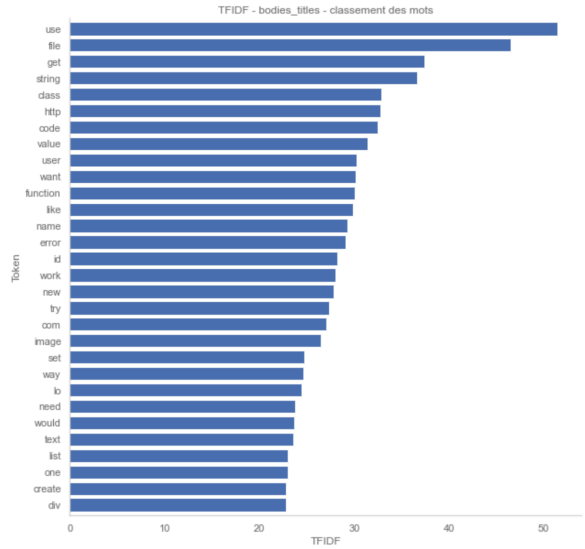
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

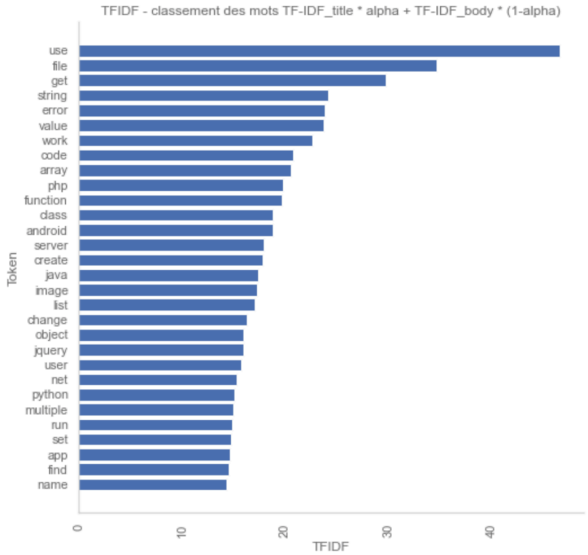
Implémentation
La librairie sci-kit learn propose la classe de vectorisation TfidfVectorizer que nous utilisons pour calculer la matrice TF-IDF pour le body, title et tag.
Nous commençons par le monogramme, nous continuerons avec les bigrammes

Monogrammes
Nous nous retrouvons avec une matrice tf-idf pour le body et une autre pour title.
Nous souhaitons n’avoir qu’une seule matrice à manipuler. On peut se dire que l’on peut concatener pour chaque questions le titre et le corps de message (que nous designerons par body+title).
Le calcul de score TF IDF pour les monogrammes body+title, donne le même classement que le tfidf du body seul. Par consequent, la concaténation body+title fait perdre le poid des termes contenus dans titles.
TF-IDF (Pondéré) = TF-IDF(title) * alpha + TF-IDF(body) * (1-alpha)
Nous essayons donc en créant une matrice TFIDF à partir de la somme pondérée par un coefficient alpha des TFIDF title et TFIDF Body, suivant la formule suivante :
TF-IDF (Pondéré) = TF-IDF(title) * alpha + TF-IDF(body) * (1-alpha) avec alpha =0.8 (on suppose que le titre a plus de poids pour contenir les mots clés qui résument le corps de la question).
Nous voyons que le classement des monogrammes TFIDF pondérés sont différents de du classement TFIDF du body seul et celui du TFIDF du title seul.
Voici les classements de monogrammes par valeur de TFIDF decroissants

