Défi DEFT07 : Comparaison d'approches pour la classification de textes d'opinion

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

[1] Laboratoire LGI2P, Laboratoire de Génie informatique et d'ingénierie de la production,
Ecole des Mines d'Alès, Site EERIE –parc scientifique Georges Besse, 30035 – Nîmes,
(michel.plantie, gerard.dray)@ema.fr

[2] Laboratoire LIRMM, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier,
UMR 5506, 161 rue Ada, 34392 Montpellier Cedex 5 - France,
mathieu.roche@lirmm.fr

Résumé : Nous exposons dans cet article, les méthodes utilisées pour répondre au défi DEFT 2007. 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 proposés.

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

1 Introduction

Le défi consiste comme il est indiqué sur le site : http://deft07.limsi.fr/ à évaluer en différentes opinions des textes d'opinion différents corpus en français de style et de domaine différents.

Les corpus et leurs catégories d'évaluation sont :

- Corpus 1 : critiques films, livres, spectacles et bandes dessinées,
  - Trois catégories : bon, moyen, mauvais
- Corpus 2 : critiques de tests de jeux vidéo,
  - Trois catégories : bon, moyen, mauvais
- Corpus 3 : Commentaires de révision d'articles de conférences scientifiques,
  - Trois catégories : acceptation, acceptation sous condition, rejet
- Interventions des parlementaires et du gouvernement dans les débats sur les projets de lois votés à l'Assemblée nationale
  - Deux catégories : pour, contre

Afin de pouvoir trouver les méthodes de traitements toutes les équipes avaient accès à quatre 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 classifications à partir de textes spécifiques (ici, des textes d'opinions). Précisons que notre approche dans le cadre de DEFT'07 n'utilise aucun traitement spécifique propre aux corpus. En effet, le but du challenge est d'avoir des approches génériques de classifications adaptées à des textes d'opinion. 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. Le détail des approches utilisées n'est pas donné dans cet article qui a pour but d'analyser la performance et également les contre performances des différents traitements appliqués. Bien que nos résultats soient globalement satisfaisants (situés dans la moyenne des résultats des participants), les résultats négatifs que nous avons obtenus ont été volontairement présentés dans cet article. En effet, nous estimons que ceux-ci peuvent être particulièrement intéressants pour la communauté « fouille de texte ».

Après une pésentation 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 Méthode générale

Dans ce défi nous avons considéré que le problème posé relevait de la problématique de la classification. Chaque opinion possible 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 validation croisée

Cette méthode de traitement a été utilisée telle quelle et également nous avons ajoutés dans certains cas des traitements supplémentaires afin de tenter d'améliorer les résultats.

2.1 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 lemmatisé ou lemme.

Cependant certains types grammaticaux sont éliminés : les articles indéfinis et la ponctuation faible.

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

Corpus	Nombre d'unités linguistiques (lemmes)
Corpus 1	36214
Corpus 2	39364
Corpus 3	10157
Corpus 4	35841

Cette opération est effectuée avec l'outil d'analyse syntaxique « Synapse » (Synapse, 2001).

Cette liste de lemme 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 lemmes du corpus. Chaque dimension représente un lemme. Ainsi chaque coordonnée d'un vecteur représentera le nombre d'occurrence du lemme associé à cette dimension dans le texte.

2.2 prétraitement linguistique et représentation mathématique des textes du corpus

- Lemmatisation : Chaque texte subit le même prétraitement linguistique que précédemment c'est-à-dire une lemmatisation. Ainsi chaque texte est transformé en une suite de lemme.

- Filtrage grammatical : Dans une deuxième étape certains types grammaticaux sont éliminés. Dans un processus où il s'agit de différentier des appréciations positives et négatives nous avons choisi de conserver tous les lemmes exceptés : les articles indéfinis et la ponctuation faible. Nous pensons que ces deux éléments n'ont pas d'incidence sur la tonalité du texte. Et surtout tous les autres types grammaticaux sont succeptibles d'exprimer des nuances d'opinions ou des contributions à des opinions. Nous avons donc conservé les lemmes 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.

