Défi DEFT08 : Classification de textes en genre et en thème : Votons utile !

Michel Plantié[1], Mathieu Roche[2], Gérard Dray[1]

[1] Laboratoire LGI2P, Ecole des Mines d'Alès, Site EERIE
Parc scientifique Georges Besse, 30035 – Nîmes,
(michel.plantie, gerard.dray)@ema.fr
[2] Laboratoire LIRMM, UMR 5506,
161 rue Ada, 34392 - Montpellier Cedex 5,
mathieu.roche@lirmm.fr

Résumé : Nous exposons dans cet article, les méthodes utilisées pour répondre au défi DEFT 2008. Après une présentation succincte de la méthode générale incluant les différents types de classifications utilisés, les résultats obtenus sont détaillés et analysés. Plusieurs tentatives d'améliorations des résultats initiaux sont enfin proposées.

Abstract: This paper explains the methods used for the DEFT 08 challenge. First we briefly present our general method including the different classification types used, and then we detail and analyze the results. Several attempts to improve the initial results are exposed.

Mots-clés : Classification, fouille de texte, Machine à Vecteurs Support, SVM, Naïve Bayes, Loi Multinomiale, sélection d'attributs, Apprentissage.

Keywords: Classification, text mining, Support Vector Machine (SVM), Naïve Bayes, Multinomial law, attribute selection, Machine Learning.

1 Introduction

Le défi consiste comme il est indiqué sur le site : http://deft08.limsi.fr/ à la prise en compte des variations en genre et en thème dans un système de classification automatique.

Par cette évaluation, le défi cherche à explorer les améliorations possibles d'un système de classification thématique par la prise en compte du genre. Ceci conduit à tester d'une part les utilisations du genre et du thème dans une classification automatique de documents, et d'autre part la robustesse d'une classification thématique vis-à-vis du genre.

Pour cette tâche, deux collections de documents de genres différents ont été constituées, l'une journalistique et l'autre encyclopédique, ayant en commun un certain nombre de catégories thématiques. Ce que nous mettons ici sous le terme genre renvoie à un

ensemble de textes partageant des propriétés liées au domaine d'activité, à des pratiques et au support utilisé pour ces textes.

Les corpus et leurs catégories d'évaluation sont :

- tâche 1 : un corpus d'articles de deux genres différents (journal Le Monde LM, Wikipédia W) d'un ensemble A (Economie ECO, Sport SPO, Art ART, Télévision TEL) de catégories thématiques avec un double étiquetage, l'un en genre et l'autre en catégorie thématique,

- tâche 2 : un corpus d'articles du journal Le Monde et d'articles de Wikipédia d'un ensemble B de catégories thématiques (France FRA, Société SOC, International INT, Livres LIV), différent de l'ensemble A, avec un simple étiquetage en catégorie thématique.

Afin de pouvoir trouver les méthodes de traitements toutes les équipes avaient accès à deux corpus d'apprentissage. Les corpus de test ont ensuite été fournis par les organisateurs du défi. Ainsi, le résultat de chaque équipe sur les données test a été évalué.

Un tel défi permet d'estimer globalement la qualité des méthodes de classification à partir de textes spécifiques (ici, des textes étiquetés en genre et en thème). Précisons que notre approche dans le cadre de DEFT'08 n'utilise aucun traitement spécifique propre aux corpus. En effet, le but du challenge est d'avoir des approches génériques de classifications. Notre approche générale a donc été intégralement appliquée sur chacun des corpus. Ainsi, la spécificité, notamment linguistique, de chacun des textes d'opinion (tournures de phrases, richesse du vocabulaire, etc.) n'a pas réellement été prise en compte dans notre approche.

Cet article qui se veut assez technique dans la présentation des résultats développe succinctement les méthodes appliquées et les résultats obtenus avec ces dernières. Cet article a pour but d'analyser la performance et également les contre performances des différents traitements appliqués. Bien que nos résultats soient très satisfaisants (situés au dessus de la moyenne des résultats des participants), certains résultats négatifs que nous avons obtenus sont volontairement présentés dans cet article. En effet, nous estimons que ceux-ci peuvent être particulièrement intéressants pour la communauté « fouille de texte ».

