Rapport de projet :

Classification de textes

présenté par :

Yuyan QIAN

Master 1
« Langue et informatique »
Méthodologie de la recherche en informatique

Enseignantes :

Laurence Devillers,
Nour El Houda Ben Chaabene

Table des matières

Table des figures

Liste des tableaux

Introduction

De nos jours, la classification de texte est l'une des tâches de traitement du langage naturel les plus courantes. La détection de spam, la catégorisation de genre textuel, la thématisation d'article, l'analyse de sentiment...notre monde numérique s'organise autour de ces buts précisés et détaillés. Vu que l'omniprésence de cette automatisation, la découverte concernant la chaîne de traitement de classification joue un rôle indispensable dans la compréhension de « Machine Learning » et TAL. De même, l'évaluation des résultats prédits est une composante aussi inégligeable.

Cette tâche consiste à entraîner la machine de sorte qu'elle pourrait catégoriser les textes à l'aide de divers algorithmes. Et le but de notre projet semestriel est de mettre en place un apprentissage supervisé et d'effectuer l'évaluation en comparant les méthodes de classifieurs : naïve bayésiennes, arbres de décision, forêts d'arbres aléatoires, SVM (Support Vector Machine). Les jeux de données choisis sont beaucoup variés allant des Tweets [1] à SPAMS [2], avec Brown [3] en complément. Tweets et SPAMS ont les classes dichotomiques : 0 et 1 (pour Tweets, 0 signifie un sentiment positif et 1 représente un sentiment négatif ; pour SPAM, 0 et 1 sont des valeurs booléennes, Vrai indique les mails spam tandis que False démonte les mails ham qu'on souhaite recevoir. Au contraire, le corpus Brown servit à la classification polytomique puisqu'il comporte quinze thèmes.

La structure proposée du rapport sera présentée comme suit : Dans le premier chapitre nous introduisons l'état de l'art. Le deuxième chapitre vise à présenter les jeux de données utilisées. Le troisième chapitre est dédié à la présentation des différentes méthodes d'apprentissage automatique supervisée ainsi que leurs avantages et leurs inconvénients. Et dernier lieu, nous présenterons les résultats obtenus, et nous avons également introduit les différents moyens d'évaluation d'un classificateur. Cette partie permettra d'évaluer les

1. https ://www.kaggle.com/datasets/hbaflast/french-twitter-sentiment-analysis
2. https ://github.com/Balakishan77/Spam-Email-Classifier
3. https ://www.kaggle.com/datasets/nltkdata/brown-corpus

performances des différentes approches implémentées dans une conclusion.

Chapitre 1

État de l'art

Comme ce que nous avons mentionné dans la partie d'introduction, la classification de texte est un point chaud de l'IA. Ses pratiques sont en plein essor non seulement dans le monde industriel mais aussi dans la recherche, par exemple, le jeu de données de sentiments de movie reviews de Stanford[4]. Aller plus loin, nous voyons aussi le Breast Cancer Wisconsin (Diagnostic) Data Set[1] dans le domaine de la médecine.

Il existe de nombreuses approches pour cette tâche en NLP, allant des méthodes basées sur les règles statistiques du texte aux méthodes basées sur l'apprentissage automatique. [1]. La méthode « traditionnelle » se base globalement sur les données mathématiques calculées selon le corpus, qui comportent le nombre de mot-clé, celui de fréquence par mot, etc. Celle-ci est un type d'analyse compréhensible, efficace et ainsi superficielle.

Plus profondément, les approches neuronales (Deep Learning) vont explorer les textes d'une manière autonomique. Le réseau de neurones convolutionnel est un exemple construit avec des couches de neurones. Personnellement, ce type de traitement a besoin de connaissance solide de mathématiques de sorte qu'il devient plus difficile à mettre en œuvre.

Sur le même dataset SPAM, des chercheurs ont testé plusieurs classifieurs comme naïve bayésienne[5], réseau de neurones[3] et machine à vecteurs de support (SVM)[6]. Presque tous les cas ont la performance excellente, soit la précision totale est plus que 95 pourcentages. Ce type de traitement permet à la fois de soulager les humains et affirmer l'efficacité de classification thématique.

Vu les diverses caractéristiques de textes, les modèles de classification se varie d'une manière flexible. Et les plus courants de nos jours sont les réseaux de neurones, les machines à vecteurs de support et les arbres de

1. D'après le cours magistral 1 M1SOL030 - Modèles de Linguistique Computationnelle - Ibtihel Ben Ltaifa-Corina Chutaux

décision. Les réseaux de neurones, en particulier les réseaux de neurones convolutionnels, ont été largement utilisés pour la classification de text. Les machines à vecteurs de support, quant à elles, sont particulièrement efficaces pour traiter de grandes quantités de données.