2.3 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 voyons les diminutions de la dimension de l'espace vectoriel associée à chaque corpus.

Corpus	**Nombre initial d'unités linguistiques**	**Nombre d'unités linguistiques Après réduction**
Corpus 1	36214	704
Corpus 2	39364	2363
Corpus 3	10157	156
Corpus 4	35841	3193

2.4 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 expaces 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 performance des classifieurs.

2.5 Choix de la méthode de classification

Une fois réduit l'espace vectoriel 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 tenté 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 cinq méthodes de classification. Notons que la plupart de ces méthodes est décrite de manière précise dans (Plantié, 2006).

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.
- La classification par la méthode des réseaux RBF (Radial Basis Function)
- La classification par arbre de décision de type C4.5
- La classification par la méthode probabiliste fondée sur la loi de Dirichlet lissée

2.5.1 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.

2.5.2 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 effectués des essais avec plusieurs de ces méthodes.

2.5.3 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).

2.5.4 Classifieur par la méthode des arbre de décision de type C4.5

Cette technique utilise l'approche bien connue de (Quinlan, 1993) et qui est également présentée dans (Plantié, 2006). Elle permet d'élaborer un arbre de décision sur l'ensemble des mots clés constituant l'espace vectoriel de représentation des textes. nous l'avons décrite dans (Plantié, 2006).

2.5.5 Classifieur par la méthode probabiliste de Bayes combiné à la loi de Dirichlet

Cette technique utilise le même principe que la méthode probabiliste de Bayes/multinomiale décrite précédemment mais remplace la loi de Bayes et la loi Multinomiale par une loi de Dirichlet lissée comme précisé dans (Nallapati Ramesh, 2006).

2.6 Évaluation des performances de la classification par validation croisée

La validation croisée est une technique d'évaluation permettant de valider une méthode de classification en particulier. Cette approche ne construit pas de modèle utilisable mais sert à estimer l'erreur réelle d'un modèle selon l'algorithme suivant (figure 1) :

Validation croisée (S;x) : *// S est un ensemble, x est un entier*
Découper S en x parties égales S1, … , Sx
Pour i de 1 à x
Construire un modèle M avec l'ensemble S - Si
Evaluer une mesure d'erreur ei de M avec Si
Fin Pour

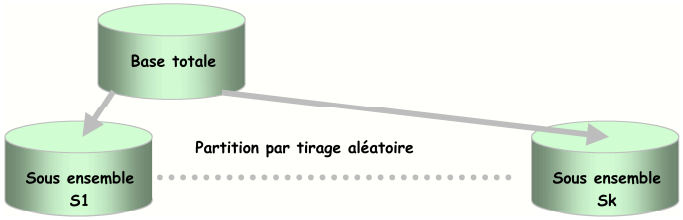

Figure 1 : Processus de validation croisée

Si la taille des *Si* est de un individu, on parle alors de validation par « leave one out ».

En général le nombre *x* de parties est fixé à 10.

Dans notre approche nous avons utilisée la méthode de validation croisée sur l'ensemble des vecteurs « 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.

2.7 Mesure de performances de la classification

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).

3 Résultats obtenus avec la méthode générale

Nous allons présenter ici les résultats obtenus tout d'abord en validation croisée sur le corpus d'apprentissage, puis les résultats sur les corpus de tests fournis dans le cadre du défi DEFT.

Nous allons présenter ces résultats par corpus.

Dans les tableaux présentés ci-dessous, il existe un différence notable entre ceux obtenus par la méthode de validation croisée et ceux obtenus sur les corpus de test. Cette 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 2074 textes dont :

309 textes classés : 0 (*mauvais*)

615 textes classés : 1 (*moyen*)

1150 textes classés : 2 (*bien*)

Ce corpus est déséquilibré, la dernière catégorie comporte deux fois plus 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.

Voici le tableau des résultats obtenus.