Après une présentation de notre méthode générale détaillée en section 2, la section suivante décrit les résultats obtenus. Enfin, la section 4 propose des méthodes additionnelles qui ont également été testées dans le cadre du défi mais qui n'ont malheureusement pas été toujours satisfaisantes. Enfin, la section 5 développe quelques perspectives à notre travail.

2 Processus global

Dans ce défi nous avons considéré que le problème posé relevait de la problématique de la classification. Chaque thème et chaque genre représente une catégorie et la tâche se traduisait donc en une procédure pour attribuer des candidats à une catégorie prédéfinie.

La méthode de traitement générique que nous avons utilisée comprend cinq étapes détaillées ci-après.

Étape 1 : prétraitement linguistique : recherche des unités linguistiques du corpus.

Étape 2 : prétraitement linguistique et représentation mathématique des textes du corpus

Étape 3 : sélection des unités linguistiques caractéristiques du corpus.

Étape 4 : choix de la méthode de classification

Étape 5 : Évaluation des performances de la classification par découpage ensemble apprentissage / test

2.1 Représentation des textes

2.1.1 Recherche des unités linguistiques de chaque corpus

Ce prétraitement consiste à extraire du corpus toutes les unités linguistiques utilisées pour la représentation des textes de ce corpus.

Dans notre méthode, une unité linguistique est un mot non lemmatisé.

Nous avons considéré que les mots non lemmatisés portaient davantage d'information que les lemmes associés. Cette information est de nature à améliorer les résultats de classification.

Nous extrayons donc tous les mots pour chaque corpus. Cela donne pour chaque corpus :

Corpus	**Nombre de textes**	**Nombre d'unités linguistiques (mots)**
Corpus 1	15225	191857
Corpus 2	23550	255161

Cette opération est effectuée avec l'outil d'analyse « weka » (Weka Project, 2002-2005).

Cette liste de mots pour chaque corpus constituera donc ce que nous nommerons un « index ».

Chaque texte sera représenté par un vecteur de « compte ». L'espace vectoriel de représentation est constitué par un nombre de dimension égal au nombre de mots du corpus. Chaque dimension représente un mot. Ainsi chaque coordonnée d'un vecteur représentera le nombre d'occurrence du mot associé à cette dimension dans le texte.

2.1.2 prétraitement linguistique et représentation mathématique des textes du corpus

Dans notre méthode nous n'utilisons pas de lemmatisation et nous n'appliquons aucun filtrage grammatical. Dans un processus où il s'agit de différentier des thèmes et des genres nous avons choisi de conserver tous les mots. Tous les types grammaticaux sont susceptibles d'exprimer des nuances en genre et en thème ou des contributions à ces catégories. Nous avons donc conservé les mots associés à tous ces types grammaticaux.

Vectorisation : Enfin la dernière étape consiste à transformer en vecteur d'occurrence chaque texte. Les dimensions de l'espace vectoriel étant l'ensemble des lemmes du corpus. Chaque coordonnée d'une dimension représente donc le nombre d'apparition dans le texte considéré du lemme associé à cette dimension.

2.1.3 Sélection des unités linguistiques caractéristiques du corpus

L'ensemble des textes d'un corpus et donc les vecteurs associés constituent dans notre approche l'ensemble d'apprentissage qui permettra de calculer un classifieur associé. L'espace vectoriel défini par l'ensemble des lemmes du corpus d'apprentissage et dans lequel sont définis ces vecteurs comporte un nombre important de dimensions. Par suite,

les vecteurs de chaque texte de l'apprentissage peuvent avoir de nombreuses composantes toujours nulles selon certaines de ces dimensions. On peut donc considérer que ces dimensions n'ont aucune incidence dans le processus de classification et peuvent même ajouter du bruit dans le calcul du classifieur entraînant des performances moindres de la classification.

Pour pallier cet inconvénient, nous avons choisi d'effectuer une réduction de l'index afin d'améliorer les performances des classifieurs. Nous utilisons la méthode très connue présentée par Cover qui mesure l'information mutuelle associée à chaque dimension de l'espace vectoriel (Cover & Thomas, 1991).

Cette méthode expliquée en détail dans (Plantié, 2006) permet de mesurer l'interdépendance entre les mots et les catégories de classement des textes.

