DEFT2008

Actes du quatrième DÉfi Fouille de Textes

Proceedings of the Fourth DEFT Workshop

13 juin 2008

Avignon, France

DEFT2008

Actes du quatrième DÉfi Fouille de Textes

13 juin 2008
Avignon, France

Comités

Comité de programme

Patrick Paroubek (LIMSI–CNRS), *président*

Catherine Berrut (LIG)

Fabrice Clérot (France Telecom)

Guillaume Cleuziou (LIFO)

Matthieu Constant (Univ. Marne-la-Vallée)

Halima Dahmani (CEA–LIST)

Béatrice Daille (LINA)

Marc El-Bèze (LIA)

Patrick Gallinari (LIP6)

Eric Gaussier (Xerox Research)

Thierry Hamon (LIPN)

Fidélia Ibekwe-SanJuan (ELICO)

Pascal Poncelet (LGI2P)

Christophe Roche (LISTIC)

Mathieu Roche (LIRMM)

Bernard Rothenburger (IRIT)

Pascale Sébillot (IRISA)

Yannick Toussaint (LORIA)

François Yvon (LIMSI–CNRS)

Comité d'organisation

Martine Hurault-Plantet (LIMSI–CNRS), *co-responsable*

Cyril Grouin (LIMSI–CNRS), *co-responsable*

Jean-Baptiste Berthelin (LIMSI–CNRS)

Sarra El Ayari (LIMSI–CNRS)

Sylvain Loiseau (LIMSI–CNRS)

Table des matières

Présentation et résultats

Actes du quatrième défi fouille de texte, DEFT2008, Avignon, France, 13 juin 2008.
Proceedings of the Fourth DEFT Workshop, DEFT2008, Avignon, France, 13th June 2008.
Pages 3-12

TALN 2008, Avignon, 9–13 juin 2008

Présentation de DEFT'08 (DÉfi Fouille de Textes)

Les membres du Comité d'Organisation de DEFT'08 :
Cyril Grouin[1] Jean-Baptiste Berthelin[1] Sarra El Ayari[1]
Martine Hurault-Plantet[1] Sylvain Loiseau[1]
(1) LIMSI-CNRS – BP133 – 91403 Orsay Cedex
{cyril.grouin, jean-baptiste.berthelin, sarra.elayari,
martine.hurault-plantet, sylvain.loiseau}@limsi.fr

Résumé. Dans le cadre de la campagne d'évaluation annuelle DEFT *(défi fouille de textes)*, la quatrième édition a pour objet l'identification de catégories textuelles en genre et en thème. Nous avons utilisé des articles provenant de deux sources, *Le Monde* et Wikipédia, chaque article ayant été rattaché à l'une des neuf catégories extraites de ces corpus. Cet article présente l'objectif de la tâche, les corpus utilisés ainsi que les prétraitements effectués sur ces corpus. Nous reviendrons également sur les tests manuels que nous avons réalisés pour mesurer la faisabilité de la tâche. Enfin, nous détaillerons les mesures utilisées pour évaluer les résultats des participants.

Abstract. Within the annual DEFT evaluation campaign (where DEFT is for "Défi Fouille de Texte"), the fourth edition involves the identification of textual categories in genre and thema. We used articles from two sources : *Le Monde* (a French daily newspaper), and Wikipedia. Each of these articles is connected with one of the nine categories we extracted from both corpora. This paper describes the aim of the task, the corpora that are used, and the preprocessing that has been applied to them. We also examine the way in which we performed manual tests in order to measure the feasibility of the task. Finally, we give the detail of the measurements that served to evaluate the competitors' results.

Mots-clés : Campagne d'évaluation, fouille de textes, catégorisation, genre, thème, indices de confiance.

Keywords: Evaluation campaign, text mining, categorization, genre, confidence indicators.

1 Introduction

L'édition 2008 de DEFT[1] s'inscrit dans le cadre de la conférence JEP/TALN organisée en Avignon du 9 au 13 juin 2008. Le thème retenu cette année concerne l'identification du genre d'un document parmi deux possibilités (journalistique et encyclopédique) et la catégorisation de chaque document parmi neuf catégories thématiques.

Les participants devaient effectuer deux tâches distinctes :
– Tâche n° 1 : identification du genre et du thème (parmi un choix de quatre thèmes) ;
– Tâche n° 2 : identification du thème uniquement (parmi un choix de cinq thèmes, distincts de ceux utilisés dans la tâche 1) mais portant néanmoins sur les deux genres de documents rassemblés pour ce défi.

2 Présentation des corpus

Les corpus ont été constitués à partir de deux sources distinctes : *Le Monde* et Wikipédia en français, constituant ainsi les deux genres de textes à identifier. Par genre, nous entendons des contextes d'écriture différents, en l'occurrence un contexte journalistique (*Le Monde*) et un contexte d'encyclopédie collaborative sur l'Internet (Wikipédia).

Pour chacun de ces corpus, nous avons relevé neuf catégories communes (rubrique dans laquelle a paru un article du *Monde* ou catégorie sous laquelle a été classé un article de Wikipédia) dont nous donnons ci-dessous des représentants de chaque corpus. Par catégorie, nous entendons un ensemble d'articles traitant de la même thématique.

– Art : articles consacrés à l'art et à la culture (danse, peinture, sculpture, théâtre) ;
– Économie : articles consacrés à l'économie et aux entreprises ;
– France : articles de politique nationale française ;
– International : articles de politique internationale ou nationale (sauf politique française) ;
– Littérature : articles relatifs aux livres (critiques, parutions) et à la littérature ;
– Sciences : articles consacrés aux sciences ;
– Société : articles consacrés aux problèmes de société ne relevant pas du domaine politique ;
– Sports : articles traitant du sport (rencontres, résultats, personnalités) ;
– Télévision : articles consacrés à la radio et à la télévision (programmes, fonctionnement).

