CAMPAGNE D'EXPÉRIENCES

classification supervisée

L**** L** // L***** S******

Le manque de temps a induit un manque de recul sur le contenu des tests (les SVM prennent un temps infini, ce qui ne permet aucun ajustement). On aura aussi intérêt à reprendre les stratégies non supervisées. L'analyse avec LDA a été commencée, mais les résultats ne sont pas pertinents. Ils sont disponibles en fin de rapport.
On attend de pouvoir apporter des améliorations.

Ce rapport concerne des prédictions effectuées sur deux ensembles de données, des discours présidentiels et des descriptions de films. On s'intéresse ainsi à l'impact du prétraitement et de l'algorithme utilisé pour l'apprentissage sur la qualité de ces prédicitions. La différence majeure entre les deux corpus est qu'il s'agit de textes respectivement oraux et écrits ; y a-t-il des conséquences sur les approches à avoir ?

Les intuitions mises en œuvre ici sont très largement inspirées du rapport [Ayres, 2014] fourni en exemple à suivre. Les scores quant à eux ont pour objectif (facile) d'être dépassés. On cherchera à isoler les meilleures performances, indépendamment du temps de traitement, pour les soumettre au nom du binôme.

Performances en test finales :
~95% avec SVC, mix bigrammes, TFIDF + retrait des stopwords, 55000 mots/classe.

CHIRAC, MITTERRAND

La première tâche de ce travail porte sur des discours de MM. Chirac et Mitterrand, avec l'ambition de rattacher chaque phrase à celui qui l'a prononcé. Les tournures des phrases comptent probablement encore plus que les sujets traités sur des périodes historiques proches et des situations politiques relativement similaires. On s'intéresse donc entre autres à l'ouverture des phrases ou aux groupes de mots que l'on trouve souvent réunis.

On détaille ici la démarche suivie (en théorie) pour cette tâche et la suivante.

APPROCHES D'APPRENTISSAGE

Les algorithmes disponibles dans `scikit.learn` peuvent être testés tour à tour à la recherche du meilleur résultat. Les plus courants pour un apprentissage supervisé sont Naive Bayes (`MultinomialNB`) et SVM (`SVC`). Ce sont des techniques que nous maîtrisons du fait des UE précédentes, et nous ne pourrons rien tenter d'autre pour le moment.

L'approche Naive Bayes considère des mots supposément indépendants dans le texte, ce qui n'est jamais vrai ; c'est une hypothèse simplificatrice qui permet de gagner du temps et qui donne parfois d'excellents résultats en améliorant la portabilité du modèle. L'approche SVM écarte cette hypothèse d'indépendance. Elle repère toute interaction entre les mots pour établir les classes, ce qui la rend aussi plus lente. Le degré d'intrication de ces interactions dépend du noyau choisi (le linéaire est le plus simple, c'est celui que nous lançons ici, la justification de ce choix vient en page 5).

La lenteur des SVM étant rédhibitoire, les tests effectués avec sont moins exhaustifs et s'appuient en partie sur ce que nous avons tenté pour les données Movies. Et puis, il arrive qu'avec un bon paramétrage, un classifieur bien plus simple et plus rapide (soit Naive Bayes) donne un résultat satisfaisant sans gain significatif de performance.

REPRESENTATION DES DONNEES

Se pose ensuite la question de la représentation des textes. Il y a deux options, `BagofWords` (ou *unigrams*) et `Ngrams`, où le paramètre n ne dépassera pas 2 pour nous (on travaille alors sur une mixture d'unigrammes et de bigrammes). Nous allons entraîner chacun de nos classifieurs (`NB` et `SVC`) avec ces deux variantes pour comparer les résultats.

Bag of Words considère tous les mots comme indépendants : on a cassé la structure de la phrase et effacé la syntaxe (mise à part la flexion si absence de racinisation). Les n-grammes quant à eux ont l'avantage de représenter la suite des termes en contexte, ce qui permet de repérer les tics de langage d'un orateur, ce qu'on cherche à faire. Malheureusement, considérer des n-grammes augmente exponentiellement le volume des données et donc le temps de traitement ; on s'en tient donc aux bigrammes pour un bon compromis. A voir, de nouveau, lequel des deux choix est le meilleur pour les performances.