		Validation Croisée			**Jeu de test**
Type de classifieur		Précision	Rappel	Fscore	Fscore
RBF-Network	Classe 0	0.927	0.737	0.821	**0.4715**
	Classe 1	0.713	0.704	0.708	
	Classe 2	0.835	0.887	0.86	
Naive Bayes Multinomial	Classe 0	0.735	0.776	0.755	0.5902
	Classe 1	0.635	0.56	0.595	
	Classe 2	0.806	0.845	0.825	
SVM	Classe 0	0.729	0.708	0.718	0.6102
	Classe 1	0.602	0.575	0.588	
	Classe 2	0.796	0.821	0.808	
Dirichlet	Classe 0	0.4224	0.9545	0.5857	
	Classe 1	0.7034	0.4365	0.5387	
	Classe 2	0.8581	0.7431	0.7965	

Comme indiqué précédemment nous constatons une chute importante des résultats sur le corpus de test. Nous pouvons tirer quelques enseignements des résultats précédents :

- le classifieur RBF-Network est plus performant sur le jeu d'apprentissage mais son résultat chute très fortement sur le jeu de test, plus fortement que les autres classifieurs. Il est donc plus sensible à l'apparition de nouvelles données.
- Le classifieur de Bayes Multinomial est très sensible au déséquilibre de population des individus. Il est cependant plus robuste sur les données de test.
- Le classifieur SVM donne les meilleurs résultats.
- Le classifieur Dirichlet n'est pas meilleur que les deux précédents.

Remarque : Les résultats sur les jeux de test pour les classifieur de Bayes et SVM ne sont pas officiels, ils ont été effectués après l'échéance. Compte tenu du nombre très limité de soumissions possibles nous avons préféré soumettre d'autres résultats fondés sur des méthodes combinés.

Les résultats sur ce corpus sont moyens (62% de fscore au maximum). Nous avons tenté d'améliorer ce score par une méthode fondée sur les synonymes (voir ci-après).

Le résultat du classifieur RBF-Network a été publié dans le jeu 1 de nos soumissions.

3.2 Corpus 2

Le corpus d'apprentissage comporte 2537 textes dont :

497 textes classés : 0 (*mauvais*)

1166 textes classés : 1 (*moyen*)

874 textes classés : 2 (*bien*)

Ce corpus est légèrement déséquilibré, la catégorie 1(moyen) comporte deux fois plus d'individus que la première. Voici le tableau des résultats obtenus.

		Validation Croisée			**Jeu de test**
Type de classifieur		Précision	Rappel	Fscore	Fscore
SVM	Classe 0	0.825	0.774	0.799	**0.7829**
	Classe 1	0.799	0.842	0.82	
	Classe 2	0.866	0.834	0.849	
RBF-Network	Classe 0	0.912	0.789	0.846	**0.5475**
	Classe 1	0.782	0.927	0.849	
	Classe 2	0.906	0.751	0.821	
Naive Bayes Multinomial	Classe 0	0.792	0.819	0.805	0.7416
	Classe 1	0.812	0.815	0.814	
	Classe 2	0.862	0.841	0.851	
Dirichlet	Classe 0	0.5490	0.9671	0.7004	
	Classe 1	0.9319	0.4957	0.6472	
	Classe 2	0.7653	0.9185	0.8349	

La chute des résultats sur le corpus de test est moins importante pour la méthode SVM.

Nous pouvons tirer quelques enseignements des résultats précédents :

- le classifieur RBF-Network est équivalent à SVM sur le jeu d'apprentissage mais son résultat chute très fortement sur le jeu de test.
- Le classifieur de Bayes Multinomial donne de bons résultats malgré le déséquilibre de population des individus.
- Le classifieur SVM donne les meilleurs résultats.
- Le classifieur Dirichlet n'est pas meilleur que les deux précédents.

Remarque : Les résultats sur les jeux de test pour le classifieur de Bayes ne sont pas officiels, ils ont été effectués après l'échéance, nous avons préféré soumettre d'autre résultats fondés sur des méthodes combinés.

Le résultat du classifieur SVM a été publié dans le jeu 1 de nos soumissions.