Dans le tableau suivant nous présentons ces dimensions des espaces vectoriels obtenus pour chaque corpus.

Corpus	**Nombre initial d'unités linguistiques**	**Nombre d'unités linguistiques Après réduction**
Corpus 1	191857	12719
Corpus 2	255161	22106

2.1.4 Construction des vecteurs réduits de l'ensemble des textes de chaque corpus

Une fois les « index » de chaque corpus obtenus, nous considérons chaque mot clé sélectionné dans cet index comme les nouvelles dimensions des nouveaux espaces vectoriels de représentation des textes de chaque corpus. Les espaces vectoriels en question comporteront donc un nombre de dimensions largement réduit. Ainsi pour chaque corpus nous calculerons les vecteurs d'occurrence de chaque texte associé à l'index du corpus considéré.

Nous nommerons les vecteurs ainsi calculés : les vecteurs « réduits ».

L'utilisation de cette réduction d'index permet d'améliorer grandement les performances des classifieurs.

2.2 Choix de la méthode de classification

Une fois l'espace vectoriel réduit nous procédons au calcul du modèle de classification. Ce modèle sera ensuite utilisé pour l'évaluation des textes du jeu de test.

Nous avons utilisé plusieurs méthodes de classification. Elles sont fondées sur quatre méthodes principales.

Nous avons également testé d'autres procédures de classification dont les performances se sont révélées moins intéressantes.

Le choix de la procédure de classification s'est fait sur chaque ensemble d'apprentissage ou corpus. La sélection fut très simple, nous avons conservé la méthode de classification la plus performante pour un corpus donné. Les mesures de performances sont décrites ci après.

Nous décrivons brièvement ci-après les trois méthodes de classification. Notons que la plupart de ces méthodes est décrite de manière précise dans (Plantié, 2006; Plantié, Roche, & Dray, 2008).

En voici la liste :

- La classification probabiliste utilisant la combinaison de la loi de Bayes et de la loi multinomiale,
- La classification par les machines à vecteurs support S.V.M type SMO.
- La classification par les machines à vecteurs support S.V.M type Libsvm.
- La classification par la méthode des réseaux RBF (Radial Basis Function)
- La classification par boosting sur le classifieur de Bayes

2.2.1 Classifieur de Bayes Multinomial

Cette technique (Wang, Hodges, & Tang, 2003) est classique pour la catégorisation de textes nous l'avons décrite dans (Plantié, 2006). Elle combine l'utilisation de la loi de Bayes bien connue en probabilités et la loi multinomiale. Nous avons simplement précisé le calcul de la loi à priori en utilisant l'estimateur de Laplace pour éviter les biais dus à l'absence de certains mots dans un texte.

2.2.2 Classifieur par la méthode des Machines à Vecteurs Support (S.V.M.)

Cette technique (Joachims, 1998) a été décrite dans (Plantié, 2006). Elle consiste à délimiter par la frontière la plus large possible les différentes catégories des échantillons (ici les textes) de l'espace vectoriel du corpus d'apprentissage. Les vecteurs supports constituent les éléments délimitant cette frontière.

Plusieurs méthodes de calcul des vecteurs support peuvent être utilisées comme indiqué dans (Platt, 1998) :

- une méthode linéaire
- une méthode polynomiale
- une méthode fondée sur la loi gaussienne normale
- une méthode fondée sur la fonction sigmoïde

Nous avons essentiellement utilisé la méthode linéaire et celle fondée sur la loi.

2.2.3 Classifieur par la méthode des réseaux RBF (Radial Basis Function)

Cette technique implémente un réseau de neurones à fonctions radiales de base. Elle utilise un algorithme de « clustering » de type « k-means » (MacQueen., 1967) et utilise au dessus de cet algorithme une régression linéaire. Les gaussiennes multivariables symétriques sont adaptées aux données de chaque « cluster ». Toutes les données numériques sont normalisées (moyenne à zéro, variance unitaire).

Cette technique est présentée dans (Parks & Sandberg, 1991).

