PARCOURS DATASCIENTIST

PROJET 6

SYSTÈME DE SUGGESTION DE TAGS POUR STACKOVERFLOW

janvier 7, 2019

CONTENTS

Table of Contents

1 Introduction

Cette étude présente la démarche pour la mise en œuvre d'un système de suggestion de « tags » pour la plateforme de Stackoverflow.

Stackoverflow permet à ses utilisateurs de poster des problèmes que ces derniers rencontrent dans la mise en œuvre de solutions dans le domaine des **technologies de l'information**. Des aidant répondent à ces questions. Questions et réponses forment une base de connaissances.

Pour retrouver facilement des problèmes similaires et enrichir la base de connaissances, des « tags » sont proposés pour chacune des questions posées. Ces éléments d'information, les tags, contiennent l'information nécessaire (et idéalement, suffisante) pour retrouver une classe de problèmes **similaires**.

Pour suggérer des tags liés à a question postée, plusieurs approches sont proposées dans cette étude. Ce document présente la construction et l'analyse des modèles de données qui alimentent les différents algorithmes mis en œuvre. Il présente aussi les méthodes mises en œuvre pour la résolution du problème.

Sont mises en œuvre dans cette étude:

- Pour la standardisation des POSTs, des algorithmes de traitement du langage naturel implémentés dans la librairie NLP écrite en Python.
- Des méthodes dites statistiques, basées sur le décompte des TOKEN formant POST .
- Des algorithmes d'apprentissage dits non supervisés, appliqués aux POSTs standardisés. Ces algorithmes trouvent leur pertinence dans le fait que les POSTs constituent un ensemble de données non structurées.
- Des algorithmes d'apprentissage dits supervisés, pour lesquels les TAG_A sont les données annotées à partir desquels la fonction de prédiction (de suggestion) de TAG_S est entraînée.

Le corps de ce document présente l'analyse du modèle de données et les processus de construction de la représentation des jeux de données.
Les annexes de ce document présentent avec quelques détails les algorithmes de machine leanrning supervisés et non supervisés mis en œuvre dans cette étude.

2 Formulation du problème : classification multi-étiquettes

L'ensemble des questions sélectionnées pour l'étude forment le **corpus** du modèle de données.

Une question est pour partie formulée en langage naturel. Identifiées par des balises dédiées, des parties d'une question peuvent être exprimées dans un autre langage comme un langage de programmation, un langage de description des données.

Un tag associé à une question est emprunté au domaine technique auquel la question se réfère. Un n'appartient pas nécessairement au langage naturel.

Le problème posé est formulé comme celui de classifier des questions en fonction des tags qui leurs sont attribués. Cette classification est ensuite utilisée pour suggérer les tags plus pertinents à de nouvelles questions. Du fait qu'une question peut se voir attribuer plusieurs tags, la classification est ici abordée sous l'angle d'une **classification multi-étiquettes**.

3 Standardisation du modèle de données

Le processus de standardisation permet de décrire tous les POST avec le même nombre de caractéristiques.

3.1 Notations et définitions

Pour la suite de l'exposé on note :

- Description ou Body : la partie détaillée d'une question

- Titre ou Title: le titre de la question
- POST : une question et/ou une question + le titre associé à la question.
- TAG_S : les tags suggérés par un algorithme mise en œuvre dans cette étude;
- TAG_A : les tags assignés à un POST par un utilisateur. Ils sont récupérés de la base de données de Stackoverflow.
- $NTAG_{S,A}$:le nombre de tags suggérés qui correspondent aux tags assignés.
- TAG_R : les tags de référence sont tous les tags référencés dans la base de données de Stackoverflow
- $NTAG_S$: le nombre de tags suggérés pour un POST
- $NTAG_A$: le nombre de tags assignés à un POST
- $NTAG_E$: le nombre de tags assigné à un POST par un système pressenti expert. Le système pressenti expert utilisé dans l'étude est l'algorithme Word2Vec.
- N_{POST} : Le nombre de POST

Pour la suite de l'exposé, le mot POST sera employé en place de question et/ou question+titre.

3.2 Standardisation : traitement NLP

Chaque POST est constitué :

- d'une description
- d'un titre
- d'un ensemble de TAG_A , tags assignés par l'utilisateur.

Le traitement NLP de standardisation décrit ci-dessous n'est pas appliqué aux TAG_A.
Ce traitement consiste à ne retenir que les mots des titre et description contenant de l'information significative.
Ce traitement consiste en :

- le filtrage des caractères de ponctuation en préservant les caractères de ponctuation accolés à un mot;
- le filtrage des mots liés à la description de méta-données comme les balises du langage XML
- le filtrage des caractères isolés non alpha-numériques.
- Le filtrage des caractères numériques pris comme mots
- le filtrage des verbes
- le filtrage des mots pris dans un dictionnaire « stopword » disponibles dans la librairie NLTK.
- le filtrage des mots appartenant aux formules de politesse ; les POST sont une forme de demande d'aide ; les formules de politesse améliorent le retour des aidant sans apporter d'information au sujet.
- le filtrage des séquences de mots des langages dit « machines » ;
- le filtrage des balises de description XML.

Du fait des traitements implémentés dans les algorithmes mis en œuvre dans cette étude, les procédés de filtrage lexicaux par lexémisation et « stemming » n'ont pas été appliqués au corpus.

Le modèle de données obtenu est une suite de mots, chacun d'eux portant une **information significative** sur le POST duquel ils sont extraits.

Les résultats des algorithmes alimentés par ces données normalisés vont dépendre de façon critique de cette étape.

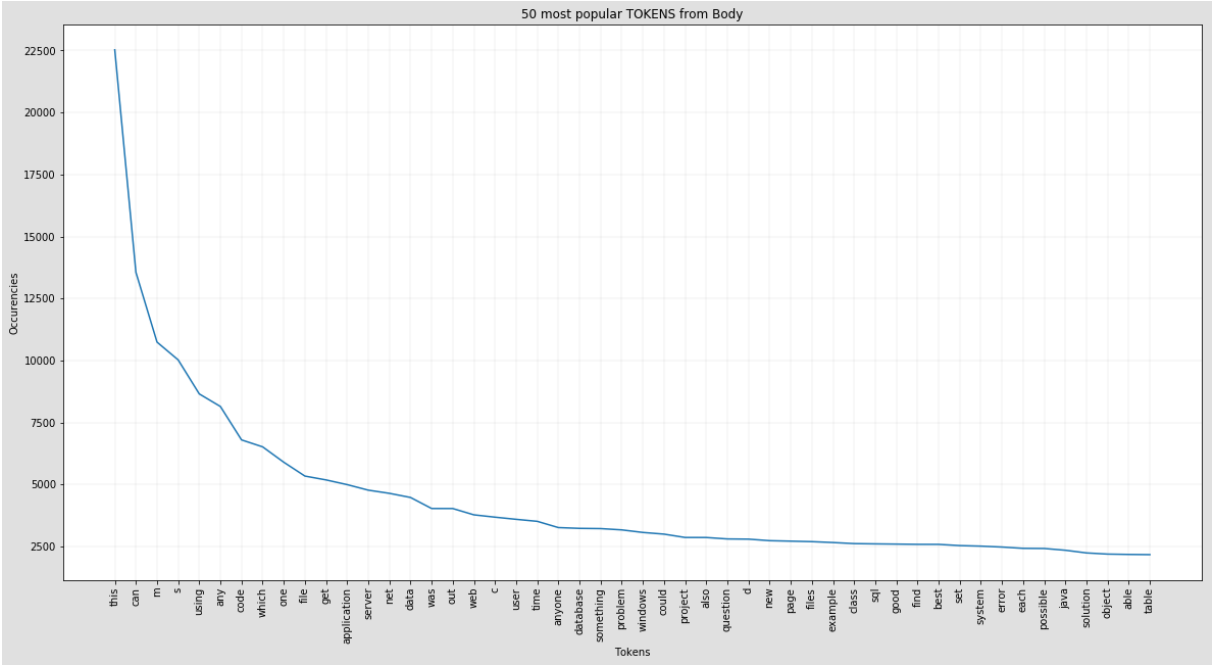
3.3 Tokenization : word embedding

Cette étape consiste, pour chaque POST du corpus, à le discrétiser en une suite de « tokens ». Ce processus, encore nommé « word embedding », est interprétable comme le plongement d'un POST sur le vocabulaire du corpus. Cette suite de Tokens peut être apparentée à une suite de mots dérivés du langage naturel. Pour rappel, le langage naturel est celui dans lequel les description et titre ont été exprimés.

3.4 Analyse du modèle standardisé

L'analyse consiste à étudier les descriptions titres et tags du corpus.