Chapitre 2

Jeu de données

Nous introduisons dans cette section l'étude de trois jeux de données : le premier est constitué de tweets (1526724 instances ans ce corpus dont 771604 négatifs et 755120 positifs), le second est constitué des courriels électroniques (5728 mails en total dont 4360 spams et 1368 hams. Le dernier Brown Corpus (Brown University Standard Corpus of Present-Day American English) est composé des phrases originales, les phrases tagged et les tags (57340 instances et 15 classes [1]).

	label	text
0	0	- Awww, c'est un bummer. Tu devrais avoir davi...
1	0	Est contrarié qu'il ne puisse pas mettre à jou...
2	0	J'ai plongé plusieurs fois pour la balle. A ré...
3	0	Tout mon corps a des démangeaisons et comme si...
4	0	Non, il ne se comporte pas du tout. je suis en...

FIGURE 2.1 – Exemple du corpus Tweet

Le jeu de données « French Twitter Sentiment Analysis » sur Kaggle contient des tweets en français avec leur sentiment associé (Voir Figure2.1). Il a été collecté à partir de l'API de Twitter et comprend environ 1.5 million tweets. Les tweets sont classés comme positifs, négatifs en fonction de leur contenu émotionnel. Ce jeu de données peut être utilisé pour entraîner

1. belles lettres, religion, mystery, reviews, government, adventure, humor, editorial, learned, lore, fiction, hobbies, romance, news, science fiction

des modèles de reconnaissance de sentiments sur les données de Twitter en français.

Le jeu de données « Spam Email Classifier » sur GitHub contient des courriels classés en tant que spam ou non spam [2]. Il comprend environ 2500 courriels, dont la moitié sont des courriels indésirables (spam) et l'autre moitié des courriels légitimes (non spam ou ham). Ce jeu de données peut être utilisé pour entraîner des modèles de détection de spam afin de filtrer automatiquement les courriels non désirés dans les boîtes de réception des utilisateurs.

	filename	para_id	sent_id	raw_text	tokenized_text	tokenized_pos	label
0	cd05	0	0	Furthermore/rb ,/, as/cs an/at encouragement/n...	Furthermore , as an encouragement to revisioni...	rb , cs at nn in nn nn , pps rb bez jj to vb c...	religion
1	cd05	0	1	The/at Unitarian/jj clergy/nns were/bed an/at ...	The Unitarian clergy were an exclusive club of...	at jj nns bed at jj nn in vbn nns -- cs at nn ...	religion

FIGURE 2.2 – Exemple du corpus Brown

Le Brown Corpus est un jeu de données de texte en anglais comprenant plus de un million de mots issus de diverses sources, telles que des journaux, des magazines et des romans (Voir Figure 2.2). Ses données par raport à Tweet et Spam sont plus riches où J'ai tiré la colonne Tokenized text à analyser.Il a été compilé en 1961 par Henry Kucera et W. Nelson Francis de l'Université Brown et est considéré comme l'un des premiers exemples de jeu de données de traitement du langage naturel. Le Brown Corpus est divisé en 15 catégories de textes, telles que des articles de journaux, des reportages, des récits, etc. Il a été utilisé pour étudier l'utilisation des mots et des phrases en anglais et pour entraîner des modèles de traitement du langage naturel.

Les informations plus détaillées sont organisées dans le tableau 2.1.

Corpus	**Instances**	**Classes**	**Phrases**	**Mots**	**Caractères**
Tweet	1526724	2	2572315	25979719	119084472
Spam	5728	2	91012	1878460	13351968
Brown	57340	15	61435	1175396	9518440

TABLE 2.1 – Taille de trois corpus

Les trois jeux de données sont des fichiers csv. En raison des différentes tailles, j'ai rencontré un problème avec de longs temps d'exécution lors de la

2. la même forme que l'exemple du corpus Tweets

lecture des données. La solution que j'ai trouvée sera analysée en détail dans la section Méthodologie.

En outre, la proportions entre des jeux de train et de test est répartie également, soit 30% pour *test* et 70% pour *train*. En chiffre, j'ai 1718 mails de test, ce qui fait que le train représente 4009 mails. En même temps, 458017 tweets de *test* contre 1068707 *trains*.

Chapitre 3

Méthodologie

Ce travail est généralement divisé en quatre étapes successives. Parmi celles-ci, la vectorisation et la classification sont centrales et, en fonction des caractéristiques de chaque donnée, j'ai pratiqué les traitements nuancés.