D'un autre côté, on se demande s'il faut coder chaque mot ou bigramme avec sa fréquence brute ou avec son score TF/IDF. Nous mènerons des essais sur les deux. Le codage TF/IDF peut sembler une mauvaise approche, parce qu'il écarte les mots fréquents (les tics) en les pondérant avec un coefficient moindre ; mais ce sont ceux des deux présidents ensemble, il arrive donc comme nous le verrons en seconde partie qu'il permette un meilleur rendement.

CHOIX DE PRETRAITEMENT

On obtient finalement huit variantes à tester avec le produit scalaire de {`MultinomialNB`, `SVM`}, {`BagofWords`, `NGrams`} et {`CountVectorizer`, `TfIdfVectorizer`).

Mais ces approches ne sont rien sans une préparation adéquate du texte brut. Pour attaquer la base de données, on commence par un prétraitement avec trois outils majeurs : la racinisation, ou réduction de chaque mot à sa base ; l'unification de la casse en minuscules, pour s'alléger de la question du découpage des phrases ; et enfin, la suppression des mots vides (ou *stopwords*) trop courants pour être significatifs. On s'est étonné.e.s déjà en TME de l'impact parfois négatif de ce prétraitement (qui était pour nous la chose systématique à faire) sur les performances. Nous avons quelques explications.

|||| Dans ces circonstances, la casse du texte est à conserver. Elle permet de visualiser les débuts de phrase des présidents – un outil important pour un orateur. Il ne fait aucun doute que Chirac et Mitterrand ont des manières différentes de placer l'emphase, le passage en minuscules est donc à éviter.
|||| La racinisation permet de regrouper les mots en familles sémantiques. Si elle permet de réduire la taille des données, ce qui est bienvenu, elle est essentielle en analyse de sentiments et donc tout à fait d'actualité pour des critiques de films ; mais ici, il s'agit de départager deux orateurs dans leur façon d'utiliser la langue, et c'est bien un cadre où les terminaisons comptent.
|||| Les mots vides n'ont pas d'importance capitale. Les retirer, c'est retirer du bruit. Les deux présidents utilisent sans doute à peu près également les termes simples et courants du dictionnaire. La liste des mots vides a été fournie dans le package `nltk`, et elle ne contient pas les termes trop évidents comme '`et`' ou '`de`' mis en lumière dans le rapport Ayres. Reste que ce qui est ou non un mot vide n'est pas toujours très intuitif : par exemple, '`cette`' n'en est pas un.

Les tests effectués prouvent la justesse de ce raisonnement. Pour les classifieurs Naive Bayes par exemple, la combinaison de traitement retenue consiste en le seul retrait des *stopwords*. – On le verra en page 3.

DEROULEMENT DES TESTS

Donc, huit stratégies algorithmiques, plus huit saveurs de prétraitement possibles en faisant le choix de raciniser ou non, de passer ou non en minuscules, et de retirer ou non les mots vides.
Ces stratégies sont testées avec un seul objectif, optimiser les performances tout en empêchant le surapprentissage. On cherche donc la dimension optimale des classes (nombre de mots à retenir dedans) : de fait, le codage en bigrammes des discours génère un total de 330 000 mots et couples de mots dont 28000 mots seuls. Hors de question de garder autant de dimensions pour déterminer le profil des présidents, puisque l'idée d'une modélisation est de simplifier et de s'appliquer à d'autres données. On teste donc chacune des variantes jusqu'à 100 000 mots par pas de 5000 pour bigrammes et jusqu'à 28000 (la totalité) par pas de 1000 pour sac de mots.

Pour juger de la qualité d'un modèle, le choix de la métrique est important. Pour nous, l'évaluation se fait grâce au score F1 qui compense le déséquilibre des classes, comme expliqué en cours. Le classifieur doit avoir à la fois un bon rappel et une précision correcte : il faut pallier le biais vers Chirac qui représente ici plus de 600% de Mitterrand (ce qui veut dire que sans F1, un classifieur qui classe tout en Chirac aurait un très bon score de base).

Maintenant que nous avons une métrique convenable, pour éviter le surapprentissage, il nous faut créer une pipeline et mettre en place une validation croisée. Cela permet de s'entraîner tour à tour sur des éléments différents des données et d'isoler à chaque fois un ensemble d'auto-évaluation. La performance en test à l'aveugle n'en sera que meilleure. Le paramètre conseillé en cours de ML était de cinq validations croisées, ce qu'on applique ici.