Le résultat du classifieur RBF-Network a été publié dans le jeu 3 de nos soumissions.

Notons par ailleurs, que les résultats que nous avons obtenus pour la soumission 1 sur ce corpus sont de très bonne qualité (0.78) comparativement au fscore moyen du défi sur ce corpus (0.6604 +/- 0.086). Une analyse plus approfondie serait nécessaire pour expliquer un tel résultat très positif sur le corpus des critiques de tests de jeux vidéo qui est d'une taille conséquente.

3.3 Corpus 3

Le corpus d'apprentissage comporte 881 textes dont :
227 textes classés : 0 (*rejet*)
278 textes classés : 1 (*acceptation sous conditions*)
376 textes classés : 2 (*acceptation*)
Ce corpus est assez équilibré, la catégorie 2(*acceptation*) comporte 50% d'individus en plus que la 1.
Voici le tableau des résultats obtenus.

		Validation Croisée			**Jeu de test**
Type de classifieur		Précision	Rappel	Fscore	Fscore
SVM	Classe 0	0.733	0.604	0.662	**0.4782**
	Classe 1	0.645	0.57	0.605	
	Classe 2	0.673	0.803	0.732	
RBF-Network	Classe 0	0.503	0.758	0.605	
	Classe 1	0.91	0.44	0.594	
	Classe 2	0.645	0.693	0.668	
Naive Bayes Multinomial	Classe 0	0.6452	0.819	0.805	0.4914
	Classe 1	0.812	0.815	0.814	
	Classe 2	0.862	0.841	0.851	
Dirichlet	Classe 0	0.5985	0.696	0.6436	
	Classe 1	0.6234	0.5035	0.5571	
	Classe 2	0.6768	0.7093	0.6927	

La chute des résultats sur le corpus de test est conséquente pour tous les classifieurs.
Nous pouvons tirer quelques enseignements des résultats précédents :

- le classifieur RBF-Network est presque équivalent à SVM sur le jeu d'apprentissage mais son résultat chute très fortement sur le jeu de test.
- Le classifieur de Bayes Multinomial donne les meilleurs résultats.
- Le classifieur Dirichlet n'est pas meilleur que les deux précédents.

Remarque : Les résultats sur les jeux de test pour le classifieur de Bayes ne sont pas officiels, ils ont été effectués après l'échéance, nous avons préféré soumettre d'autre résultats fondés sur des méthodes combinés.
Le résultat du classifieur SVM a été publié dans le jeu 1 de nos soumissions.

3.4 Corpus 4

Le corpus d'apprentissage comporte 17299 textes dont : 10400 textes classés : 0 (*contre*), 6899 textes classés : 1 (*pour*).
Ce corpus est un peu déséquilibré, la catégorie 1 comporte 30% d'individus en moins que la première.
Voici le tableau des résultats obtenus.

		Validation Croisée			**Jeu de test**
classifieur		Précision	Rappel	Fscore	Fscore
RBF-Network	Classe 0	0.822	0.61	0.701	**0.6179**
	Classe 1	0.577	0801	0.671	
C4.5 Quinlan	Classe 0	0.503	0.758	0.605	**0.5940**
	Classe 1	0.503	0.758	0.605	
SVM	Classe 0	0.806	0.874	0.839	0.6907
	Classe 1	0.782	0.684	0.73	
Naive Bayes Multinomial	Classe 0	0.8	0.813	0.806	0.6855
	Classe 1	0.711	0.694	0.702	
Dirichlet	Classe 0	0.855	0.7375	0.7919	
	Classe 1	0.6727	0.8118	0.7357	

La chute des résultats sur le corpus de test est conséquente pour les classifieurs SVM et Naïve Bayes Multinomial. Par contre le classifieur RBF chute peu sur le corpus de test.

Nous pouvons tirer quelques enseignements des résultats précédents :

- Le classifieur SVM donne les meilleurs résultats.
- Le classifieur de Bayes Multinomial est très proche du meilleur.
- le classifieur RBF-Network chute un peu sur le jeu de test.
- Le classifieur Dirichlet est le meilleur sur une des classes.
- Le classifieur utilisant les arbres de décisions (C4.5) a des performances inférieures à tous les autres (nous avons constaté ce phénomène sur l'ensemble des corpus).

