



Mémoire de certification

Conduire un projet de sciences de données

RS 1527

Challenge Kaggle : Quora Pairs

Charroux Alain

Data Science SAP3

DS SAP3 – 2019/2020

Introduction

Quora Questions Pairs est un des challenge proposés par Kaggle.

Quora est une société web qui se propose de répondre à tout genre de question . Le volume traité est important (300 millions d'utilisateurs par mois). Le business model de Quora est fondé sur la publicité en ligne (les questions posées permettent de cibler efficacement les publicités). La concurrence étant importante, il est crucial pour Quora de continuer à capter ses utilisateurs et donc de pouvoir répondre très rapidement et de façon pertinente à toute question. Une voie naturelle consiste à réutiliser les réponses déjà fournies et validées et donc à identifier automatiquement une question déjà posée.

C'est l'objet de ce challenge : en n'ayant à sa disposition que le texte de paires de questions, identifier les paires de questions identiques.

Ce challenge faisant l'objet d'un prix, les compétiteurs ont été nombreux (plus de 3000 équipes et 20000 soumissions) et fait preuve d'une grande créativité avec parfois d'importants moyens. On peut essayer de clustérer les solutions proposées en:

- Feature engineering 'simple' non sémantique
 - nb de mots,
 - longueurs des mots,
 - fréquence des caractères,
 - toutes sortes de métriques sur les questions ..
- Features complexes sémantiques issues du NLP (Natural Language Processing)
 - Essentiellement, utilisation de bags of words, tfidf, doc2vec,... pour créer une signature vectorielle de chaque question
 - Calculs de distance/proximité entre les vecteurs obtenus
- Utilisation de ces features par des modèles à bases de réseaux neuronaux
- Méthodes ensemblistes pour mettre en compétition plusieurs modèles

S'il est relativement facile avec des méthodes 'simples' d'avoir un résultat moyen à ce challenge, elles plafonnent très vite et se placer dans le peloton de tête est difficile. Les équipes gagnantes ont par exemple, utilisé des réseaux neuronaux très avancés ou mis en compétition plus de 500 modèles...

La stratégie adoptée pour construire le modèle prédictif a été volontairement humble et itérative :

1. Choix d'une feature non sémantique pour constituer un modèle 'naïf' de référence. La feature choisie est volontairement simple et facile à appréhender.
2. Exploration et ajout de variantes autour de cette feature élémentaire
3. Ajout de features sémantiques plus complexes.

Les modèles à bases de réseaux de neurones n'ont pas été envisagés, faute de compétence. De même, faute de temps, une approche ensembliste n'a pas été testée.

Description du challenge

Données fournies

Training

Le dataset de training comporte 404290 lignes et 6 colonnes

Champ	Type	Description	Example
id	entier	N° de paire	404286
qid1	entier	N° unique de question	18840
qid2	entier	N° unique de question	155606
question1	texte	Texte complet de la question 1	Do you believe there is life after death?
question2	texte	Texte complet de la question 2	Is it true that there is life after death?
is_duplicate	Entier : 1 ou 0	La variable target	1

Test¹

Le dataset de test comporte 2345796 lignes et 3 colonnes :

Champ	Type	Description	Example
Test_id	entier	N° de paire	2345793
question1	texte	Texte complet de la question 1	What are some famous Romanian drinks (alcoholic & non-alcoholic)?
question2	texte	Texte complet de la question 2	Can a non-alcoholic restaurant be a huge success?

Evaluation

Les soumissions sont évaluées en calculant le [logloss](#) entre les valeurs prédites et les valeurs exactes. Plus cette valeur est petite, meilleur est donc le score.

Autres informations

- Toutes les données de training sont des données réelles venant de Quora. Les questions sont en anglais.
- Le champ *is_duplicate* est le fruit d'un '*consensus raisonnable*' entre experts mais n'est pas exact à 100%.
- Les données de challenge sont un mélange inconnu de données réelles et de données générées. Rien n'est dit sur la façon de générer ces lignes artificielles et elles ne sont pas comptées dans le score.

¹ Par la suite, on parlera du dataset du *challenge* pour éviter la confusion avec le dataset de test qu'il est naturel de constituer à partir du dataset de training.

La compétition

- 3295 équipes
- 20815 soumissions
- Meilleur score : 0.112770. Pire score : 28.5 (!)

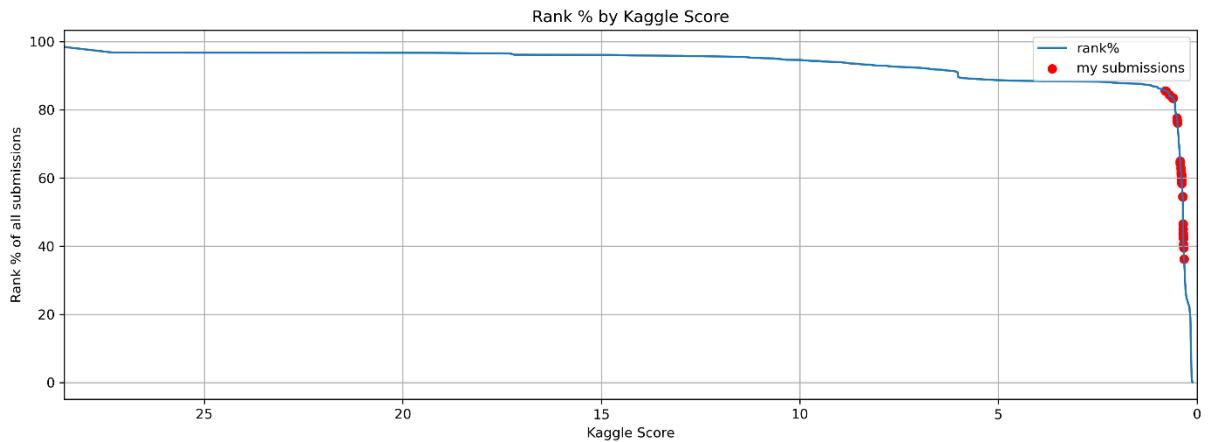


Figure 1 Toutes les soumissions

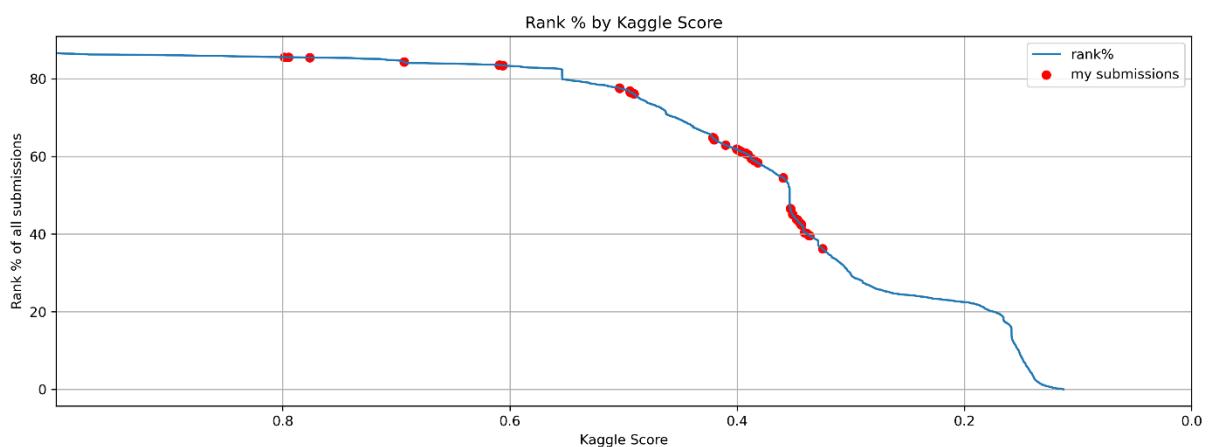


Figure 2 Les soumissions avec un logloss<1

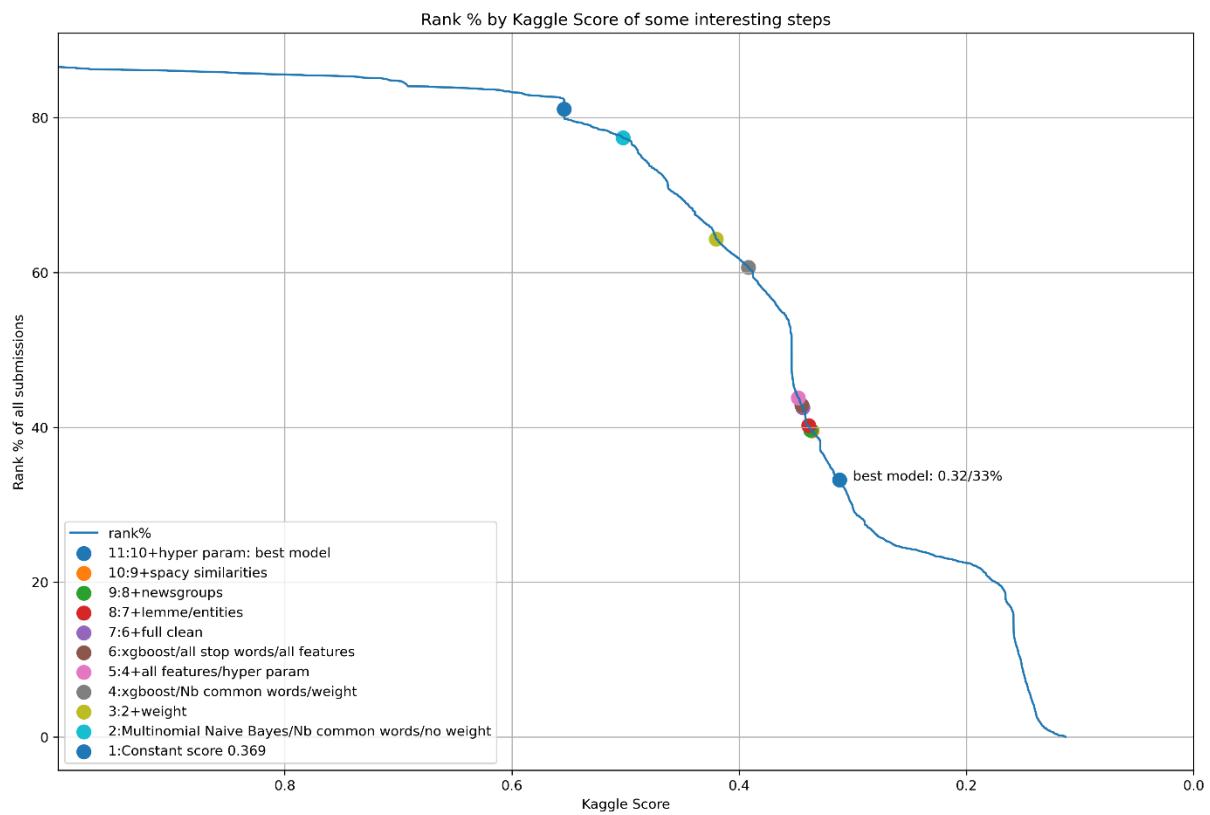


Figure 3 Quelques étapes remarquables

Exploration et analyse des datasets

Exploration du corpus de textes

Le dataset de training fournit environ 800 000 questions. Le dataset de challenge fournit 4 600 000 questions.

Une exploration manuelle du dataset de training permet de comprendre quelques caractéristiques de base sur le contenu des questions :

- Elles sont en général courtes.
- Il y a des fautes d'orthographe et des erreurs de frappe
- Les sujets sont très variés.

	question1
0	What is the step by step guide to invest in share market in india?
1	What is the story of Kohinoor (Koh-i-Noor) Diamond?
2	How can I increase the speed of my internet connection while using a VPN?
3	Why am I mentally very lonely? How can I solve it?
4	Which one dissolve in water quickly sugar, salt, methane and carbon di oxide?
5	Astrology: I am a Capricorn Sun Cap moon and cap rising...what does that say about me?
6	Should I buy tiago?
7	How can I be a good geologist?
8	When do you use シ instead of ツ?
9	Motorola (company): Can I hack my Charter Motorola DCX3400?

Figure 4 Training: 10 premières questions

Avec une telle volumétrie, il n'est pas facile d'appréhender le contenu de ce corpus : il n'y a pas de thématique principale. Les sujets sont très variés, le contexte apporté par chaque question très limité et le vocabulaire global très large.

Visualiser et se familiariser avec un tel corpus de textes est en soi un challenge.

Nous avons essayé d'explorer cet ensemble de questions en utilisant quelques outils standards de NLP (*Natural Language Processing*)

Utilisation d'informations sémantiques

Aucune information de haut niveau (quels sont les sujets les plus cités ?, sur quoi ou qui s'interroge-t'on ?,...) n'est disponible. Il faut soit :

- l'injecter en utilisant une source externe. On va essayer d'identifier le newsgroup auquel appartiendrait la question au moyen d'un modèle externe
- la déduire à partir du contenu des questions. On va essayer de détecter les entités nommées.

Newsgroups

Le modèle newsgroups (cf. Annexe) a permis de trouver le newsgroup le plus probable (*parmi une sélection limitée*) pour chaque question. En appliquant un threshold sur la probabilité d'identification de 0.9 , on obtient la répartition suivante:

newsgroup	% train newsgroup	% challenge newsgroup
computers	34,136047	36,265086
forsale	0,071982	0,098607

politics	13,483624	11,766493
religion	17,318326	15,61031
science	29,728757	28,8925
sport	2,558649	3,992037
vehicles	2,702614	3,374967

Figure 5 Répartition des newsgroups identifiés sur train et challenge

Ceci fournit un autre moyen de comparer le contenu des 2 datasets :

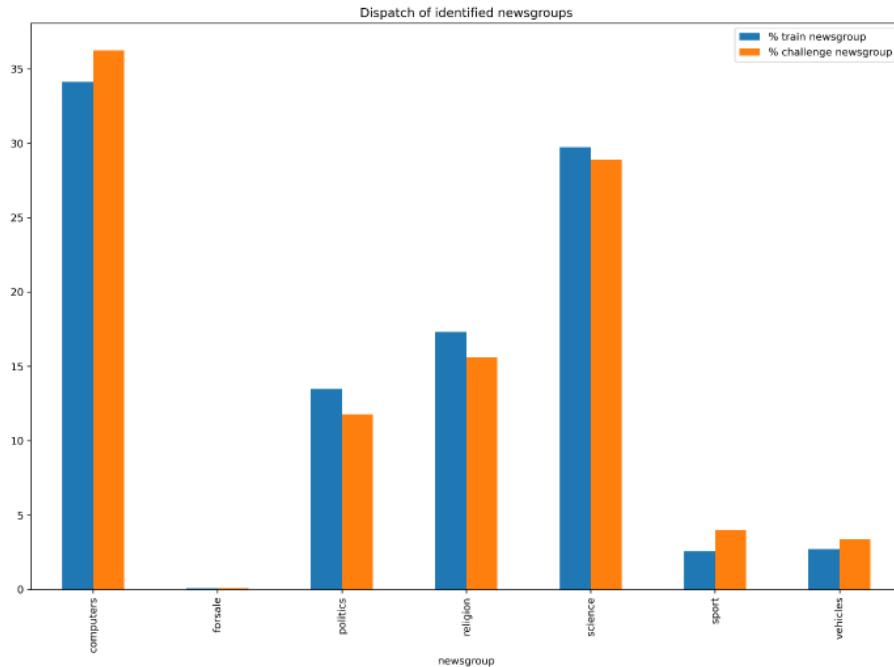


Figure 6 Distribution des newsgroups identifiés

Un test de *kolmogorov-Smirnow* le confirme ($p\text{-value} > 0.96$) : la distribution des newsgroups (quand on est capable de les identifier) est identique dans le dataset de training et le dataset de challenge.

Entités nommées (NERP)

Des outils utilisant des réseaux de neurones entraînés sur des volumes de textes très importants sont disponibles et facilement utilisables. Ils procèdent à une analyse très détaillée d'un texte et, entre autres, produisent une liste des *entités remarquables*. La librairie *spacy* détecte :

Code Spacy	Description	% train	% challenge
GPE	Countries, cities, states	9,89	9,53
PERSON	People, including fictional	9,88	9,59
PRODUCT	Objects, vehicles, foods, etc. (not services)	0,44	0,46
ORG	Companies, agencies, institutions, etc.	6,19	6,96
DATE	Absolute or relative dates or periods	6,60	6,72
NORP	Nationalities or religious or political groups	4,28	4,43
WORK_OF_ART	Titles of books, songs, etc.	0,04	0,03
LANGUAGE	Any named language	0,94	1,29
EVENT	Named hurricanes, battles, wars, sports events, etc.	0,40	0,34
FAC	Buildings, airports, highways, bridges, etc.	0,23	0,24
LOC	Non-GPE locations, mountain ranges, bodies of water	1,02	0,90
NA	No entity detected	60,09	59,49

Figure 7 Liste des entités nommées reconnues par spacy

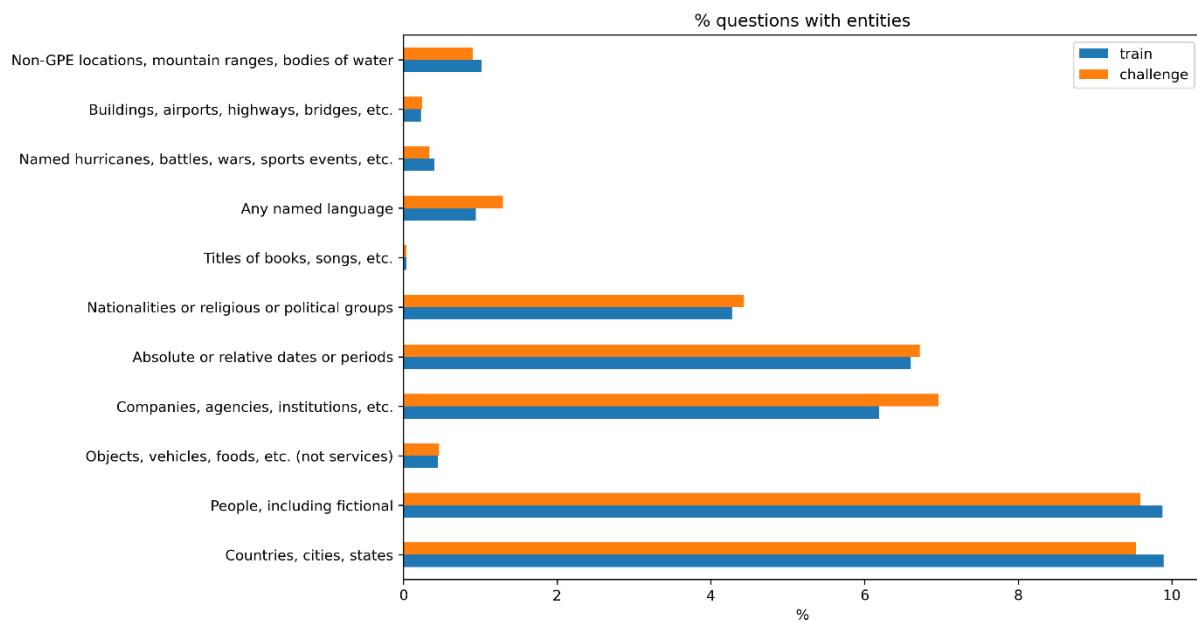


Figure 8 Répartition des catégories d'entité

Un test de Kolmogorov-Smirnov ($p\text{-value}>0.99$) confirme que la distribution des entités (celles que l'on est capable d'identifier) est la même dans le dataset de training et de challenge.