3 Préparation des données

3.1 Corpus « Le Monde »

Le corpus du journal *Le Monde* nous a été fourni par la société ELDA[2]. Chaque article est enregistré dans un fichier distinct au format XML qui reproduit la structure organisationnelle du journal en distinguant les éléments constitutifs de l'article (titraille, chapô, texte de l'article)

[1] `http://deft08.limsi.fr/`
[2] ELDA : Evaluations and Language resources Distribution Agency, `www.elda.org`

des méta-données associées (date de publication, secteur de rédaction, éléments d'indexation). Le secteur de rédaction nous fournit la catégorie thématique de l'article.

Afin de disposer d'un nombre équivalent d'articles dans chaque catégorie entre les corpus du *Monde* et de Wikipédia, nous n'avons utilisé qu'une sous-partie du corpus d'origine : tous les articles de l'année 2004 pour les catégories *Art, Économie, France, International, Société* et *Télévision*, les articles des années 2004 et 2005 pour les catégories *Littérature* et *Sports*, et l'ensemble des articles de la période 2003 à 2006 pour la catégorie *Sciences*.

Dans un premier temps, nous avons listé tous les articles publiés sous l'une des neuf catégories prédéfinies (voir section 2) en nous fondant sur le secteur de rédaction renseigné dans chaque fichier. Nous avons ensuite converti au format texte brut les articles de cette liste faisant plus de 300 caractères en effaçant les mentions explicites au journal (indications de copyright et références à un article paru dans une édition antérieure du quotidien). Seuls les articles détaillant les résultats du Loto sportif ont été éliminés de ce nouvel ensemble.

3.2 Corpus « Wikipédia »

La constitution du corpus d'articles de Wikipédia a été réalisée à partir de la version complète de la base, datée d'octobre 2007.

Extraction des articles Une différence importante entre le système de catégorisation en rubriques du *Monde* et le système de catégorisation de Wikipédia est que le premier résulte en une partition des articles alors que dans le second, un article peut appartenir à plusieurs catégories à la fois. Nous avons donc utilisé plusieurs stratégies pour extraire de Wikipédia un ensemble d'articles qui soit une partition en catégories.

Dans un premier temps, nous avons représenté l'ensemble des catégories de Wikipédia sous la forme d'un graphe qui organise les catégories en sous-catégories et super-catégories (voir graphe 1). Afin de collecter les articles, nous sommes partis des catégories les plus représentatives de chacune des neuf thématiques définies pour le défi (par exemple, *Économie* et *Société* sur le graphe). À ce niveau, nous constatons qu'il n'existe que peu d'articles directement rattachés à ces catégories racines.

À partir de ces catégories racines, nous avons parcouru le graphe et collecté tous les articles situés sous ces catégories racines ou sous l'une de leurs sous-catégories, en n'allant pas plus loin que trois niveaux de sous-catégories. Nous avons en effet constaté qu'en s'éloignant de la catégorie racine, d'une part, la spécification thématique décroît (notamment par le biais d'une sur-spécialisation de l'article), et d'autre part, le nombre de connexions vers des catégories hétérogènes augmente (et renforce le risque qu'un article dépende de deux catégories racines).

Pour nous assurer que chaque sous-corpus soit bien contrasté sur le plan thématique, nous avons supprimé toutes les sous-catégories qui relient deux catégories racines ainsi que les articles rattachés à ces catégories (par exemple, les sous-catégories *Organisation sociale, Industrie* et *Entreprise* sur le graphe 1, qui sont à l'intersection des catégories racines *Économie* et *Société*).

Seuls les articles de plus de 300 caractères et d'au-moins une année d'existence ont été conservés, dans la perspective d'éliminer les ébauches d'articles.