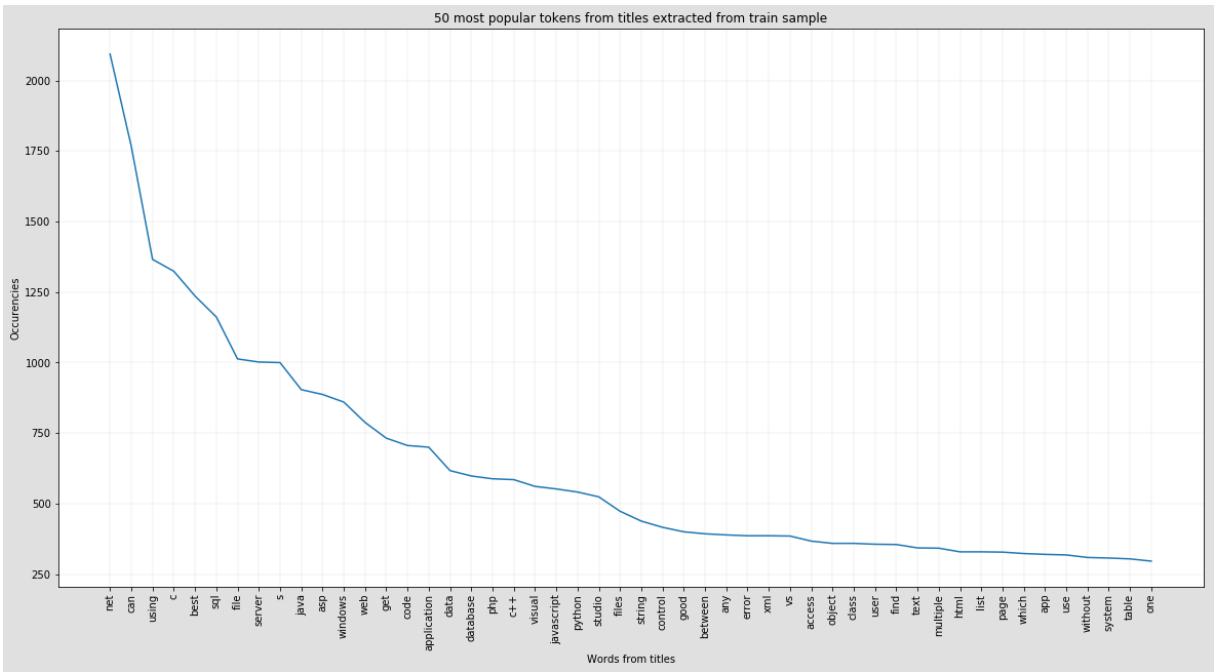
3.4.1 Fréquence des tokens issus des descriptions de l'ensemble des POST



Ce diagramme présente les 50 Tokens les plus populaires issus du processus de standardisation représentant les descriptions du dataset. En appelant VOCAB l'ensemble de tous les TOKEN du corpus de POST, on obtient les grandeurs statistiques suivantes :

- $P(TAG_R|VOCAB_{Body})=0.18$: Le vocabulaire du corpus contient près de 20 % des TAG de référence.
- $P(VOCAB_{Body} \cap TAG_R|TAG_R)=0.14$: Près de 15 % des TAG de référence se trouvent dans le vocabulaire de la description. Cette seconde statistique est indicative. Le dataset sur lequel cette mesure a été faite contient 25K POST, contre près de 50K TAG_R

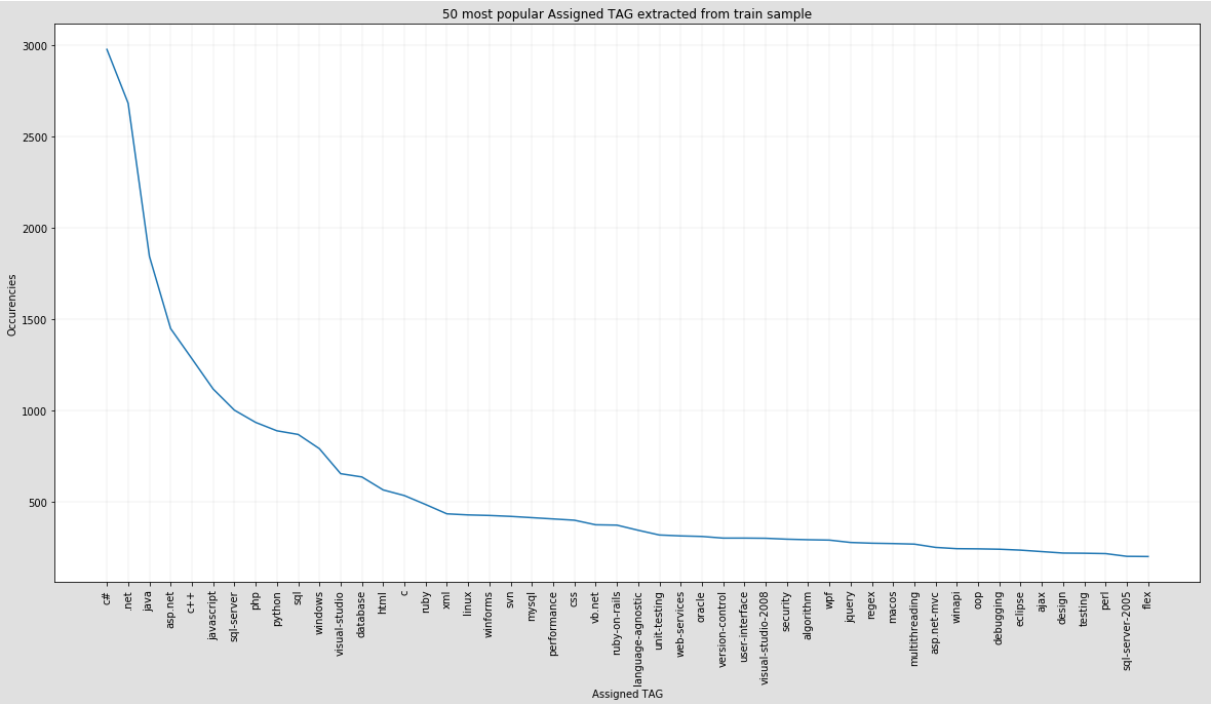
3.4.2 Fréquence des tokens issus des titres de l'ensemble des POST



- $P(TAG_R|VOCAB_{Title})=0.40$: Le vocabulaire du corpus de titres contient plus de 40 % des TAG_R .

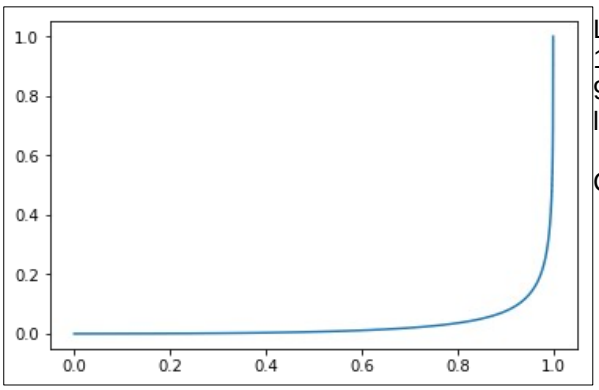
- $P(VOCAB_{Title} \cap TAG_R|TAG_R)=0.1$: Moins de 10 % des TAG de référence se trouvent dans le vocabulaire décrivant les Titres des POST. Cette mesure de probabilité sur les TOKEN des titres est à ramener à la mesure du même type sur les TOKEN des descriptions.

3.4.3 Fréquence des TAG assignés issus de l'ensemble des POST



- $P(TAG_R|TAG_A)=0.93$:pour l'échantillon de 25K POST, la distribution des TAG_R sur les TAG_A n'est pas complète. Certains TAG assignés issus du corpus n'appartiennent pas à la liste des TAG de référence de Stack Over Flow. Lors de l'entraînement du modèle avec des algorihtmes de machine learning, 7 % des TAG de référence ne pourront être suggérés.

- $P(TAG_A \cap TAG_R|TAG_R)=0.12$: sur ce même échantillon, les TAG assignés ne représentent que 10 % des TAG de référence. Ce faible taux n'a pas d'impact important sur la représentativité de l'échantillon, au vu des courbes de Lorenz et des coefficients de GINI décrits ci-dessous.