Donc, pour chacun des huit classifieurs, et pour chacune des huit combinaisons de paramètres, la dimension qui sera finalement retenue est celle pour laquelle la performance en validation croisée est la plus convaincante (avec un compromis entre qualité des résultats et dimension des classes.) On aura donc huit champs graphiques portant chacun huit courbes dont on cherche les maxima ; on isolera pour chacun la meilleure de ces courbes, puis l'on choisira entre les huit stratégies finalistes le mélange technique + prétraitement qui semble donner le meilleur résultat. C'est la seule qui accèdera au post-traitement.

Les comptes-rendus sont présentés en page suivante. Les scores F1 obtenus sont tous supérieurs à 93%.

CONCLUSIONS ET CRITIQUES

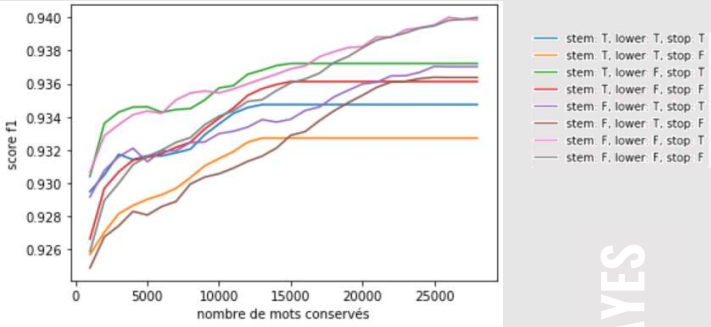
Il manque finalement quatre tests de SVC qui sont trop longs à faire. Nous avons donc choisi des paramètres un peu hasardeux pour générer au moins une courbe. Mais le temps mis à faire le calcul est récompensé : les scores de base de l'essai SVC sont meilleurs que le meilleur score de Naive Bayes.

Dans les essais Naive Bayes, le `TFIdfVectorizer` rend le prétraitement inutile. Les mots les plus courants du corpus sont déjà dépréciés dans la pondération. À ce moment, nul besoin de retirer les mots vides (pire, le faire diminue le score). Pour notre essai sur SVC avec le même *vectorizer*, nous avons maintenu ces mots vides ; peut-être avons-nous eu tort, nous ne le saurons pas.
Cet essai limité par le temps disponible a quand même donné les meilleurs résultats (en bas à gauche). La performance optimale est donc obtenue pour SVC, TFIDF, bigrammes, retrait des mots vides, et il n'est pas sage de lancer un calcul sur 80000 dimensions là où l'on peut en garder bien moins pour un résultat comparable : le choix final est donc un équilibrage à 55000.

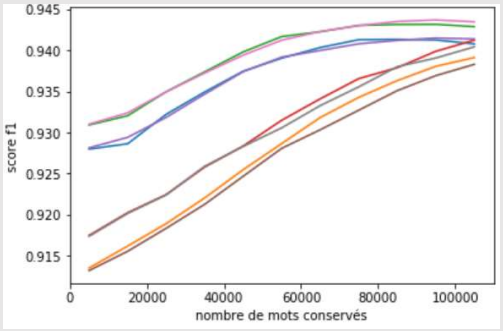
Un lissage a ensuite servi à retirer des incohérences ponctuelles en se fondant sur la présentation en blocs de discours. Si nous l'avions appliqué à d'autres stratégies, nous aurions pu obtenir des résultats encore meilleurs (selon le principe des escaliers *prévus pour la montée*). Toujours est-il que le gain de performance est d'environ 3% sur les données d'entraînement. Ne sachant rien sur la distribution d'où sont tirées les données et ne pouvant pas supposer qu'il s'agit de la même, nous avons cependant préféré oublier le lissage fréquentiel. Voilà d'où viennent nos prédictions.
Une remarque : les discours sont balisés. Il est possible de les isoler par groupes de lignes. Un seul locuteur est à l'origine de chacun de ces groupes de lignes. Avec un classifieur qui a raison 95% du temps, étendre la « valeur majoritaire » qu'on lit pré-lissage à l'intérieur des groupes peut être une bonne idée. Mais les 5% d'erreur seraient étalés si malchance, et ce serait aussi tricher.