Nuages de mots

Les 2 outils précédents permettent de se faire une idée du contenu global des questions mais l'information fournie est loin d'être complète. Par exemple, pour 60 % des questions, il n'a pas été possible de détecter une entité nommée.

Les nuages de mots permettent de visualiser les mots présents dans un corpus de textes ainsi que leur fréquence : plus le mot est fréquent, plus il est gros dans le diagramme. Ces diagrammes sont censés fournir une bonne vision globale même si il faut en pratique se fixer un nombre relativement restreint de mots à afficher.

Nuage de mot sans filtre

Afficher brutalement un nuage sur tout le corpus non filtré est facile mais ne rend guère de service.

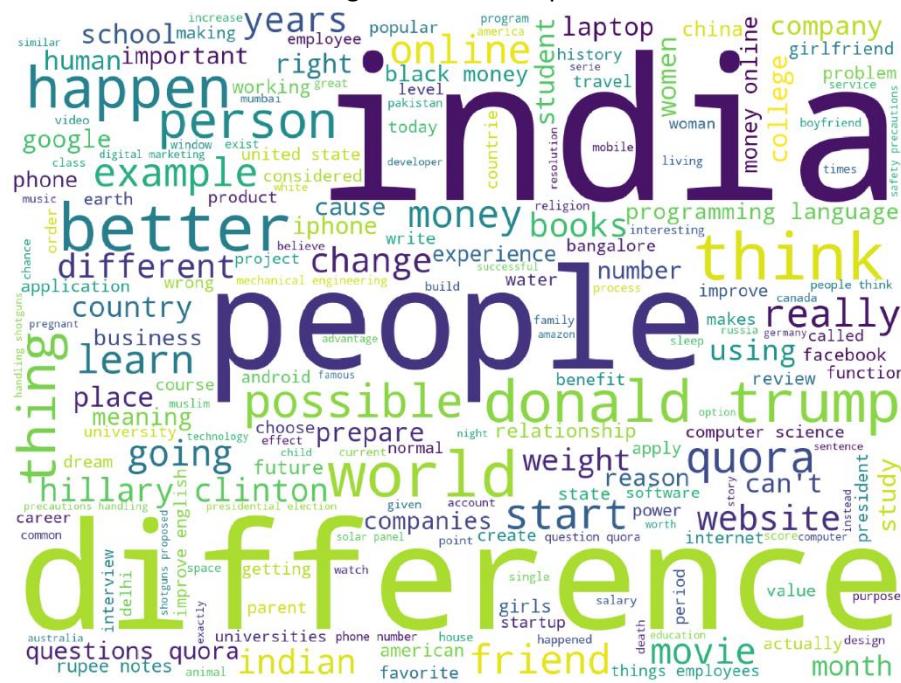


Figure 9 wordcloud: aucun filtre

On comprend tout de même quelques faits basiques :

- Quora est très utilisé en Inde
 - Les données datent de 2017 : Trump, Clinton sont des sujets chauds

Si l'on génère le même nuage à partir de questions nettoyées et lemmatisées de façon à réduire drastiquement le vocabulaire, on obtient à peine mieux :



Figure 10 wordcloud Training : question1 nettoyée et lemmatisée

On peut au moins tenter de comparer visuellement avec les données de challenge et essayer de se convaincre que les 2 datasets utilisent le même vocabulaire :

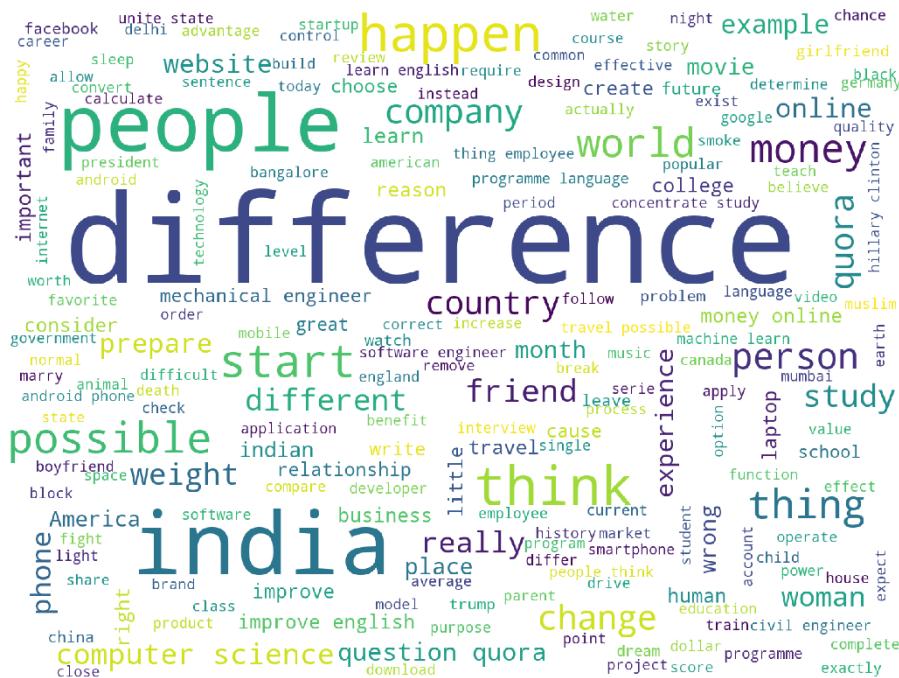


Figure 11 wordcloud Challenge: question1 nettoyée et lemmatisée

Filtrage par l'utilisation de données sémantiques de haut niveau

On peut également tenter de filtrer le dataset avec les informations trouvées précédemment et de se focaliser sur un thème donné.

Par exemple, un nuage de mots sur le thème '*politics*' :

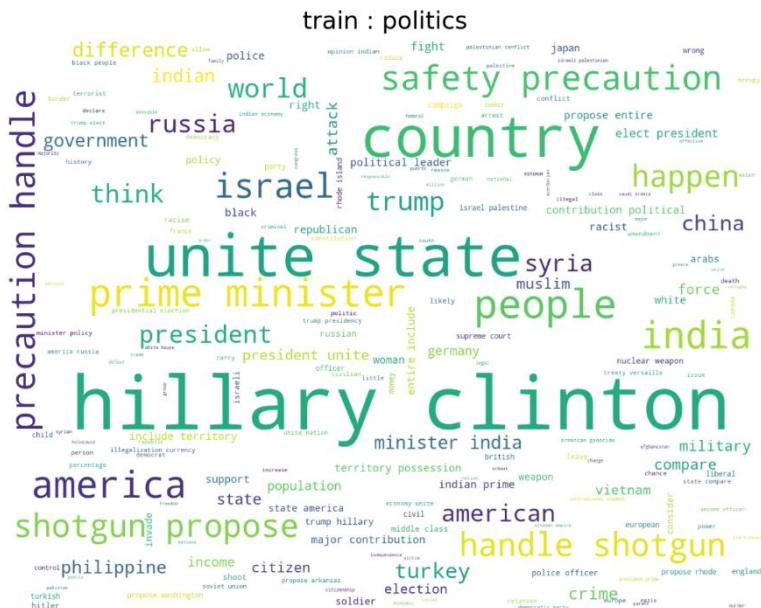


Figure 12 wordcloud politics

Filtrage par l'utilisation des entités nommées

Les entités nommées et catégorisées permettent également de créer des nuages de mots plus ciblés :

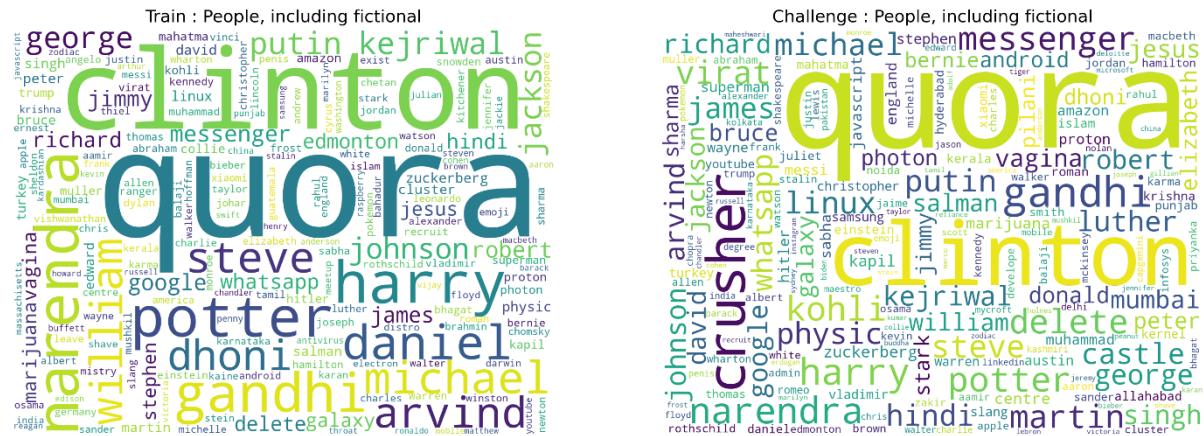


Figure 13 wordcloud PERSON in train and challenge

L'information affichée est toujours aussi copieuse...

On peut croiser les 2 informations newsgroup et entités :

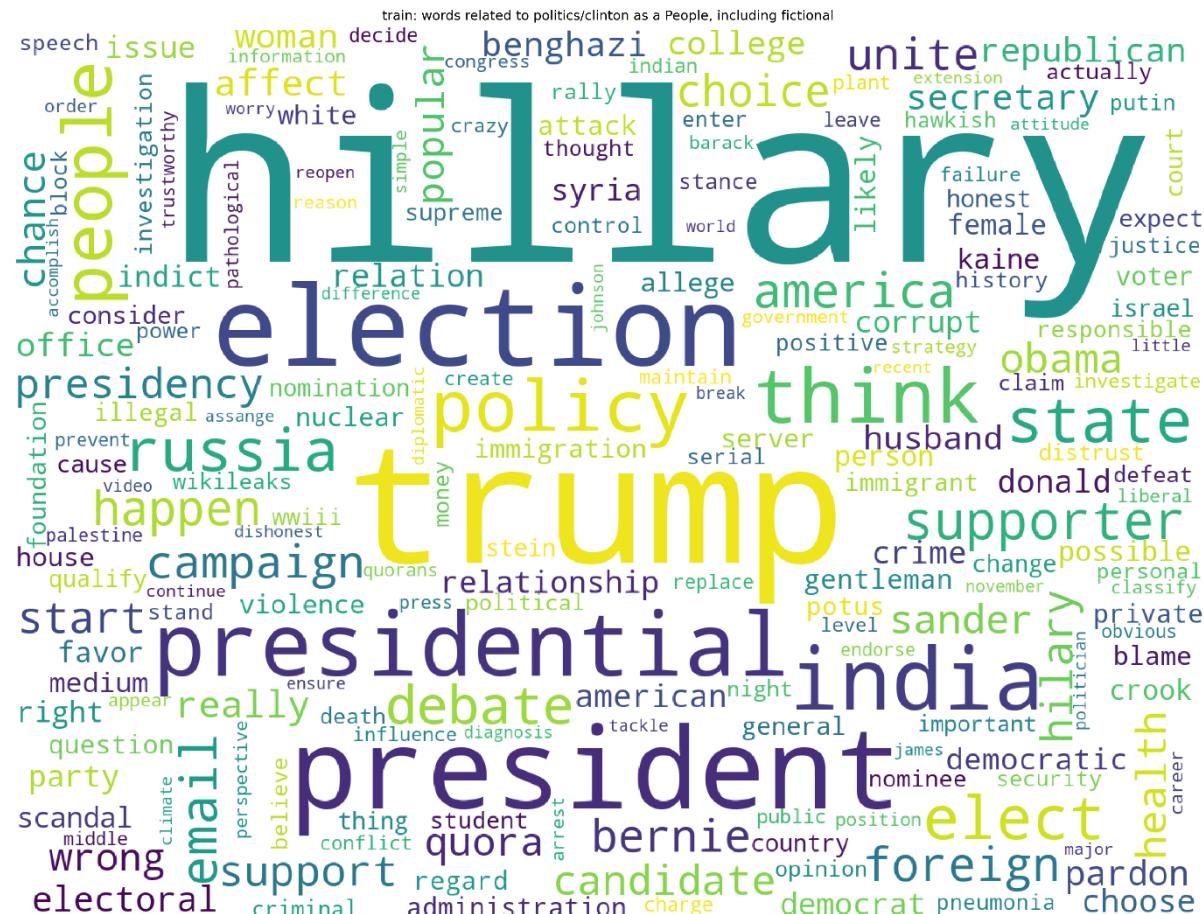


Figure 14 wordcloud questions politiques citant Clinton en tant que personne

En rajoutant des filtres exploitant ces informations sémantiques, on commence à générer des nuages de mots plus lisibles.

Toutefois, il ne faut pas oublier que l'information *newsgroup+entité* est globalement très peu disponible : on finit très rapidement par ne plus trouver de réponses correspondant à un filtre.

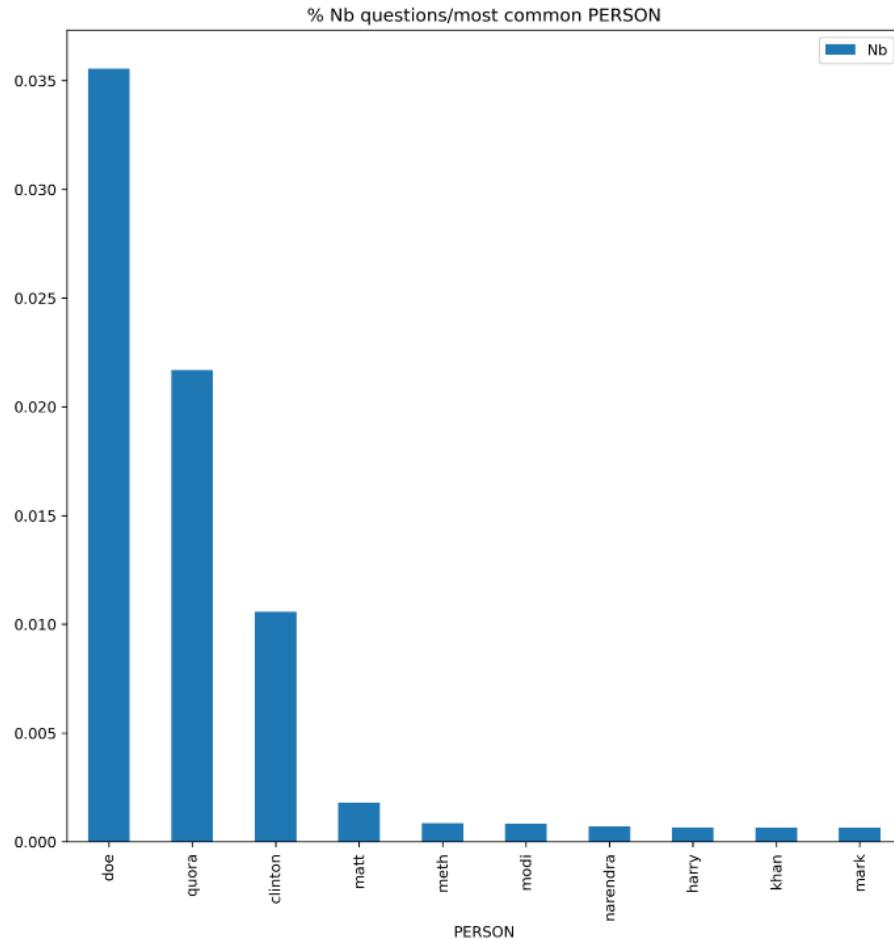


Figure 15 personnes les plus citées dans le training

Par exemple , Clinton, la troisième personne la plus citée ne représente déjà que 0.05 % des questions quand on filtre par le newsgroup '*politics*'...

On peut conclure que dans notre contexte, les nuages de points s'ils sont faciles à générer, ferment peu sur le contenu des questions soit parce qu'ils ne filtrent pas assez l'information, soit parce qu'ils la filtrent trop.

Recherche de topics : LDA (*Latent Dirichlet Allocation*)

Cet algorithme permet de découvrir l'ensemble des topics représentant une collection de documents. Avec cet ensemble de topics, on peut par exemple :

- Réduire un gros corpus de textes à des mots-clefs
- Commencer à résumer des textes
- Labelliser automatiquement de nouveaux textes à partir des topics appris.
- Comparer des documents
- ...

Techniquement :

LDA représente un topic comme un ensemble de mots associés à des probabilités et un document comme une combinaison linéaire de topics: '*la phrase 2 est 40% topic 1 et 60% topic 2*', le mot 'héritage' ayant une probabilité 0.7 d'être dans le topic 1, le mot famille, une probabilité 0.3, etc, ... LDA permet de découvrir ces topics. Une fois les topics déterminés, tout document peut être représenté par son vecteur dans l'espace des topics et une algèbre simple est possible avec ces vecteurs. Par exemple, calculer la distance entre 2 documents est possible et fournit une mesure de la similarité entre ces 2 documents.

C'est bien sur une voie possible pour ce challenge et elle fait partie des options finalement étudiées mais ici, les topics ont été recherchés pour tenter d'analyser le contenu des questions.

