Syllabs@DEFT2018 : combinaison de méthodes de classification supervisées

Chloé Monnin[1] Olivier Querné[1] Olivier Hamon[1]
(1) Syllabs, 35-37 rue Chanzy, 75011, France
monnin@syllabs.com, querne@syllabs.com, hamon@syllabs.com

RESUME

Nous présentons la participation de Syllabs à la tâche de classification de tweets dans le domaine du transport lors de DEFT 2018. Pour cette première participation à une campagne DEFT, nous avons choisi de tester plusieurs algorithmes de classification état de l'art. Après une étape de prétraitement commune à l'ensemble des algorithmes, nous effectuons un apprentissage sur le seul contenu des tweets. Les résultats étant somme toute assez proches, nous effectuons un vote majoritaire sur les trois algorithmes ayant obtenus les meilleurs résultats.

ABSTRACT

Syllabs@DEFT2018: Combination of Supervised Classification Methods
This paper describes Syllabs' participation in the task of tweet classification in the transport domain during DEFT 2018. For this first participation in a DEFT campaign, we have chosen to test several state-of-the-art classification algorithms. After a pre-processing step shared by all the algorithms, training is made based only on tweet content. The results being quite close, we realise a majority pool on the three best algorithms.

MOTS-CLES : Classification, SVM, régression logistique
KEYWORDS: Classification, SVM, Logistic Regression

1 Introduction

Les réseaux sociaux prenant une place croissante sur le Web et avec l'avènement de messages courts à des fins de communication, contextualiser l'information est de première importance. Le domaine de la classification vise à organiser, voire hiérarchiser, des *choses* (connaissances, concepts, objets, etc.) selon des classes établies au préalable. Cette classification permet, entre autres utilisations, de filtrer ces *choses* pour focaliser une utilisation, une étude, un traitement, etc.

Dans le cadre de la campagne DEFT 2018 (Paroubek et al., 2018), Syllabs a participé à la première tâche qui a pour objectif de classifier des tweets, selon qu'ils traitent du transport ou non. Dans le cadre de cette tâche un corpus de près de 70 000 tweets annotés nous a été fourni, que nous avons utilisé pour l'entraînement de 5 classifieurs supervisés, et après application de 5 prétraitements différents. De l'ensemble des 25 combinaisons, nous en avons tiré les 4 meilleurs, auquel s'est ajouté un vote des 3 meilleurs, pour soumettre 5 *runs* à la tâche de classification.

Dans un premier temps, nous revenons sur la description de la tâche à laquelle nous avons participé, puis nous décrivons notre méthode de travail, les données et classifieurs utilisées et les résultats obtenus sur un corpus de développement à partir de plusieurs combinaisons de modèles et prétraitements sur les données. Enfin, nous présentons les résultats obtenus sur le corpus de test avant de conclure.

2 Méthode

Nous avons concentré nos travaux sur la première tâche de DEFT, ce qui nous a permis de réaliser plusieurs combinaisons de tests afin de fournir un classifieur ayant les meilleurs résultats sur un corpus de développement.

Outre une sélection des données qui distingue corpus d'entraînement et de développement, notre méthode repose sur des principes simples, à savoir :

1. Prétraitement des données : nous avons testé 5 combinaisons de prétraitements divers.
2. Entraînement de 5 classifieurs supervisés sur le corpus d'entraînement appliqués à chaque prétraitement.
3. Observation des résultats, éventuellement adaptation des étapes 1 et 2.
4. Sélection de 4 *runs* issus des combinaisons prétraitement/modèle ayant obtenues les meilleurs résultats sur le corpus de développement.
5. Ajout du résultat d'un vote majoritaire sur les 3 meilleurs classifieurs en plus des 4 runs sélectionnés à l'étape 4.

Ainsi, 5 *runs* ont été soumis à la première tâche DEFT.

2.1 Sélection des données

Afin d'entraîner les classifieurs et de réaliser nos tests, nous avons séparé le corpus de tweets initial en deux corpus distincts. Le premier, servant de corpus d'entraînement, contient 90 % du corpus initial (soit 62 024 tweets). Le second, utilisé comme corpus de développement, contient les 10 % restant (soit 6 890 tweets).