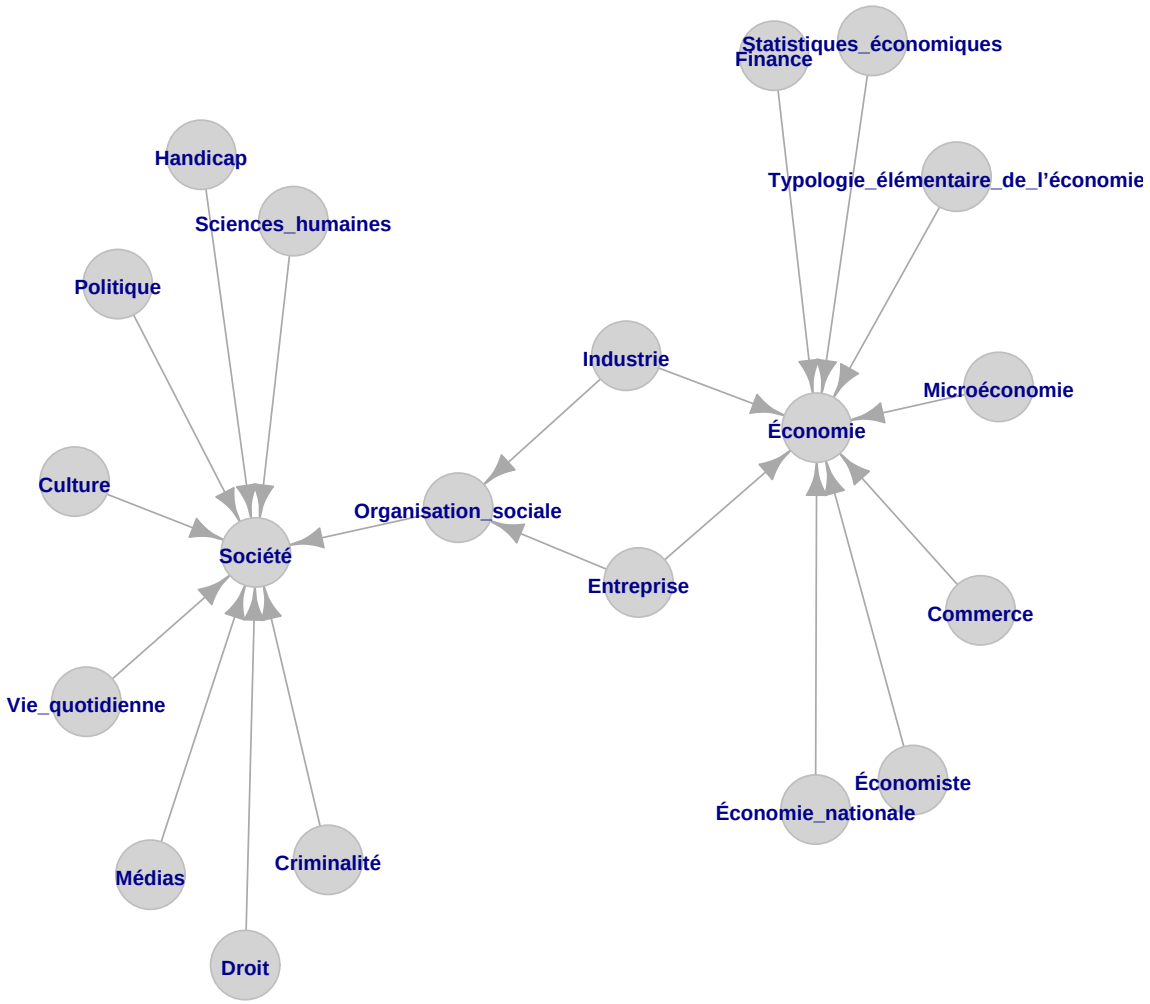

FIG. 1 – Graphe du premier sous-niveau de catégories Wikipédia.

Conversion des articles Nous avons ensuite procédé à une étape de conversion du format des articles, d'abord du format Wikipédia au format TEI[3] par le biais du convertisseur wiki2tei[4], puis du format TEI au format texte brut. La première étape de conversion nous permet de conserver les informations et la syntaxe Wiki tout en bénéficiant d'un format XML plus facilement manipulable. La seconde étape de conversion du format XML au format texte brut a ensuite été réalisée au moyen de requêtes XSLT.

Pour la version finale des articles, nous avons éliminé tous les éléments qui ne relèvent pas directement du contenu textuel des articles (tableaux, listes, bibliographies, tables des matières) ainsi que les titres préformatés utilisés par Wikipédia (« liens externes », « voir aussi », etc).

[3]TEI : Text Encoding Initiative (`www.tei-c.org`), consortium international qui définit et maintient depuis 1994 un standard adapté à la représentation de textes.

[4]Wiki2tei, par Bernard Desgraupes et Sylvain Loiseau : `http://wiki2tei.sourceforge.net/`

3.3 Constitution des corpus

Pour chaque ensemble de catégories relatif à une tâche (voir section 2), nous avons produit les corpus d'apprentissage et de test sur la base d'une répartition respectivement fixée à 60% et 40% du total des articles. Nous avons par ailleurs procédé à une répartition aléatoire des articles en termes de genre et de catégorie d'appartenance dans les différents corpus de test et d'apprentissage.

TÂCHE 1	Appr.	Test
Art	5 767	3 844
Économie	4 630	3 085
Sports	3 474	2 315
Télévision	1 352	1 352

TÂCHE 2	Appr.	Test
France	3 326	2 216
International	5 305	3 536
Littérature	4 576	3 049
Sciences	6 565	4 375
Société	3 778	2 517

FIG. 2 – Proportion d'articles par catégorie pour chacune des tâches.

3.4 Évaluation manuelle de la tâche

Afin de valider le choix de la tâche pour cette édition du défi, nous avons procédé à une évaluation manuelle d'un échantillon du corpus auprès de 4 juges humains. Le corpus à évaluer se composait de 30 articles du *Monde* et 30 articles de Wikipédia. Seuls le titre et le corps de l'article ont été conservés, les tableaux ayant été éliminés de ces fichiers. Les marques distinctives d'appartenance à l'un ou l'autre de ces corpus ont été enlevées (références au *Monde* et bandeaux d'informations utilisés par Wikipédia).

Le test s'est déroulé de la manière suivante : chaque article étant enregistré dans un fichier distinct, les évaluateurs ont eu pour consigne d'identifier le genre et la catégorie sous laquelle chaque article a paru. Tous les articles ont été regroupés en un seul ensemble ; autrement dit, les évaluateurs ont dû choisir parmi toutes les catégories et non parmi deux sous-ensemble de quelques catégories chacun.

Ce test a été réalisé sur une première sélection de 8 catégories :

Le Monde	Wikipédia
Carnet	*Personnalité*
Économie	*Économie*
France	*Politique de la France*
International	*Politique nationale*, moins la catégorie *Politique de la France*
Sciences	*Sciences*
Société	*Société*, moins les sous-catégories *Politique, Personnalité, Sport, Médias*
Sports	*Sport*
Télévision	*Télévision*

FIG. 3 – Correspondance entre catégories du Monde et de Wikipédia sur les 8 catégories utilisées lors du test.

3.4.1 Résultats

Les résultats des juges humains en termes de rappel et précision se sont révélés excellents sur l'identification du genre (F-scores compris entre 0,94 et 1,00) et plutôt bons sur l'identification des catégories (F-scores compris entre 0,66 et 0,82).

	1	**2**	**3**	**4**
Genres	1,00	0,98	0,97	0,94
Catégories	0,79	0,77	0,82	0,66

FIG. 4 – F-scores obtenus par les juges humains sur l'identification du genre et des catégories.

Nous avons par ailleurs confronté ensemble les résultats des juges humains grâce au coefficient κ (Carletta, 1996), coefficient qui permet de mettre en évidence les taux d'accord entre juges.

Juge	**Réf.**	**1**	**2**	**3**	**4**
Réf.		1,00	0,97	0,93	0,87
1	1,00		0,97	0,93	0,87
2	0,97	0,97		0,90	0,83
3	0,93	0,93	0,90		0,87
4	0,87	0,87	0,83	0,87	

Juge	**Réf.**	**1**	**2**	**3**	**4**
Réf.		0,56	0,52	0,60	0,39
1	0,56		0,69	0,75	0,55
2	0,52	0,69		0,71	0,61
3	0,60	0,75	0,71		0,52
4	0,39	0,55	0,61	0,52	

FIG. 5 – Coefficient κ entre juges humains et la référence.
Identification du genre (tableau de gauche) et des catégories (tableau de droite).

Ces coefficients démontrent l'excellent accord des juges entre eux ainsi qu'avec la référence pour l'identification du genre, et font état d'accords modérés à bons pour l'identification des catégories. Ces résultats nous ont confortés dans le choix du thème de ce défi.