Les résultats demandent une certaine expertise pour être analysés et même, produits. Par exemple, la détermination du nombre idéal de topics est délicate. La notion de *divergence Kullback-Leibler* permet d'évaluer la qualité associée au nombre de topics trouvés et de comparer 2 décompositions.

En pratique, les implémentations disponibles imposent de fixer a priori le nombre de topics et de qualifier la décomposition obtenue avec une métrique de *cohérence* facile à utiliser. De plus, des interfaces graphiques spécialisées sont disponibles pour naviguer mais les résultats restent délicats à interpréter.

Les résultats correspond à la décomposition en 10 topics sur *le texte brut* peuvent être visualisés avec un outil graphique complexe. Sommairement :

- Plus les cercles sont éloignés, meilleure est la qualité de la décompositions (les 'sujets' trouvés ne se recoupent pas).
- Plus le cercle est grand, plus important est le mot dans le topic

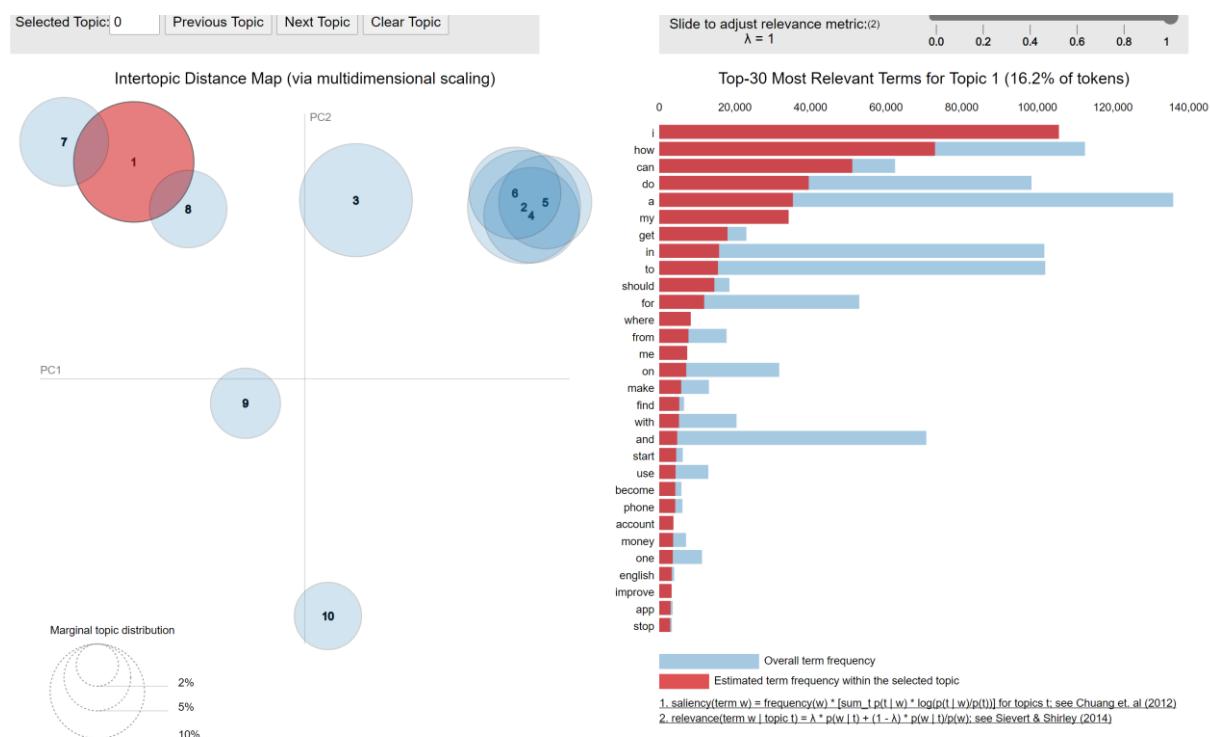


Figure 16 Interface de visualisation LDA 10 topics sur texte brut

On s'aperçoit rapidement que, parce qu'aucun filtrage des mots courants n'a été effectué, ce qui a été trouvé comme topics fréquents correspond en fait à des patterns de questions : '*How can I*', '*What is a*',...

Si on effectue la même analyse en ayant éliminé tous les stop words, nettoyé le texte et procédé à une lemmatisation, on obtient une décomposition de meilleure qualité et nettement moins orientée sur les syntaxes les plus fréquentes :

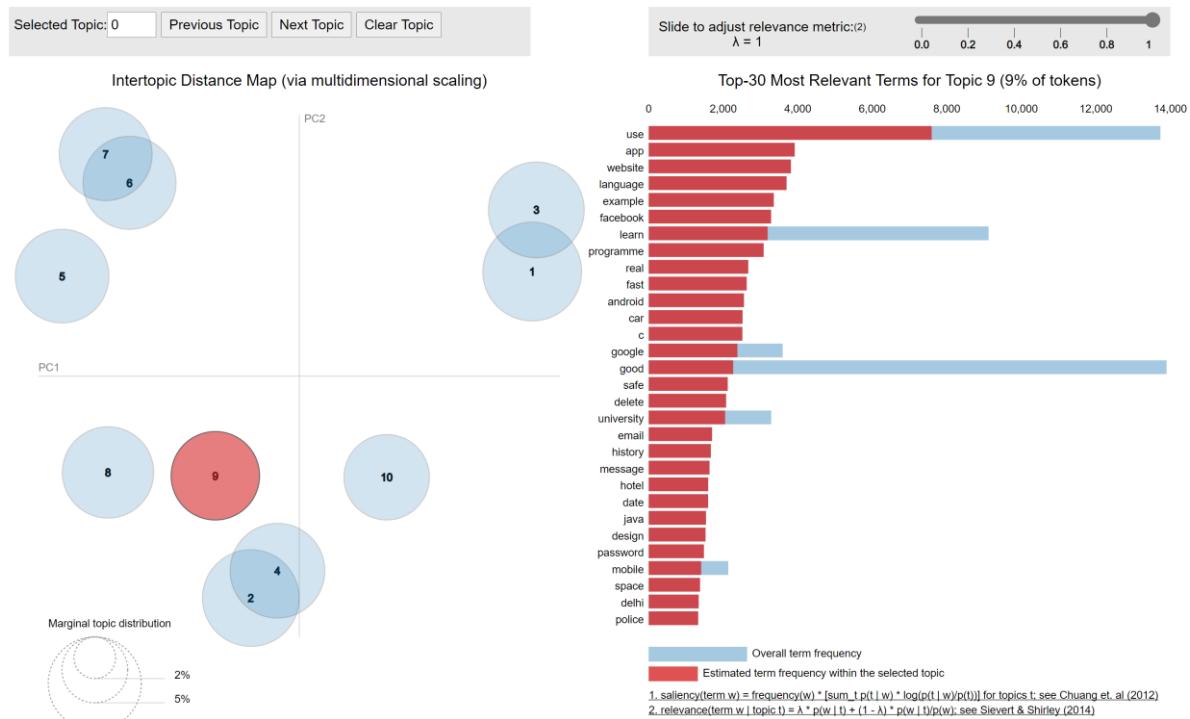
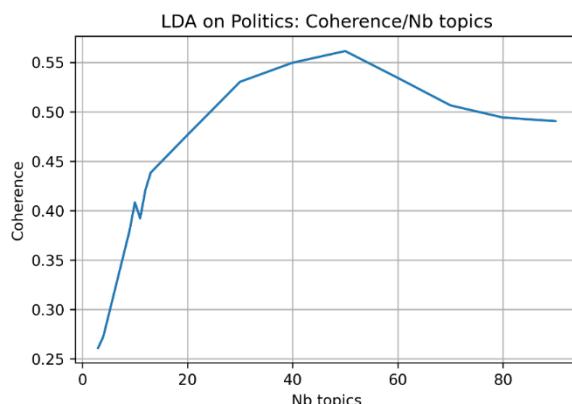


Figure 17 Interface de visualisation LD1 10 topics sur texte lemmatisé

Les résultats sont un peu plus clairs mais restent difficiles à comprendre.

Une analyse plus poussée en filtrant le dataset permet d'obtenir des topics plus intéressants :

- On analyse seulement les questions identifiées avec le newsgroup '*politics*'
- On recherche le nombre de topics correspondant à la meilleure décomposition (en minimisant la métrique de cohérence). Pour le newsgroup '*politics*', il ne faut pas moins de 50 topics.



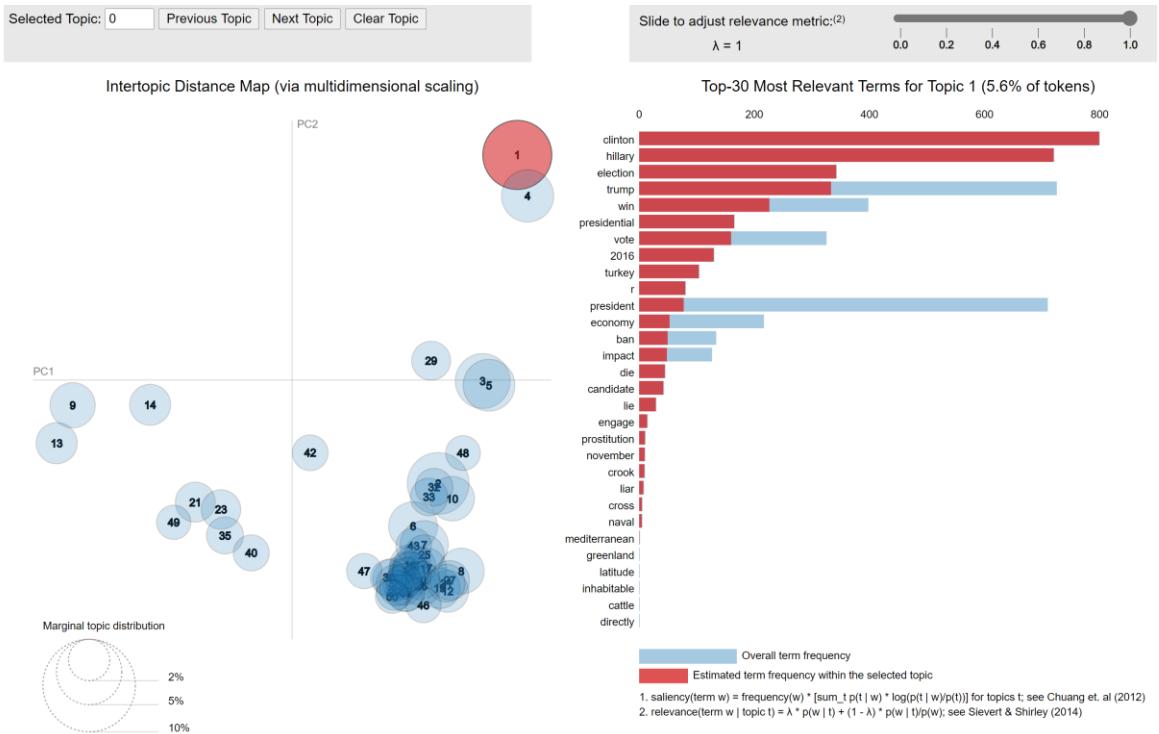


Figure 18 Interface de visualisation LDA 50 topics sur politics

Dans ce contexte, les premiers topics trouvés sont nettement plus clairs même si l'outil graphique fourni par la bibliothèque *gensim* reste complexe.

La bibliothèque *textacy* fournit une autre implémentation de LDA et des graphes nettement plus abordables. Une limite d'implémentation restreint malheureusement le nombre de topics visualisables à une dizaine.

On obtient en effet la représentation graphique des topics (liste complète en annexe C) nettement plus claire:

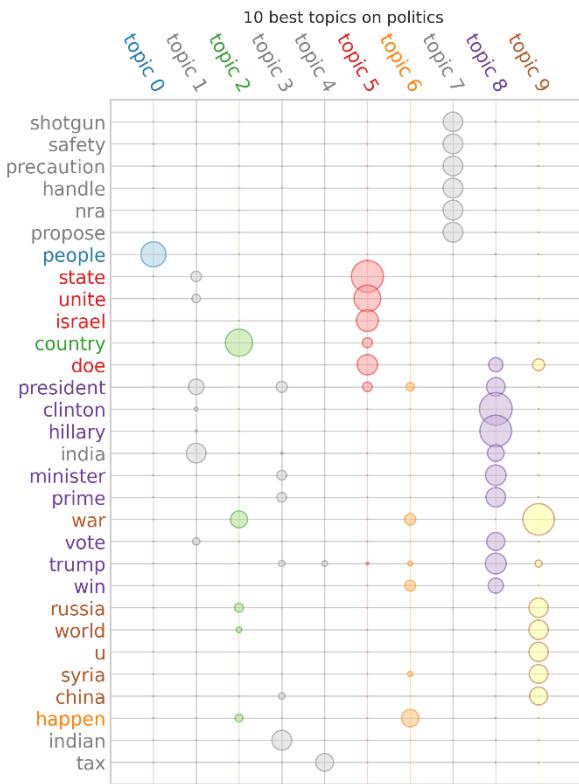


Figure 19LDA (textacy) 10 topics sur politics

LDA+TSNE (*T-Distributed Stochastic Neigborhood Embedding*)

TSNE est une méthode récente de réduction de dimension concurrente de l'ACP (*Analyse en Composante Principale*).

Les deux méthodes essayent de projeter au mieux des données dans un espace avec beaucoup de dimensions (le nombre de topics dans notre cas) vers un espace à peu de dimensions (typiquement 2) accessible à l'interprétation.

ACP maximise la variance le long de la composante principale et considère les autres composantes comme du bruit : la structure globale est conservée en perdant de l'information sur la structure locale.

TSNE en utilisant des mathématiques complexes, réussit à trouver les meilleures projections tout en préservant les structures locales et arrive donc à représenter correctement les clusters détectés par LDA.

En pratique :

- On utilise LDA pour trouver un ensemble de n topics. La pertinence et l'interprétabilité de cette décomposition en topics restent difficiles.
- On utilise TSNE pour projeter en 2d les vecteurs de topics des documents. La projection déforme l'espace des points avec 2 propriétés remarquables :
 - Les points d'un même topic restent proches
 - Les points sont tout de même bien séparés dans la projection s'ils sont bien séparés initialement.

Dans notre cas, on obtient de belles images à partir des topics trouvés précédemment mais l'empilage de 2 méthodes mathématiques complexes rend l'interprétation des résultats hors de notre portée et n'aide finalement pas à se faire une idée du contenu du corpus de texte.

Pour le newsgroup *politics* avec 10 topics, on obtient le graphe :

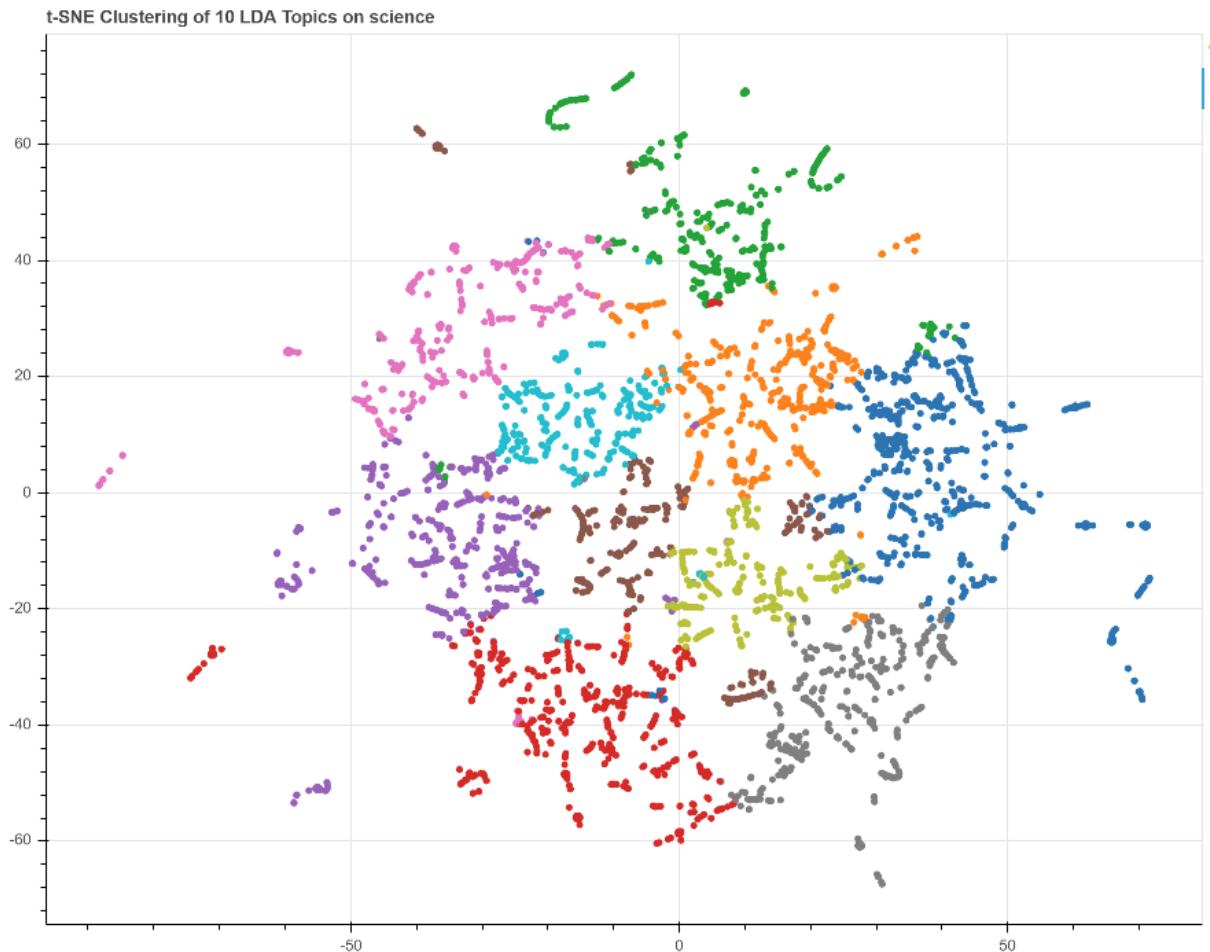


Figure 20TSNE Politics - 10 topics (4121 questions)

L'exploration d'un tel corpus de textes reste un problème : les outils testés sont peu performants dans le contexte de ce challenge : énormément de sujets très différents avec peu de textes pour chaque sujet, des textes très courts,...