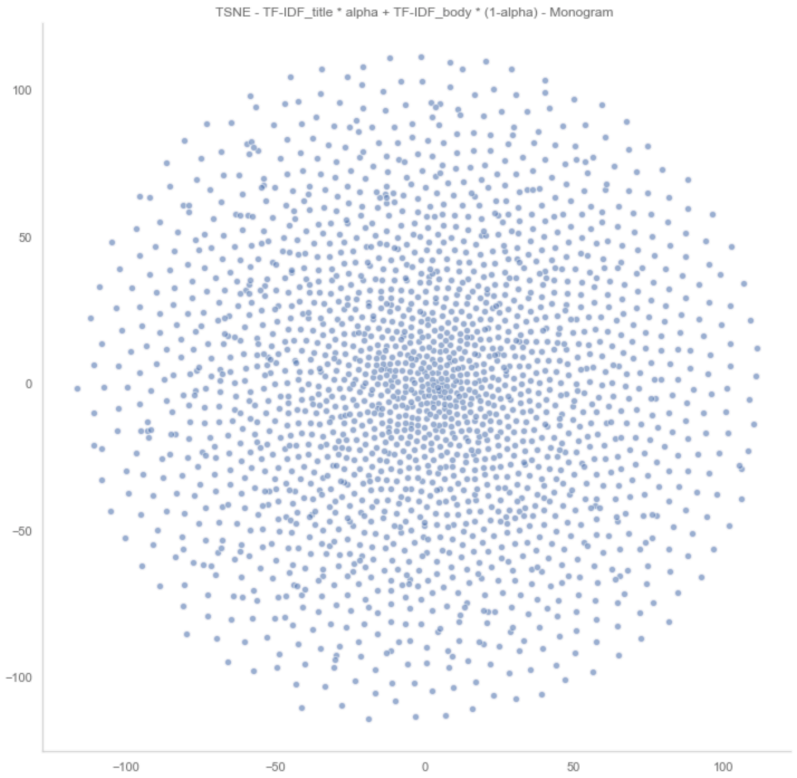




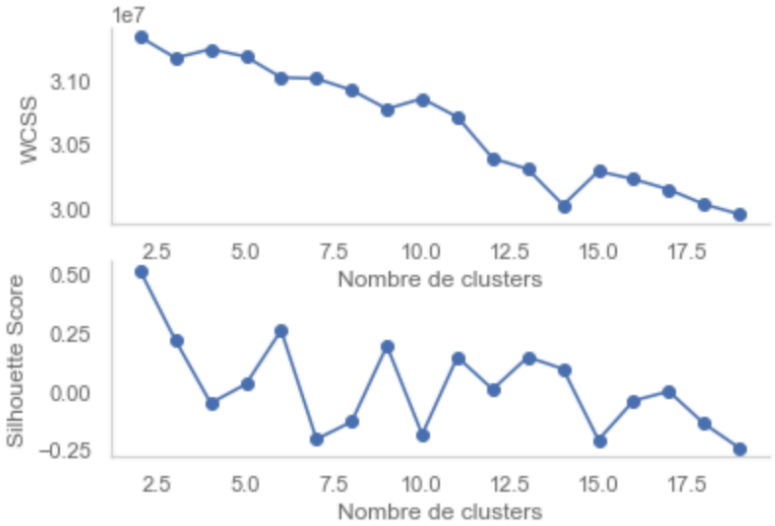




La visualisation TSNE correspondante (qui permet de réduire le nombre de dimensions tout en conservant la similarité des points) suggère beaucoup de petits regroupements



Au vu de ces regroupements, nous faisons une recherche de cluster par kmeans en calculant le Within Cluster Sum of Square(la somme des carré entre chaque point et le centroide d'un cluster). En minimisant cette valeur, les distances entre les points d'un cluster et de leur centroide tend vers 0, pour former des clusters parfaits. Cependant, cela n'a plus de sens avec un nombre elevé de clusters car nous aurons autant de clusters que d'observations.Il existe donc une valeur seuil pour K que nous pouvons trouver en utilisant le graphe du coude, nombre de clusters au dela duquel la distance wcss diminue faiblement. Cette méthode, lorsqu'elle ne présente pas de coude peut ne pas suffire.
La valeur de silhouette mesure à quel point un point est similaire à son propre cluster (cohésion) par rapport à d'autres clusters (séparation). Un score de 1 correspond a des clusters parfaitement délimités, un score de 0.5 correspond à des clusters indistincts les uns des autres (les points sont à cheval entre plusieurs centre de clusters différents) et un score de 0 correspond à des points rattachés à des mauvais clusters.
Dans le cas du TDF-IDF, le clustering K-Means ne présente ni de coude ni de score de silhouette supérieur à 0.5. Par conséquent le clustering n'est pas concluant.