3.1 Lecture

Pour lire les tweets, les spams et les brown, j'ai introduit les fonctions de la librairie « pandas » [1] afin de les transformer en matrice. L'objet renvoyé par la fonction « pivot » [2] est de type « DataFrame ». Ensuite, la grande taille de dataset Tweet ont engendré l'exécution hyper lente. Pour avancer, une méthode spécifique qui pourrait servir à lire les bases de données importantes, à savoir Chunksize [3], ont mise en place. Chaque fois, une partie de données est lue à l'aide de cette fonction. En outre, après la lecture, il y a une combinaison de chaque lecture intitulée Concat. Vu que les corpus sont caractérisés par les mots-clés, les mots apparus deviennent mon objet d'analyse. Pour extraire ces caractéristiques, je me suis concentré sur la fréquence de mots. Voici les Étapes de la chaîne de mon traitement : Vectoriser les textes, pondérer les mots, ajouter les prétraitements.

1. https ://pandas.pydata.org/docs/reference/api/pandas.read_csv.html
2. https ://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pivot.html
3. https ://github.com/shachi01/dask_in_python_ml/blob/master/efficient_read_csv.ipynb

3.2 Vectorisation

La méthode CountVectoriser permets de représenter les données sous forme numérique. Afin d'enrichir mes données et d'améliorer la performance, j'ai fait appel à la pondération Tf-idf en remplaçant CountVectoriser avec TfIdfvectoriser. Les paramètres de la méthode vectorisation se composent des pre-traitements : Stopwords a une liste intégrée de mot-vide anglais. Pour le corpus majoritairement français (Tweet), j'ai tenté « Stopwords = None » et « max-df = 0.8 » afin de détecter et filtrer automatiquement les mot-vide à la base des contenus du corpus. Effectivement, ce coup d'essai a dégradé les résultats, soit la précision calculée qui sera présentée dans l'évaluation. Cela signifie que cette façon d'éliminer les mots vides ne sont pas effectives pour les dataset Tweet.

3.3 Classification

Le site sklearn fournit une carte de comment choisir un algorithme en fonction des données [4]. Et sur cette base, j'ai choisi 5 classifieurs : MultinomialNB de classifieur bayésien, arbres de décisions, LogisticRegression, LinearSVC de SVM et Perceptron.

A propos de chaque algorithmes, j'ai trouvé que leurs propres paramètres avaient une influence positive pour la bonne classification, par exemple pour Tweet et Brown, l'augmentation de la profondeur maximale de l'arbre de décision [5] améliorera continuellement la précision des résultats de la classification. Cependant, au fur et à mesure que la profondeur augmente, la machine prend plus de temps pour terminer l'apprentissage.

3.4 Evaluation

La façon la plus simple d'évaluer son modèle est de le tester sur des données dont on connaît déjà les labels et de comparer les résultats du modèle aux vraies valeurs des labels. [6]. Dans mon projet, plusieurs mesures de l'efficacité des classifieurs sont présentées : en total trois modules d'évaluation.

Le premier s'appelle Classification report et il se présente sous la forme d'un tableau. On y trouve toutes les mesures pour évaluer une classification, comme la précision (proportion des éléments pertinents parmi l'ensemble des

4. https ://scikit-learn.org/stable/tutorial/machine_learning_map/
5. max_depth
6. https ://ledatascientist.com/introductionalacategorisationdetextes/

éléments proposés), le rappel (proportion des éléments pertinents proposés parmi l'ensemble des éléments pertinents), le support (nombre d'instances concernées), la micro f-mesure (qui prend en compte la taille des classes) et la macro f-mesure (qui ne tient pas compte de la taille des classes). Le deuxième module s'appelle Confusion matrix et est plus concis, il présente les résultats bruts sous la forme d'un tableau à deux entrées. Le troisième module s'appelle precision recall fscore support et celui-ci présente les mêmes résultats que le Classification report, mais sous une autre forme. Enfin, en plus de ces deux méthodes d'évaluation, j'ai aussi noté la quantité de temps passée par chaque modèle, dans ce cas, l'évaluation se passe plus pertinente.

Chapitre 4

Résultats

Dans le but de bien présenter les résultats de mon projet, j'ai organisé les meilleures scores dans les tableaux avec les paramètres modifiés.