3.4.2 Consistance des catégories

Nous avons par ailleurs mesuré la consistance de chaque catégorie en mettant en évidence le rappel et la précision obtenue par l'ensemble des évaluateurs sur chacune des catégories. Cette mesure a été produite sur la base d'une seconde évaluation réalisée par des juges humains, et portant sur un ensemble plus large de catégories (ajout des catégories *Art* et *Littérature*).

La classification des catégories par précisions décroissantes est la suivante : *Sports* (1,00%), *International* (0,80%), *France* (0,76%), *Littérature* (0,76%), *Art* (0,74%), *Télévision* (0,71%), *Économie* (0,58%), *Sciences* (0,33%), *Société* (0,26%). Il en ressort qu'aucun des documents classés dans la catégorie *Sport* n'a été mal classé, alors qu'à l'inverse, les catégories *Sciences* et *Société* sont celles qui ont posé le plus de problèmes.

La classification obtenue sur les rappels décroissants varie légèrement : *International* (0,87%), *Économie* (0,80%), *Sports* (0,75%), *France* (0,70%), *Art* (0,62%), *Littérature* (0,49%), *Télévision* (0,46%), *Société* (0,42%), *Sciences* (0,33%). Ainsi, les articles de la catégorie *International* sont ceux qui ont été le mieux identifiés. Ce classement confirme par ailleurs la difficulté ressentie par les juges humains vis-à-vis des catégories *Société* et *Sciences*.

3.4.3 Ajustement du défi

Les résultats obtenus à l'issue des différents tests réalisés auprès de juges humains nous ont permis, d'une part, de sélectionner les catégories thématiques à conserver pour le défi, et d'autre part, de définir les deux ensembles de catégories utilisés pour chaque sous-tâche.

Alors que nous avions retenu la catégorie *Carnet* (biographies de personnes célèbres) lors du premier test humain, nous avons décidé de l'abandonner dans la suite du défi pour deux raisons. En premier lieu, nous nous sommes rendus compte qu'il s'agissait plutôt d'un genre, le genre « biographie », plutôt qu'une catégorie thématique. D'autre part, nous avons éprouvé quelques difficultés à affecter à une seule catégorie les articles qui pouvaient potentiellement relever de deux catégories, par exemple dans le cas de la biographie d'un sportif qui relèverait des catégories *Carnet* et *Sports*.

En second lieu, nous avons procédé à une répartition des catégories pour chaque sous-tâche sur la base d'un équilibre entre catégories jugées faciles et difficiles par les évaluateurs humains :
– *Art, Économie, Sports, Télévision* pour la sous-tâche combinant reconnaissance de genre et de catégorie ;
– *France, International, Littérature, Sciences, Société* pour la sous-tâche de reconnaissance seule des catégories. Pour ce second ensemble, nous avons fait le choix de rassembler trois catégories assez proches sur le plan thématique (*France, International* et *Société*).

4 Déroulement du défi

Six équipes ont participé à l'édition 2008 du défi, dont une constituée de jeunes chercheurs :
– GREYC (Caen) : F. Rioult, Th. Charnois, Y. Mathet et A. Doucet ;
– LGI2P (Nîmes) et LIRMM (Montpellier) : M. Plantié, G. Dray et M. Roche ;
– LIA (Avignon) : J. M. Torres-Moreno, M. El Bèze, F. Béchet, E. Sanjuan, P. Peinl et P. Bellot ;
– LIA (Avignon) : E. Charton, R. Acuna-Agost, N. Camelin et R. Kessler, *jeunes chercheurs* ;
– LIFO (Orléans) et INaLCO (Paris) : G. Cleuziou et C. Poudat ;
– LIP6 (Paris) : D. Buffoni et A.-P. Trinh.

4.1 Organisation du défi

4.1.1 Corpus d'apprentissage

Les corpus d'apprentissage ont été diffusés à partir du 16 janvier 2008. Comme pour la précédente édition du défi, nous avons autorisé les participants à utiliser des bases de connaissances mais nous avons exclu la possibilité d'utiliser d'autres corpus d'apprentissage que ceux fournis.

4.1.2 Corpus de test

La phase de tests a été élaborée selon les mêmes modalités que l'année précédente : une fenêtre de trois jours comprise dans la période du 17 au 28 mars 2008, les participants ayant le choix du premier jour de cette phase de tests. L'ensemble des candidats s'est porté en faveur de la seconde semaine pour soumettre les résultats.

4.2 Évaluation des résultats

4.2.1 Définition du F-score utilisé pour le classement final

Chaque fichier de résultat a été évalué en calculant le F-score de chacun des corpus avec $\beta = 1$.

$$\mathrm{F}_{\text{score}}(\beta) = \frac{(\beta^2 + 1) \times \text{Précision} \times \text{Rappel}}{\beta^2 \times \text{Précision} + \text{Rappel}}$$

Lorsque le F-score est utilisé pour évaluer la performance sur chacune des n classes d'une classification, les moyennes globales de la précision et du rappel sur l'ensemble des classes peuvent être évaluées de 2 manières (voir (Nakache & Métais, 2005)) :

– La micro-moyenne qui fait d'abord la somme des éléments du calcul – vrais positifs, faux positifs et négatifs – sur l'ensemble des n classes, pour calculer la précision et le rappel globaux ;
– La macro-moyenne qui calcule d'abord la précision et le rappel sur chaque classe i, puis en fait la moyenne sur les n classes.

Dans la micro-moyenne chaque classe compte proportionnellement au nombre d'éléments qu'elle comporte : une classe importante comptera davantage qu'une petite classe. Dans la macro-moyenne, chaque classe compte à égalité.

Micro-moyenne

$$\text{Précision} = \frac{\sum_{i=1}^{n} TPi}{\sum_{i=1}^{n} (TPi + FPi)} \quad \text{Rappel} = \frac{\sum_{i=1}^{n} TPi}{\sum_{i=1}^{n} (TPi + FNi)}$$

Macro-moyenne

$$\text{Précision} = \frac{\sum_{i=1}^{n} \left(\frac{TPi}{(TPi + FPi)} \right)}{n} \quad \text{Rappel} = \frac{\sum_{i=1}^{n} \left(\frac{TPi}{(TPi + FNi)} \right)}{n}$$

Avec :
– TPi = nombre de documents correctement attribués à la classe i ;
– FPi = nombre de documents faussement attribués à la classe i ;
– FNi = nombre de documents appartenant à la classe i et non retrouvés par le système ;
– n = nombre de classes.