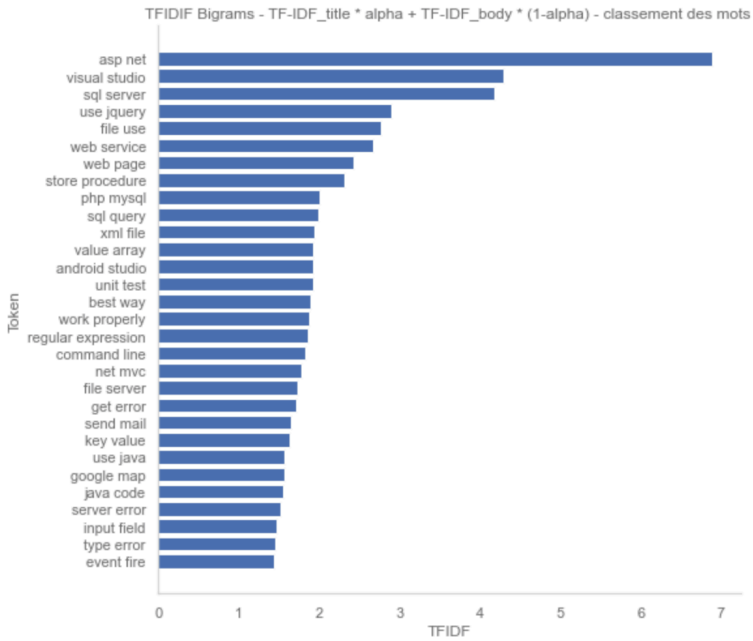


Pour affiner le sens des questions, nous utiliserons les N-Grammes dans la suite.

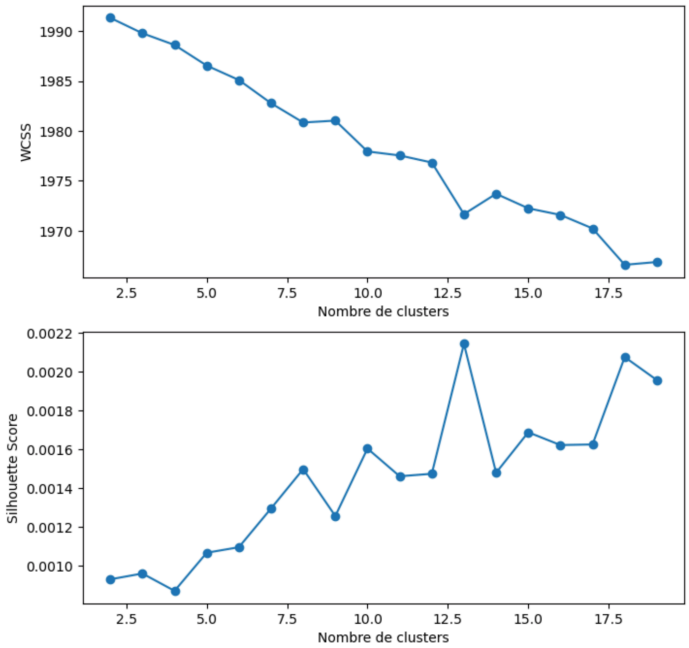
Les N-grammes capturent le contexte dans lequel les mots sont utilisés ensemble. Par exemple, il peut être judicieux de considérer des bigrammes comme « New York » au lieu de les diviser en mots individuels comme « New » et « York »

Considérez la phrase "J'aime danser sous la pluie"
Voir les cas Uni-Grammes, Bi-Grammes et Tri-Grammes ci-dessous.
UNIGRAM : "j", "aime", "danser", "sous", "la", "pluie"
BIGRAM : "J'aime", "aime danser", "danser sous", "sous la", "la pluie"
TRIGRAM : « J'aime danser sous », «danser sous la», « sous la pluie»

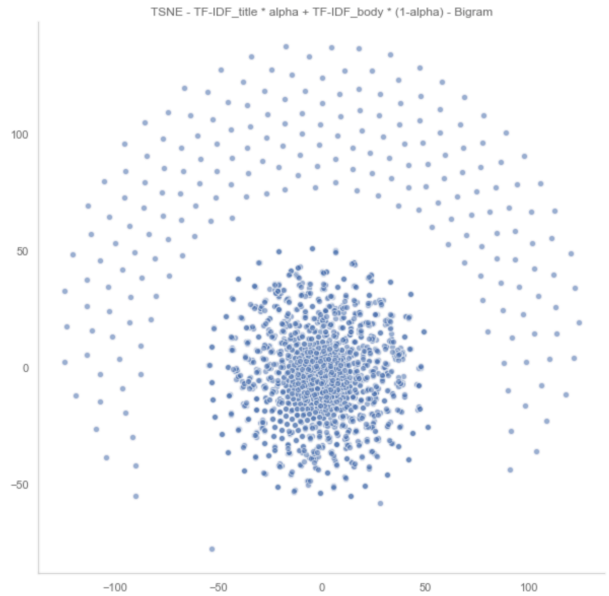
Bi Grammes
L'analyse par bi-grammes donne les résultats suivants :



Les groupes de bi-grammes comme les couples de mot « visual studio », « web service », « sql query » sont plus intuitifs pour suggérer des thématiques que de simple monogrammes.