MULTINOMIAL NAIVE BAYES

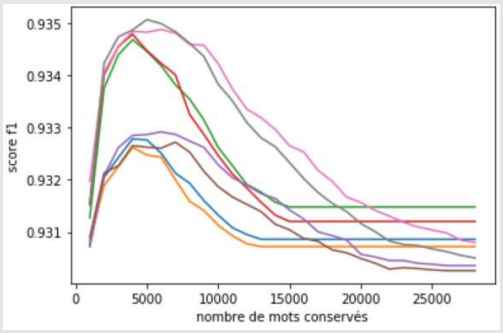
CountVectorizer, unigrammes
/Mots vides+stem., tradeoff estimé à 12000 mots conservés.



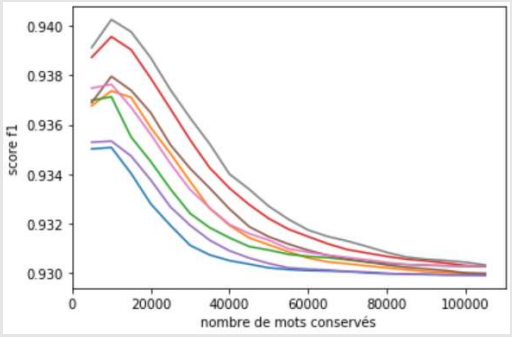
CountVectorizer, mix bigrammes
Retirer mots vides, tradeoff estimé à 70000 mots conservés.



TfIDFVectorizer, unigrammes
Ne rien faire, pic net estimé à 6000 mots conservés.



TfIDFVectorizer, mix bigrammes
Ne rien faire, pic net estimé à 10000 mots conservés.

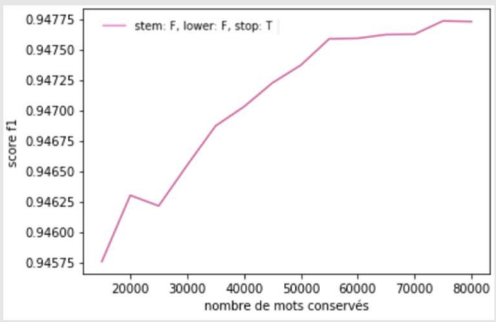


(L'aplatissement des courbes unigrammes à partir d'un seuil donné est dû à la disparition des mots vides.)
Sur les SVM, les tests sont beaucoup moins exhaustifs du fait du temps de traitement.

SUPPORT VECTOR CLASSIFIER

CountVectorizer, unigrammes
/Mots vides + stem., tradeoff estimé à ... mots conservés.

TfIDFVectorizer, mix bigrammes
Retirer mots vides, tradeoff estimé à 55000 mots conservés.



Impact du lissage
Application de la stratégie optimale à la base d'entraînement

94%

de prédictions justes
(score F1)

Situation après lissage
Gain de quelques % pour la performance

97%

de prédictions justes
(score F1)

avec suppression des lettres isolées
et application des fréquences de classe en entraînement.
Attention, seule la suppression des lettres isolées est appliquée pour le fichier des prédictions soumises (car on ne peut pas faire l'hypothèse assurée que les textes sont issus de la même distribution).

WORDCLOUD ENTRAÎNEMENT

Style Chirac

Style Mitterrand

En conclusion, voici les mots censément discriminants entre MM. Chirac et Mitterrand, avec un peu de nettoyage : pour la lisibilité, l'affichage est allégé des stopwords classiques auxquels on ajoute `c'est` et `cette`. (À enrichir des bigrammes si amélioration permise).