Les catégories étant inégalement réparties dans les corpus, nous avons choisi de calculer le F-score global avec la macro-moyenne pour que les résultats sur chaque classe comptent de la même manière quelle que soit la taille de la classe.

Par ailleurs, dans la mesure où plusieurs classes peuvent être attribuées au même document avec des indices de confiance, nous avons établi les règles suivantes d'attribution d'une classe à un document pour le calcul du F-score strict.

Un document est attribué à la classe i si :

– Seule la classe i a été attribuée à ce document, sans indice de confiance spécifié ;
– La classe i a été attribuée à ce document avec un meilleur indice de confiance que les autres classes. S'il existe plusieurs classes possédant l'indice de confiance le plus élevé, alors nous retiendrons celle qui sera la première d'entre elles dans la balise `<EVALUATION>`.

Dans le calcul de ce F-score, l'indice de confiance n'est pris en compte que pour sélectionner la catégorie attribuée à un document.

4.2.2 Définition du F-score pondéré par l'indice de confiance

Un système de classification automatique peut attribuer à un document une distribution de probabilité sur les différentes classes au lieu de lui attribuer une seule classe. L'indice de confiance est cette probabilité pour un document d'appartenir à une catégorie donnée.

Le F-score pondéré par l'indice de confiance sera utilisé à titre indicatif pour des comparaisons complémentaires entre les méthodes mises en place par les équipes.

Dans le F-score pondéré, la précision et le rappel pour chaque classe sont pondérés par l'indice de confiance. Ce qui donne :

$$\text{Précision}_i = \frac{\sum_{\text{attribué correct. } i=1}^{\text{Nombre attribué correct. } i} \text{indice de confiance}_{\text{attribué correct.}_i}}{\sum_{\text{attribué } i=1}^{\text{Nombre attribué } i} \text{indice de confiance}_{\text{attribué}_i}}$$

$$\text{Rappel}_i = \frac{\sum_{\text{attribué correct. } i=1}^{\text{Nombre attribué correct. } i} \text{indice de confiance}_{\text{attribué correct.}_i}}{\text{nombre de documents appartenant à la classe } i}$$

Avec :

– Nombre attribué correct.$_i$: nombre de documents attribué correct.$_i$ appartenant effectivement à la classe i et auxquels le système a attribué un indice de confiance non nul pour cette classe ;
– Nombre attribué$_i$: nombre de documents attribués$_i$ auxquels le système a attribué un indice de confiance non nul pour la classe i.

Le F-score pondéré est ensuite calculé à l'aide des formules du F-score classique (voir section 4.2.1).

4.2.3 Algorithme utilisé pour désigner le vainqueur de DEFT'08

Les équipes ont été classées en fonction des rangs obtenus sur l'ensemble des sous-tâches et en considérant chaque soumission comme atomique. Le rang d'une soumission est donc égal à la somme des rangs associés au F-score classique de cette soumission sur chaque sous-tâche. Ainsi, c'est le classement pour chaque sous-tâche qui compte, et non les valeurs cumulées du F-score. L'algorithme utilisé est présenté ci-après :

début
  Pour chaque sous-tâche **faire**
    / Score : liste qui associe à chaque couple (équipe, soumission) son F-score */*
    Score(soumission, équipe) = F-score(sous-tâche, soumission, équipe)
    / Tri de la liste Score dans l'ordre décroissant du F-score */*
    Score trié(soumission, équipe) = tri(Score(soumission, équipe))
    / Tableau des rangs obtenus par chaque soumission de chaque équipe, pour la sous-tâche considérée */*
    Rang[sous-tâche][soumission][équipe] = rang(Score trié(soumission, équipe))
  fin Pour
  Pour chaque équipe ayant soumis **faire**
    / Somme, sur toutes les sous-tâches, des rangs obtenus pour chaque soumission */*
    Rang global[soumission][équipe] = $\sum_{\text{sous-tâche}}$ rangs[sous-tâche][soumission][équipe]
    / Choix de la meilleure soumission (rang le plus faible) */*
    Rang[équipe] = $\min_{\text{soumission}}$(rangs[soumission][équipe])
  fin Pour
  / Choix du vainqueur : équipe dont le rang est le plus faible */*
  ÉquipeV telle que : Rang[ÉquipeV] = $\min_{\text{équipe}}$(Rang[équipe])
fin

FIG. 6 – Algorithme pour désigner le vainqueur

5 Conclusion

Dans cet article, nous avons présenté les différents corpus utilisés et les méthodes que nous avons mobilisées pour constituer les corpus de test et d'apprentissage. Nous avons également détaillé les différents tests qui ont été réalisés entre juges humains, ces tests nous ayant permis, d'une part d'affiner la tâche de cette campagne, et d'autre part de mesurer la faisabilité de la tâche. Enfin, nous avons rappelé les différentes étapes du déroulement de ce défi en insistant notamment sur les mesures utilisées pour évaluer et classer les résultats des participants.

Remerciements

Nous remercions la société ELDA pour la mise à disposition grâcieuse du corpus du Monde et nos partenaires : le CNRS, l'AFIA, l'ATALA et Wikipédia.

Références

CARLETTA J. (1996). Assessing agreement on classification tasks : the kappa statistics. *Computational Linguistics*, **2**(22), 249–254.

NAKACHE D. & MÉTAIS E. (2005). Evaluation : nouvelle approche avec juges. In *INFORSID*, p. 555–570, Grenoble.