L'utilisation de données externes ou l'extraction des entités nommées, même si elles sont partielles permet toutefois de répondre à une question importante : le vocabulaire employé dans les 2 datasets est le même.

Analyse des datasets de training et de challenge

Le corpus de textes ayant été étudié indépendamment de toute problématique de prédiction, un ensemble de propriétés basiques ont été vérifiées pour évaluer la qualité des datasets et les comparer dans l'objectif d'effectuer une prédiction de la target *is_duplicate*.

Qualité des données

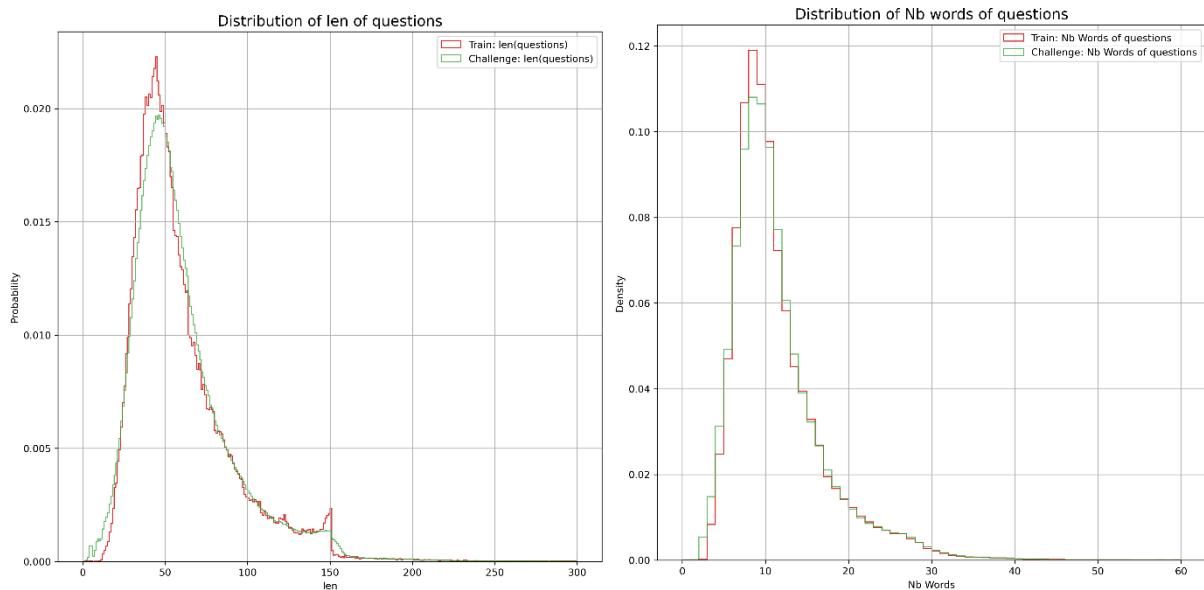
Property	Train	Challenge
Nb unique id	404290	2345796
Nb pairs	404290	2345796
Nb duplicate=1	149263	
Distinct values in duplicate	0 ou 1	
% duplicate = 1	36,92	
Nb unique qid1	236581	
Nb unique qid2	253733	
Nb qid1=qid2	0	
Nb question1=question2	0	60
Nb lower(question1)=lower(question2)	1	64
mean(len(question1))	59,54	60,12
mean(len(question2))	60,11	60,02
Nb issues in Unicode encoding of question 1	1336	8229
Nb issues in Unicode encoding of question 2	1247	7922
% issues in Unicode encoding of question 1	0,003	0,004
% issues in Unicode encoding of question 2	0,003	0,003
Nb question1 empty or missing	1	2
Nb question2 empty or missing	2	4
Nb pairs (A,B,1)+(B,C,1)	271490	
Transitivity: Nb (A,B,1)+(B,C,1)=>(A,C,?)	206517	
Transitivity KO: Nb (A,B,1)+(B,C,1)=>(A,C,0)	111	

Les erreurs détectées sont minimes et faciles à corriger et surtout, les données sont de bonne qualité et permettent de faire une classification sur le champ *is_duplicate* :

- La target est bien binaire sans valeur manquante.
- Les identifiants sont bien uniques (même s'ils ne sont a priori d'aucune utilité pour ce challenge).
- La quasi-totalité des questions est remplie.
- Il n'y a pas de cas triviaux (égalité évidente des questions).
- Les paires sont cohérentes par transitivité

Propriétés élémentaires des questions²

Property	Train				Challenge				Distribution similaire
	Min	Max	Mean	Std Dev	Min	Max	Mean	Std Dev	Stat/p-value Kolmogorov-Smirnov
Length	0	1169	59,8	31,9	0	1176	60	31,6	0.025/0
Nb words	0	237	11	5,9	0	238	11	5,8	0.020/0
Mean length of words	0	36	4,6	0,8	0	51	4,6	8,6	0.034/0



Les tests de similarité confirment ce qui est annoncé dans la description du challenge : les questions du dataset de challenge n'ont pas les mêmes propriétés que celles du dataset de training (ce qui n'empêche pas qu'elles puissent utiliser le même vocabulaire).

Données cachées

Utilisation de la définition du logloss

Une première donnée cachée et potentiellement utilisable pour le challenge est fournie par le processus d'évaluation lui-même.

Dans le dataset de training, il y a 36,9 % de questions dupliquées. La target n'est évidemment pas fournie pour le dataset de challenge.

Toutefois, le logloss étant le critère d'évaluation, si on fournit une prédiction constante, on peut estimer ce ratio pour le dataset de challenge en utilisant le score renvoyé par Kaggle.

$$\text{logloss} = -\frac{1}{N} \sum_{i=1}^N y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))$$

Par exemple, avec une prédiction constante de 0.369, on obtient une note kaggle de 0.55410 et on doit alors résoudre :

² Pour ces propriétés, on a fusionné question1 et question2 dans les deux datasets

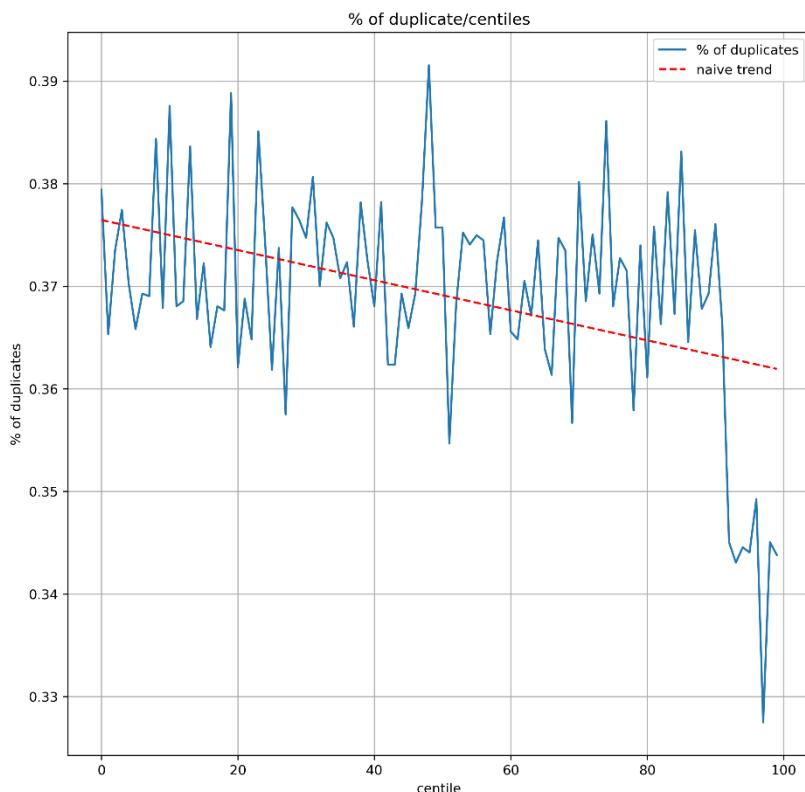
$$0.55410 = -1/N \sum_{i=1}^N y_i \log(0.369) + (1 - y_i) \log(1 - 0.369)$$

Le ratio de questions dupliquées est de 36,9 % dans le dataset de training et de 17,5 % dans le dataset de challenge.

Il faudra donc ajuster en conséquence le processus d'entraînement pour que les prédictions finales respectent ce ratio.

Information cachée dans le champ id

Le champ id est une clé unique croissante de paire de questions entre 0 et 404290. En agrégant le champ id par centiles, on remarque une légère tendance cachée :



On a appliqué 2 tests permettant de vérifier si une série temporelle est stationnaire ou pas :

- *Augmented Dickey-Fuller Unit Root Test*: stat=-2.809, p=0.057
- *Kwiatkowski-Phillips-Schmidt-Shin*: stat=0.453, p=0.054

Pour ces 2 tests, l'hypothèse H0 ie *la série est probablement non stationnaire* est acceptée.

Dans le dataset de train, plus le champ id est grand, moins une paire de questions a de chance d'être dupliquée.

Ceci est probablement, le résultat de l'amélioration des processus internes de Quora.

En pratique, même si plusieurs équipes gagnantes se sont servies de cette tendance pour ajuster finement leurs modèles, nous n'avons pas cherché à exploiter cette propriété.

Machine Learning

Recalibration du dataset de training

Nous avons vu que le % de paires identiques observées dans le dataset de training est de 36,9% alors que, par déduction, il est de 17,46 % dans le dataset de challenge. Il faut procéder à une recalibration du ratio cas positif/cas négatifs avant de pouvoir entraîner un modèle prédictif. Pour cela, on peut :

- Changer l'échantillonnage : soit on supprime des paires équivalentes (downsampling des cas positifs) soit on duplique certaines paires non équivalentes (upsampling des cas négatifs).
- Changer le poids de chaque ligne : on affecte un poids différent pour les cas positifs et les cas négatifs. Les poids choisis sont calculés pour arriver à un ratio global de 17,5% .

Nous avons choisi d'utiliser la méthode des poids : le dataset de training est déjà assez gros ce qui élimine l'upsampling et on n'a pas envie de 'perdre' des questions, ce qui élimine le downsampling.

Les poids utilisés partout sont :

	Formule	Valeur
Ratio courant cas positifs	(Nb is_duplicate=1)/Nb rows	0.369
Ratio désiré cas positifs		0.1746
Poids Cas positif : Is duplicate = 1	Ratio désiré cas positifs/ Ratio courant cas positifs	0.473
Poids Cas négatif : is_duplicate=0	(1- Ratio désiré cas positifs)/(1- Ratio courant cas positifs)	1.308

Utilisation d'une feature simple : les mots communs

Rien d'autre n'est disponible que le texte pour alimenter un modèle prédictif sur la target *is_duplicate*. Tout l'objet du challenge consiste donc à calculer un ensemble de features équivalent à l'attribut *is_duplicate*.

Un critère brutal d'égalité est : *2 questions sont équivalentes si elles sont constituées des mêmes mots*. Appliqué strictement, ce critère est équivalent à 'le texte des questions est le même' et devient inutile. Un critère plus souple serait : *2 questions sont équivalentes si elles emploient plutôt les mêmes mots*.

Nous avons donc défini un ensemble de 8 compteurs et ratios simples tentant de décrire à quel point une paire de questions utilise les mêmes mots :

Champ	Description
nb_words_question1	
nb_words_question2	
nb_common_words	Nombre de mots communs
nb_common_words/nb_words_question1	% de mots communs avec question2 dans question1
nb_common_words/nb_words_question2	% de mots communs avec question1 dans question2
nb_words_question1-common_words	Mots non communs
nb_words_question2-common_words	Mots non communs
nb_common_words/(nb_words_question1+nb_words_question2)	Ratio global sur la paire

Globalement, on essaye de quantifier à quel point une question porte sur le même sujet que l'autre (mots communs) et à quel point elle parle aussi d'autre chose (mots non communs). On a également introduit quelques ratios pour essayer de limiter les facteurs d'échelle.

Dans un premier temps, ces mesures sont volontairement calculées sur le texte brut de chaque question. Aucun prétraitement n'est effectué.

Analyse univariée des variables ajoutées et évaluation a priori de leur intérêt

Même si on espère que c'est la combinaison de plusieurs variables qui va permettre de construire un bon modèle, la courbe de roc pour chaque variable par rapport à la target rest un indicateur de l'intérêt de cette variable pour modéliser la target. Plus l'AUC (*Area Under Curve*) est grand, plus la variable est intéressante pour expliquer la target (*indépendamment de toute autre donnée*).

Variable	AUC	Corrélation avec is_duplicate
nb_common_words/(nb_words_question1+nb_words_question2)	0.734	0.370
nb_common_words/nb_words_question2	0.725	0.359
nb_common_words/nb_words_question1	0.723	0.358
nb_common_words	0.680	0.225
nb_words_question2	0.420	0.160
nb_words_question1	0.416	0.154
nb_words_question2-common_words	0.323	0.301
nb_words_question1-common_words	0.320	0.295

L'intérêt des plus fortes valeurs d'AUC est confirmé quand on affiche les distributions de chaque variable par rapport à la target. Pour les grandes valeurs, on voit que les distributions sont partiellement séparées, pour les valeurs faibles, les courbes se superposent et sont donc plus difficilement séparables. De même, les plotbox et diagrammes en violons sont une autre façon de représenter la séparabilité de la target par rapport à une variable.

Sans surprise, la variable avec la meilleure AUC et donc le meilleur ‘pouvoir de séparabilité’ est **nb_common_words/(nb_words_question1+nb_words_question2)**.