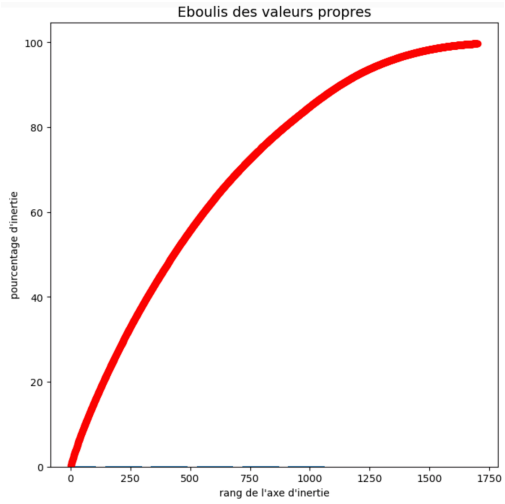
La recherche de cluster par clustering K-means sur les bigrammes donne un score de silhouette très faible, proche de 0 même en augmentant le nombre de cluster.

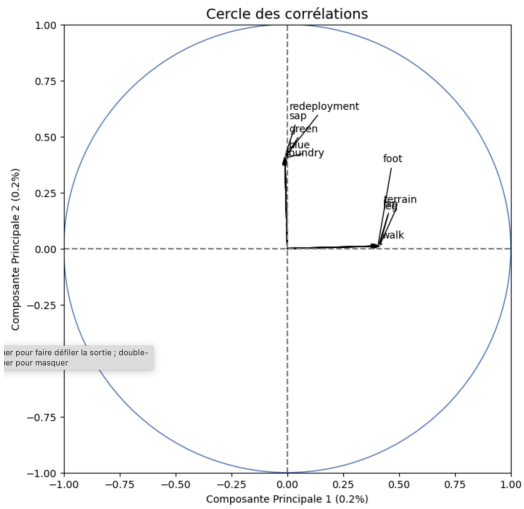


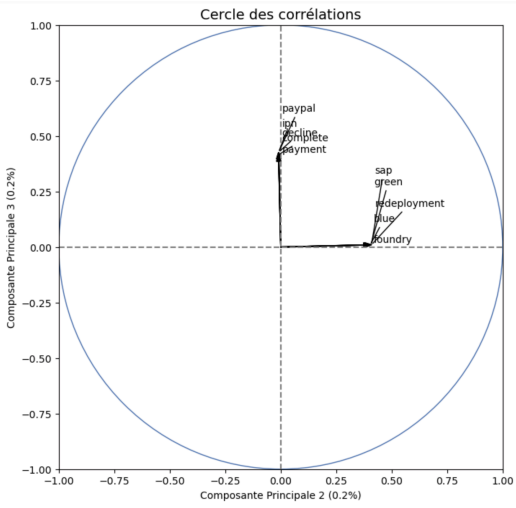
La visualisation TSNE met en évidence un ensemble bien distinct de points en forme d'anneau autour d'un nuage de points au centre constitués de sous-ensembles plus compacts.
Bien que le clustering K-means ne soit pas concluant, la visualisation TSNE présente des ensembles plus marqués et le classement TFIDF pondéré est plus évocateur que pour les monogrammes. Nous poursuivrons les tests de modelisations sur la matrice TDFIDF pondérée sur les corps et titres de questions.

Analyse en Composantes Principales

Nous tentons aussi d'effectuer une réduction dimensionnelle des 2296 termes pour identifier des ensembles thématiques via une ACP.