Résultats de l'édition 2008 du DÉfi Fouille de Textes

Martine Hurault-Plantet[1] Jean-Baptiste Berthelin[1] Sarra El Ayari[1]
Cyril Grouin[1] Sylvain Loiseau[1] Patrick Paroubek[1]
(1) LIMSI-CNRS – BP133 – 91403 Orsay Cedex
{martine.hurault-plantet, jean-baptiste.berthelin, sarra.elayari,
cyril.grouin, sylvain.loiseau, patrick.paroubek}@limsi.fr

Résumé. Cet article présente les résultats obtenus par les participants de l'édition 2008 du défi fouille de textes (DEFT). Ces résultats se révèlent particulièrement élevés et homogènes entre chaque participant, avec une réussite accrue sur l'identification du genre par opposition à l'identification des thèmes. Dans cet article, nous revenons sur l'ensemble des résultats en opposant les F-scores stricts aux F-scores de confiance ; nous mettons également en avant l'incidence du score de confiance sur les résultats. Enfin, nous présentons les méthodes utilisées par les participants.

Abstract. This article presents the results obtained by the participants to the 2008 edition of the DEFT text-mining challenge. These results appear to be both high and homogeneous between any two participants, and successes are greater in genre identification as aopposed to topic identification. In this article, we survey the totality of the results, contrasting strict F-scores with confidence F-scores ; we also emphasize the incidence of confidence F-scores upon results. Finally, we present the methods used by the participants.

Mots-clés : F-score, rappel, précision, front de Pareto, tf*idf, représentation de textes, classification de textes.

Keywords: F-score, recall, precision, Pareto front, tf*idf, text representation, text classification.

Introduction

Comme lors de la précédente édition du défi, chaque candidat avait la possibilité de soumettre jusqu'à trois résultats. Chaque soumission a été considérée comme étant un ensemble indissociable portant sur les deux tâches. Chaque soumission comportait donc 3 résultats de catégorisation : la catégorie en genre et la catégorie thématique des documents du corpus de la tâche 1 et la catégorisation thématique des documents du corpus de la tâche 2.

Pour toutes les soumissions, nous avons calculé le F-score strict (avec $\beta = 1$) pour chaque résultat de catégorisation puis, sur la base de ces calculs, nous avons défini la meilleure soumission de chaque équipe. Nous avons ensuite procédé au classement final des équipes en ne prenant en compte que la meilleure soumission de chacun des participants.

1 F-scores stricts

1.1 Résultats des participants

Cette nouvelle édition du défi a révélé l'excellence des résultats obtenus par l'ensemble des participants – hormis quelques rares accidents – et ce, quelle que soit la sous-tâche considérée (tableau n° 1).

Équipe par ordre d'inscription	Soumission	T1 genre	T1 cat	T2 cat	Confiance
J. M. Torres-Moreno (LIA)	1	0.958	0.859	0.859	oui
J. M. Torres-Moreno (LIA)	2	0.981	0.883	0.872	oui
J. M. Torres-Moreno (LIA)	3	0.980	0.854	0.880	oui
M. Plantié (LGI2P/LIRMM)	1	0.971	0.853	0.858	non
M. Plantié (LGI2P/LIRMM)	2	0.970	0.852	0.852	non
M. Plantié (LGI2P/LIRMM)	3	0.955	0.823	0.828	non
E. Charton (LIA)	1	0.980	0.875	0.879	non
E. Charton (LIA)	2	0.959	0.809	0.662	non
E. Charton (LIA)	3	0.980	0.844	0.853	non
D. Buffoni (LIP6)	1	0.951	0.804	0.874	oui
D. Buffoni (LIP6)	2	0.973	0.879	0.874	oui
D. Buffoni (LIP6)	3	0.976	0.894	0.876	oui
G. Cleuziou (LIFO/INaLCO)	1	0.937	0.790	0.821	non
F. Rioult (GREYC)	1	0.964	0.849	0.838	oui
F. Rioult (GREYC)	2	0.856	0.672	0.328	non
F. Rioult (GREYC)	3	0.964	0.672	0.815	non

FIG. 1 – F-scores stricts ($\beta = 1$) pour toutes les soumissions sur chaque tâche avec indication d'utilisation de l'indice de confiance dans les résultats soumis. La meilleure soumission de chaque équipe apparaît sur une ligne grisée.

Les résultats sur l'identification du genre, certes limitée à deux choix possibles, se sont révélés excellents. L'identification des catégories a également produit de très bons résultats, comme l'attestent les F-scores stricts obtenus par les participants :

1. Identification du genre (tâche n° 1) : les F-scores stricts des participants sont compris entre 0,856 et 0,981 avec trois soumissions où le F-score strict s'établit à 0,980 ;
2. Identification de la catégorie (tâche n° 1) : les F-scores sont compris entre 0,672 et 0,894 ;
3. Identification de la catégorie (tâche n° 2) : les F-scores sont compris entre 0,328 et 0,880.

Pour chaque tâche, nous constatons l'extrême homogénéité des résultats entre les différents participants. En dehors de quelques rares soumissions moins bonnes, les F-scores stricts se tiennent dans des intervalles assez restreints, comme en témoignent les nuages de points représentés sur les graphiques du Front de Pareto (voir graphiques n° 5, 6 et 7).

En comparaison avec l'édition 2007 de DEFT (Paroubek *et al.*, 2007), il apparaît que les résultats des participants sont meilleurs cette année, et surpassent même les évaluations manuelles effectuées par les organisateurs.