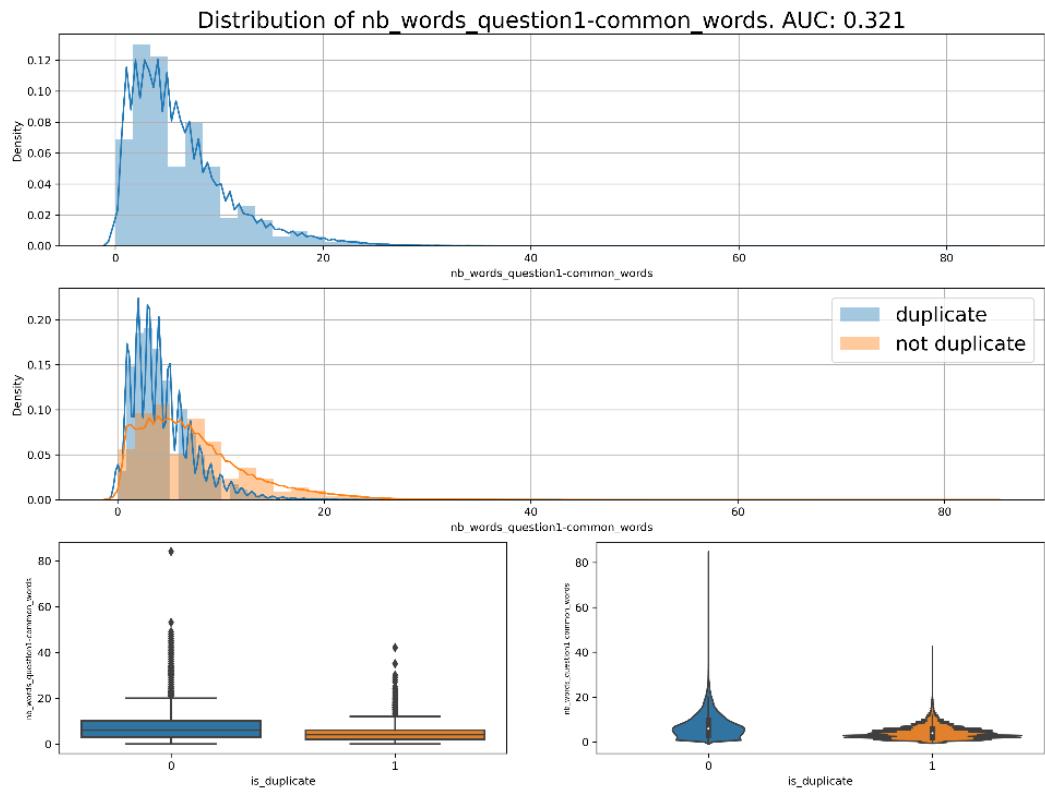


Figure 21 Distribution de la variable avec la plus petite AUC

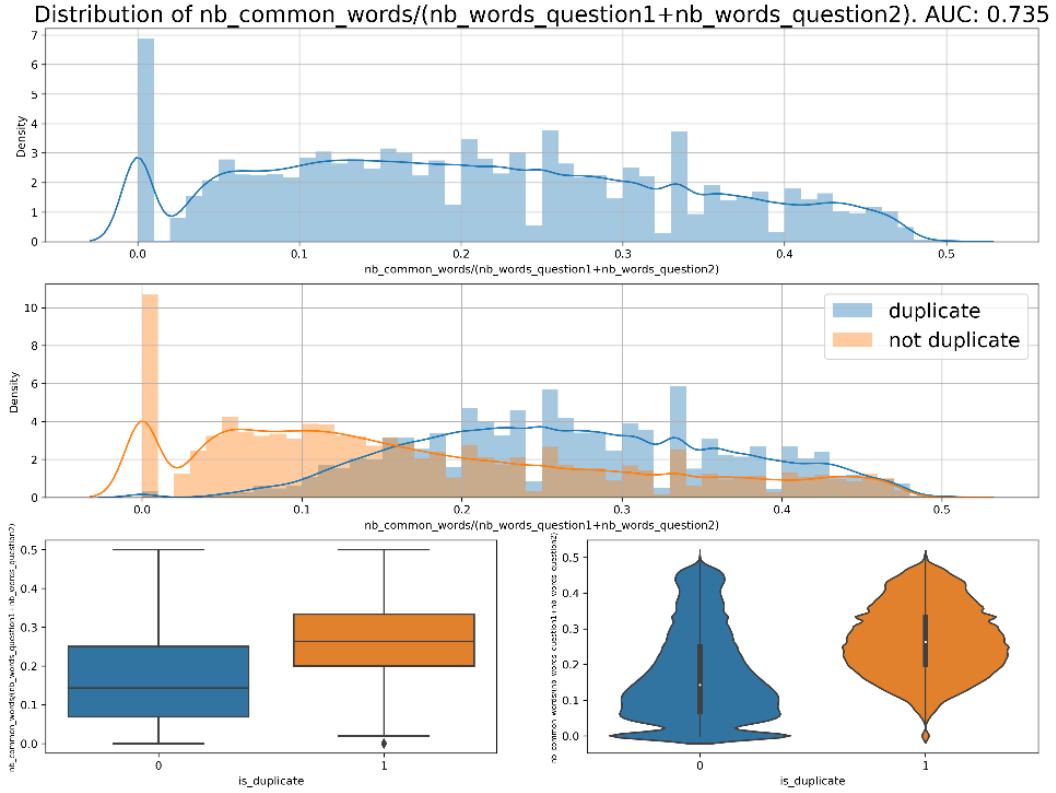


Figure 22 Distribution de la variable avec la plus grande AUC

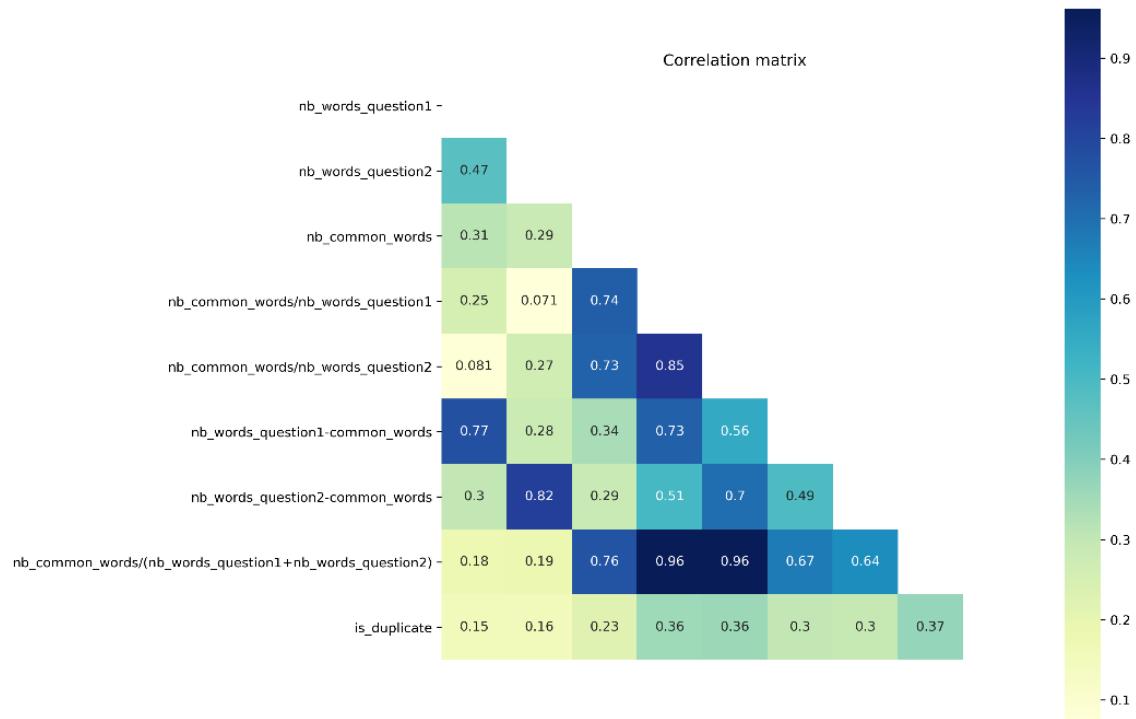


Figure 23 Matrice de corrélation

La matrice de corrélation montre évidemment des corrélations entre les variables créées. L'algorithme choisi pour le premier modèle de référence a été XGBoost pour sa simplicité de mise en œuvre et sa réputation de résistance aux variables corrélées.

XGBoost

La librairie XGBoost implémente l'algorithme [Gradient boosting decision tree](#). Le *Boosting* est une technique ensembliste où de nouveaux modèles sont ajoutés pour corriger les modèles précédents. Les modèles sont ajoutés séquentiellement jusqu'à ce que l'erreur ne diminue plus. *Gradient boosting* utilise un algorithme de descente de gradient pour minimiser le risque empirique à chaque ajout de modèle (des arbres de décision):

$$R_n(f) = \frac{1}{n} \sum_{i=1}^n l(Y_i, f(X_i))$$

Où $l(Y_i, f(X_i))$ mesure l'erreur entre la prévision $f(x)$ et l'observation y et l une fonction différentiable et convexe. L'algorithme renvoie une suite récursive d'estimateurs :

$$f_n(x) = f_{n-1}(x) + \lambda h_n(x)$$

Où $h_n(x)$ est une règle 'faible'. En pratique, les règles faibles sont le plus souvent des arbres avec peu de coupures.

XGBoost (eXtreme Gradient Boosting) est une implémentation moderne et performante de l'algorithme de gradient boosting³.

Un premier modèle a été réalisé en utilisant la librairie XGBoost sur une partition 80%,20% du dataset de training. Le logloss obtenu sur l'ensemble de test est de [0.3841](#) et le score kaggle (logloss sur le dataset de challenge) est de [0.39206](#).

Pour ce qui concerne la performance au sens Kaggle, ces résultats sont médiocres mais encourageants au vu de la simplicité des opérations réalisés. Toutefois, une analyse des métriques usuelles de classification montre que les résultats sont, en pratique, trompeurs :

	precision	recall	F1-score
0	0.83	1	0.90
1	0.57	0.02	0.03
Accuracy			0.82
Macro avg	0.70	0.51	0.47
Weighted avg	0.78	0.82	0.75

Default XGBoost on basic features; Confusion Matrix with default threshold

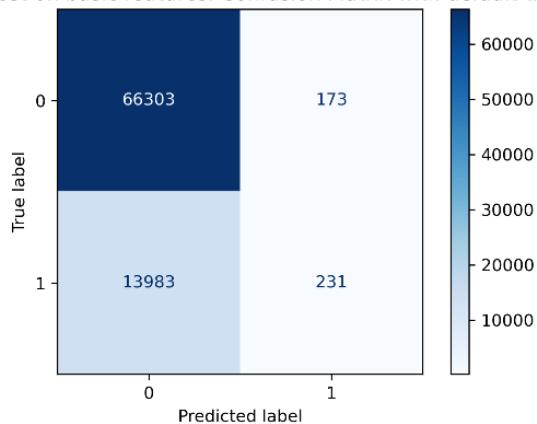


Figure 24 Matrice de confusion (features basiques & default threshold)

L'accuracy est de [0.82](#) mais en appliquant le threshold usuel de 0.5 sur les probabilités pour décider de la classe, on ne prédit que [0.767 %](#) de questions dupliquées (la proportion réelle dans le dataset de training est de [37.71 %](#)). Toutes les probabilités générées étant inférieures à 0.5, le modèle ne prédit que des paires non dupliquées, ce qui suffit à avoir une accuracy correcte.

³ <https://www.youtube.com/watch?v=Vly8xGnNiWs> ou <http://datascience.la/xgboost-workshop-and-meetup-talk-with-tianqi-chen/>

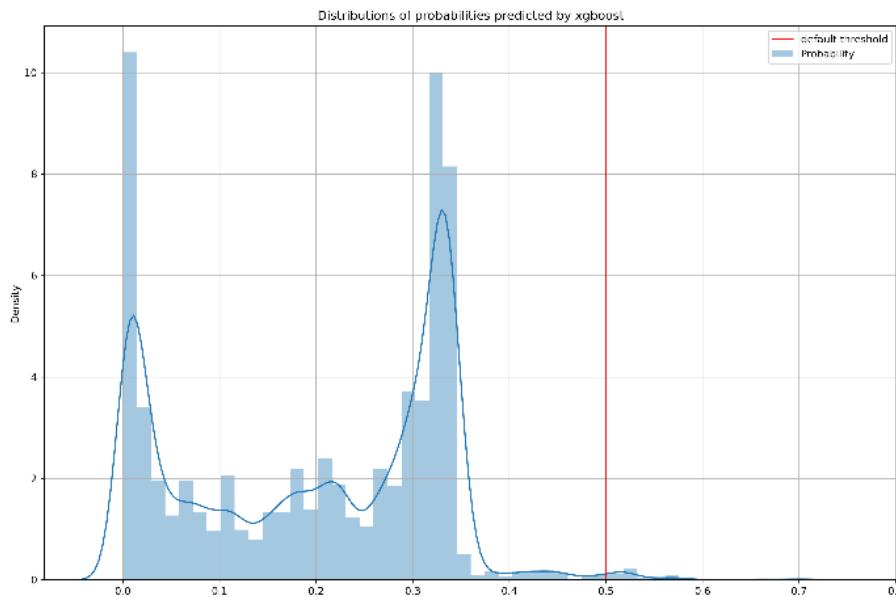


Figure 25 Features basiques: distribution des probabilités

Nous avons continué à jouer le jeu du meilleur score Kaggle mais l'utilisabilité des modèles obtenus doit être challengée et éventuellement corrigée par le biais d'une procédure de calibration : le score kaggle n'est pas forcément une mesure de l'utilisabilité d'un modèle pour générer des décisions.

Ajout de features non sémantiques

La notion de mots communs est une *feature non sémantique* (non sémantique dans la mesure où aucune compréhension du texte n'est nécessaire, il s'agit de simples comptages) simple permettant de construire un premier modèle. En procédant à un prétraitement du texte des questions, nous avons pu ajouter d'autres features non sémantiques qui sont des variations sur l'idée initiale.

Preprocessing

Les features initiales ont été volontairement calculées sur le texte brut des questions : aucun prétraitement n'a été appliqué (y-compris le passage en minuscules des questions). Les prétraitements (nettoyage, stop words, lemmatisation,...) sont des opérations classiques en NLP et nous avons voulu mesurer leur impact.

Stop words

En examinant les listes de mots non partagés entre questions, on s'aperçoit que de nombreux mots très communs polluent ces listes. Par exemple : *a, is what, not, the, ...* Ce sont ce qu'on appelle les *stop words* de la langue anglaise. Les librairies de NLP fournissent un dictionnaire de stop words et nous avons recalculé les features après passage en minuscules et élimination de ces mots inutiles (en fusionnant 2 dictionnaires différents).

Nettoyage

Toujours en continuant à examiner la liste des mots non partagés, on remarque des mots non reconnus à cause :

- D'une ponctuation. Par exemple, toutes les questions finissent par un ? accolé au dernier mot.
- Des formes abrégées de la langue anglaise : *what's* n'est pas reconnu comme *what is*
- De fautes de frappes : *intially* pour *initially*
- De la présence d'unités de mesures : *100,000rs* est en fait *100000 roupies*

- D'abréviations : *kms* pour *kilometers*...
- ...

Par tâtonnement et également en appliquant des préconisations standards de kaggle⁴, nous avons défini et appliqué environ 50 règles de nettoyage sur le texte des questions.

Lemmatisation

La lemmatisation est une opération permettant retrouver le mot racine (le *lemme*) d'un autre mot.

Par exemple:

- what will -> what be,
- trainings -> train
- ...

Cette opération de lemmatisation en agrégeant toutes les variantes d'un mot sur un unique mot racine permet de réduire encore la liste des mots non partagés entre questions et donc de calculer de nouvelles features ajustées.

	No stop words	+nltk stopwords	+sklearn stopwords	+clean	+lemme
nb_words_question1	0.416846	0.406615	0.403675	0.400973	0.392604
nb_words_question2	0.421823	0.412457	0.410945	0.407867	0.400255
nb_common_words	0.682908	0.726849	0.725406	0.675331	0.650918
nb_common_words/nb_words_question1	0.728826	0.771614	0.771865	0.736529	0.732026
nb_common_words/nb_words_question2	0.731133	0.768482	0.768014	0.730916	0.732026
nb_words_question1-common_words	0.314964	0.289178	0.284100	0.292211	0.288846
nb_words_question2-common_words	0.317145	0.295027	0.291187	0.299165	0.295969
nb_common_words/(nb_words_question1+nb_words_question2)	0.741120	0.780440	0.779861	0.748105	0.748420

Figure 26 AUC des features après preprocessing

Les AUC a priori des features ajustées changent de façon complexe ce qui rend leur interprétation hasardeuse. Toutefois, les relations et ordres de grandeurs sont conservés.

Par construction, ces nouvelles features sont fortement corrélées entre elles, ce qui est évidemment confirmé par la matrice de corrélation :

⁴ <https://www.kaggle.com/currie32/the-importance-of-cleaning-text>

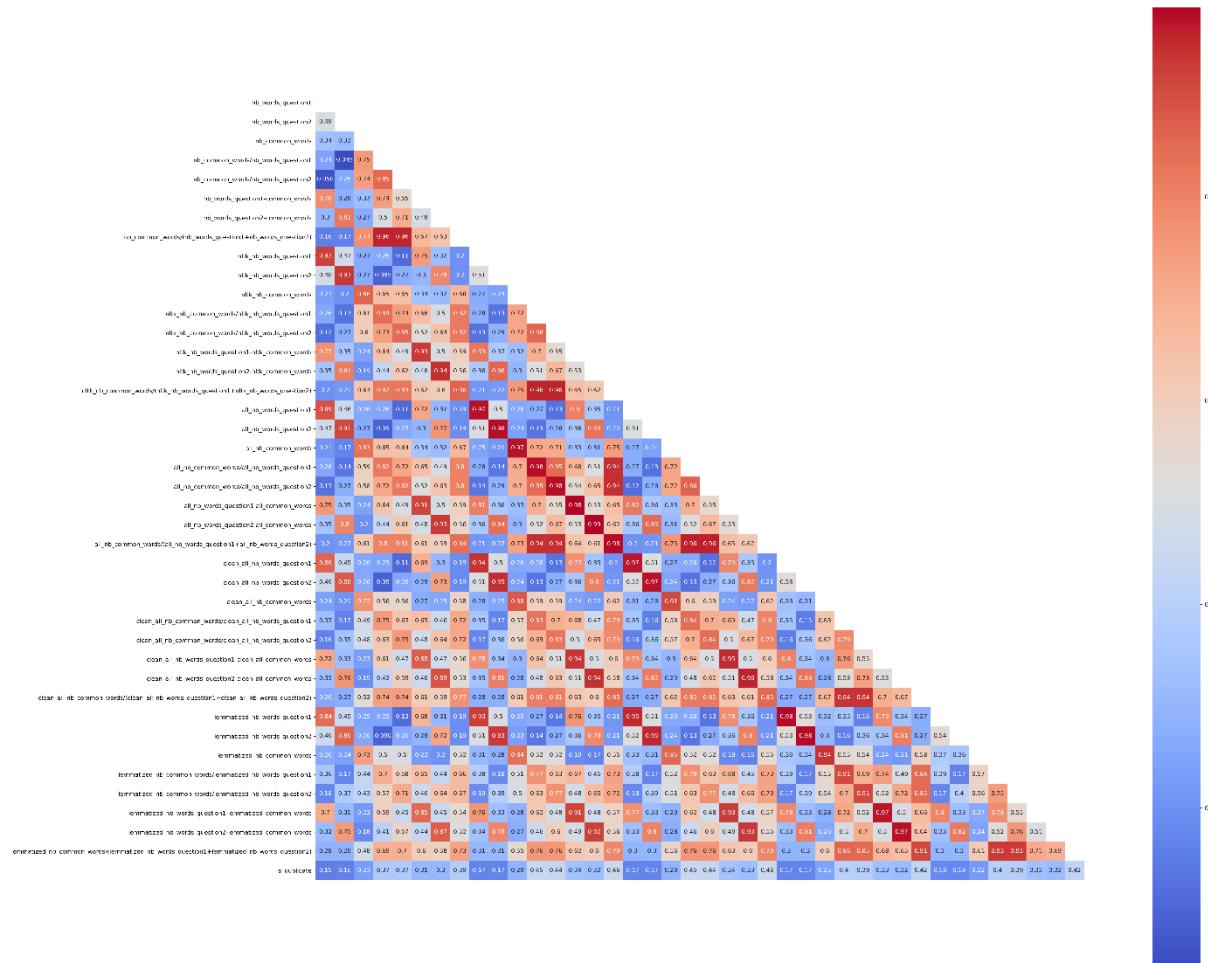


Figure 27 Matrice de corrélation: features non sémantiques

Dans l'objectif de tirer parti de toute brique d'information et *de ne rien perdre de leurs interactions*, nous avons décidé de laisser à la disposition de l'algorithme XGBoost tous les nouveaux indicateurs pour arriver à un total de 40 indicateurs simples.

Si l'on s'en tient au critère d'évaluation de kaggle ie le logloss, nos essais confirment l'importance des interactions entre features : même des features avec une corrélation entre elles aussi importante que 0.97 permettent d'améliorer légèrement le logloss du modèle et ne peuvent être éliminées a priori .