Remarque : Les résultats sur les jeux de test pour les classifieurs de Bayes et SVM ne sont pas officiels, ils ont été effectués après l'échéance, nous avons préféré soumettre d'autre résultats fondés sur des méthodes combinés.

Le résultat du classifieur RBF-Network a été publié dans le jeu 1 de nos soumissions.

Le résultat du classifieur C4.5 Quinlan a été publié dans le jeu 3 de nos soumissions.

3.5 Explication de la différence entre résultats en validation croisée et sur le corpus de test

La réduction de la taille de l'index est fondée sur l'appartenance de textes à des classes. Si nous réduisons l'index sur l'ensemble du corpus d'apprentissage, alors cela suppose que le vocabulaire utilisé dans les jeux d'apprentissage est exhaustif ou presque et que les jeux de tests n'utiliseront pas de mots différents de ceux appartenant à l'ensemble d'apprentissage.

Cette hypothèse est généralement fausse.

La procédure de validation croisée que nous avons utilisée utilise des vecteurs qui sont calculé sur l'index réduit de tout l'ensemble d'apprentissage.

Cette procédure est incorrecte car la réduction d'index doit être effectuée uniquement sur la partie de l'ensemble d'apprentissage qui est utilisée pour calculer le modèle de chaque sous ensemble utilisé dans chaque itération de la validation croisée.

Ainsi la procédure de validation croisée se transforme comme suit :

// S est un ensemble, x est un entier
Découper S en x parties égales S1, … , Sx
Pour i de 1 à x
Réduire l'index sur l'ensemble S- Si
Calculer les vecteurs avec l'index réduit obtenu
Construire un modèle M avec l'ensemble S - Si sur les vecteurs réduits
Evaluer une mesure d'erreur ei de M avec Si
Fin Pour

Dans notre cas x vaut 10.

En utilisant cette procédure nous avons constaté que les résultats en fscore sur l'ensemble d'apprentissage sont quasiment identiques à ceux obtenus en test et cela sur les quatre corpus testés.

4 Méthodes additionnelles pour améliorer les résultats

Nous avons tentés plusieurs approches pour améliorer les résultats. Elles sont de cinq types :

- Filtrage préliminaire des textes
- Ajout de synonymes
- Procédure de vote de classifieurs
- Utilisation des fonctions grammaticales des mots pour le calcul de l'index.
- Utilisation de bi-grammes en lieu et place de lemmes.

4.1 Filtrage préliminaire des textes

Ce traitement a été tenté uniquement sur le corpus 1. Nous avons effectué une procédure préliminaire pour élaguer les phrases considérer comme inutiles dans le corpus 1.

En lisant les textes du premier corpus nous avons constaté que une part non négligeable de ceux-ci contenaient à la fin du texte une partie commençant par l'expression : « Notre avis : ».

Nous avons considéré que la partie de texte qui suivait était uniquement constitué de phrases de jugement de valeur.

Ainsi nous avons utilisés ces phrases pour extraire dans tous les textes de l'ensemble d'apprentissage les phrases exprimant un jugement de valeurs.

Nous avons déjà traité cette problématique et montré notre approche de traitement dans (Plantié, 2006). Elle consiste à considérer un ensemble d'apprentissage contenant deux catégories :

- les phrases exprimant un jugement de valeur
- les phrases n'exprimant pas un jugement de valeur.

Le problème revient alors d'éliminer dans un texte les phrases n'exprimant pas un jugement de valeur, que l'on pourra considérer comme des phrases inutiles ou non pertinentes. Cette problématique reprend le thème du défi DEFT 2005 pour séparer des phrases de « Chirac » de celles de « Mitterand ».