WORDCLOUD PREDICTIONS

Style Chirac

Style Mitterrand

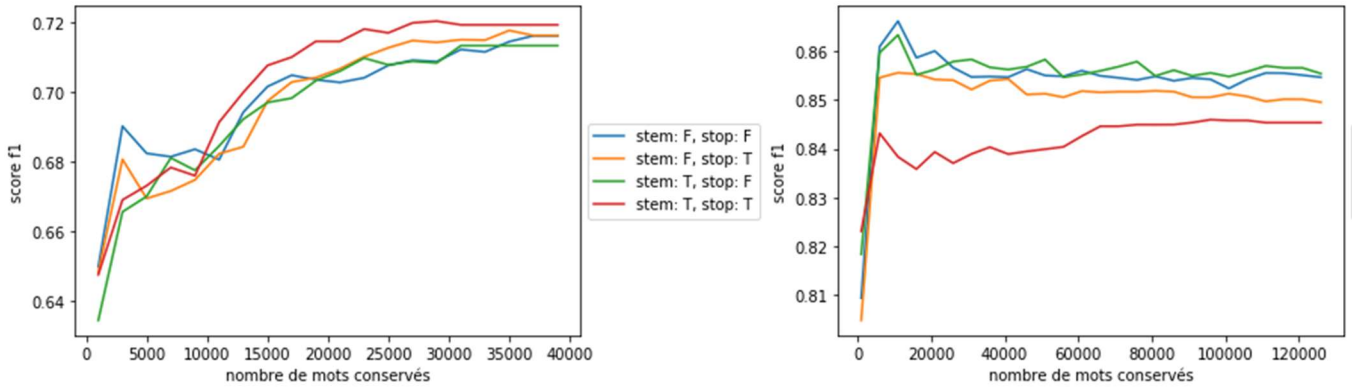
Performances en test finales :
~85% avec SVC, mix bigrammes, TFIDF
+ stemming ; 15000 mots par classe.

CRITIQUES DE FILMS

Le second essai concerne des données beaucoup plus diverses, des critiques de films. Il y en a cependant moins (2000 dont 1000 positives, 1000 négatives). Intuitivement, on se dit qu'on va s'intéresser un peu plus aux mots employés pour décrire une opinion. Mais les tournures ont elles aussi de l'importance.

On ne réexplique pas les tenants et les aboutissants de NaiveBayes et SVM, qui seront aussi nos techniques choisies sur cette partie. Les possibilités de codage sont toujours les mêmes (comptage, TF/IDF, unigrammes et bigrammes). En revanche, dans la phase de prétraitement, le changement de casse n'est pas une option : il a déjà été fait par défaut. C'est gagner du temps avec quatre courbes à tracer par lancement au lieu de huit.

Ajoutons ici une justification au choix du kernel linéaire comme outil de classification pour le classifieur SVM. Il n'y a que deux classes, avis positif et avis négatif, pour lesquelles une frontière linéaire peut se montrer largement suffisante. De fait, à l'essai, le noyau RBF (Radial Basis Function) qui est appliqué par défaut sur le SVC et qui peut prendre en compte des interdépendances plus complexes entre les classes donne des résultats beaucoup moins probants. Contrairement à celle des présidents, la taille de la base nous a permis de le vérifier rapidement et confirme qu'aller au plus simple est souvent le mieux sur des cas aussi basiques.



A gauche, la version RBF. À droite, le kernel linéaire que nous avons gardé.

CONCLUSIONS ET CRITIQUES

Le score moyen de l'ordre de 85% ne sera pas dépassé.
Les résultats sont présentés en page suivante.

On notera que sur tous les graphes, la chute des scores est générale (~10%) par rapport au cas des discours présidentiels : il s'agit pourtant des mêmes techniques. En fait, ici, plusieurs locuteurs s'expriment avec des styles bien différents, là où Chirac et Mitterrand restent bien les mêmes. Même si l'on résume les opinions des scripteurs à deux options binaires, les nuances sont plus complexes selon la source du message. Il est donc plus difficile pour un algorithme de s'en sortir.

On s'explique aussi la différence avec le rapport Ayres, où l'auteur avoue que ses résultats, plus élevés que les nôtres (environ 5% de différence), sont biaisés. En utilisant une `Pipeline`, nous avons remédié à son problème de l'apprentissage des TF/IDF sur la totalité des données : le dictionnaire est recalculé comme il faut à chaque étape de la validation croisée. Pour une même technique utilisée, l'existence de cette pipeline ne peut que diminuer le score en évitant le surapprentissage.