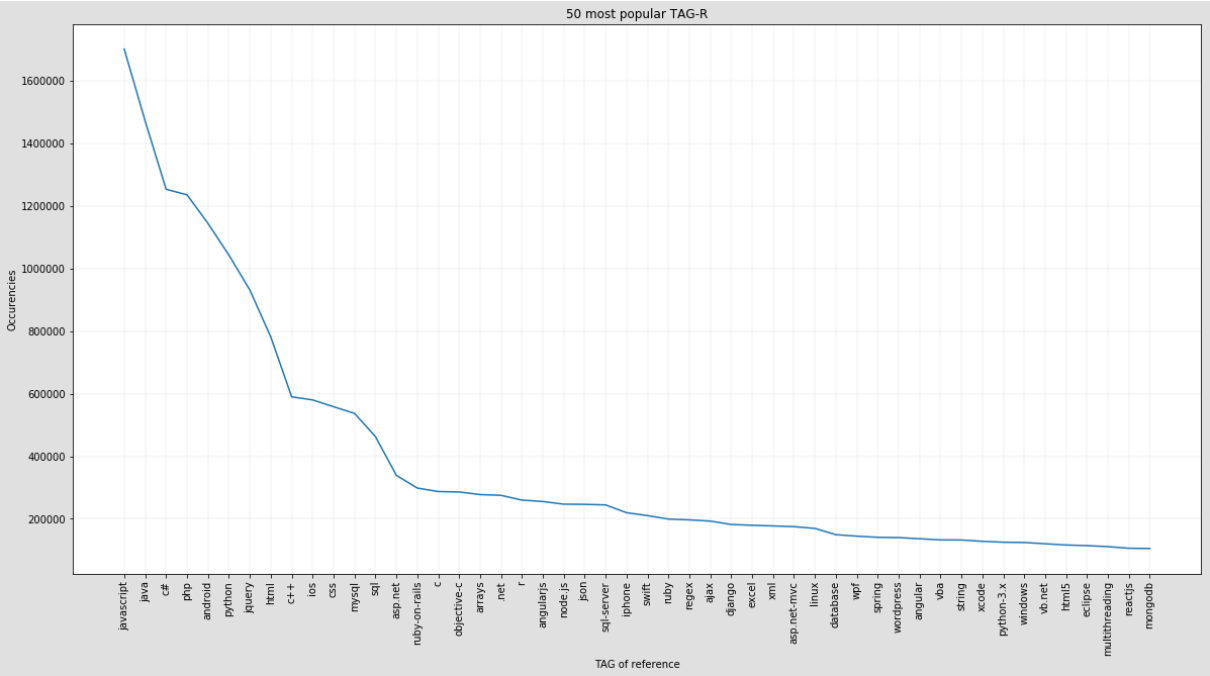


Courbe de Lorenz pour les TAG_A de 25K POST

La courbe ci-contre montre qu'à peine plus de 10 % des TAG assignés représentent près de 90 % de l'ensemble des TAG assignés dans tous les POST de l'échantillon.

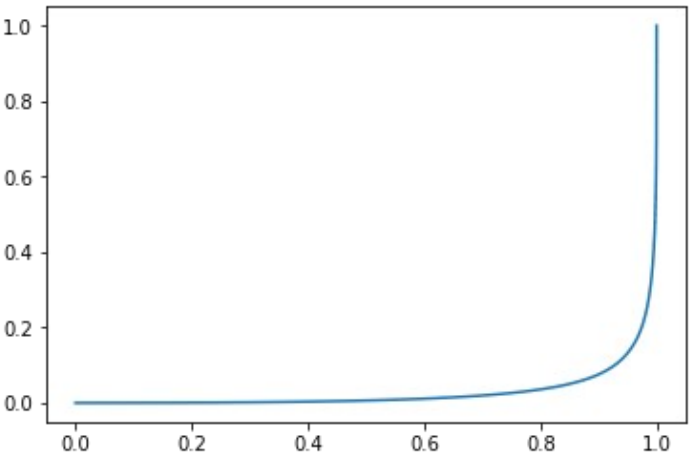
Coefficient de GINI : 0.91

3.4.4 Fréquence des TAG de référence



La courbe de Lorenz ci-contre montre que 10 % des TAG de référence représentent plus de 90 % de l'ensemble des TAG assignés à l'ensemble des POST insérés dans la base de données de Stackoverflow.

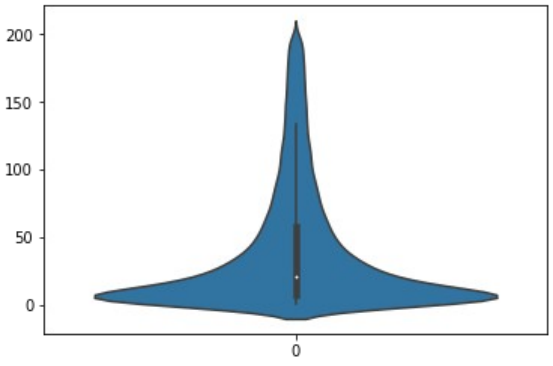
Coefficient de GINI : 0.94



Courbe de Lorenz des TAG de référence

3.4.5 Distribution des TAG assignés sur l'échantillon d'entraînement

Le diagramme de distribution et de densité des TAG_A assignés montre un profil de distribution de forme identique aux TAG_R



Distribution et densité des TAG_A

Ordre de popularité	TAG_A	TAG_R
1	C#	JAVASCRIPT
2	NET	JAVA
3	ASP NET	C#
4	C++	PHP
5	JAVASCRIPT	ANDROID
6	SQL SERVER	PYTHON
7	PHP	HTML
8	PYTHON	C++
9	SQL	MYSQL
10	WINDOWS	SQL

Le diagramme des occurrences des TAG_A dans l'échantillon a la même allure que celui des TAG_R. Il est cependant noté que les TAG les plus populaires sont dissemblables. Ce fait peut s'interpréter comme un **effet d'obsolescence des technologies** les plus populaires. La liste des TAG_A est issue de POST moins récents que ceux référencés dans la liste de référence TAG_R.

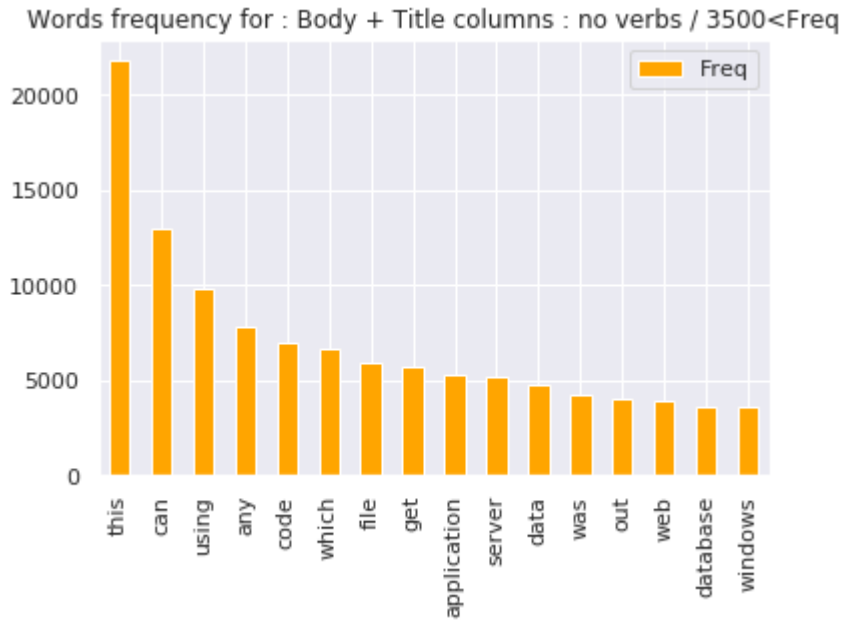
4 Analyse du modèle de Stackoverflow

4.1 Analyse des distribution des données du modèle

Les distributions des TAG_R de référence et des TAG_A assignés montrent des profils de courbes semblables. Si on suppose que les TAG assignés par les utilisateurs reflètent les thèmes abordés dans les descriptions des questions, ce fait témoigne de l'invariance de la loi de distribution des POST dans le temps.

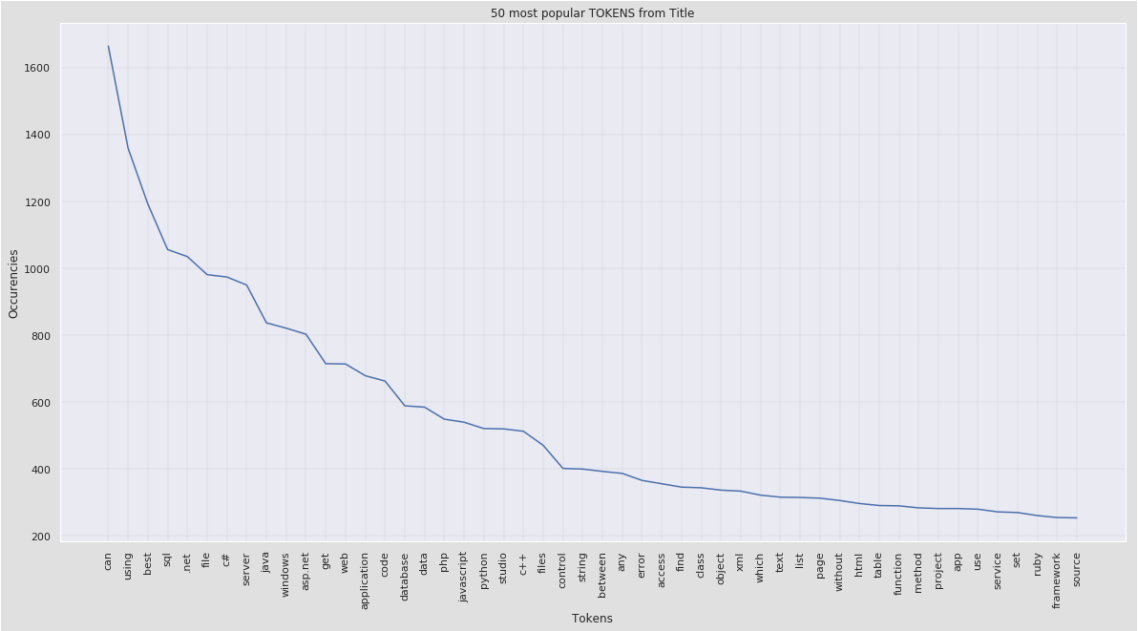
Cependant, on constate une dissemblance entre les valeurs des TAG les plus populaires. Ceci s'interprète comme le fait que l'échantillon choisie ne représente pas les POST en cours.