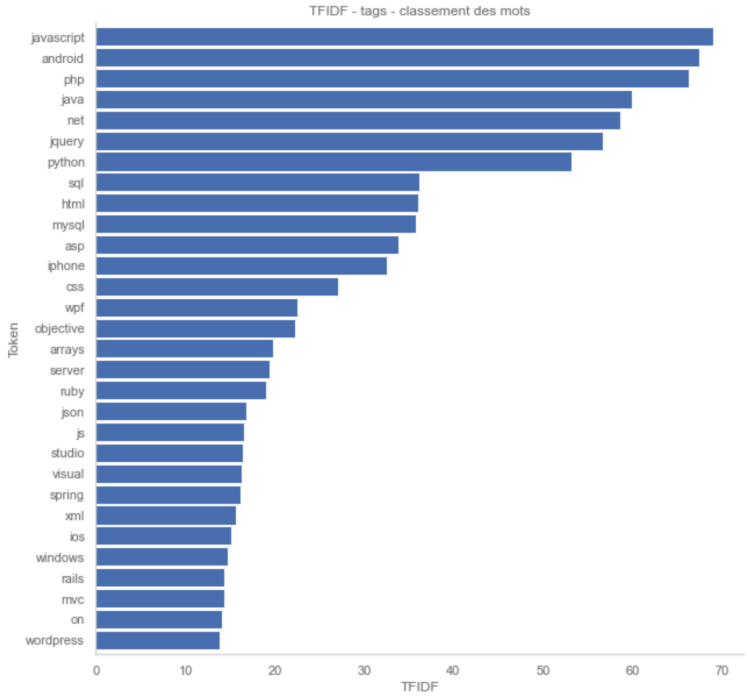




L'éboulis des valeurs propre montre qu'il faut près de 1500 composantes pour approcher expliquer 100% de la variance. Ce qui représente une réduction peu efficace, compte tenu que chaque composante n'explique qu'au plus 0,2% de la variance.
Compte tenu des résultats peu concluants sur la modélisation non supervisée, nous allons chercher à modéliser à l'aide des tags.

Labellisation

Pour ce faire, il nous faut convertir les tags en classes.
Ensuite, nous calculons la somme des valeurs TF IDF par tags sur le corpus de questions et les classons par total décroissant. Nous utiliserons l'ordre dans le classement comme identifiant de classe.



Modélisation (supervisée)

Au préalable, nous séparons notre jeu de données en données d'entrainement et de test, pour respectivement entrainer le modèle et évaluer les performances du modèle.
Pour mesurer la performance du modèle nous calculerons l'exactitude des prédictions sur les vrais positifs et les vrais négatifs par rapport à l'ensemble des classes prédites (accuracy), mais également la similarité des classes réelles et celles prédites par le modèle(score ARI), valant 0 quand la classification entre 2 clusters est aléatoire et 1 lorsque les 2 clusters sont identiques (sans tenir compte des permutations).
Avec ces métriques, nous allons essayer de trouver la modélisation supervisée la plus efficace.

Bayes Naif

La méthode de classification naïve de Bayes est basée sur le théorème de Bayes. Il est qualifié de « naïf » car il suppose l'indépendance entre chaque paire de caractéristiques des données. Il permet de calculer la probabilité à postériori des classes y pour chaque variable x, en s'appuyant sur la probabilité à priori des effectifs de chaque classe.
L'exactitude obtenue sur les données de test est plutôt faible (près de 50%) et le score ARI de stabilité, très faible, proche de 0.

Metrique	Accuracy	ARI
Score	0.541667	0.078659

Random Forest

Le modèle de Random Forest construit plusieurs arbres de décision et les agrège pour obtenir une prédiction plus précise et plus stable. La forêt qu'il construit est une collection d'arbres de décision, formés avec la méthode ensembliste de bagging.

Metrique	Accuracy	ARI
Score	0.815	0.38884

L'exactitude obtenue sur les données de test est plutôt bonne (près de 80%) mais le score ARI de similitude entre les ensembles réels et prédits est faible, très inférieure à 1.

Réseau de neurones

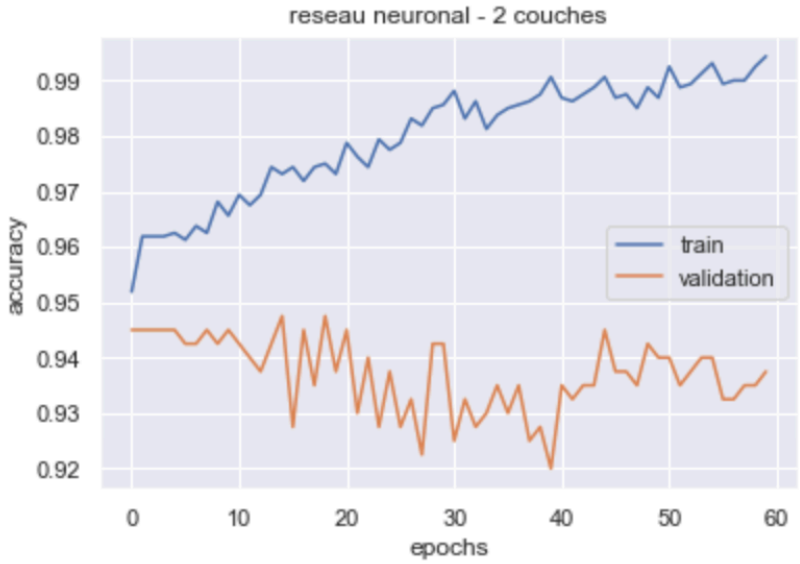
Un réseau neuronal monocouche ne peut être utilisé que pour représenter des fonctions linéairement séparables. Cela signifie des problèmes très simples où, par exemple, les deux classes d'un problème de classification peuvent être soigneusement séparées par une ligne.
Pour notre problématique de classification multiclasse, nous utilisons donc un réseau de neurones à 2 couches avec la fonction d'activation softmax en sortie pour obtenir des valeurs de probabilité pour chaque classe en sortie, et relu dans les couches cachées.