Nous constituons donc un nouvel ensemble d'apprentissage constitué uniquement des textes et contenant l'expression « Notre avis : ». On constitue alors deux ensembles :

- classe 1 : les phrases de ces textes hors jugement de valeurs(n'appartenant pas à la rubrique « Notre avis : »)
- classe 2 : les phrases « jugement de valeur » (appartenant à la rubrique débutant par « Notre avis : »)

Un fois constitué cet ensemble d'apprentissage, nous calculons alors un nouveau classifieur. Ce classifieur interviendra sur les autres textes du corpus 1 (hors ceux contenant « Notre avis : ») pour éliminer les phrases étant classées « hors jugement de valeurs » ou classe 1. Le meilleur classifieur dans cette tâche a été le « Naïve Bayes Multinomial ». Nous avons obtenu pour ce classifieur un fscore de 84% par procédure de validation croisée.

Avec cette procédure nous obtenons donc un **nouveau** corpus 1, dans lequel chaque texte est une réduction du texte initial. Chaque texte ne contient que la partie considérée comme jugement de valeurs.

Puis nous appliquons sur ces textes la méthode générale présentée au chapitre précédent.

Hélas tous les tests que nous avons effectués en utilisant les différents classifieurs présentés précédemment donnent des résultats fscore inférieur d'environ 5 à 10%. Nous n'avons donc pas présenté de résultats pour cette méthode.

4.2 Ajout de synonymes

Comme nous l'avons expliqué dans la section précédente les textes du corpus de test comportent certains mots de vocabulaire qui ne sont pas obligatoirement présent dans le corpus d'apprentissage. Afin de prendre en compte ces nouveaux mots nous proposons d'utiliser les synonymes.

Nous avons déjà dans nos travaux rencontré cette problématique et montré notre approche de traitement dans (Plantié, 2006).

Voici l'algorithme :

```
Programme synonyme (texte)
    Établir la liste des lemmes (texte)
    Pour chaque lemme du texte à analyser :
           Rechercher dans l'index du corpus le lemme
           Si le lemme est présent : ne rien faire
           Sinon(le lemme est absent) :
                  Rechercher la liste des synonymes de (lemme)
                  Pour chaque synonyme de la liste
                     Si le synonyme est présent dans la liste des lemmes du corpus
                                Associer le lemme à l'indice de ce lemme du corpus
                                Fin du pour
                     Sinon rien
                  Fin Pour
           Fin si
    Fin Pour
```

Ainsi chaque mot inconnu est associé à un mot de l'index. Les vecteurs de chaque texte sont alors calculés selon la procédure ci-dessus. Nous pouvons alors utiliser la méthode du chapitre précédent.

Hélas tous les tests que nous avons effectués en utilisant les différents classifieurs présentés précédemment donnent des résultats fscore inférieur d'environ 10 à 15%. Nous n'avons donc pas présenté de résultats pour cette méthode.

4.3 Procédure de vote

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

Nous avons tentés deux approches :

- le vote à la majorité simple
- le vote tenant compte du fscore de chaque classifieur

4.3.1 Vote à la majorité simple

Nous avons appliqué cette procédure pour les quatre corpus.

Le principe est le suivant :

Nous prenons les résultats de trois classifieurs. Pour chaque texte évalué nous retenons la réponse qui emporte la majorité de 2 au moins sur 3.

Dans le tableau qui suit nous montrons les classifieurs retenus et les résultats obtenus :

Corpus	Classifieur 1	Classifieur 2	Classifieur 3	Résultat Fscore Sur jeu de test
Corpus 1	RBF-Network 8 clusters par classe	Naïve Bayes Multinomial	RBF-Network 6 clusters par classe	**0.4231**
Corpus 2	SVM	RBF-Network 4 cluster par classe	Naïve Bayes Multinomial	**0.7325**
Corpus 3	SVM	RBF-Network 4 clusters par classe	Naïve Bayes Multinomial	**0.4421**
Corpus 4	Naïve Bayes Multinomial	RBF-Network 8 cluster par classe	C4.5 Quinlan	**0.6706**

Ces résultats sont inférieurs à nos meilleurs résultats sur chacun des corpus. Les procédures de vote à la majorité sont donc peu convaincantes. Ces résultats ont été publiés dans le jeu 2 de nos soumissions.