2.2.4 Classifieur par la méthode adaboost sur le classifieur Naive Bayes Multinomial

Ce classifieur a pour objectif de doper les performances d'un classifieur associé par l'utilisation de la méthode Adaboost M1 (Yoav & E., 1996). Cet algorithme améliore souvent de façon importante les résultats d'un classifieur mais quelquefois déprécie les résultats. Dans le cas du classifieur de Bayes nous avons constaté que les résultats de Adaboost étaient souvent légèrement meilleurs.

2.3 Évaluation des performances de la classification par Apprentissage / Test

La classification par découpage apprentissage test est une technique d'évaluation permettant de valider une méthode de classification en particulier. Nous avons utilisé cette méthode plutôt que la validation croisée. Les temps de calculs étaient trop long en validation croisée.

Cette approche construit un modèle incomplet non utilisable mais sert à estimer l'erreur réelle d'un modèle selon l'algorithme suivant (figure 1) :

Apprentissage/Test (S;x) *:*
// S est un ensemble,
Découper S en 2 parties S1, S2 (S1=80% de S ; S2= 20% de S)
Effectuer la réduction d'index sur S1, et la propager sur S2
Construire un modèle M avec l'ensemble S1
Evaluer une mesure d'erreur ei de M avec S2

Processus Apprentissage / test

Dans notre approche nous avons utilisée la méthode sur l'ensemble des vecteurs « non réduits » d'un corpus. L'objectif que nous nous sommes fixés dans le cadre du défi est d'évaluer nos résultats à partir du seul corpus d'apprentissage disponible. Ceci nous a aidé à adapter les paramètres les plus pertinents.

Pour évaluer la performance d'un procédé de classification nous utilisons la mesure préconisée dans le cadre du défi DEFT07 c'est à dire le « fscore ». Il s'agit de la moyenne harmonique de la précision et du rappel. Ces deux mesures sont bien connues, et une explication complète de ces mesures est écrite dans (Plantié, 2006).

2.4 Système de vote de classifieurs

Afin d'améliorer les scores obtenus précédemment nous avons utilisé des procédures de vote.

Nous avons testé plusieurs approches :

- le vote de 6 classifieurs :
  - Naive Bayes Multinomial, SVM-SMO, -SVM-Libsvm, Adaboost , Complément Naive Bayes
- le vote de 5 classifieurs :
  - Naive Bayes Multinomial, SVM-SMO, SVM-Libsvm, RBFnetworks, Adaboost , Complément Naive Bayes
- le vote de 4 classifieurs
  - Naive Bayes Multinomial, SVM-SMO, Adaboost , Complément Naive Bayes
- le vote de 2 classifieurs
  - Naive Bayes Multinomial, SVM-SMO,

Vote majoritaire

Nous avons appliqué cette procédure pour les deux corpus.

Le principe est le suivant :

Nous prenons les résultats de deux classifieurs ou plus. Pour chaque texte évalué nous retenons la réponse qui emporte la majorité.

Vote tenant compte du fscore de chaque classifieur

Nous avons utilisé les résultats du rappel et de la précision pour chaque classifieur afin de trouver une procédure de vote. Nous avons utilisé cette procédure sur le corpus 1 et sur le corpus 3.

Dans le corpus 1 nous avons sélectionné pour chaque classe le classifieur ayant le meilleur résultat de précision sur cette classe.

Ainsi à chaque classe correspondait un classifieur. Nous avons utilisé deux classifieurs pour cette procédure de vote : SVM, et Naïve Bayes Multinomial.

Dans le corpus 3 nous avons sélectionné pour chaque classe le classifieur ayant le meilleur résultat de rappel sur cette classe.

Ainsi à chaque classe correspondait un classifieur. Nous avons utilisé deux classifieurs pour cette procédure de vote : RBF-Network, et SVM.

Les résultats obtenus sur les ensembles d'apprentissage sont moins bons que les systèmes de vote précédents. Nous ne les avons pas utilisés sur les jeux de tests.

3 Résultats obtenus avec la processus global

Nous allons présenter ici les résultats obtenus sur les corpus d'apprentissage et les corpus de tests fournis dans le cadre du défi DEFT'08.

Nous allons présenter ces résultats par corpus.

Dans les tableaux présentés ci-dessous, il existe peu de différence entre ceux obtenus par la phase d'apprentissage et ceux obtenus sur les corpus de test. Cette absence de différence est expliquée à la fin de ce chapitre.