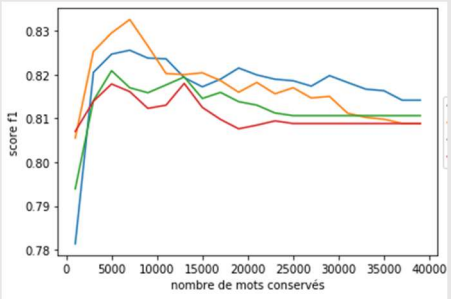
Mais Ayres pensait que le fait de calculer les TF/IDF sans pipeline et donc sur les données de test expliquait les performances supérieures de cette stratégie par rapport aux autres. Or, nous prouvons ici qu'elle est tout simplement la meilleure.
Le choix retenu est donc TF/IDF, mix bigrammes, SVM ((cf. page suivante, dernier graphique à droite).
Pour générer les prédictions finales, ce graphique nous donnait le choix entre *aucun prétraitement* et une *racinisation*, du fait des courbes associées qui sont à peu près identiques. Le choix de la racinisation a été fait puisqu'il diminue légèrement la taille des données à traiter. En revanche, aucun traitement n'a été fait contre les mots vides, car ils peuvent indiquer un certain état d'esprit du scripteur (les courbes pour cette stratégie montrent d'ailleurs que les retirer est la pire solution).

Après coup, nous avons pensé à d'autres éléments pertinents, comme à relever la ponctuation des critiques pour les catégoriser. Après vérification, la ponctuation survit à la racinisation, à la vectorisation et n'est pas considérée comme un mot vide ; nous sommes donc ravies qu'elle ait été prise en compte.

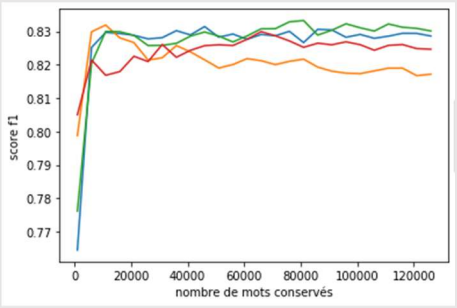
NB : par contre, la présence de majuscules dans les données de test (et pas dans les données d'apprentissage) rend la conversion en minuscules nécessaire par défaut pour la prédiction, afin d'adapter ce que le modèle « sait » à ce qu'il voit. On décide aussi d'ignorer le mot « /br » qui est certes omniprésent dans les critiques, mais qui était aussi totalement absent des données de train et qui pourrait de ce fait parasiter des facteurs plus importants.

MULTINOMIAL NAIVE BAYES

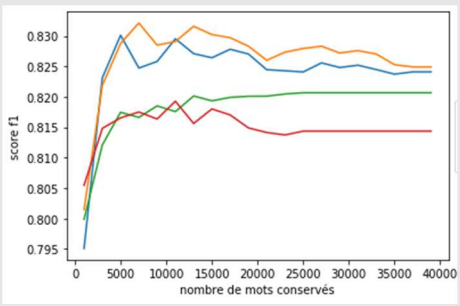
CountVectorizer, unigrammes
Retirer mots vides, pic net à 8000 mots conservés.



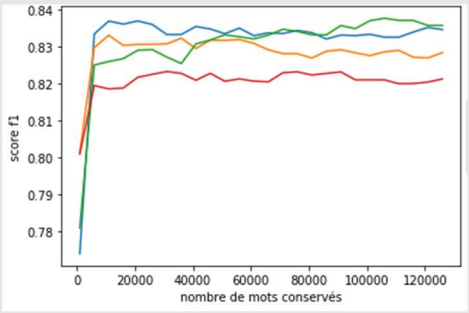
CountVectorizer, mix bigrammes
Tout comparable, tradeoff estimé à 80000 mots conservés.



TfIDFVectorizer, unigrammes
Retirer mots vides, pic net à 7000 mots conservés.



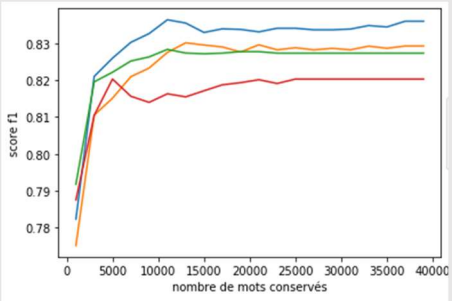
TfIDFVectorizer, mix bigrammes
Ne rien faire, tradeoff estimé à 20000 mots conservés.



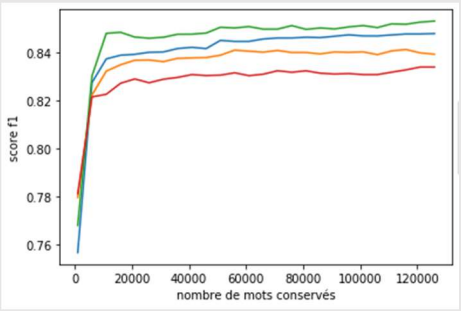
Généralement, sur cette base, les courbes s'aplatissent à partir d'une certaine valeur : inutile d'aller plus loin.