4.3.2 Vote tenant compte du fscore de chaque classifieur

Nous avons tenté d'utiliser 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 : RBF-Network, et Naïve Bayes Multinomial.

Algorithme :

```
Affectation d'un classifieur à une classe, en prenant le classifieur donnant max de la
précision sur le corpus d'apprentissage.
Programme voteprecision (texte)
Si le classifieur en question affecte le texte en cours de traitement à cette classe
    alors nous prenons ce résultat.
Sinon nous prenons le classifieur suivant et l'on recommence la procédure.
Si aucun classifieur n'affecte le document en question à sa classe de prédilection,
    alors nous prenons le meilleur des deuxième choix.
```

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.

Algorithme :

```
Affectation d'un classifieur à une classe, en prenant le classifieur donnant max du rappel
sur le corpus d'apprentissage.
Programme voterappel (texte)
Si le classifieur en question affecte le texte en cours de traitement à cette classe
    alors nous prenons ce résultat.
Sinon nous prenons le classifieur suivant et l'on recommence la procédure.
Si aucun classifieur n'affecte le document en question à sa classe de prédilection,
    alors nous prenons le meilleur des deuxième choix.
Fin Si
```

Voici les résultats obtenus sur le jeu de test :

Corpus	Type de vote	Résultat Fscore Sur jeu de test
Corpus 1	Classe affectée au Classifieur ayant le Maximum de précision	**0.4715**
Corpus 3	Classe affectée au Classifieur ayant le Maximum de rappel	**0.4416**

Ces résultats sont inférieurs à nos meilleurs résultats sur chacun des corpus. Les procédures de vote à en tenant compte de la précision et du rappel sont donc peu convaincantes. Ces résultats ont été publiés dans le jeu 3 de nos soumissions.

4.4 Utilisation des fonctions grammaticales des mots pour le calcul de l'index

Dans le cadre de DEFT'07, nous avons appliqué un prétraitement supplémentaire. Ainsi, avant d'effectuer la classification des textes, nous avons cherché à améliorer les traitements « linguistiques » des textes. Dans ce but, les termes (groupes de mots respectant des patrons syntaxiques spécifiques) ont été extraits et exploités.

Avant d'extraire la terminologie, une première étape consiste à apposer des étiquettes grammaticales à chacun des mots du corpus. Une telle tâche a été effectuée avec l'étiqueteur de Brill (E. Brill, 1994) qui utilise des règles lexicales et contextuelles apprises à partir d'un corpus annoté.

Le tableau ci-dessous montre un fragment du corpus de relectures d'articles issu de DEFT'07 qui a été étiqueté avec (E. Brill, 1994).

Texte d'origine : **L'auteur devrait essayer de cibler...**	
Texte étiqueté : **L'/DTN :sg auteur/SBC :sg devrait/VCJ :sg essayer/VNCFF de/PREP cibler/VNCFF...**	
DTN : Déterminant de groupe nominal **SBC** : Substantif, nom commun **VCJ** : Verbe, conjugué	**VNCFF** : Verbe, non conjugué, infinitif **PREP** : Préposition **sg** : singulier

A partir des quatre corpus de DEFT'07 étiquetés, des patrons syntaxiques spécifiques peuvent être utilisés afin d'extraire les termes nominaux propres à chacun des corpus (termes de type "Nom Nom", "Adjectif Nom", "Nom Adjectif", "Nom Préposition Nom"). L'extraction de la terminologie a été menée avec le système(Roche, Heitz, Matte-Tailliez, & Kodratoff, 2004). Précisons que de nombreux travaux s'appuient sur l'utilisation de patrons syntaxiques pour extraire la terminologie (Bourigault & Fabre, 2000; Daille, 1994; Jacquemin, 1999).