Avec les mêmes paramètres initiaux pour l'algorithme xgboost, nous obtenons pour chaque groupe de feature non sémantique une progression constante:

	logloss
basic	0.3784
Basic+nlkt stop words	0.3398
Basic+nlkt+ sklearn stop words	0.3390
Basic+nlkt+ sklearn stop words + clean	0.3373
Basic+nlkt+ sklearn stop words + clean + lemmes	0.3323

Figure 28 xgboost : logloss par groupe de features non sémantiques

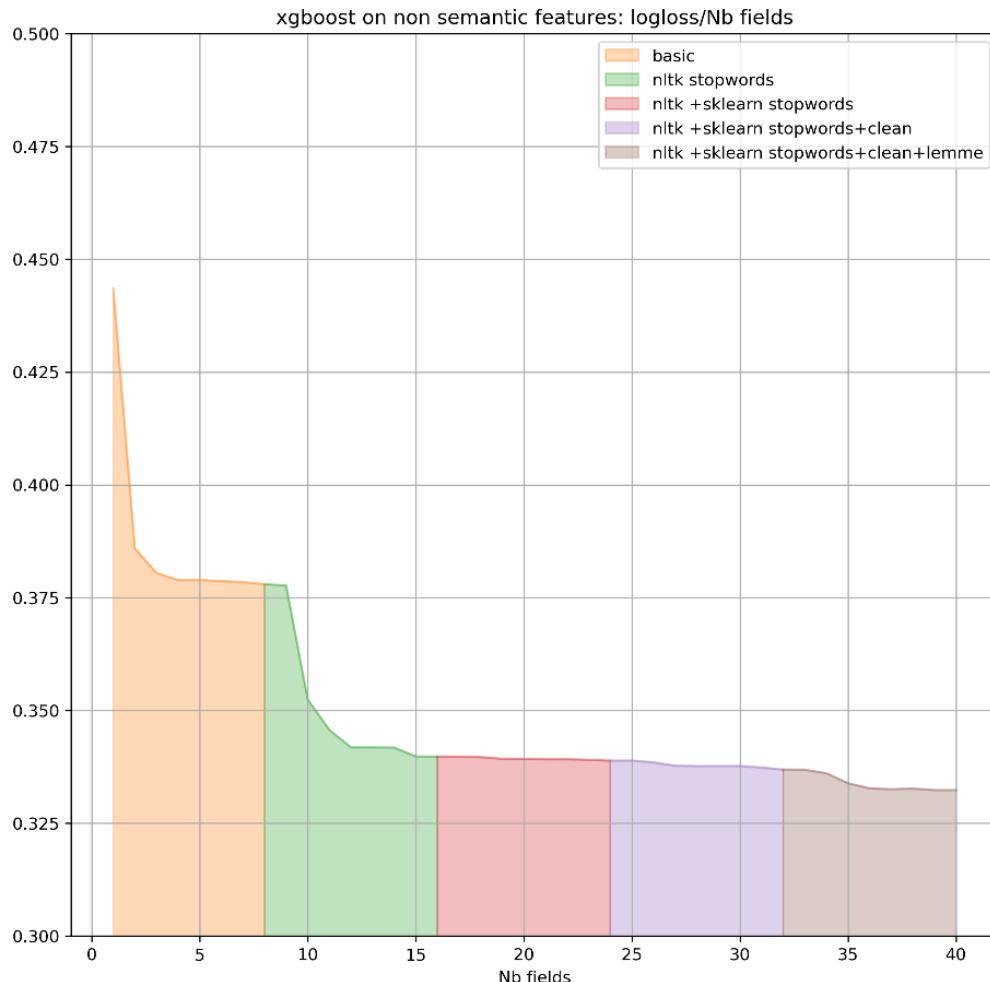


Figure 29 xgboost: logloss / non semantic feature

Ajout de features sémantiques

L'idée d'exploiter la notion de mots communs pour approximer la target *is_duplicate* est simple mais ne permet que d'arriver à un *logloss* de 0.3323 et à un score kaggle de 0.344, ce qui correspond à un score dans les 40 premiers % des meilleurs scores. Pour faire mieux, nous avons décidé d'ajouter des features sémantiques plus complexes et plus difficiles à produire.

Entités nommées

Les bibliothèques de NLP les plus puissantes permettent d'effectuer des analyses très fines sur un texte : analyse grammaticale complète mais aussi reconnaissance des entités nommées (cf Figure 7

Liste des entités nommées reconnues par *spacy*). Ces entités nommées - quand elles sont détectées – sont des informations sémantiques fortes. Nous les avons utilisées comme une aide à l'exploration du dataset mais elles peuvent être utilisées en tant que feature pour notre problème : *si 2 questions citent les mêmes entités alors elles ont plus de chance d'être identiques (elles 'parlent' des mêmes choses)*.

Parmi les catégories détectées par *spacy*, nous avons conservé les catégories d'entités suivantes qui nous paraissaient les moins ambiguës :

Category	Description
GPE	Countries, cities, states
PERSON	People, including fictional
PRODUCT	Objects, vehicles, foods, etc. (not services)
ORG	Companies, agencies, institutions, etc.
DATE	Absolute or relative dates or periods
NORP	Nationalities or religious or political groups
WORK_OF_ART	Titles of books, songs, etc.
LANGUAGE	Any named language
EVENT	Named hurricanes, battles, wars, sports events, etc.
FAC	Buildings, airports, highways, bridges, etc.
LAW	Named documents made into laws.
LOC	Non-GPE locations, mountain ranges, bodies of water

Pour chaque catégorie d'entité, nous avons calculé 4 indicateurs permettant de mesurer à quel point les questions ‘parlent’ des mêmes entités.

Indicateur	Description
Nb_entities_<Category>_question1	Nombre d'entites de la catégorie dans la question 1
Nb_entities_<Category>_question2	Nombre d'entites de la catégorie dans la question 2
Nb_entities_common_<Category>	Nombre d'entites de la catégorie communes entre question1 et question2
ratio_nb_entities_common_<Category>	Ratio nb entites communes/nb entités totales pour la catégorie

Cela permet d'ajouter un total de 48 nouveaux indicateurs. Pour le dataset de train, la librairie *spacy* détecte des entités dans 44% des paires.

Utilisation d'une source externe : newsgroups

Dans notre recherche de features sémantiques riches et simples à produire, nous avons décidé d'utiliser un dataset standard : le dataset newsgroup20. Ce dataset contient un extrait du contenu de 20 newsgroups. Pour les 18000 posts disponibles, on dispose du texte et du nom du newsgroup.

Ce dataset sert de bases à de nombreux exemples dans le NLP. L'annexe A détaille comment nous avons construit un modèle simple de classification à partir de ce dataset. Ce modèle est appliqué à chaque question et permet de générer la probabilité d'appartenance à 6 catégories de newsgroups. Nous avons utilisé ces informations seuillées pour aider à l'exploration des datasets mais ces probabilités peuvent être appliquées à notre problème : si les 2 questions ont une forte probabilité d'appartenir au même newsgroup, elles ont plus de chances d'être dupliquées (elles portent sur le même sujet).

Nous avons donc ajouté les 6 probabilités ⁵générées par le modèle newsgroup en tant que features sémantiques de notre modèle.

Similarité

Le problème posé dans ce challenge n'est fondamentalement pas nouveau : la recherche de documents similaires est une application standard du NLP et une opération de base d'un moteur de recherche. A ce titre, des mesures de similarité sont disponibles plus ou moins facilement dans les bibliothèques de NLP. Sommairement :

- On associe à chaque mot un vecteur de poids. Ce vecteur sémantique '*signature*' peut être calculé simplement en fonction de la fréquence d'occurrence dans le texte et dans le corpus de texte (*tf-idf*) ou de façon beaucoup plus complexe par l'intermédiaire de réseaux de neurones entraînés sur de très grands corpus (*word2vec, doc2vec, ...*).
- Une arithmétique simple est possible avec les vecteurs sémantiques associés à chaque mot. Par exemple, le vecteur d'un document est la somme des vecteurs des mots qui le constituent.
- De même, il est possible de comparer ou calculer simplement la distance entre 2 vecteurs sémantiques.

La bibliothèque *spacy* implémente directement cette notion de similarité entre 2 documents. Même si cette mesure semble être très proche de la solution à ce challenge kaggle, il faut s'attendre à ce qu'elle ne soit pas parfaite.

Ceci est confirmé par l'AUC a priori de la similarité calculée par *spacy* qui est seulement de 0.7013 et donc comparable à la qualité a priori de nos autres indicateurs. Une exploration manuelle de la matrice de confusion associée à la similarité confirme également que la similarité entre 2 questions n'est pas leur équivalence.

Modèle final

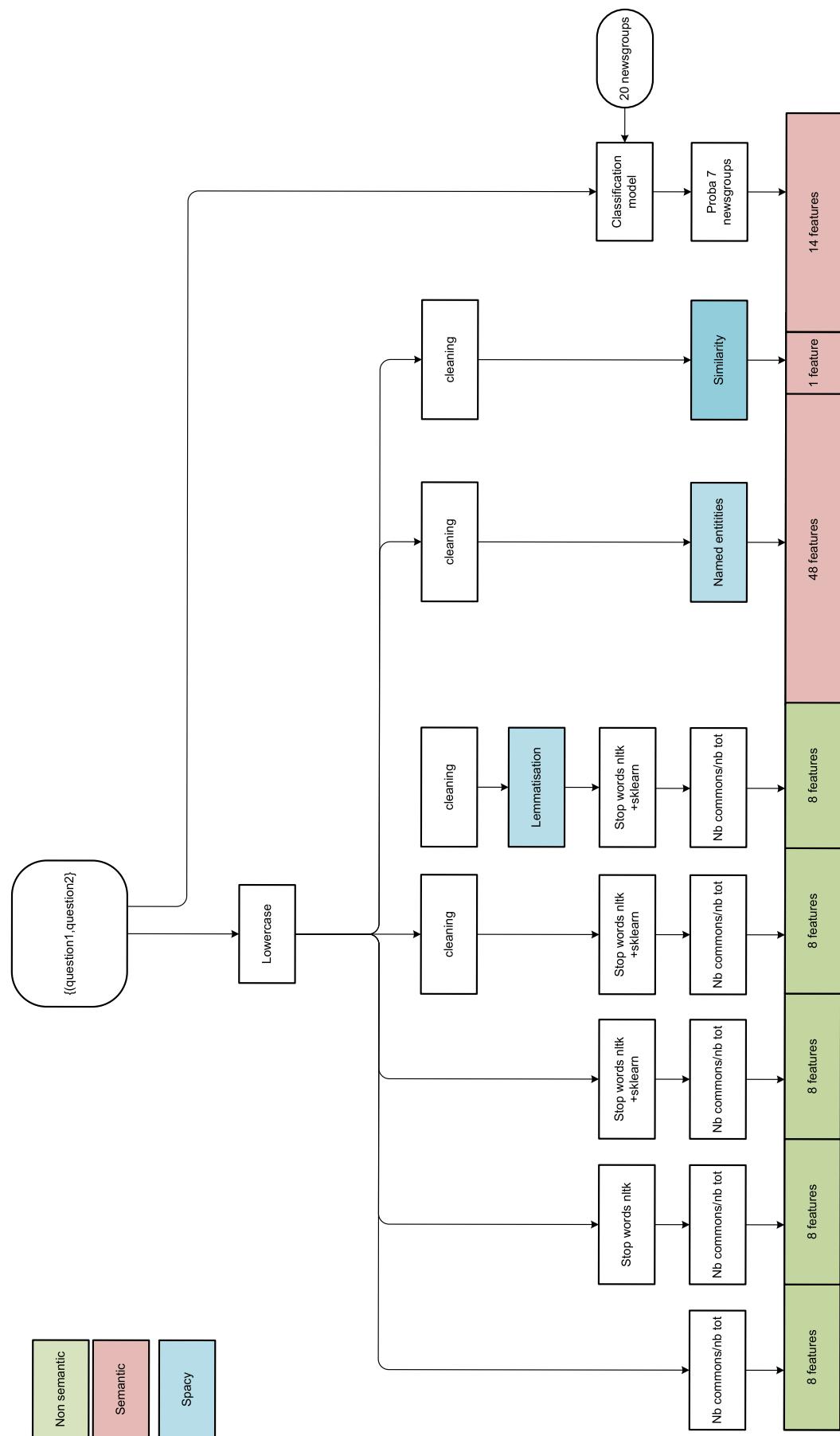
Le modèle final intègre donc finalement 103 features non sémantiques (variantes sur le nombre de mots communs) ou sémantiques (entités, newsgroup et similarité).

En utilisant les mêmes paramètres de base de l'algorithme xgboost, on obtient

	logloss
Basic+nltk+ sklearn stop words + clean + lemmes + entités	0.3309
Basic+nltk+ sklearn stop words + clean + lemmes + entités + newsgroups	0.3302
Basic+nltk+ sklearn stop words + clean + lemmes + entités + newsgroups + similarité	0.3291

⁵ Aucun seuil n'est appliqué.

Figure 30 103 features



Optimisation des hyper paramètres

Le modèle final intègre donc les 103 features avec un paramétrage basique de l'algorithme xgboost.

En utilisant la bibliothèque *hyperopt*, nous avons calculé 500 modèles explorant un espace important d'hyper paramètres pour obtenir finalement notre meilleur modèle avec une amélioration importante du logloss et du score kaggle :

Paramètres XGBoost	
colsample_bytree	0.7
eta	0.05
gamma	0.85
max_depth	11
min_child_weight	3.0
n_estimators	868
subsample	0.95
Résultats	
logloss	0.2867
Score kaggle	0.31176
Rank %	33.2

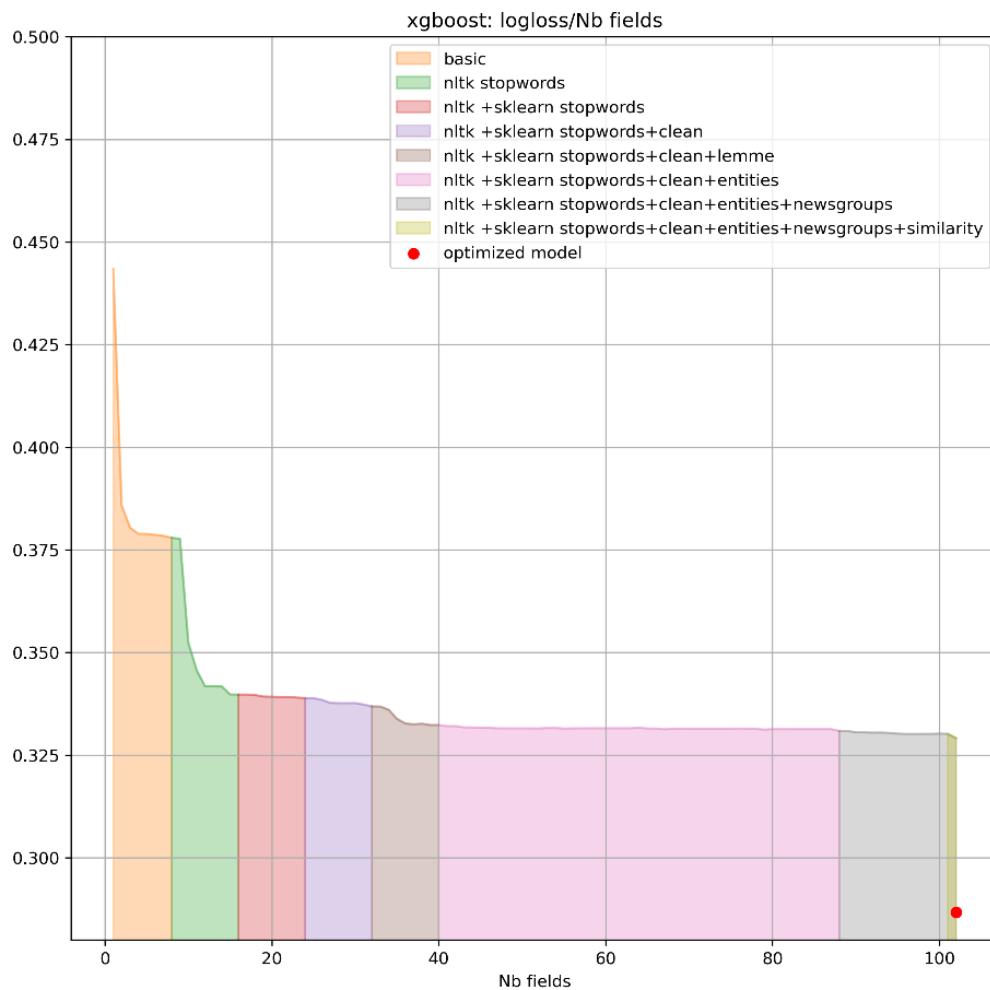


Figure 31 xgboost: logloss / all features

Evaluation

En manipulant des concepts simples ou des fonctions directement fournies par les bibliothèques de NLP, nous avons donc obtenu un modèle qui se place dans le premier tiers des meilleurs modèles. Ceci au sens du score kaggle.

Ce score kaggle plutôt satisfaisant correspond-il à un bon modèle ?

Si on examine les métriques standards d'une classification ainsi que la matrice de confusion, on remarque comme auparavant, que les probabilités sont sous estimées et qu'en utilisant le seuil standard de 0.5, on prédit trop peu de paires dupliquées.

	precision	recall	F1-score
0	0.88	0.96	0.92
1	0.68	0.40	0.51
Accuracy			0.83
Macro avg	0.73	0.80	0.75
Weighted avg	0.86	0.83	0.84

Uncalibrated XGBoost on all features: Confusion Matrix with default threshold

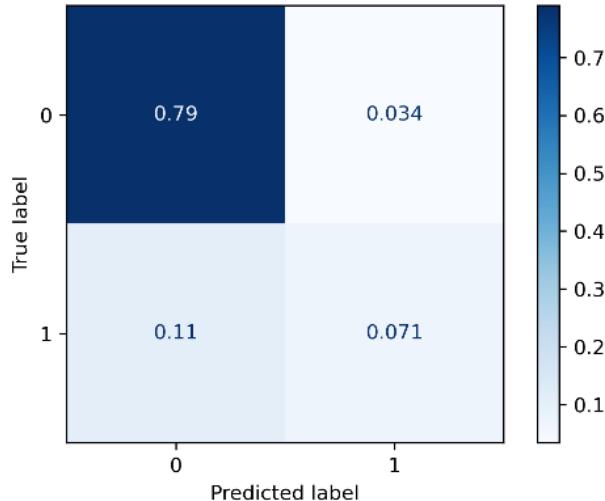


Figure 32 Modèle non calibré: matrice de confusion

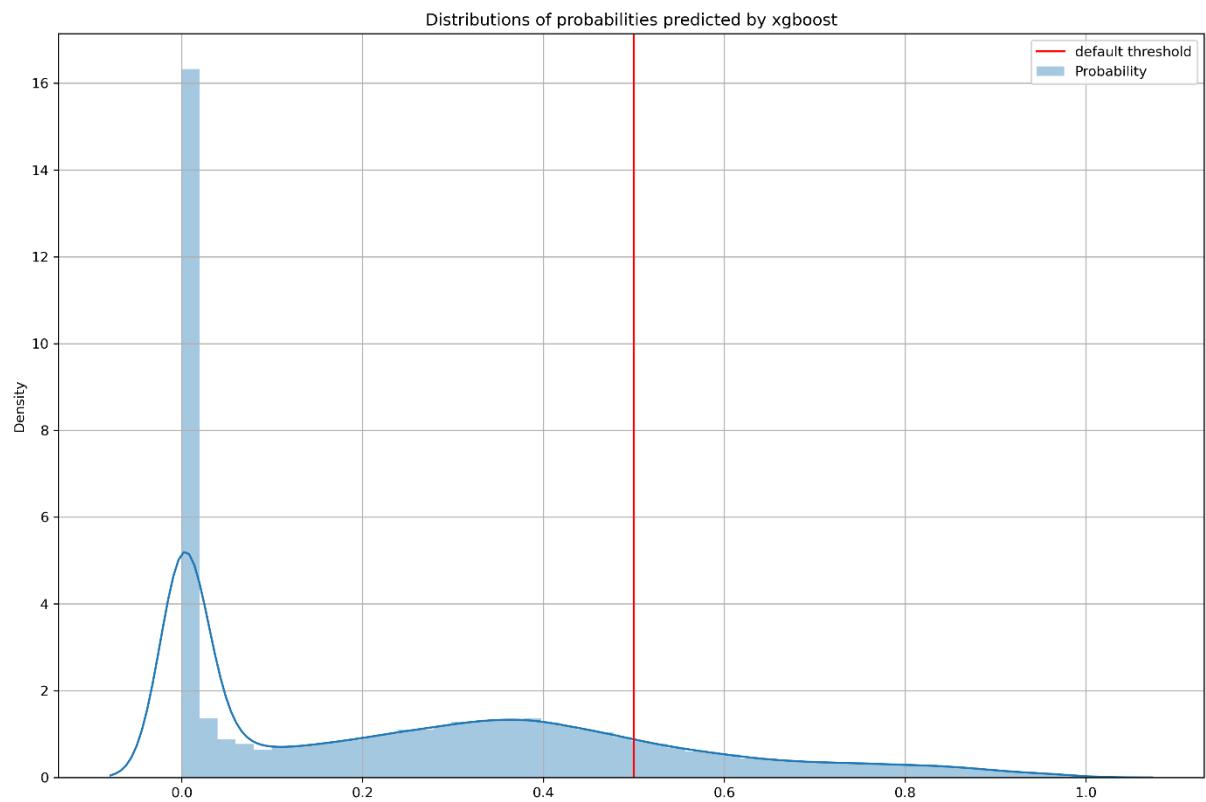


Figure 33 modèle non calibré: distribution des probabilités

La courbe de confiance confirme bien que les probabilités prédictes sont sous estimées par le modèle xgboost.