Nous ajoutons également une couche dropout par couche de neurones pour limiter le suraprentissage, avec une perte de 30% par couche dropout.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_layer (Dense)          (None, 2964)              8788260
_________________________________________________________________
dropout (Dropout)            (None, 2964)              0
_________________________________________________________________
hidden_layer_1 (Dense)       (None, 26)                77090
_________________________________________________________________
dropout_1 (Dropout)          (None, 26)                0
_________________________________________________________________
hidden_layer_2 (Dense)       (None, 26)                702
_________________________________________________________________
dropout_2 (Dropout)          (None, 26)                0
_________________________________________________________________
output_layer (Dense)         (None, 31)                837
=================================================================
Total params: 8,866,889
Trainable params: 8,866,889
Non-trainable params: 0
_________________________________________________________________
```

A partir de 60 itérations, ce modèle dépasse 99% d'exactitude dans ses prédictions en entrainement. Sur la phase de validation, il atteint un peu moins de 95% d'exactitude.

Metrique	Accuracy	ARI
Score	0.983333	0.834407



Sur les données de test, le score d'exactitude est de plus de 99% et le score ARI autour de 83%.

Résultats

Les score d'exactitude et ARI du réseau neuronal sont bien meilleurs que les modélisations précédentes. Nous privilegions donc ce modèle pour notre moteur de suggestion de tag.

Conclusions

Dans ce rapport, nous avons proposé une méthodologie pour categoriser automatiquement des questions techniques sur Stackoverflow. La méthodologie s'appuie dans un premier temps sur l'acquisition et la préparation des données puis sur la constitution d'une matrice TFIDF pondérée sur les valeurs TFIDF des corps et des titres des questions. Cette matrice TFIDF pondérée est utilisée pour identifier les ensembles éventuels par apprentissage non supervisé (K-Means, ACP).

Ensuite la comparaison des scores d'exactitude et ARI sur les différents apprentissages supervisés (Naive Bayes, Random Forest, Réseau neuronal) sur les tags permet de déterminer le meilleur modèle prédictif sur un jeu de données réduit (2000 questions) afin de limiter les temps de calculs. Pour la suite, la méthodologie peut être étendue sur un jeu de données à volumétrie réelle.

Dans cette étude, le réseau neuronal à 2 couches avec dropout a présenté les meilleures performances et sera le modèle choisi pour le moteur de suggestion de tags à réaliser pour le projet 6 du parcours DataScientist d'Openclassrooms.

Notebooks et sources:
https://github.com/8huit/OC_DS_P6

Références

Blueprints for Text Analytics Using Python (2020) - Jens Albrecht, Sidharth Ramachandran, Christian Winkler – O'Reilly Media
Part 3: Topic Modeling and Latent Dirichlet Allocation (LDA) using Gensim and Sklearn - https://www.analyticsvidhya.com/blog/2021/06/part-3-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/
Traitement Automatique du Langage Naturel en Français (TAL) - https://maelfabien.github.io/machinelearning/NLPfr/#
End-to-end topic modelling in python: Latent Dirichlet Allocation (LDA) - https://morioh.com/p/2a01aa34c758
tSNE vs. UMAP: Global Structure - https://towardsdatascience.com/tsne-vs-umap-global-structure-4d8045acba17
t-SNE Corpus Visualization - https://www.scikit-yb.org/en/latest/api/text/tsne.html
Cours OpenClassromms - Analysez vos données textuelles - https://openclassrooms.com/fr/courses/4470541-analysez-vos-donnees-textuelles
Multiclass classification using scikit-learn - https://www.geeksforgeeks.org/multiclass-classification-using-scikit-learn/