La courbe de distribution des TOKEN a une allure similaire à celle de la distribution des TAG assignés. Ce fait met en évidence la corrélation entre les TAG assignés aux POST et la description des POST.



On constate que la standardisation fait apparaître plus fréquemment es mots du langage naturel que les termes techniques.

Par ailleurs, les diagrammes ci-dessous mettent en évidence le fait que les titres contiennent plus de TOKEN issus du langage technique que les descriptions. Ces TOKEN sont potentiellement des TAG suggérés.



La représentation en cloud word de la tranche de fréquence des TOKEN issus des descriptions.

L’ambiguïté entre les TOKEN issus du langage naturel et les termes techniques conduisent à sur-représenter les premiers comme CAN, à la fois verbe et protocole industriel.

4.2 Tests d’inférence statistique

Les tests de Kolmogorov/Smirnov et de Shapiro/Wilk donnent une p-value proche de 0.
L’hypothèse nulle étant peu vraisemblable, nous prenons le parti de mener cette étude en considérant que les sources à l’origine des données ne sont pas Gaussiennes.
Ce parti-pris a pour incidence d’écarter l’algorihme génératif GMM pour la mise en œuvre d’une méthode de classification et de suggestion de tags par clustering.

5 Vectorisation : représentation numérique du corpus

Chaque POST du corpus issu du processus de normalisation est représenté par une suite de mots, des tokens.

L'étape qui suit consiste à obtenir une représentation numérique des POST du corpus. Pour ce faire, le corpus standardisé est transformé en une table de nombre réels, $[M_{IJ}]$. Les lignes $I \in \{1,..,N\}$ de cette table identifient chacune un POST et chaque colonne $J \in \{1,..,K\}$ identifie chacune un mot du **vocabulaire** du corpus. Le vocabulaire du corpus est un sous-ensemble de l'ensemble de mots uniques issus de l'ensemble des POST du corpus.

Les éléments M_{IJ} sont des nombres réels obtenus par des techniques de comptage, de co-occurence ou de TF-IDF. Ces deux dernières techniques sont brièvement décrites dans les sections qui suivent.

En résultat de ce processus, chaque POST est représenté numériquement sous la forme d'un « vecteur » dans « l'espace » de mots du vocabulaire du corpus.

Les termes « matrice », « espace » et « vecteur » sont utilisés ici en abus de langage. Le vocabulaire du corpus n'engendre pas, à priori, « d'espace vectoriel » au sens propre de ce terme. Dans la suite, les termes « matrice » et « table » seront utilisés sans distinction.

Cette représentation numérique du corpus va permettre d'appliquer à ce dernier des algorithmes pour résoudre le problème tels que formulé en 2.

5.1 Numérisation par co-occurrences

La co-occurence consiste à numériser en POST en considérant un tuple de mots. On fait l'hypothèse que les tuples de mots ont une relation sémantique au sens du problème de classification formulé en 2.

5.1.1 Cas particulier : co-occurence de deux mots du corpus

L'expression idiomatique « expression régulière » prend un sens singulier pour le corpus étudié. Ce couple de mots étant un idiomatisme du corpus, il sera vraisemblablement présent dans plusieurs POST. On obtient une mesure de cette vraisemblance en calculant la probabilité d'observer ce couple dans cet ordre : $P(Reguliere|Expression) = \frac{P(Expression, Reguliere)}{P(Expression) * P(Reguliere)}$

Qui s'interprète comme la probabilité d'observer le mot « Reguliere » devant le mot « Expression » : c'est la fréquence du couple de mots pris dans n'importe quel ordre, divisée par la fréquence d'apparition de chacun des mots seuls composant l'expression. Dans ce calcul de vraisemblance, le mot «Expression» est un paramètre de l'observation du mot «Reguliere».

La valeur de cette probabilité est corrélée au sens de cette expression dans le corpus.

5.2 Généralisation de la co-occurence

En généralisant, on calcul ainsi la co-occurence de 1 à M mots.
Pour un tuple de M mots (MOT1) , la vraisemblance d'un mot sachant les autres mots est le produit des vraisemblances :

$$P(MOT_M | MOT_{1,...}, MOT_{M-1}) = \prod_{I=1}^{I=M-1} \frac{P(MOT_M, MOT_I)}{P(MOT_M) * P(MOT_I)}$$

5.3 Numérisation TF-IDF

Dans le cas d'un token d'un POST à numériser, la technique TF-IDF consiste à pondérer la fréquence d'apparition de ce token dans un POST par un terme inverse de sa similarité dans les autres POST du corpus.

Plus le token apparaît dans d’autres POST, moins il est significatif pour le POST en question. Et inversement, moins il apparaît dans les autres POST, plus il caractérise le POST en question.
Ce principe peut être généralisé a n’importe quel tuple de mots.

Un TF-IDF (1,2) représentera numériquement les POST du corpus à la fois par des mots uniques numérisés et les couples de mots numérisés.

6 Conclusions

La performance des résultats est fortement dépendante du type de mesure de précision adoptée. La méthode consistant à comparer mot à mot les TAG suggérés aux TAG assignés est drastique. Il peut être envisagé d'utiliser l'algorithme Fuzzy-wuzzy pour réaliser des mesures de précision.

La pertinence des résultats des algorithmes supervisés dépendent étroitement de la capacité des utilisateurs à assigner des tags aux questions qu'ils postent.
L'utilisation des algorithmes non supervisés, permettent de s'abstraire de ce biais potentiel. En effet, ces algorithmes sont focalisés sur des données non structurées sans tenir compte à priori des tags apposés par les utilisateurs. De par ce fait, l'utilisation d'un algorithme comme W2VEC peut tenir rang de système expert pour évaluer la pertinence des tags apposés par les utilisateurs. Il permet aussi, d'affiner l'évaluation des différentes méthodes utilisées en permettant, en plus du calcul de précision, le calcul du « recall ».

La performance des modèles pourrait être améliorée en évaluant des jeux de données avec un nombre de POST bien plus important. D'une façon générale, les algorithmes de machine learning sont sensibles à la fois à la complexité des modèles de données et à leur dimension.

La relative faible performance des méthodes mettant en œuvre l'optimisation des marges de séparation pour la classification peuvent s'expliquer par :

- le fait que ces méthodes mettent en œuvre des méthodes de gradient conjugué, sensibles à la colinéarité des vecteurs formant la matrice (POST X Features). La meilleur performance de l'algorithme de régression logistique, qui, par le coefficient de régularisation, atténue les effets de la colinéarité.
- Le problème de classification n'est pas complètement linéaire ; l'optimisation des erreurs induites par le cas non séparable ne suffit pas pour obtenir une classification performante.

La faiblesse de performance du modèle W2VEC s'explique de par le faible nombre d'observations mis en œuvre pour entraîner le modèle. Les réseaux de neurones, dont fait partie W2VEC, nécessitent un volume de données important pour les entraîner.

Dans le domaine des technologies de l'information, la mise en œuvre de procédés de traitement du langage naturel présente des caractères délicats. Les traitements de la ponctuation, de la lemmatization ou des caractères non alpha-numériques doivent être entrepris avec circonspection sous peine de détruire de l'information utile dans le jeux de données aux prix de la dégradation des performances des modèles de machine learning.

ANNEXES

DESCRIPTION DES MÉTHODES MISES EN ŒUVRE

7 Annexes : Benchmark des algorithmes

7.1 Méthodologie de benchmark

Les mêmes dataset d'entraînement et de tests sont utilisés pour l'ensemble des méthodes mises en œuvre.
Pour l'algorithme W2VEC, le dataset standardisé est utilisé. Pour les autres algorithmes, les représentations numériques des dataset d'entraînement et de test sont utilisés.

7.1.1 Calcul de la précision

Pour chacun des algorithmes mis en œuvre, un calcul de précision est réalisé sur un ensemble de données pour lesquels les TAG_A assignés sont connus.