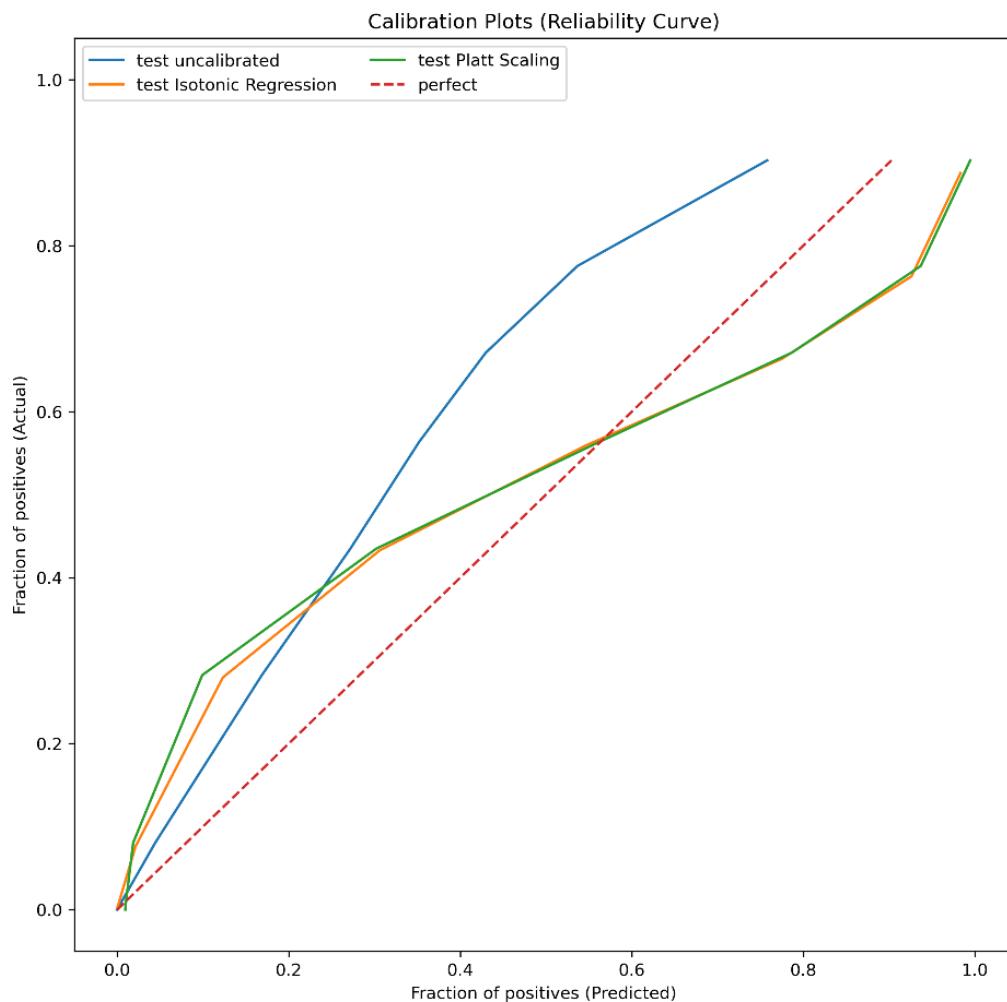


Figure 34 courbes de confiance

En procédant à une calibration des probabilités, (nous avons choisi la méthode *Platt scaling* pour sa simplicité), nous obtenons bien les 17 % attendus de paires dupliquées dans le dataset de challenge amis au prix d'un logloss à 0.4.

Les métriques de classification et le matrice de confusion deviennent :

	precision	recall	F1-score
0	0.94	0.85	0.89
1	0.68	0.75	0.61
Accuracy			0.83
Macro avg	0.73	0.80	0.75
Weighted avg	0.86	0.83	0.84

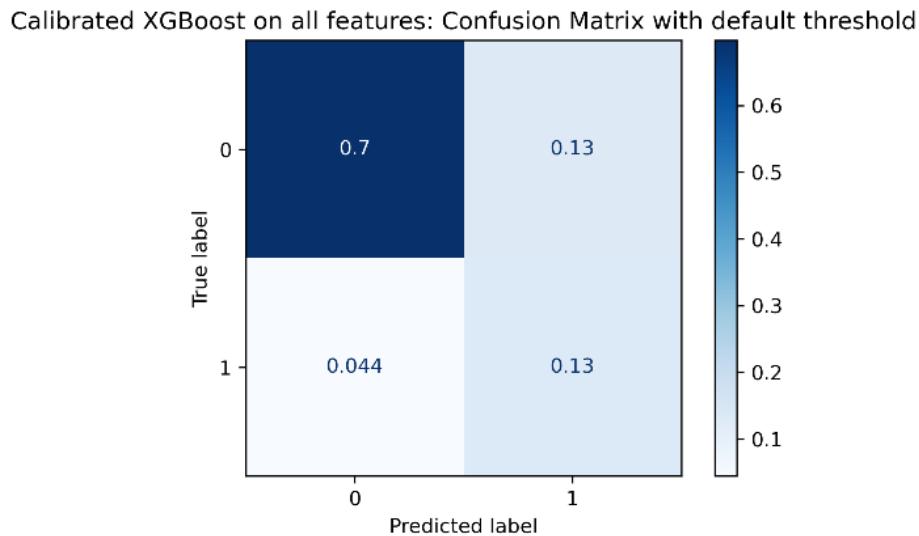


Figure 35 Matrice de confusion après calibration

Un bon score kaggle n'implique donc pas un bon modèle pour les décisions de classification : le logloss favorise les décisions prudentes (les paires identiques sont plus rares). Sans plus d'informations sur le coût des erreurs de classification, il est difficile de trancher sur la pertinence d'une recalibration des probabilités prédictes mais le modèle ne doit pas être utilisé aveuglément pour produire des décisions.

Simplification du modèle final : sélection de variables

Le modèle final utilise les interactions entre 103 indicateurs. Certains de ces indicateurs sont lourds à calculer (plus d'une $\frac{1}{2}$ heure pour l'indicateur de similarité en utilisant 8 cores). D'autres transportent a priori très peu d'information.

A cause de la complexité des interactions, il est difficile de procéder objectivement à des éliminations a priori. Plusieurs stratégies ont été essayées.

Les variables les plus importantes d'après le modèle

En utilisant les métriques internes d'importance du model xgboost global, nous avons sélectionné les 39 variables les plus importantes. En pratique, la sélection comporte :

- Toutes les probabilités d'appartenance à un newsgroup
- La similarité
- Quelques entités : EVENT, PERSON, LANGUAGE
- Quelques features non sémantiques : tous les ratio Nbre de mots communs/(nombre total de mots)

Uniquement les features sémantiques

- Toutes les entités
- Toutes les probabilités d'appartenance à un newsgroup

Les features 'faciles' à calculer

- Toutes les features non sémantiques hors lemmatisation
- Toutes les probabilités d'appartenance à un newsgroup

Stratégie	logloss	Score Kaggle
Meilleur modèle	0.2867	0.31176
39 features les plus importantes	0.2927	0.31516
63 Features sémantiques	0.3665	0.39054
46 Features 'faciles'	0.3010	0.32422

On constate expérimentalement :

- Il n'y a pas décidément pas de feature ou de groupe de features magique : les interactions apportent beaucoup.
- Les features associées aux entités peuvent être largement expurgées. En ne conservant que les catégories d'entités les plus peuplées, on conserve de bonnes performances.
- Les informations apportées par les newsgroups sont importantes
- Un modèle uniquement constitué de couteuses features sémantiques (ce qui inclus la similarité censée être très proche de notre problème) a des performances très inférieures à un modèle constituées uniquement de features simples.

En termes de ratio coût/performances, un modèle basé uniquement sur les features 'faciles' ('comptages simples hors lemmatisation+newsgroups) se dégage du lot.

Conclusion & Perspectives

Le challenge proposé par Kaggle est motivant mais effectivement difficile. Notre meilleur modèle nous projette (de peu) dans le premier tiers des meilleurs résultats. Indépendamment de la modestie du résultat final, nous avons pu au cours de cette étude :

- Manipuler des techniques standard pour explorer des corpus de textes et nous rendre compte de la difficulté de leur utilisation dans le cadre de ce challenge
- Evaluer l'intérêt de features simples pour construire des modèles passables
- Mesurer concrètement l'importance d'un bon processus de nettoyage des textes
- Utiliser les services avancés de bibliothèques NLP pour améliorer nos modèles
- Mesurer l'importance des techniques d'optimisation d'hyper paramètres
- Appréhender les difficultés inhérentes à l'évaluation d'un modèle

Pour aller plus loin, les pistes principales nous paraissent être :

- Ajout de nouvelles features : les possibilités sont nombreuses : indicateurs divers sur le vocabulaire utilisé, bi/trigrammes, utilisation directe de word2vec, doc2vec, Toutefois, en explorant les résultats fournis par les équipes gagnantes, il semble qu'il n'y ait pas de feature magique pour ce challenge : la détection des interactions entre features est la clef.
- *Utilisation d'algorithmes plus puissants capables de découvrir plus d'interactions.* Au prix d'une complexité importante, les réseaux neuronaux ont eu les meilleurs résultats dans ce challenge mais d'autres variantes de xgboost sont intéressantes.
- *Utilisation de techniques ensemblistes.* Indépendamment de la technique utilisée pour fabriquer les modèles, il est intéressant de mettre en collaboration plusieurs modèles en espérant que leur déficiences se compensent. Les différentes façons de gérer ces ensembles de modèles sont nombreuses et ont été abondamment explorées par les équipes gagnantes. Un danger connu de ces méthodes ensemblistes est le risque d'overfitting mais les méthodes évitant cet écueil sont bien documentées et abordables.

Annexes

Annexe A Modèle newsgroup

Les post de 20 newsgroups ont été collectés (18000 posts) et sont la base d'un grand nombre de démonstrations standards dans le monde du NLP. Pour chaque post, le titre du newsgroup est fourni.

Newsgroup
alt.atheism
comp.graphics
comp.os.ms-windows.misc
comp.sys.ibm.pc.hardware
comp.sys.mac.hardware
comp.windows.x
misc.forsale
rec.autos
rec.motorcycles
rec.sport.baseball
rec.sport.hockey
sci.crypt
sci.electronics
sci.med
sci.space
soc.religion.christian
talk.politics.guns
talk.politics.mideast
talk.politics.misc
talk.religion.misc

Ces newsgroups ont été fusionnés pour obtenir une liste plus réduite avec plus de textes dans chaque newsgroup:

Newsgroup
religion
computers
forsale
vehicles
science
politics

On construit un pipeline pour créer un modèle de classification multi-classe :

1. Comptage des mots (n-grams 1 & 2)
2. Génération des tf-Idf
3. Modèle Multinomial Naive Bayes
4. Optimisation des Hyper paramètres pour trouver classiquement alpha=0.01 comme valeur optimale

On obtient un modèle de classification avec une accuracy de 0.8.

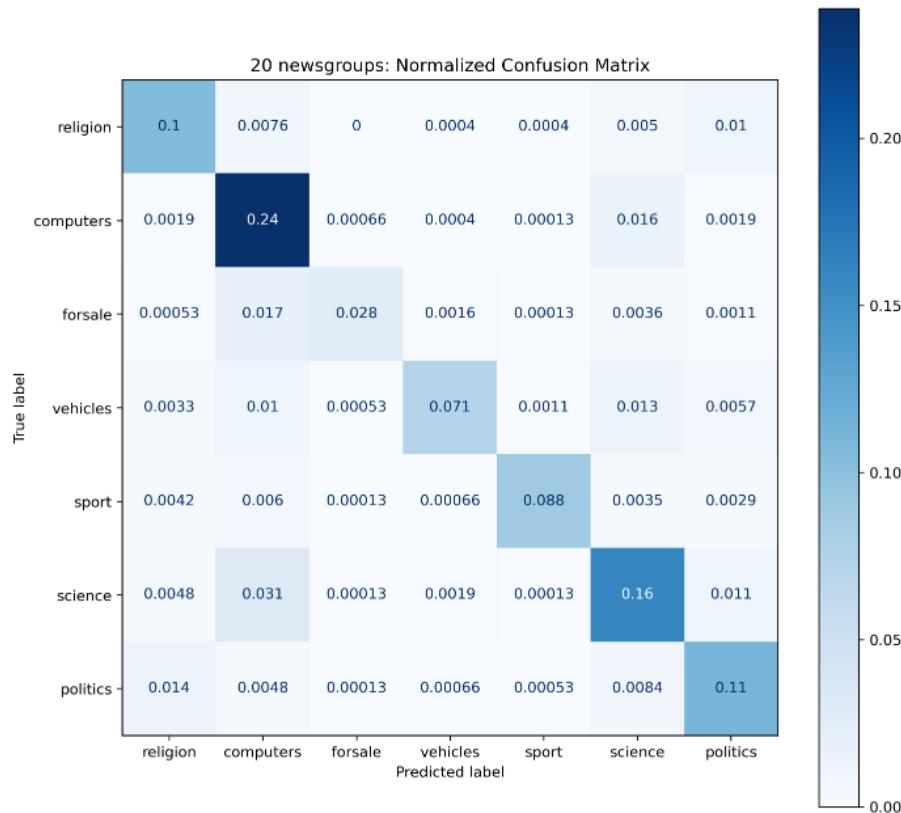


Figure 36 Matrice de confusion du modèle newsgroup

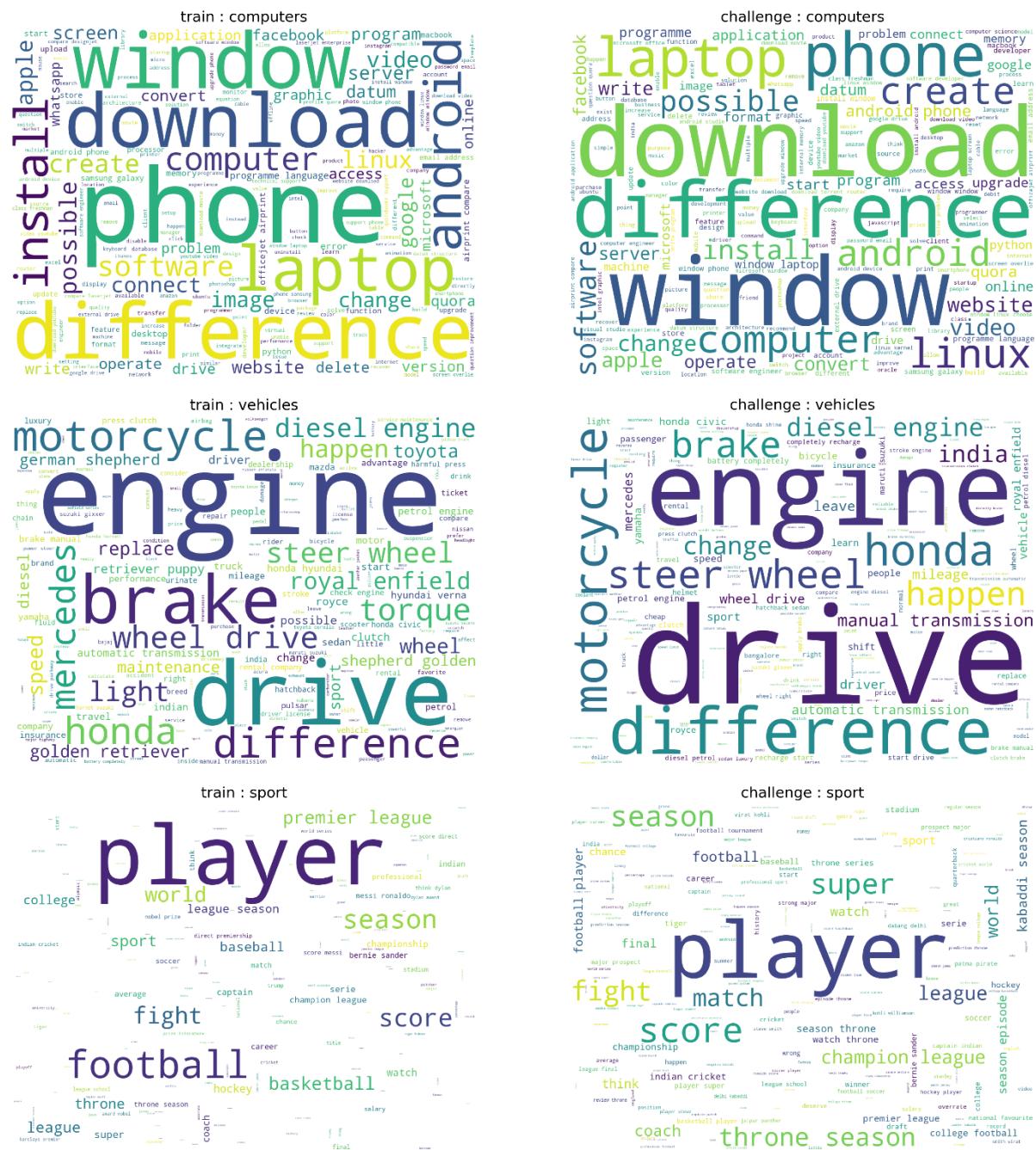
Ce modèle est appliqué aux datasets de training et de challenge pour :