La Table 1 présente les statistiques d'annotation sur les corpus d'entraînement et de développement, ainsi que le nombre total de tweets (nous avons ajouté les statistiques sur le corpus de test comme comparaison).

Corpus	**#tweets**	**#Transport**	**#Non-transport**
Entraînement	62 024	31 889	30 136
Développement	6 890	3 580	3 311
Test	7 815	-	-

TABLE 1 : prétraitements testés

La répartition des annotations transport et non-transport semble bien répartie que ce soit pour le corpus d'entraînement ou de développement. La taille du corpus de développement est comparable à celui de test.

2.2 Prétraitement

L'analyse de tweets nécessite *a priori* des traitements spécifiques dus à la nature particulière des données. L'information transmise y est réduite à son plus simple concept et brièveté de chaque tweet rend cette information difficile à interpréter. En particulier, le bruit est l'une des plus importantes caractéristiques à filtrer. Dans les tweets, le bruit prédomine généralement sur le contenu pertinent, mais les plus petits extraits d'information peuvent avoir leur importance. Ainsi, la préparation et le prétraitement des données sont des éléments essentiels ayant un impact sur la suite des traitements.

Afin d'étudier cet impact, nous avons mis en œuvre 5 prétraitements différents (*PT1 à 5*). Ceux qui nous fourniront les meilleurs résultats combinés aux classifieurs seront susceptibles d'être conservés :

- **Tokenisation** : le découpage des tweets a été réalisé à l'aide du tokenizer nltk (Loper & Bird, 2002)[1] ;
- **Suppression des URLs** : toutes les chaînes de caractères identifiées comme étant une URL sont supprimées ;
- **Suppression de la casse** : tous les caractères sont convertis en minuscule ;
- **Suppression de la ponctuation** : tous les caractères de ponctuation sont supprimés ;
- **Suppression des mots vides** : les mots vides sont supprimés, à partir d'une liste interne d'environ 300 termes ;
- **Pondération du lexique transport** : la pondération du lexique transport utilise une liste de plus de 1 100 termes et expressions résultant de l'analyse manuelle du corpus d'entraînement. Cette pondération consiste à doubler ces termes et expressions lors de la construction des modèles de classification.

La Table 2 présente les différentes combinaisons de prétraitements sélectionnées[2].

Prétraitements	PT1	PT2	PT3	PT4	PT5
Tokenisation	X	X	X	X	X
Suppression des URLs	X	X	X	X	X
Suppression de la casse		X	X	X	X
Suppression de la ponctuation (hors @ et #)		X	X	X	X
Suppression des mots vides			X		X
Pondération du lexique transport				X	X

TABLE 2 : prétraitements testés

[1] http://www.nltk.org/api/nltk.tokenize.html

[2] Nous n'avons pas jugé utile de combiner l'ensemble des traitements, certains de ces traitements nous semblant indispensables.

2.3 Construction des modèles

Nous avons choisi de tester 5 solutions de classification supervisée, parmi les plus courantes pour ce type de tâche. Pour chacune de ces solutions, à l'exception de l'approche naïve, nous avons fait varier leurs paramètres.

1. **Classification bayésienne naïve (CBN)** : Nous avons utilisé un premier classifieur naïf comme point de comparaison pour évaluer les autres classifieurs. Celui-ci utilise la distribution de la variable de Bernoulli[3].
2. **Machine à vecteurs de support[4] (MVS)** (*Support Vector Machine – SVM*) : Les noyaux RBF et linéaire ont été testés, le second ayant été laissé de côté car moins performant. Nous avons fait varier le paramètre de *C* entre des valeurs de 10^{-1} et 10^{5}.
3. **Régression logistique[5] (RL)** *(Logistic Regression)* : Nous avons joué sur les contraintes du modèle en faisant varier le paramètre de *C* entre des valeurs de 10^{-1} et 30.
4. **Arbre de décision[6] (AD)** (Decision Tree) : Nous avons réalisé des tests utilisant deux stratégies différentes de séparation des nœuds, *random* et *best*.
5. **Descente de gradient stochastique[7] (DGS)** (*Stochastic Gradient Descent – SGD)* : Nous avons fait varier le terme de régularisation avec les paramètres *alpha* (10^{-4} et 10^{-6}) et *penalty* (*l2*, *elasticnet*), ainsi que le nombre d'itération sur l'ensemble d'entraînement (10 à 80).