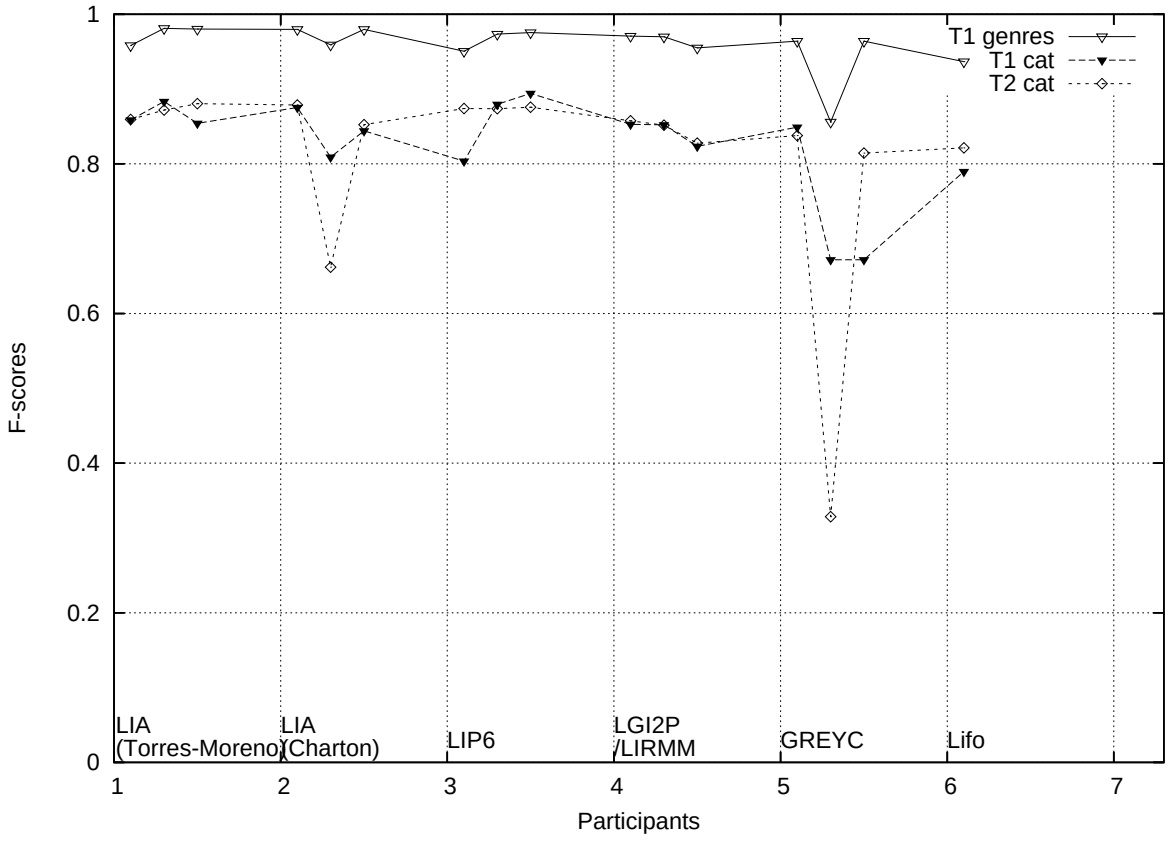

FIG. 2 – F-score strict ($\beta = 1$) pour l'ensemble des soumissions de chacun des candidats.

1.2 De meilleurs résultats que les juges humains...

Lors de la préparation de la tâche, nous avons procédé à des tests d'évaluation entre juges humains. Ces tests se sont révélés particulièrement bons et nous ont confortés dans le choix de cette tâche. À la réception des résultats des participants, force est de constater la supériorité des machines au regard des résultats obtenus par ces dernières sur les évaluateurs humains ! Cette constatation se révèle davantage sur l'identification des catégories.

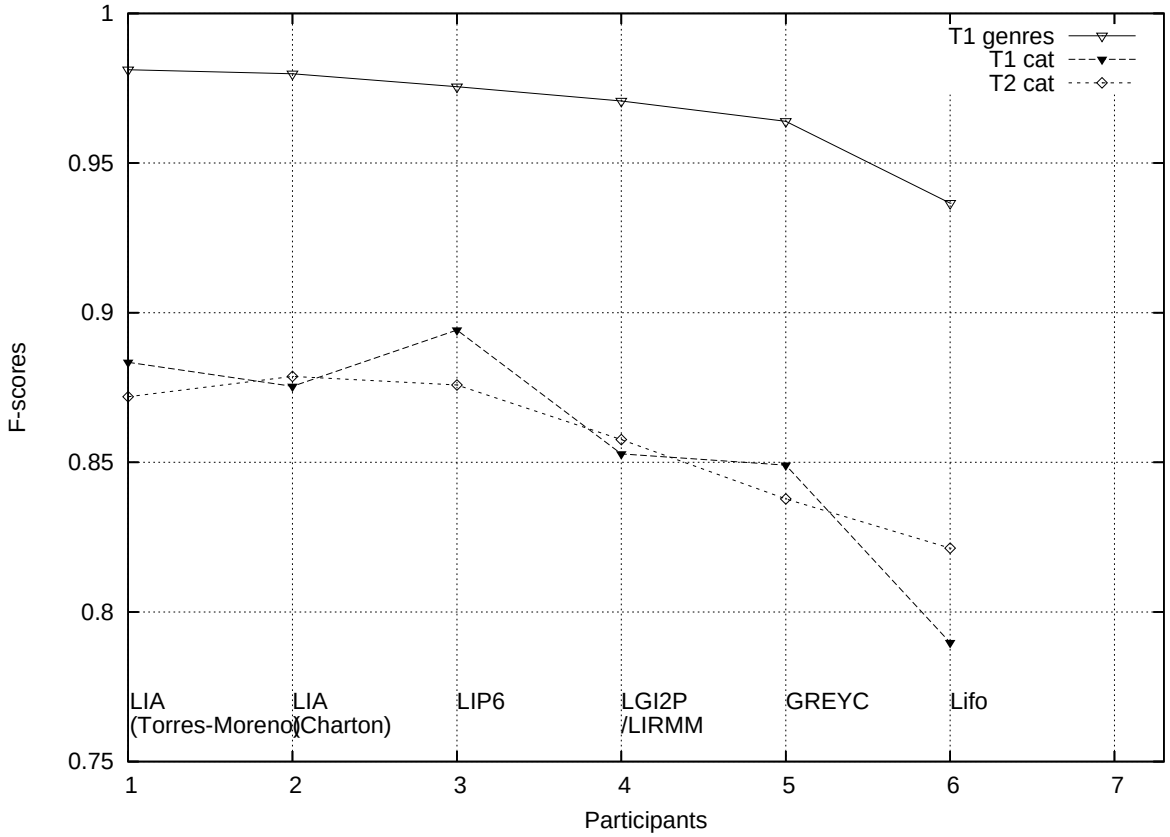

FIG. 3 – F-score strict ($\beta = 1$) pour les meilleures soumissions de chacun des candidats.