3.1 Corpus 1

En utilisant la méthode générale présentée précédemment nous avons sélectionné plusieurs classifieurs performants.

Le corpus d'apprentissage comporte 15225 textes dont :

5767 textes ART, 4630 textes ECO, 3474 textes SPO, 1354 textes TEL,

Ce corpus est un peu déséquilibré, la dernière catégorie comporte deux fois moins d'individus que les autres. Le déséquilibre entre les tailles des catégories pose souvent des difficultés pour obtenir de bons scores de classement. En effet si la performance sur la classe la plus volumineuse est faible en pourcentage de fscore le nombre d'échantillons mal classés devient important et les performances sur les autres classes deviennent bien plus faibles.

Dans le cas d'un corpus déséquilibré la performance de l'ensemble dépend en grande partie de la performance obtenue sur la catégorie comportant le plus grand nombre d'échantillons.

Nous avons effectué la détection du genre et du thème en même temps pour le corpus 1, c'est-à-dire que nous avons considéré 8 catégories 4 thèmes x 2 genres.

	Genre / thème	**Jeu de test**			**Jeu d'apprentissage**
Type de classifieur		Précision	Rappel	Fscore	Fscore Genre + thème
5 classifieurs	genre	97,139%	97,004%	**97,072%**	90,34%
	thème	88.503%	82,288%	**85,282%**	
6 classifieurs	genre	97,071%	96,923%	**96,997%**	89,98%
	thème	88.326%	82,314%	**85,214%**	
SVM-SMO	genre	95,600%	95,400%	**95,500%**	97,30%
	thème	85,388%	79,493%	**82,335%**	90,04%

Les résultats en apprentissage sur les deux premiers classifieurs sont affichés sur 8 catégories mélangeant genre et thème. Le dernier résultat a été effectué par détermination du genre et du thème séparément.

- le classifieur par vote à 5 classifieurs est très performant à la fois sur le jeu d'apprentissage et le jeu de test.
- le classifieur par vote à 6 classifieurs est un peu moins performant que le précédent à la fois sur le jeu d'apprentissage et le jeu de test.
- Le classifieur SVM donne des résultats inférieurs aux systèmes fondés sur un vote. Nous constatons une différence significative sur la détection du thème entre le jeu d'apprentissage et le jeu de test par ce classifieur.

3.2 Corpus 2

En utilisant la méthode générale présentée précédemment nous avons sélectionné plusieurs classifieurs performants.

Le corpus d'apprentissage comporte 23550 textes dont :

5767 textes ART, 4630 textes ECO, 3474 textes SPO, 1354 textes TEL,

Ce corpus est un peu déséquilibré, la dernière catégorie comporte deux fois moins d'individus que les autres.

Nous avons utilisé les mêmes classifieurs que pour le corpus 1.

	Jeu de test			**Jeu d'apprentissage**
Type de classifieur	Précision	Rappel	Fscore	Fscore Genre + thème
5 classifieurs	85,927%	85,589%	**85,758%**	86,16%
4 classifieurs	85.326%	85,032%	**85,179%**	87,28%
SVM-SMO	82,614%	82,916%	**82,765%**	84,53%

Les classifieurs sont dans l'ordre des soumissions.

- le classifieur par vote à 5 classifieurs est très performant à la fois sur le jeu d'apprentissage et le jeu de test.
- le classifieur par vote à 4 classifieurs est un peu moins performant que le précédent sur le jeu de test alors qu'il est plus performant sur le jeu d'apprentissage. Nous n'avons pas utilisé le vote sur 6 classifieurs pour des raisons de performances.
- Le classifieur SVM donne des résultats inférieurs aux systèmes fondés sur un vote. Nous constatons peu de différence entre le résultat sur le jeu d'apprentissage et le jeu de test.

Les résultats montrent que les méthodes par vote sont plus robustes que la méthode SVM seule. En effet les résultats sur les jeux de tests sont proches de ceux sur le jeu d'apprentissage pour les méthodes par vote. Les méthodes par vote sont vraiment utiles à deux niveaux : amélioration des performances et robustesse.