2.4 Résultats sur les données de développement

L'ensemble des 25 combinaisons (5 prétraitements différents, 5 classifieurs différents) a été évalué sur le corpus de développement en terme de précision (ligne du haut), rappel (ligne du milieu) et f1-mesure (ligne du bas).

Les résultats sont présentés dans la Table 3.

Lorsque l'on observe les résultats de l'évaluation sur notre corpus de développement dans leur ensemble, deux choses en particulier sautent aux yeux. Tout d'abord, les prétraitements n'améliorent pas beaucoup les performances, voire les abaissent. Ensuite, les écarts entre les classifieurs ne sont pas très importants.

Parmi les prétraitements, les suppressions de la casse et de la ponctuation affaiblissent les performances des classifieurs, de même que la pondération à partir d'un lexique transport (ce qui n'est pas le cas pour tous les classifieurs). Au contraire, la suppression des mots vides semble améliorer les performances, même si là aussi ce n'est pas le cas pour tous les classifieurs.

Il faut également noter que le vote majoritaire, de manière surprenante et mis à part pour quelques cas, n'apporte pas d'amélioration significative des résultats.

3 http://scikit-learn.org/stable/modules/naive_bayes.html#bernoulli-naive-bayes
4 http://scikit-learn.org/stable/modules/svm.html#classification
5 http://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
6 http://scikit-learn.org/stable/modules/tree.html#classification
7 http://scikit-learn.org/stable/modules/sgd.html

Modèle	PT1	PT2	PT3	PT4	PT5
Classification bayésienne naïve (CBN)	0,836 0,817 0,819	0,836 0,817 0,819	0,841 0,819 0,821	0,834 0,812 0,815	0,833 0,811 0,814
Machine à vecteurs de support (MVS)	0,864 0,837 0,840	0,862 0,834 0,837	0,845 0,824 0,827	0,859 0,825 0,829	0,847 0,822 0,825
Régression logistique (RL)	0,863 0,835 0,838	0,861 0,835 0,838	0,865 0,837 0,840	0,865 0,837 0,840	0,850 0,824 0,827
Arbre de décision (AD)	0,801 0,796 0,797	0,797 0,793 0,794	0,808 0,801 0,802	0,795 0,790 0,791	0,802 0,796 0,797
Descente de gradient stochastique (DGS)	0,824 0,807 0,809	0,812 0,802 0,803	0,775 0,775 0,775	0,813 0,800 0,802	0,804 0,794 0,796
Vote CBN+MVS+RL	0,867 0,837 0,840	0,864 0,836 0,839	0,864 0,836 0,839	0,860 0,828 0,832	0,853 0,825 0,828

TABLE 3 : résultats obtenus sur le corpus de développement en précision (1re ligne), rappel (2e ligne) et f1-mesure (3e ligne)

3 Résultats sur les données de test

A partir des résultats obtenus dans la section précédente, nous avons sélectionné les 4 meilleures combinaisons de prétraitements et modèle :

- PT1 + Machine à vecteurs de support ;
- PT1 + Régression logistique ;
- PT3 + Régression logistique ;
- PT3 + Machine à vecteurs de support.

En plus de ces combinaisons, nous avons ajouté un *run* de vote majoritaire entre les 3 meilleurs classifieurs sur le prétraitement PT3 :

- Vote PT3 + {Classification bayésienne naïve, Machine à vecteurs de support, Régression logistique}

Les résultats qui nous ont été rendus par les organisateurs de la tâche sont présentés dans la Table 4. Après réception des résultats, nous avons réalisé que notre système de vote n'était pas correct car il prenait en compte des pondérations approximatives fournies par les classifieurs. C'est pourquoi nous ajoutons une dernière ligne dans les résultats, qui ne fait pas partie des résultats officiels mais qui correspond à un vote majoritaire corrigé dont les résultats ont été calculés après la fin de la campagne.