Outre les termes nominaux qui ont été extraits et utilisés lors de l'étape de classification, nous nous sommes également intéressés à l'extraction des termes adjectivaux et adverbiaux. En particulier, les termes de type "Adverbe Adjectif" peuvent se révéler particulièrement pertinents pour certains corpus. A titre d'exemple, les termes *encore préliminaire*, *encore insuffisant*, *très significatif*, *difficilement compréhensible* obtenus à partir du corpus de relectures d'articles peuvent être assez discriminants pour classifier certains textes. Notons cependant que le corpus de relectures contient de nombreuses fautes d'orthographe (fautes d'accents, caractères manquants, etc.). Une telle situation peut expliquer les résultats décevants en utilisant uniquement la terminologie pour les tâches de classification. Une correction de ces fautes aurait pu significativement améliorer les tâches d'extraction de la terminologie et donc de classification. De plus, le nombre de termes extraits peut se révéler assez faible pour certains textes. Ceci met alors en défaut les méthodes statistiques qui ont été mises en oeuvre dans le cadre du défi.

Nous avons ensuite utilisé la méthode générale présentée dans la section précédente sur le corpus 1. Ainsi, nous avons considéré la liste des termes 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. Le nombre de termes extraits peut se révéler assez faible pour certains textes ce qui réduit significativement la taille de l'index. Ceci met alors en défaut les méthodes statistiques qui ont été mises en oeuvre dans le cadre du défi. Egalement les termes sélectionnés après réduction d'index ne sont pas suffisamment significatifs pour représenter la diversité des textes. Ceci peut expliquer les résultats fortement dégradés avec cette méthode.

4.5 Utilisation de bi-grammes en lieu et place de lemmes

Nous avons tenté d'extraire les bi-grammes du corpus mais uniquement dans un premier temps les bi-grammes : Adjectif-Adverbe ou inversement. 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 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.

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

Nous aurions souhaité poursuivre cette expérience par l'extraction de tous les bi-grammes du corpus et ensuite appliquer la procédure de réduction d'index. Hélas le temps nous a manqué.

5 Conclusion et perspectives

Ce défi fut passionnant. Nous avons cependant manqué de temps et surtout de machines très performantes en terme d'espace mémoire notamment. Certains algorithmes en effet ont pris plusieurs heures pour donner des résultats.

Nous avons passé en revue plusieurs méthodes de classification. Mais nous devons approfondir nos essais sur les différentes méthodes de classification. Pour améliorer nos résultats nous devons également effectuer des prétraitements plus poussés, car les classifieurs montrent leurs limites sur la plupart des corpus. En particulier nous fondons de nombreux espoirs sur :

- une application plus approfondie de la méthode de Dirichlet,
- la combinaison de classifieur plus précise et plus adaptée que les procédures de vote simple présentées ici,
- L'utilisation plus généralisée des bi-grammes,
- La combinaison de méthodes fondées sur les lemmes et sur les bi-grammes.

Références

Bourigault, D., & Fabre, C. (2000). Approche linguistique pour l'analyse syntaxique de corpus. *Cahiers de Grammaires, 25*, 131-151.

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

Daille, B. (1994). *Approche mixte pour l'extraction automatique de terminologie : statistiques*

lexicales et filtres linguistiques. Unpublished Ph. D. thesis, Université Paris 7.

E. Brill, I., Vol. (1994). Some advances in transformation-based part of speech tagging. *AAAI, 1*, 722-727.

Jacquemin, C. (1999). *Syntagmatic and paradigmatic representations of term variation.* Paper presented at the 7th Annual Meeting of the Association for Computational Linguistics (ACL'99).

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.

Nallapati Ramesh, M. T., Robertson Stephen. (2006). *The Smoothed-Dirichlet distribution : a new building block for generetive topic models*.

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.

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.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. (Morgan Kaufmann ed.). San Mateo (CA US) Morgan Kaufmann.

Roche, M., Heitz, T., Matte-Tailliez, O., & Kodratoff, Y. (2004). *EXIT : Un système itératif pour l'extraction de la terminologie du domaine à partir de corpus spécialisés.* Paper presented at the JADT'04.

Synapse. (2001). Synapse Analyser: Synapse. "Synapse Analyser." http://synapse-fr.com, .

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