4 Méthodes additionnelles pour améliorer les résultats

Nous avons testé plusieurs approches pour améliorer les résultats. Elles sont de deux types :

- Lemmatisation préliminaire des textes et Utilisation des fonctions grammaticales des mots pour le calcul de l'index.
- Utilisation de bi-grammes en addition des unigrammes.

4.1 Lemmatisation préliminaire des textes

Ce traitement a été expérimenté uniquement sur le corpus 1. Nous avons lemmatisés tous les textes du corpus avant la vectorisation des textes. Dans le même temps nous avons éliminés les articles indéfinis et les ponctuations faibles.

Hélas tous les tests que nous avons effectués en utilisant les différents classifieurs présentés précédemment donnent des résultats fscore inférieur d'environ 2 à 5%. Nous n'avons donc pas présenté de résultats pour cette méthode.

4.2 Utilisation de bi-grammes en addition des unigrammes

Nous avons extrait les bi-grammes du corpus. Cette extraction s'est effectuée avec la même méthode qu'au paragraphe précédent.

Nous avons ensuite utilisé la méthode générale présentée au chapitre précédent sur le corpus 1. C'est-à-dire que nous avons considéré la liste des unigrammes et bi-grammes extraits comme l'index du corpus à partir duquel tous les textes ont été vectorisés. Puis la procédure classique a été implémentée : réduction d'index, classification, validation croisée.

La taille des index à la fois sur le corpus 1 et quelques essais sur le corpus 2, est bien plus grande : environ 1900000 unités linguistiques. L'algorithme de réduction d'index par calcul de la différence d'entropie devient très long. L'index réduit compte 4000 unités linguistiques supplémentaires.

Nos résultats sur le corpus 1 ont montré une amélioration importante d'environ 1,5% en fscore à 91.88% sur le jeu d'apprentissage en appliquant un classifieur avec genre et thème fusionnés par système de vote à 6 classifieurs. L'amélioration sur les jeux de tests devrait être du même montant compte tenu de la robustesse des classifieurs par système de vote.

5 Conclusion et perspectives

Nos résultats sont globalement au dessus des moyennes générales, ce qui est encourageant.

Nous avons passé en revue plusieurs méthodes de classification. Les méthodes par vote de classifieurs améliorent d'environ 2 à 3% les résultats sur des classifieurs simples.

Nous souhaitons utiliser les trigrammes pour tenter encore d'améliorer les résultats. Cependant la taille des index devient alors très importante et les temps de calculs deviennent très longs. Nous devons améliorer alors les performances des algorithmes de calcul pour obtenir des résultats dans des temps raisonnables.

Références

Cover, & Thomas. (1991). *Elements of Information Theory*: John Wiley.

Joachims, T. (1998). *Text Categorisation with Support Vector Machines : Learning with Many Relevant Features.* Paper presented at the ECML.

MacQueen., J. B. (1967). *Some Methods for classification and Analysis of Multivariate Observations.* . Paper presented at the 5th Berkeley Symposium on Mathematical Statistics and Probability.

Parks, J., & Sandberg, I. W. (1991). « Universal approximation using radial-basis function networks ». In *Neural Computation* (Vol. 3, pp. 246-257).

Plantié, M. (2006). *Extraction automatique de connaissances pour la décision multicritère.* Unpublished Thèse de Doctorat, Ecole Nationale Supérieure des Mines de Saint Etienne et de l'Université Jean Monnet de Saint Etienne, Nîmes.

Plantié, M., Roche, M., & Dray, G. (2008). *Un système de vote pour la classification de textes d'opinion.* Paper presented at the 8èmes journées francophones "Extraction et Gestion des Connaissances" pp 583-588, INRIA Sophia Antipolis.

Platt, J. (1998). Machines using Sequential Minimal Optimization. . In *Advances in Kernel Methods - Support Vector Learning*: B. Schoelkopf and C. Burges and A. Smola, editors.

Wang, Y., Hodges, J., & Tang, B. (2003). Classification of Web Documents using a Naive Bayes Method. *IEEE*, 560-564.

Weka Project, U. o. w. (2002-2005). Weka: University of waikato.

Yoav, F., & E., S. R. (1996). *Experiments with a new boosting algorithm.* Paper presented at the Thirteenth International Conference on Machine Learning, San Francisco USA.