SUPPORT VECTOR CLASSIFIER

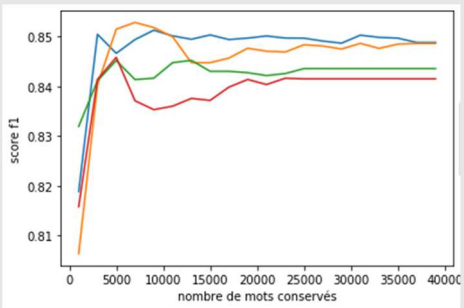
CountVectorizer, unigrammes
Ne rien faire, pic net à 10000 mots conservés.



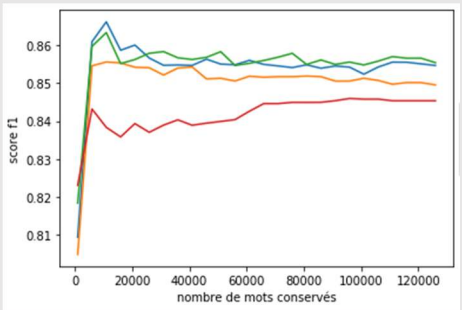
CountVectorizer, mix bigrammes
Stemming, tradeoff estimé à 50000 mots conservés.



TfIDFVectorizer, unigrammes
Retirer mots vides, pic net à 7000 mots conservés



TfIDFVectorizer, mix bigrammes
Ne rien faire ou stemming, pic net à 15000 mots conservés.



WORDCLOUD ENTRAÎNEMENT

Lexique positif

Lexique négatif

L’affichage des *wordclouds* est allégé des stopwords classiques augmentés de `br`, `film`, `movie`, `scene`, `make`, `one`, `show`, `story`, `not`. Y afficher des bigrammes aurait été possible avec un peu de code supplémentaire (à faire si amélioration permise).

WORDCLOUD PRÉDICTIONS

Lexique positif

Lexique négatif

Annexe
techniques de clustering
(non supervisées)

ESSAIS AVEC LDA

Le temps de calcul pour un seul passage est assez rapide. On ne s'intéresse pas ici à la justesse d'une prédiction, mais à ce que le modèle entraîné reconnaît comme sujets favoris chez les présidents. Les mots que nous disions *discriminants* plus haut avec les classifieurs supervisés sont en fait plus ou moins partagés par les deux présidents, à des fréquences différentes ; on pensait que les méthodes associées à un classifieur LDA (Latent Dirichlet Allocation) devaient permettre de visualiser facilement les expressions exceptionnellement importantes chez chacun.

On n'a pas vraiment tout à fait compris la façon dont il fallait définir les topics – notamment leur nombre. Plus les topics sont nombreux, plus la classification sera fine et juste. On pourrait ensuite fusionner ces topics pour savoir de quoi parle chaque président avec une technique de clustering. Le fait est qu'on n'arrive pas à savoir comment convertir ces topics épars en classification d'auteurs, ce qui serait d'ailleurs tout à fait pertinent pour notre projet PLDAC // pas assez lu de ressources pour le moment.

Comme les locuteurs traitent tous les deux de sujets variés, il n'y a presque aucune chance que tester avec seulement deux topics permette réellement de les séparer. On a essayé : on a ici jugé qu'un mot était discriminant si sa probabilité d'être dans une classe dépassait de loin celle d'être dans l'autre. On peut désormais donner et les mots probables, et les mots discriminants. Pour les premiers, on retrouve comme attendu des traits des deux locuteurs dans les deux ensembles : avis mitigé sur l'intérêt.

a c'est France cette
tout français bien nom
aujourd'hui si j'ai comme
très fait où tous
président monsieur date cela

plutôt chiraquien ?

plus aussi être pays
monde faire doit d'une
entre d'un l'europe tous
sans leurs faut développement
politique notamment paix peut

plutôt mitterrandien ?

Quant aux mots discriminants, à savoir `c'est`, `a`, `France` et `tout` selon notre code, ils ne veulent rien dire.