4.1 Avec le jeu de données de Tweet

Précision	Algorithme	min	max	stopwords	minuscules	$\max_{features}$
0.803813	LRegression	1	3	False	False	40000
0.801715	linearSvc	1	3	False	False	40000
0.790504	LRegression	1	3	False	False	50000
0.790209	LRegression	1	4	False	False	40000
0.790209	LRegression	1	1	False	False	40000
0.790196	LRegression	1	1	False	False	30000
0.789554	LRegression	1	4	False	False	20000
0.789554	LRegression	1	1	False	False	20000
0.789511	LRegression	1	1	False	True	30000

TABLE 4.1 – Résultats de Tweets

Les précisions de Tweets sont triées à l'ordre décroissante dans le tableau ci-dessous 4.1. En analysant, j'ai remarqué que la précision varie entre 0.79 et 0.80, ce qui est assez élevé. L'algorithme Logistic Regression est majoritairement utilisé dans les meilleurs cas. En conclusion, ce type de données a besoin de garder les mots vides et garder les formes de lettres. Lors que ngramme est dans une intervalle entre 1 et 3, la bonne classification est plus que celle entre 1 et 4. Cette idée corresponds parfaitement à la caractéritique française.

4.2 Avec le jeu de données de courriels

Selon la pratique de SPAM, j'ai constaté que presque tous les algorithmes ont une bonne entrînement pour les emails. J'ai eu une réussite très proche de cent pourcent. Cela indique ce type de texte est parfait comme un corpus à traiter. Dans un autre côté, ce phénomène m'a empêché d'améliorer les résultats et de découvirir les données.

Index	Précision	param	algorithme
0	0.9930	$\max_f$: 20000	linear_svc
1	0.9930	$\max_f$: 20000	Perceptron
2	0.9919	char-ngram :1-8	Perceptron
3	0.9902	word-ngram :1-2	linear_svc
4	0.9901	word-ngram :1-2	Perceptron

TABLE 4.2 – Résultats de SPAM

4.3 Avec le jeu de données Brown

Dans ce tableau, j'ai vu clairement le mauvais fonctionnemnt de tous les algorithmes testé. Perosnnellement, le problème vient de deux partie : le contenu de corpus n'est pas fait pour la catégorisation. Ensuite, de divers

Catégorie	Précision	Rappel	F1-score	Support
adventure	0.41	0.39	0.40	3278
belles_lettres	0.41	0.45	0.43	5057
editorial	0.32	0.25	0.28	2093
fiction	0.34	0.37	0.35	2975
government	0.39	0.57	0.46	2095
...	...	...	...	...
accuracy			0.42	40138
macro avg	0.41	0.37	0.38	40138
weighted avg	0.43	0.42	0.42	40138

TABLE 4.3 – Résultats de SPAM

classes n'adapte pas à la système de classification.

Conclusion

Dans ces expériences, nous avons étudié la classification de texte en utilisant plusieurs classifieurs sur des données en langue anglaise pendant plusieurs semaines. Nous avons également effectué des expériences similaires avec des données en langue française. Le classifieur qui s'est avéré le plus convaincant pour Tweet était le Logistic Regression, même si la variation des paramètres pouvait affecter ses performances. Pour les données Brown, Linaire SVC avait la meilleure performance tandis que Perceptron fonnctionn aussi mieux aue SVC dans le tri de SPAM. Nous avons constaté que la suppression des stopwords et de la ponctuation n'était pas systématiquement bénéfique pour augmenter les scores de classification, et que les algorithmes n'ont pas forcément les même meilleurs résultats face à de divers données.

Si je pouvais recommencer ce projet, j'approfondirais l'analyse du type d'article pour lequel les différents algorithmes sont les plus adaptés. En outre, il serait intéressant d'augmenter la quantité de données pour voir s'il est possible d'améliorer la qualité de la classification à cent poucent. Il serait également utile de tester davantage le pré-traitement des textes avec la machine plus compétente. En tout cas, Le corpus anglais est le corpus le plus approprié pour le traitement. Le français est beaucoup moins pertinent dans notre

classification globale. A l'avenir, le développement langagier reste à désirer.

Annexes

Dans mes codes, les parties de modification des paramètres ne sont pas pour éxecuter puisque ce sont les coups d'essai à augmenter la précision en finalisant ce projet. Et encore les parties de codage pour Tweet sont marquées le temps qu'il a pris.

Bibliographie

[1] Abien Fred M. Agarap. On breast cancer detection. In *Proceedings of the 2nd International Conference on Machine Learning and Soft Computing - ICMLSC '18*. ACM Press, 2018.

[2] W. N. Francis and H. Kucera (1964). Brown corpus.

[3] Sanaa Kaddoura, Omar Alfandi, and Nadia Dahmani. A spam email detection mechanism for english language text emails using deep learning approach. In *2020 IEEE 29th International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE)*, pages 193–198. IEEE, 2020.

[4] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[5] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes ? In *CEAS*, volume 17, pages 28–69. Mountain View, CA, 2006.

[6] Qiang Wang, Yi Guan, and Xiaolong Wang. Svm-based spam filter with active and online learning. In *TREC*, 2006.