Modèle	Précision	Rappel	F1-mesure
PT3 + RL	0,806	1,000	0,893
PT1 + RL	0,806	1,000	0,893
PT1 + MVS	0,800	1,000	0,889
PT3 + MVS	0,799	1,000	0,888
Vote PT3 + CBN+MVS+RL	0,792	1,000	0,884
Vote PT3 + CBN+MVS+RL (correctif)	*0,806*	*1,000*	*0,893*

TABLE 4 : résultats sur le corpus de test *(la ligne en italique, correction du système de vote, ne fait pas partie des résultats officiels)*

Face aux très bons scores en rappel, il est intéressant de connaître le nombre de tweets retournés par catégorie. Afin de comparer plus en détails les résultats entre classifieurs ainsi qu'avec le référentiel, la Table 5 fournit le nombre de tweets pour chaque classe.

Corpus	#Transport	#Non-transport
PT1 + MVS	4 756	3 060
PT1 + RL	4 684	3 132
PT3 + RL	4 663	3 153
PT3 + MVS	4 714	3 102
Vote PT3 + CBN+MVS+RL	4 918	2 898
Vote PT3 + CBN+MVS+RL (correctif)	4 723	3 093
Test	***3 941***	***3 875***

TABLE 5 : nombre de tweets trouvés par classe pour chacun des classifieurs

Tous nos classifieurs, sans exception, ont privilégié la classe transport. Ceci est sans doute dû à un léger sur-apprentissage de cette classe.

Les résultats du classifieur utilisant la régression logistique sont plus hauts que ceux du classifieur utilisant la machine à vecteurs de support. La différence entre les prétraitements (c.-à-d. simple tokenisation et suppression des URLs *vs* l'ajout des suppressions de la casse, de la ponctuation et des mots vide) est ténue et montre *a priori* que les classifieurs ne nécessitent pas d'effectuer de prétraitements de base pour viser une quelconque amélioration de leurs performances.

Il faut par ailleurs noter que le vote majoritaire n'améliore pas les résultats, bien qu'il confirme ceux du classifieur le plus performant.

4 Conclusion

Dans cet article nous présentons les travaux réalisés par Syllabs sur la première tâche de la campagne DEFT 2018. Ceux-ci ont été réalisés en très peu de temps, notre inscription, la première à une campagne DEFT, s'étant faite tardivement. Pour autant, nous avons obtenus des résultats état de l'art qui sont proches des meilleurs classifieurs ayant participés à cette campagne.

Nous sommes restés sur des approches classiques de classification supervisées. Sans être exceptionnels, les scores en f1-mesure sont hauts, même si cela est essentiellement dû aux scores en rappel à 1. En étudiant un peu plus en détails les résultats ainsi que les différents paramètres des classifieurs, nous pourrions sans doute augmenter légèrement la précision. De plus, c'est sans doute l'apport de données extérieures comme des lexiques ciblés sur le transport, ou la correction des tweets qui auront tendance à améliorer les performances.

Toutefois, l'un des faits les plus marquants de nos travaux est le faible impact des prétraitements sur ces résultats. Au final, l'utilisation presque originale des tweets donne des résultats convaincants, sans trop d'effort. Nous pensons par la suite étudier plus en détails ces résultats, à commencer par l'analyse de l'impact des prétraitements.

Références

PAROUBEK P., GROUIN C., BELLOT P., CLAVEAU V., ESHKOL-TARAVELLA I., FRAISSE A., JACKIEWICZ A., KAROUI J., MONCEAUX L., TORRES-MORENO J.-M. (2018) DEFT2018 : recherche d'information et analyse de sentiments dans des tweets concernant les transports en Île de France. In: Actes de DEFT. Rennes, France.

LOPER E. AND BIRD S. (2002) "NLTK: The Natural Language Toolkit," in Proc. ACL Workshop Effective Tools Methodologies Teaching Natural Language Process. Comput. Linguistics, 2002.