Avec un nombre de topics égal à 10, voici ce que dit l'algorithme (après stemming, etc.) :

tous	paix	plus	politique	être
toutes	place	tout	européenne	doit
toute	beaucoup	français	n'est	d'une
ensemble	grandes	monde	l'union	société
chaque	sécurité	d'un	dialogue	démocratie
doivent	parce	deux	donner	culture
qu'elle	ainsi	où	parlement	liberté
vivre	elles	sans	partenaires	communauté
heureux	compte	encore	réforme	trop
destin	contre	peut	peuvent	droits
vers	l'avenir	vie	cadre	lieu
avenir	entreprises	depuis	l'un	pense
responsabilités	europe	gouvernement	nouveaux	nouvelles
jour	moyens	ceux	négociations	internationale
sens	nations	d'abord	échanges	pouvoir
françaises	forte	grand	cas	république
comment	problèmes	devons	trouver	parmi
service	année	premier	système	face
avant		cœur	diversité	maintenant
débat			ministre	services

fait	aujourd'hui	a	pays	président
notamment	faut	c'est	nom	monsieur
économique	nation	france	entre	fois
l'on	jeune	cette	j'ai	s'est
toujours	grande	aussi	l'europe	justice
progrès	mieux	bien	leurs	messieurs
respect	crois	comme	dire	force
hommes	qu'ils	si	dont	nouvelle
d'être	alors	faire	chacun	conseil
social	l'histoire	qu'il	développement	mesdames
peu	celui	très	cet	première
déjà	nationale	cela	enfin	sais
autres	l'homme	date	volonté	c'est-à-dire
demain	général	temps	nouveau	élus
femmes	car	solidarité	tant	point
sociale	savez	travail	voilà	création
formation	sociaux	souvent	confiance	l'égalité
histoire	famille	rôle	également	cher
cours	seulement	sûr	dès	visite
celle	particulièrement	part	souhaite	rien

L'interprétation reste à trouver. La première colonne parle d'avenir commun, la quatrième de partenariat économique. La cinquième évoque des idéaux politiques et essaie de présenter une nation forte, ancrée sur des principes (enfin des grands mots) solides, la sixième donne un petit cours d'histoire patriotique. La dernière fait penser à une présentation de projet avec les marques classiques de la courtoisie diplomatique (mesdames et messieurs les élus, etc.). Si la tâche ne revient plus du tout à de la classification d'auteurs, nous pouvons ici donner une vague représentation de ce qu'est fondamentalement un discours politique français, mais sans plus. Elle ne sera pas très précise.

CONCLUSIONS GENERALES

et suite des événements

Nous avons pu tester diverses variantes de classifieurs supervisés (NaiveBayes et SVM sous plusieurs versions) et commencé à découvrir des méthodes de *clustering*. Les essais ont été plutôt concluants, mais le temps de traitement un peu trop long pour aller plus loin dans l'immédiat. La petite taille de la base Movies aurait pourtant permis bien d'autres tests.

Pour l'apprentissage supervisé, tenter la combinaison NBSVM dont parlent divers articles aurait pu être une bonne chose. De même pour le perceptron ou la régression logistique. Cependant, ce ne sont pas des techniques spécialisées en traitement du langage, et nous aurons encore l'occasion de les explorer en ML.

D'autres types de prétraitement n'ont pas été essayé. Il est par exemple possible de regrouper les synonymes avec des listes préétablies, ou de supprimer les mots dits « neutres » car redondants dans les deux classes (ce n'est donc pas tout à fait un tf-idf puisqu'on compare des fréquences intra-classes). La première de ces idées vaut pour faciliter un peu l'analyse de sentiments mais pas pour une tâche d'*authoring*, où le niveau de langage et le choix exact des termes est discriminant.

Comme annoncé en introduction, c'est bien la compréhension des spécificités du discours qui nous a permis d'intuiter le meilleur prétraitement – ou du moins d'expliquer nos graphiques –, preuve s'il en faut que les paramètres qualitatifs des données sont encore plus importants que leur taille et leur format.

Le plan pour une suite serait donc de tester ces autres prétraitements et d'aller plus loin avec LDA, en plus de mettre à jour les *wordclouds* avec des bigrammes (puisque c'est bien sur une base de bigrammes que nos deux prédictions ont été exportées).