On définit la précision pour un POST par la formule :

- $P_{POST} = \frac{NTAG_{SA}}{NTAG_A} = \frac{N\{TAG_A \cap TAG_S\}}{NTAG_A}$

On définit le recall pour un POST par la formule :

- $R_{POST} = P_{POST} = \frac{NTAG_{SA}}{NTAG_E} = \frac{N\{TAG_A \cap TAG_S\}}{NTAG_E}$

La précision moyenne est obtenue par :

- $P = (\sum_{POST} P_{POST}) / N_{POST}$

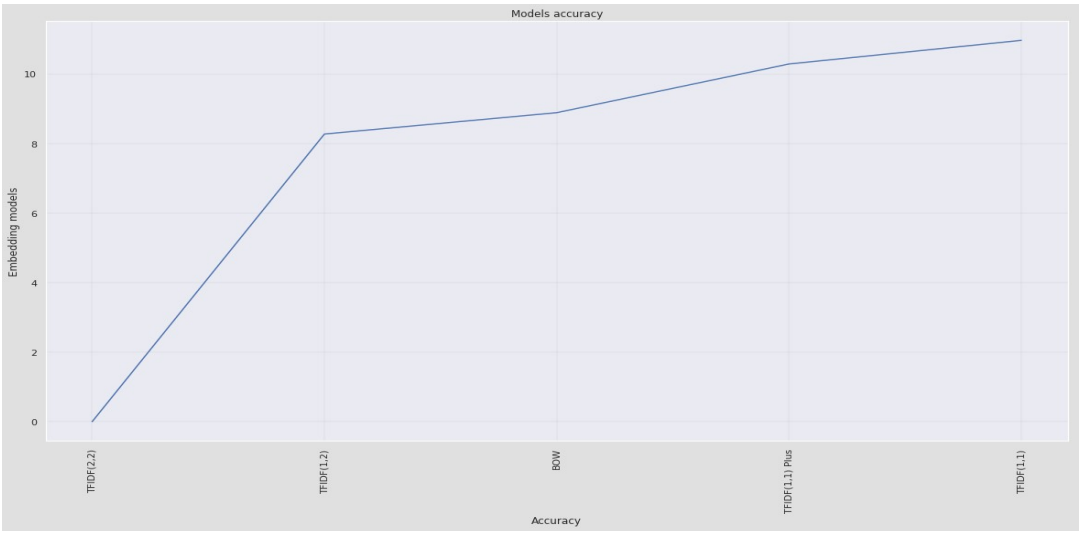
Il devient alors possible de comparer, pour un même jeux de données, les différentes performances de chacun des algorithmes évalué.

7.2 Modèles statistiques

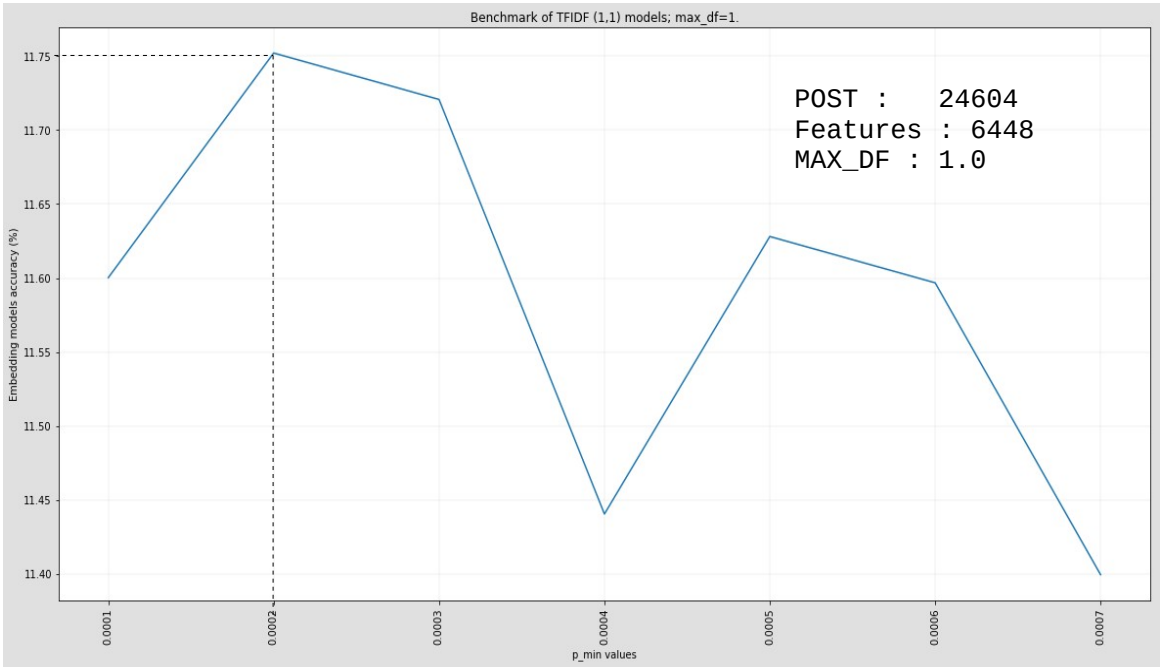
Les poids associés à chacun des mots de chaque document sont calculés par une mesure statistique discrète.
Les mesures expérimentées ici sont TF-IDF et l'occurence.

7.2.1 Modèle statistique basé sur TF-IDF

Dans ce modèle, un POST est représenté par un sous ensemble de ses composantes. Ces dernières sont choisies parmi les N valeurs les plus grandes.
En adoptant différentes valeurs de N-GRAM pour la numérisation, on obtient le benchmark suivant :



Le meilleur score de précision moyenne est obtenue pour le TF-IDF (1,1) avec un peu plus de 10 % de précision.
En faisant varier l'hyper-paramètre min_df de TF-IDF (1,1) le diagramme ci-dessous met en évidence une précision optimale pour la valeur de min_df = 2E-4 et max_df=1. Les résultats ci-dessous nous permettent de sélectionner le meilleur modèle TF-IDF pour la représentation numérique du corpus.



7.3 Les méthodes non supervisées

7.3.1 Représentation Bag Of Words des POST

Dans le cadre de cette étude, tout POST est représenté par un ensemble de mots issus d'un dictionnaire dans le modèle « Bag Of Word ». Dans ce modèle, la position des mots est indépendante de la représentation d'un document par ces mots.

On noter cette représentation : POST= $\{W_1,..,W_N\}$

7.3.2 Méthodes des clusters

Les POST, représentés sous forme de matrice a valeurs réelles, sont regroupés par similarité. La distance de similarité dépend de l'algorithme de clustering mise en œuvre.

7.3.2.1 Clustering par K-means

Cet algorithme met en œuvre l'optimisation d'une fonction quadratique qui modélise la distance moyenne de chacun des points d'un ensemble de données au centre de chacun des clusters.

Ainsi, pour un point P_p de coordonnées X, et un ensemble de clusters K donné de centroïdes $\{\mu_1,...,\mu_K\}$, la fonction suivante est optimisée :

- $f(P,K)=\sum_K \sum_{P\in K} (X_P-\mu_K)^2$ qui revient a calculer la plus petite distance de chaque point à chacun des centroïdes. Un point est déplacé d'un cluster K vers un cluster K' si $f(P,K')$ réalise un minimum.

7.3.2.2 Algorithme d'extraction des TAG du clustering : voisinage

La vectorisation du corpus est réalisée par le calcul TF-IDF. Chaque valeur de TF-IDF assignée à un mot d'un document a donc un caractère global de par le facteur IDF.
Ainsi on est en droit de faire l'hypothèse que deux documents similaires appartiendront au même cluster et, inversement, que deux documents d'un même cluster sont similaires.

Pour un POST donné, les TAG suggérés sont récupérés parmi les POST du cluster les plus similaires au POST donné.

Pour un POST donné, la suggestion de TAG issus d'un cluster K, soit $TAG_{S,K}$, consiste à :

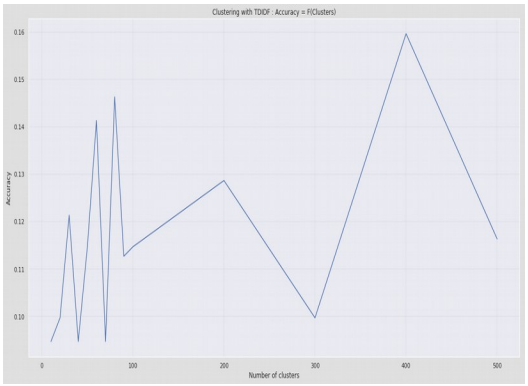
- Assigner un cluster au POST
- Calculer la distance du POST au centroïd μ_K : $d_{POST,K}=d(POST,\mu_K)$
- Récupérer, parmi les POST du cluster K, :
  - les N_{INF} POST dont la distance est inférieure à $d_{POST,K}$
  - les N_{SUP} POST dont la distance est supérieure à $d_{POST,K}$
- Les TAG suggérés sont chacun des mots des POST N_{INF} et N_{SUP} dont les valeurs TFIDF sont les plus grandes.

En prenant en compte les documents du cluster dans un voisinage et en sélectionnant 1 mot caractéristique à chacun des documents du voisinage, la notion de contexte d'un mot, supposé inhérent à sa sémantique, est introduite dans ce processus.

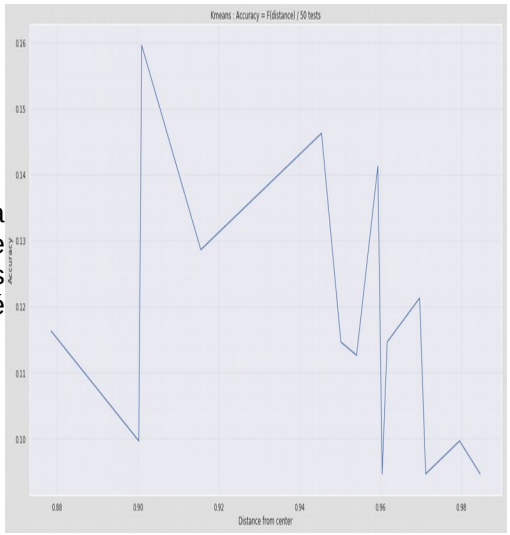
7.3.2.3 Justification de la méthode d'extraction

Le centroïde d'un cluster peut s'interpréter comme la correspondance au thème dominant du cluster. Les thèmes d'un POST sont dominés par des combinaisons des TAG assignés. Ces derniers sont largement empruntés au domaine technique plutôt qu'au langage naturel.
Le calcul de la proximité de deux documents d'un même cluster fait intervenir des TOKEN du vocabulaire qui ne génèrent pas nécessairement des thèmes de documents.
Le calcul d'extraction des TAG par la distance au centroïde privilégie la proximité des documents par leur thématique.