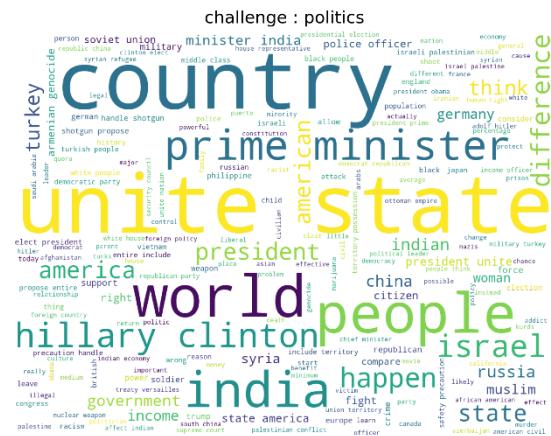
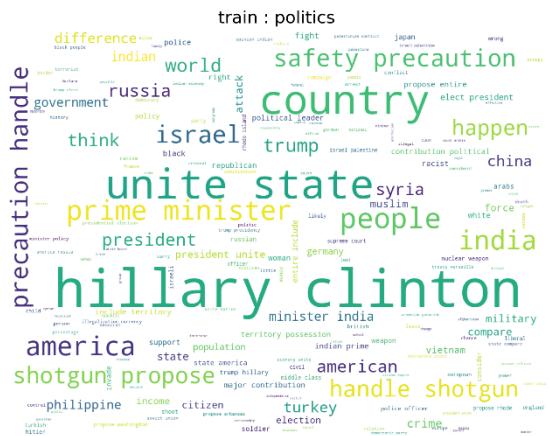
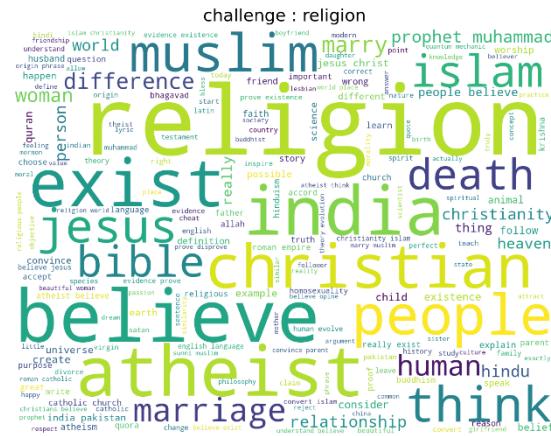
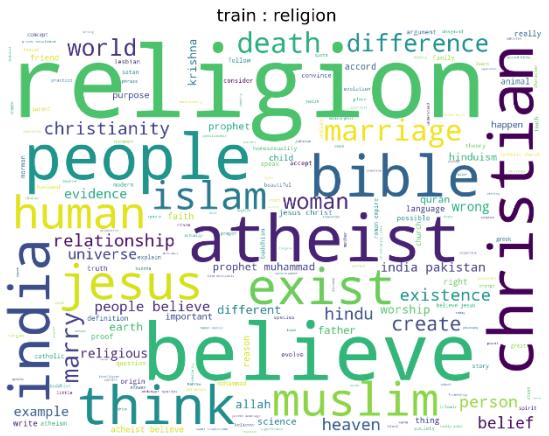
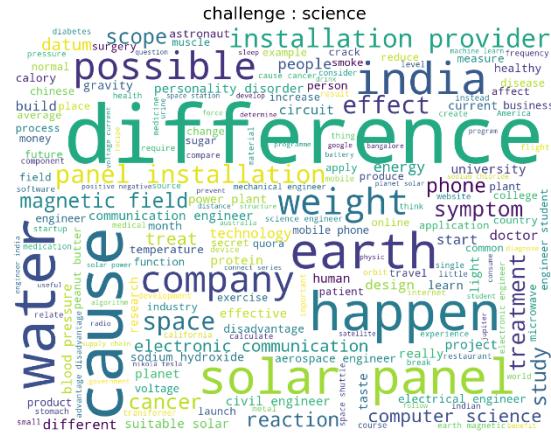
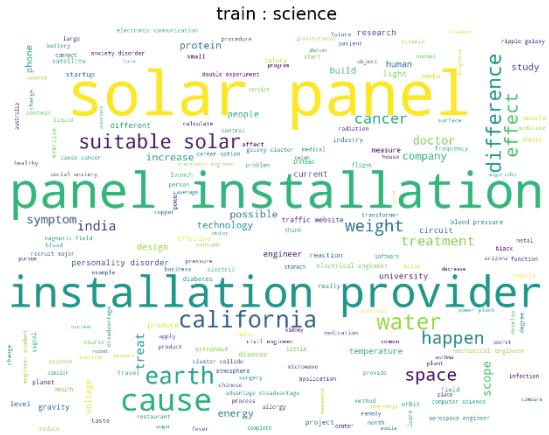
- Aider à la visualisation des textes des questions en les qualifiant avec une information sémantique de haut niveau (en appliquant un threshold de 0.9 sur les probabilités calculées)
- Rajouter une information sémantique de haut niveau et améliorer les modèles : *si le newsgroup estimé des 2 questions est le même, elles ont plus de chances d'être identiques.*

	question1	newsgroup_1
4	which one dissolve in water quickly sugar, salt, methane and carbon di oxide?	science
33	does the united states government still blacklist (employment, etc.) some united states citizens because their political views?	politics
54	how gst affects the cas and tax officers?	politics
55	how difficult is it get into rsi?	science
71	what is a narcissistic personality disorder?	science
76	how do i prevent breast cancer?	science
82	if someone wants to open a commercial fm radio station in any city of india, how much does it cost and what is the procedure?	science
90	what is the best reference book for physics class 11th?	science

Annexe B Nuages de points

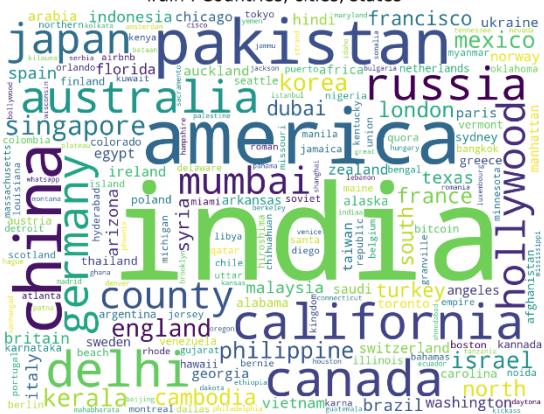
Newsgroups train/challenge



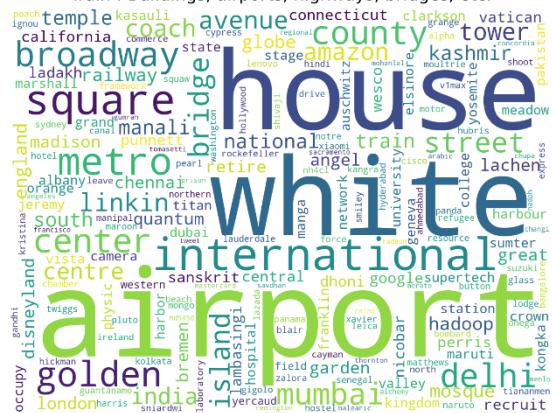


Entités training/challenge

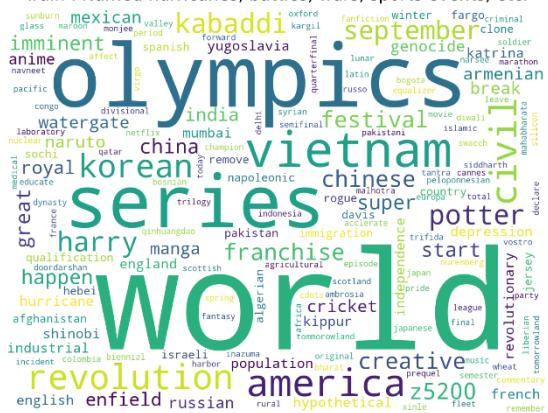
Train : Countries, cities, states



Train : Buildings, airports, highways, bridges, etc



Train : Named hurricanes, battles, wars, sports events, etc.



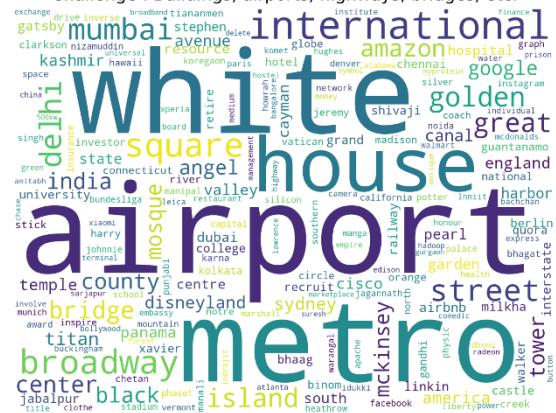
Train : Absolute or relative dates or periods



Challenge : Countries, cities, states



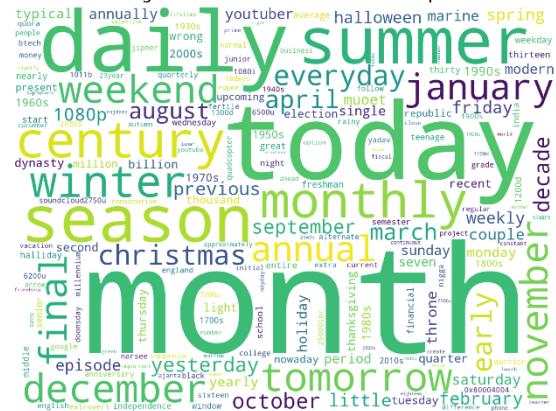
Challenge : Buildings, airports, highways, bridges, etc.

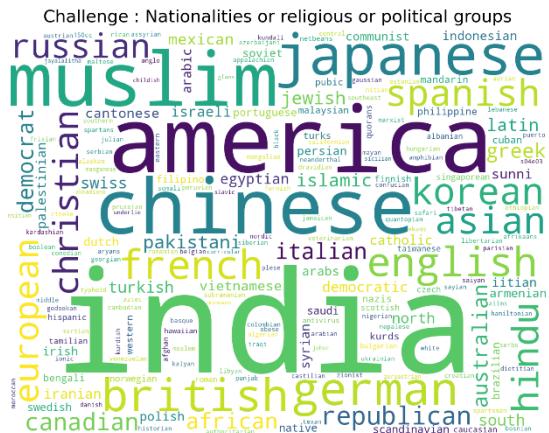
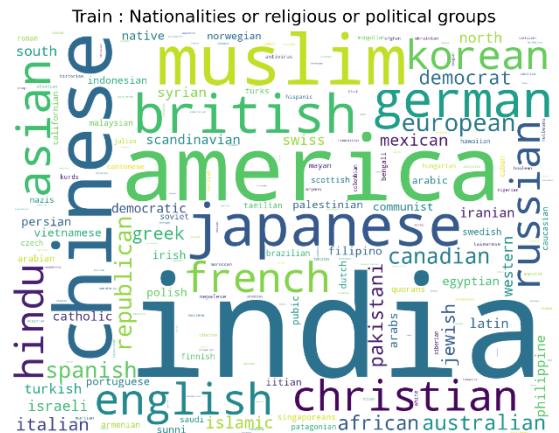
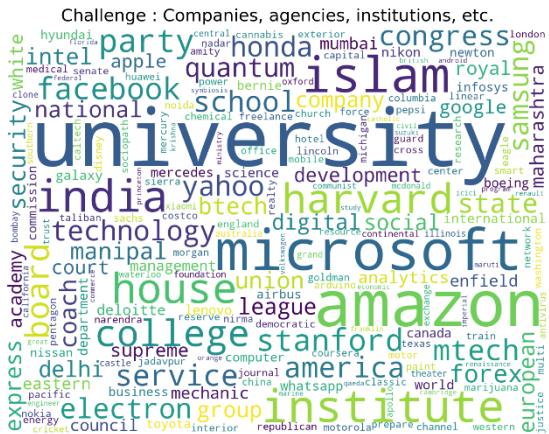
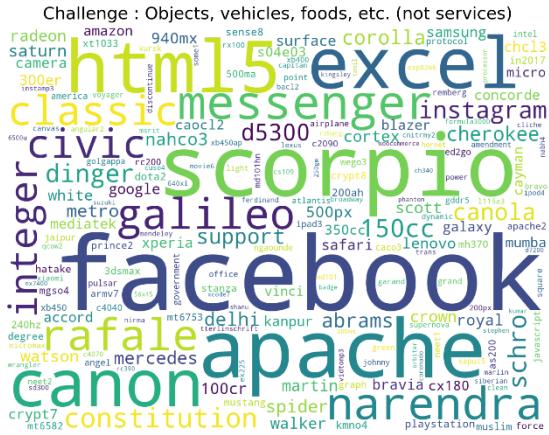
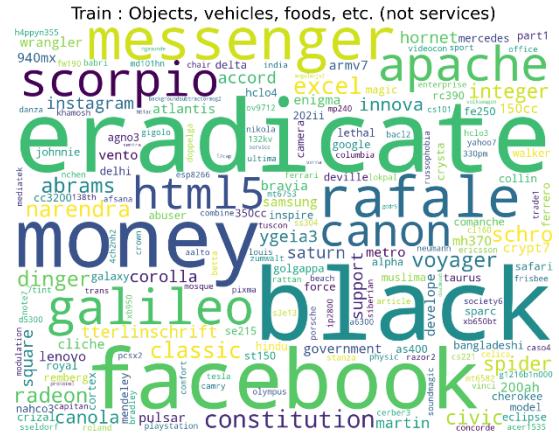


Challenge : Named hurricanes, battles, wars, sports events, etc.

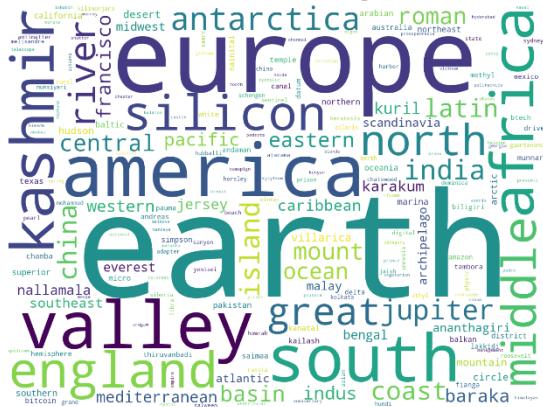


Challenge : Absolute or relative dates or periods

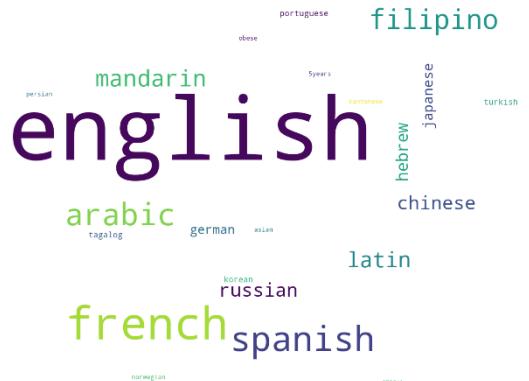




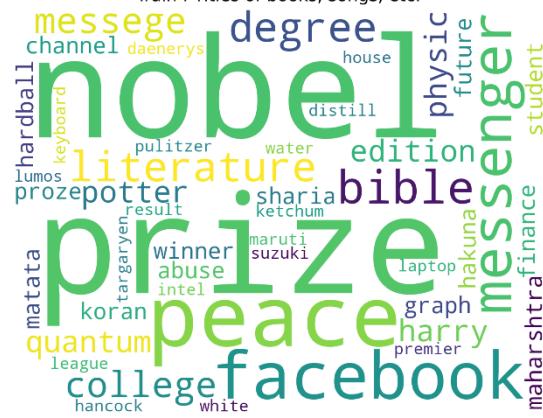
Train : Non-GPE locations, mountain ranges, bodies of water



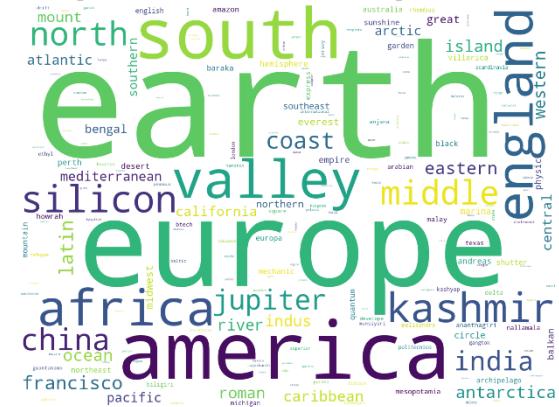
Train : Any named language



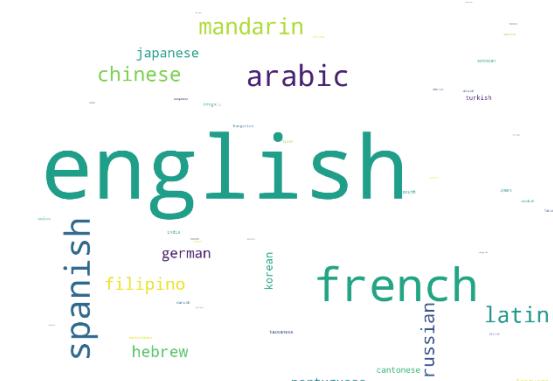
Train : Titles of books, songs, etc.



Challenge : Non-GPE locations, mountain ranges, bodies of water



Challenge : Any named language



Challenge : Titles of books, songs, etc.



Annexe C

Newgroups : LDA 10 topics