7.3.2.4 Résultats de la méthode de clustering



La courbe ci-contre montre l'évolution de la précision telle définie précédemment, ce, pour un nombre de clusters allant de 10 à 500, sérialisés comme la suite : 10,20,30,40,50,60,70,80,90,100,200,300,400,500.



La courbe ci-contre met en évidence la sensibilité de la précision à la distance du centre du cluster. Pour chaque cluster, pour chaque POST de l'échantillon de test, les moyennes des distances et des précisions ont été calculées.
Cette évolution semble indiquer que les TOKEN représentent d'autant mieux un POST qu'ils sont proches du centre d'un cluster.
Cette décroissance est d'autant plus marquée que le nombre de clusters est grand, donc, que le nombre de POST par cluster est faible.

7.3.2.5 Autres algorithmes de clustering

Cette méthode peut être généralisé à tout type de clustering, y compris GMM. Dans ce dernier cas, la distance $d(POST, \mu_K)$ pourra être remplacée par la moyenne et la variance gaussienne.

7.3.3 Méthode basée sur le processus LDA (Latent Dirichlet Allocation)

7.3.3.1 L'approche Bayesienne

Il existe des thèmes dans le corpus de POST et ces thèmes ne sont pas observables directement. Ils sont dits latents. Ces thèmes se trouvent assignés de façon aléatoire aux POST. Cette incertitude suit une loi de probabilité. On s'intéresse à la distribution de cette incertitude. Les mots composant les documents du corpus, sont eux, observables, donc, certains.

L'approche Bayesienne permet d'espérer trouver une loi de distribution à posteriori des thèmes latents sur les POST et sur les mots, à partir de la loi de distribution à priori des thèmes sur les mots des documents.

Dans une approche Bayesienne, le processus LDA va permettre de révéler les thèmes latents dans les documents du corpus à partir de l'observation des mots.

Les hypothèses propres à l'approche Bayesienne sont supposées satisfaites, à savoir :

- pour chaque POST, la probabilité d'y observer un thème T_i, soit $P(T_i|Post)$ est une variable aléatoire. La distribution de ces probabilités $P(T_i|Post)$ décrit l'incertitude d'observer chaque thème T_i dans le corpus.
- Les POST sont générés de façon indépendantes et suivent tous la même loi de distribution. On admet alors : $P(T|Post) \propto P(Post|T) * P(T)$ où :
  - $P(T|Post)$ est la probabilité à priori d'observer un thème T assigné à un POST Post.
  - $P(Post|T)$ est la vraisemblance d'observer un POST dans un thème donné
  - $P(T)$ est la probabilité à priori d'observer un thème.

Étant donné que plusieurs thèmes peuvent être assignés à chaque POST, ces relations se déclinent sur des lois de distribution multivariées :

- $P(T_1, T_{2,\ldots}, T_t|Post_{1,\ldots}, Post_d) \propto P(Post_{1,\ldots}, Post_d|T_1, T_{2,\ldots}, T_t) * P(T_1, T_{2,\ldots}, T_t)$

Comme chacun des $POST_P$ est représenté par une suite de mots $\{W_{p1,..}, W_{pN}\}$ désordonnée, on fait l'hypothèse supplémentaire que la loi jointe $P(W_{p1,..}, W_{pN})$ est invariante à toutes les permutations de la suite $\{W_{p1,..}, W_{pN}\}$.

7.3.3.2 Le processus LDA

Ce processus est mis en œuvre pour révéler la distribution de thèmes sur chaque mot du vocabulaire du corpus (de POST), et par composition, la distribution des thèmes sur chacun des POST.

La distribution de Dirichlet est utilisée dans le processus LDA du fait que ce dernier, par construction, nécessite d'échantillonner aléatoirement des distributions de thèmes sur des mots (distributions à priori). Ces échantillons suivent des lois de distribution choisies à bon escient. La distribution de Dirichlet se révèle donc particulièrement adaptée pour décrire des distributions de variables aléatoires qui sont elles même des distributions de lois de probabilités.

Le processus de Dirichlet consiste à représenter comme des distributions de Dirichlet :

- l'assignation des thèmes à chacun des POST du corpus :
  - $P(Theme|POST) = \Theta \sim Dirichlet(\alpha)$ ce, pour chacun des POST du corpus.
- l'assignation des mots d'un document sur chacun des thèmes du corpus :
  - $P(Mot|Theme) = \beta \sim Dirichlet(\eta)$

Le problème de la découverte des thèmes dans un corpus de POST dépend :

- des paramètres α et η qui sont deux distributions de variables aléatoires ;
- des mots observés : W.

Pour ce faire, deux tables, représentant des distributions de Dirichlet, sont construites de façon itérative :

- Pour chaque document du corpus, on calcul une distribution de mots du vocabulaire, notée η avec $\eta \sim Poisson(\xi)$;
- Pour chacun des thèmes $T_k avec\, k \in 1,..,K\, et\, K\, fixé$:
  - L'assignation des mots du vocabulaire $W_i avec\, i \in 1,...,N$ aux thèmes du corpus est calculée comme une distribution de Dirichlet : $P(W_i|T_k) = \beta_{ki} \sim Dirichlet(\eta)$. Cette distribution de Dirichlet est une distribution de distributions au sens où η représente la distribution des mots dans un document.
- Pour chaque document d du corpus :
  - $P(T|_{Doc})$: cette distribution est calculée selon une loi de Dirichlet(α). On obtient une table de d lignes représentant les documents du corpus et de t colonnes représentant les thèmes présents dans le corpus. Les lignes I de cette matrice, représentent les probabilités d'observer les thèmes T dans un document d. Elles sont calculées comme des distributions de Dirichlet de paramètre α : $P(Doc_I | _T) = \Pi_{k=1}^{k=t} [P^{\alpha-1}(Doc_I | T_K)]$
  - Pour chaque mot W_{di} d'un document d :
    -
  - $P(W|_T)$: distribution suivant une loi de Dirichlet de paramètre β . De par l'hypothèse Bayesienne (i.i.d conditionnellement à T multivariée) l'optimisation du calcul de cette distribution revient à optimiser le calcul de $P(T|_W)$ qui suit une loi Dirichlet(β). On obtient une table de t lignes représentant et de N colonnes, N représentant le nombre de mots du dans le vocabulaire du corpus. Pour chacune des colonne J, on a :
    $P(T_J | _W) = \Pi_{k=1}^{k=N} [P^{\beta-1}(T_J | W_k)]$

LDA : Le modèle de Dirichlet appliqué à l'émergence des thèmes latents.

Ce modèle probabiliste est dit génératif en ce sens qu'il permet de faire émerger une structure thématique dans un corpus (des thèmes du corpus).

On considère que la distribution exacte des thèmes sur les documents du corpus suit une loi de Dirichlet de paramètre α notée : Dirichlet(α) .

Le paramètre α représente la distribution moyenne de la probabilité $P(T|_{Doc})$ d'observer un thème T dans l'ensemble des documents du corpus.
Plus la valeur du paramètre α est grande, plus grande sera $P(T|_{Doc})$ pour chaque thème. Ce qui se traduira par le fait que la distribution des thèmes sur l'ensemble des documents sera d'autant plus multivariée.

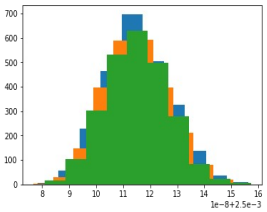


Illustration 1: Distribution des thèmes 100, 200 et 399 sur le vocabulaire

Le nombre de thèmes observables par document augmente avec le paramètre α.
Un document Doc d'indice d étant représenté par un ensemble de mots W issus du vocabulaire du corpus, on considère que la distribution des thèmes dans sur les mots d'un document d, W_d, distribution notée $P(T|_{Wd})$ suit elle, une loi de Dirichlet de paramètre β , notée Dirichlet(β). Ce paramètre, β, suit une loi de distribution de Poisson de paramètre η .

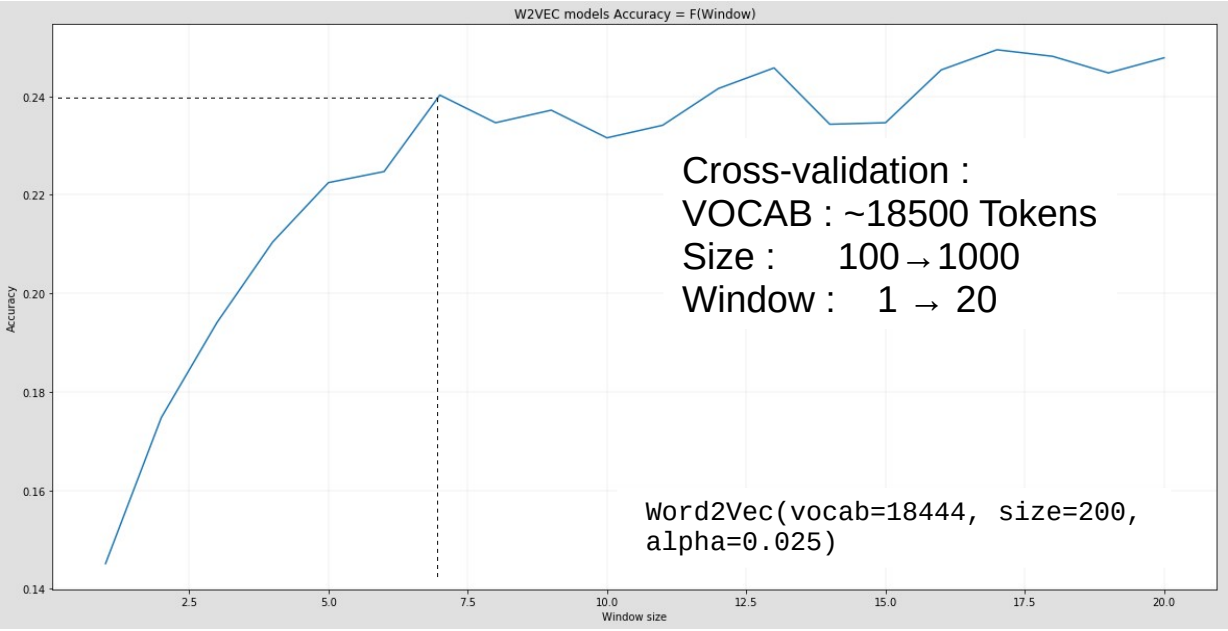
7.3.4 Méthode Word2Vec

Cet modèle projette des mots d'un corpus dans un espace vectoriel. Il devient alors possible de réaliser des opérations algébriques sur les mots.
Le modèle Word2Vec met en œuvre un réseau de neurones à 2 couches. Le principe de cet algorithme est d'être entraîné pour reconnaître un mot à partir d'un contexte ou inversement.

Le contexte d'un mot peut être représenté de deux façons :

- En Continuous Bag Of Words, le contexte d'un mot est représenté par un ensemble de mots pris dans un ordre quelconque. L'algorithme permet de déterminer un mot à partir de son contexte.
- En Continuous skipgram, l'algorithme permet de déterminer un contexte de mots à partir d'un mot désigné. Le contexte du mot désigné est représenté par un ensemble de mots pris dans un certain ordre. Les coordonnées des mots du contexte ont une magnétude d'autant plus importante que les mots du contexte sont à proximité du mot désigné.

Skip-gram est plus pertinent pour prédire des mots peu fréquents. Ce mode est utilisé pour définir une matrice de vecteurs alimentant un algorithme de M.L.



7.4 les méthodes supervisées

7.4.1 Cas de la classification mutli-variée

Chacun des algorithmes mis en œuvre réalisent une classification binaire par rapport à une classe : un TAG est assigné à une classe ou pas. Pour obtenir une classification d'un POST sur plusieurs TAG, chacun des classificateurs mis en œuvre est composé avec un opérateur « One versus Rest ». On réalise ainsi une classification successive d'un POST sur plusieurs TAG.

7.4.2 Méthodes Bayesiennes

Les méthodes basées sur le théorème de Bayes avec l'hypothèse naïve d'indépendance des variables explicatives pour une classe donnée, consistent à calculer des probabilités à posteriori $P(TAG_i|POST)$ à partir d'estimations de $P(POST|TAG_i)$ et $P(TAG_i)$.

On utilisera les fonctions de distribution de Bernoulli 7.4.5 et de Gauss 7.4.6 pour calculer ces densités de probabilités.

7.4.3 Multinomial Naive Bayes

La représentation numérique du corpus est réalisée avec l'opérateur TF-IDF. Pour rappel, dans cette représentation, l'ordre d'apparition des mots dans un document n'est pas pris en compte. C'est une représentation de type « Bag Of Words », encore notée BOW.

Pour un POST donné, on calcule la probabilité la plus élevée d'observer un TAG pour ce POST. Cette probabilité s'exprime par la relation :

- $P(TAG|POST)=\prod_{TAG_i \in \{TAG\}} P(TAG_i|POST)$ où $TAG_i, i \in [1,..,T]$ où on suppose que les observations TAG_i indépendantes les unes des autres, d'après les hypothèses Bayesiennes.

Sous ces conditions, on a la probabilité conditionnelle :

- $P(TAG_i|POST)=\frac{P(POST|TAG_i)*P(TAG_i)}{P(POST)} \propto \prod_{TAG \in \{TAG_i\}} P(POST|TAG_i)*P(TAG_i)$

Le terme $P(TAG_i)$ est calculable en estimant la fréquence d'apparition du TAG_i dans le corpus et de $P(POST|TAG_i)$ en calculant, pour le POST la fréquence de TAG_i .

C'est l'hypothèse i.i.d conditionnellement à *TAG*, qui donne à cette approche un caractère dit naïf. Il est supposé ici que chacun des TAG est indépendant des autres. Considérant cette simplification, un TAG est assigné à un POST par le classificateur naïf de Bayes indépendamment des autres TAG.

Dans le cadre du problème abordé ici, cette hypothèse ne va pas de soi. Ainsi, pour les trois TAG :

LINUX,
IOCTL,
OS,
C++

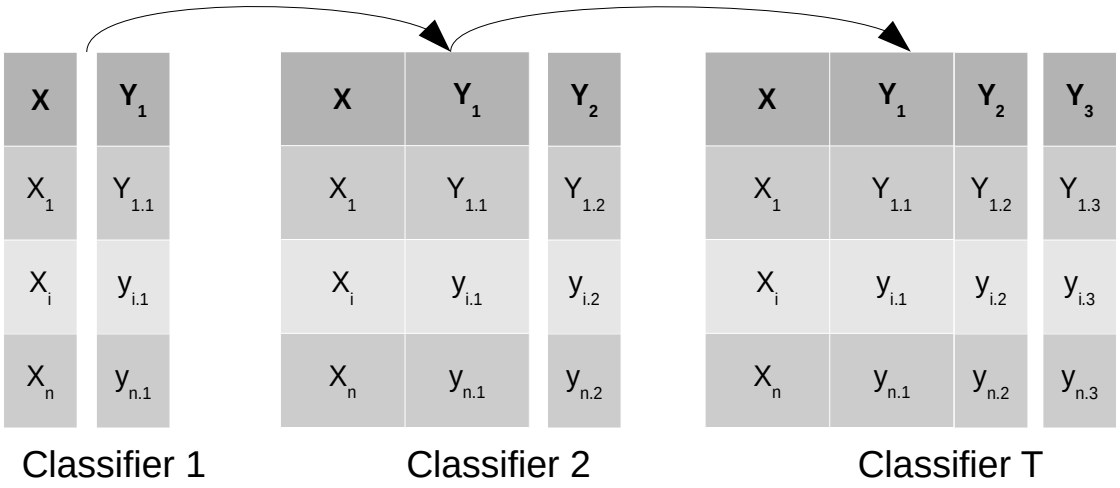
les combinaisons de ces TAG en :

LINUX, OS
LINUX, IOCTL
LINUX, C++

référencent des POST qui n'appartiennent pas au même domaine technique. C'est la combinaison des TAG qui permet de préciser le domaine dans lequel un POST s'inscrit. De par cet exemple, une combinaison de TAG s'inscrit dans un contexte sémantique pour décrire un document mettant ainsi en évidence une relation de dépendance (une corrélation) entre les TAG.

7.4.4 Chained multinomial Naive Bayse

Cet algorithme reprend les principes de la classification Naive Bayes en l'aménageant de façon à prendre en compte les liens de dépendance entre les TAG assignés.



Dans l'exemple de la classification d'un document en un nombre de TAG, T, un nombre égale T de classificateurs vont être chaînés en augmentant la dimension de la matrice du classificateur N+1 avec le vecteur TAG du classificateur N lors de son entraînement. Le classificateur final prend en compte la liaison entre tous les TAG d'un POST.

Dans le cas ou la corrélation entre les TAG est avérée, ce classificateur donnera de meilleurs résultats que le cas du classificateur du même nom non chaîné.

7.4.5 Bernoulli Naïve Bayes

Ce modèle s'appuie sur les hypothèses des lois conjointes naïves de Bayse décrit en 7.4.2 avec une fonction de décision de Bernoulli. Cette dernière prend en compte l'absence d'observation d'une variable explicative dans le calcul de la vraisemblance.

7.4.6 Gauss Naïve Bayes

La fonction de distribution pour estimer la vraisemblance tels que définie en 7.4.2 est la fonction de distribution de Gauss : $P(POST_i|TAG) = \frac{1}{\sigma_{TAG}\sqrt{2\pi}} e^{-\frac{(POST_i-\mu)^2}{2\sigma_{TAG}^2}}$ d'écart type σ_{TAG} et centré en μ.

7.4.7 Classification par la regression logistique

On fait ici l'hypothèse que le problème est linéairement séparable.
Cet algorithme de classification est dit linéaire au sens où la fonction de décision, qui permet de décider de l'appartenance d'une variable aléatoire à une classe ou pas, est obtenue en appliquant une fonction dite **logistique** à une combinaison linéaire des vecteurs du modèle de données. Cette fonction résultante traduit la plus grande probabilité d'observer l'appartenance d'un vecteur POST à une classe, ici, représentée par un TAG : $P(TAG|POST)$. **L'assignation d'un TAG à un POST revient à calculer la vraisemblance d'observer ce TAG pour ce POST**.

Les paramètres $[\beta_j]$ de cette combinaison linéaire sont « appris » avec les TAG_A , tags assignés du data-set d'entraînement. Cet apprentissage consiste à résoudre en β , un système linéaire.

Les coefficients $[\beta_j]$ de cette combinaison linéaire sont calculés en résolvant le système :

- $[TFIDF_{ij}]*[\beta_j]+\lambda g([\beta_j])=[TAG_i]$ où :
  - g est une fonction dite de régularisation des coefficients du vecteur $[\beta_j]$,
  - λ un hyper-paramètre du modèle, permettant d'en réduire la complexité (régulation de Lasso, Ridge ou Elastic net),
  - $[TFIDF_{ij}]$ est l'expression qui dérive des coefficients de la matrice TFIDF calculée en 5.3 dont le nombre de lignes correspond au nombre de POST et le nombre de colonnes correspond au nombre de TOKEN du vocabulaire du corpus.
  - $[TAG_i]$ est le vecteur cible, le tag assigné correspondant au $I^{ème}$ POST

Ce système n'ayant pas de solution analytique, il est résolu en $[\beta_j]$ par la méthode du **gradient conjugué**. La solution est obtenue par un processus itératif. Les valeurs intermédiaires à chaque itération représentent une ligne de niveau d'une fonction convexe. La ligne de niveau à l'étape suivante est atteinte en suivant la direction opposée au plus fort gradient. En fin du processus itératif, si ce dernier a **convergé**, la solution réalise alors un **minimum** de la fonction convexe du système à résoudre.

7.4.8 SGD Classifier : SVM

On fait ici l'hypothèse que le problème de classification n'est pas linéairement séparable. La classification des vecteurs de part et d'autre d'une marge définie par la fonction de décision réalise des erreurs. La fonction générale à optimiser intègre aussi l'optimisation d'erreur de classification. Cette fonction, dite de perte, minimisant l'erreur de classification est la fonction de **Hinge**. Le problème est de type **SVM**, consistant à optimiser (avec la fonction d'erreur) la largeur de la marge de séparation entre les classes de façon « souple », i.e, en imposant des contraintes quant à la position des vecteurs de support par rapport aux bords de la marge de séparation. En introduisant des multiplicateurs de Lagrange pour prendre en compte les contraintes formulées par le problème, on résout un problème dual en optimisant la fonction Lagrangienne du système. Dans ce cas de figure, le problème est résolue dans l'espace dit **dual**.
De par les dimensions du problème à résoudre, (lignes,colonnes)~(25K, 5K) le problème est résolu dans sa formulation **primale**.

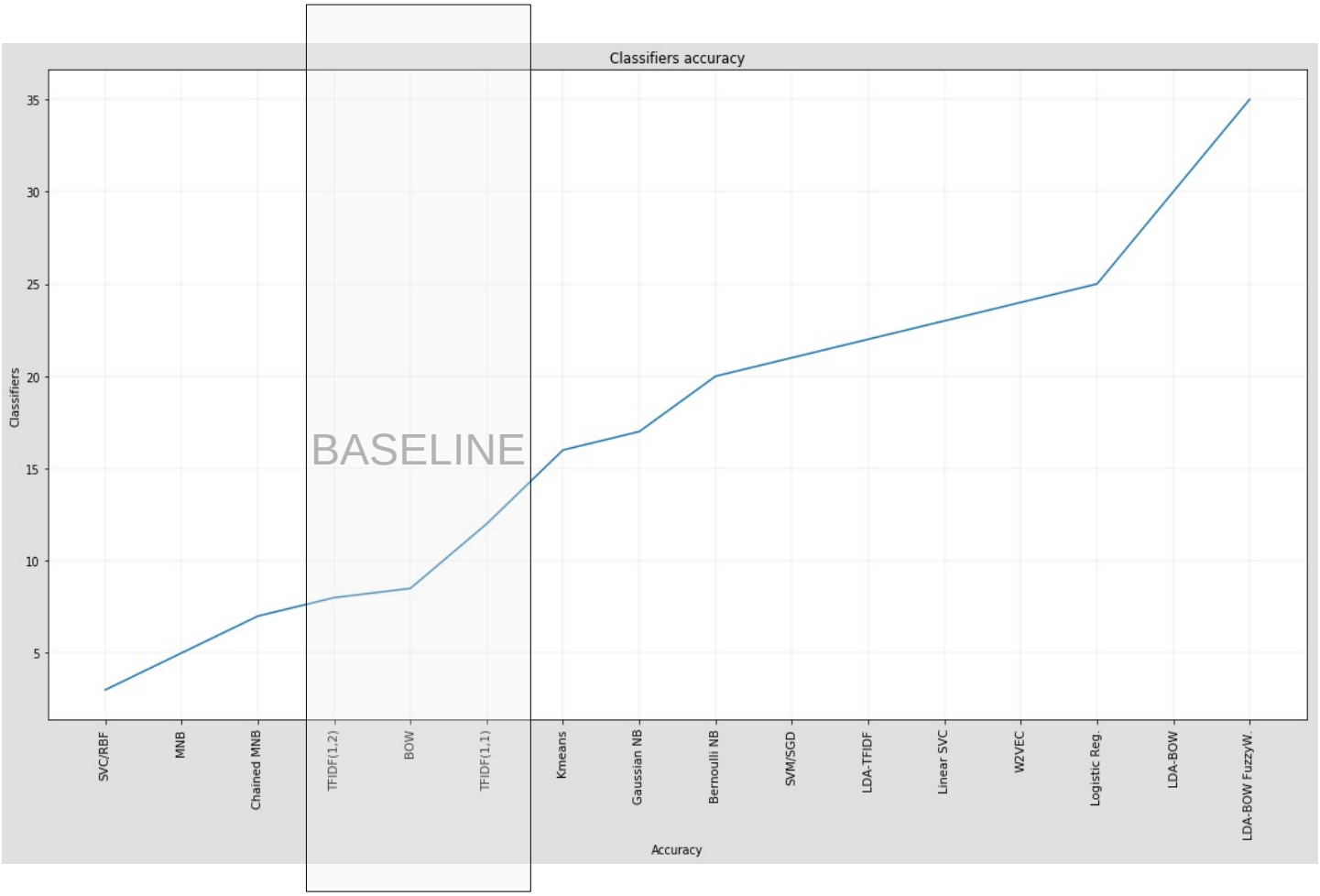
Comme la régression logistique, un hyper-paramètre de régularisation, nommé α , est mis en œuvre pour minimiser l'erreur de classification en diminuant la complexité du problème.
Une **recherche par grille** de ce paramètre nous permet d'obtenir le meilleur modèle de classification de type SVM.

Pour des problèmes de grande complexité, le **gradient stochastique** est utilisé. Cette méthode échantillonne aléatoirement un ensemble de vecteurs auxquels sont appliqués l'algorithme du gradient conjugué. Pour des données de taille importante, cette méthode de résolution se révèle plus efficace que celle du gradient conjugué.

7.4.9 SVC linéaire

Cette méthode est basée sur l'optimisation d'une marge de séparation de part et d'autre d'une fonction de décision linéaire et pour lequel le noyau issue de la formulation mathématiques pour minimiser la fonction de coût est linéaire.

8 Benchamark des méthodes mises en œuvre.



Les méthodes à noyaux non linéaires ne permettent pas d'obtenir de bonnes performances. Ces méthodes permettent de résoudre des problèmes dans des espaces fonctionnels de Hilbert sans avoir besoin de se projeter dans ces espaces. Ils mettent en œuvre des méthodes linéaires exprimées en substituant les produits scalaires par des fonctions définies positives, des noyaux.
La plus faible performance de l'utilisation d'un noyau met en évidence le caractère linéaire du problème. Ce trait est souligné par le fait qu'en général, les méthodes de classification supervisées linéaires (SVC linéaire, classification par régression logistique, SVM/SGD) donnent de meilleurs résultats que Gaussian NB, Bernoulli NB. Pour ces dernières méthodes, les vraisemblances $P(POST|TAG)$ sont supposées suivre des lois de distribution respectivement de Gauss et de Bernoulli.
Les performances du classifieur Multinomial Naive Bayse indiquent que les hypothèses de Bayes ne sont pas pertinentes pour ce type de classification.