Au niveau de l'identification du genre, les F-scores stricts évoluent entre 0,94 et 1,00 pour les juges humains et entre 0,95 et 0,98 pour les meilleures soumissions des participants au défi (voir tableau 1). Concernant l'identification des catégories, les F-scores stricts s'échelonnent entre 0,66 et 0,82 pour les évaluateurs humains contre 0,84 à 0,89 pour les meilleures soumissions (fusion des résultats obtenus pour l'identification des catégories des tâches 1 et 2).

Par cette comparaison – au-delà du nombre réduit de choix possibles dans chacune des sous-tâches –, l'intérêt d'utiliser l'outil informatique dans le domaine de la classification thématique et de l'identification de contextes d'écritures différents apparaît de manière assez évidente.

2 F-score de confiance

2.1 Résultats

L'utilisation de l'indice de confiance dans les résultats soumis par les participants pour les catégories thématiques modifie les valeurs du F-score de manière différente selon les équipes. On observe ainsi une gradation de la baisse de la valeur du F-score pondéré par rapport au F-score strict, avec une répartition assez nette des soumissions entre celles dont le F-score n'a pas ou peu baissé et celles pour lesquelles le F-score a fortement baissé :

– pas de baisse ou faible baisse : 3 soumissions sont concernées ;

– baisse moyenne : une soumission concernée (première soumission de J. M. Torres-Moreno) ;
– forte baisse : 3 soumissions concernées.

Équipe par ordre d'inscription	Soumission	T1 cat	T2 cat
J. M. Torres-Moreno (LIA)	1	0.639	0.633
J. M. Torres-Moreno (LIA)	2	0.315	0.244
J. M. Torres-Moreno (LIA)	3	0.398	0.263
D. Buffoni (LIP6)	1	0.389	0.393
D. Buffoni (LIP6)	2	0.878	0.873
D. Buffoni (LIP6)	3	0.857	0.817
F. Rioult (GREYC)	1	0.725	0.717

FIG. 4 – F-scores pondérés ($\beta = 1$) pour toutes les soumissions ayant utilisé le score de confiance.

2.2 L'incidence de l'indice de confiance sur les résultats

Les participants ont eu la possibilité d'associer un indice de confiance aux choix de la catégorie thématique ; l'identification du genre reposant sur deux choix (*Le Monde* ou Wikipédia), l'utilisation d'un indice de confiance pour cette sous-tâche n'était pas autorisé. Précisons que cet indice de confiance était proposé de manière optionnelle.

Sur les six participants du défi, trois ont utilisé l'indice de confiance pour pondérer leurs résultats. Sur ces trois participants, deux l'ont appliquée pour chaque soumission (J. M. Torres-Moreno au LIA et D. Buffoni au LIP6), l'autre équipe (F. Rioult au GREYC) ayant fait le choix de produire une soumission avec indices de confiance et deux soumissions sans indices de confiance.

L'utilisation de l'indice de confiance par les participants nous conduit a dresser trois constatations sur l'incidence de cet indice sur les résultats :

Incidence sur le classement des soumissions L'équipe ayant proposé des soumissions avec et sans utilisation de l'indice de confiance a obtenu de meilleurs résultats sur la soumission avec indice de confiance (en l'occurrence la première soumission).

Incidence sur le classement global des équipes Nous notons par ailleurs que deux des trois équipes arrivées dans les premières places de cette campagne (J. M. Torres-Moreno au LIA et D. Buffoni au LIP6) ont toutes deux utilisé les indices de confiance dans les résultats qui leur ont permis de se classer à ces niveaux.

Incidence sur le front de Pareto Enfin, nous remarquons que toutes les soumissions figurant sur le front de Pareto et la majorité des soumissions se trouvant à proximité immédiate de ce front (cf. graphiques n° 5, 6 et 7) sont également celles qui ont utilisé les indices de confiance dans les résultats.

Chaque participant ayant obtenu des résultats assez proches entre eux pour chaque sous-tâche considérée, il apparaît que le recours aux indices de confiance dans les résultats a permis de produire une différence sensible dans les classements finaux. La pondération des résultats document par document permet, *in fine*, de remonter l'ensemble des résultats d'une soumission donnée, par opposition aux soumissions qui n'utilisent pas ces indices de confiance et pour lesquelles les résultats présentés renvoient, pour chaque document, soit à une réussite (indice de confiance égal à 1), soit à un échec (indice de confiance égal à 0), sans que ces résultats ne soient pondérés par l'indice de confiance. Nous précisons que cette constatation ne porte que sur l'utilisation des indices de confiance dans les résultats soumis et que l'utilisation des outils et méthodes par chaque participant ne saurait être écartée dans la comparaison des résultats.

3 Front de Pareto

Définition Le front de Pareto est défini par l'ensemble des approches qui sont telles qu'aucune approche ne présente de meilleurs résultats pour tous les critères étudiés (rappel et précision dans le cas présent). Les approches qui ne sont pas sur le front de Pareto sont dites « dominées »[1].

Représentation graphique Le rappel est présenté sur l'axe des abscisses, la précision sur l'axe des ordonnées.

Le front de Pareto est symbolisé sur ces schémas par la ligne qui relie les meilleurs résultats entre eux. Les points sur le front de Pareto représentent les résultats qui, du point de vue du rappel, ou de la précision ou des deux à la fois, sont les meilleurs.

Les numéros aux côtés des points permettent d'identifier les équipes, un point représentant une soumission pour le corpus considéré (notez que le numéro de la soumission n'apparaît pas sur ces schémas) :

Numéro	Équipe
2	LIA : J. M. Torres-Moreno
3	LGI2P/LIRMM : M. Plantié
4	LIA : E. Charton, *équipe jeunes chercheurs*
6	LIP6 : D. Buffoni
8	LIFO/INaLCO : G. Cleuziou
10	GREYC : F. Rioult

L'ensemble des résultats étant présenté dans la tableau 1 d'une part, et les résultats étant assez groupés d'autre part, nous avons focalisé chaque graphique sur les nuages de points situés à proximité du front de Pareto. En conséquence, l'échelle utilisée diffère pour chaque graphique.

[1] `http://www.lri.fr/~aze/enseignements/bibs/2007-2008/docs/apprentissage-supervise.pdf`

3.1 Homogénéité des résultats

3.1.1 Identification du genre

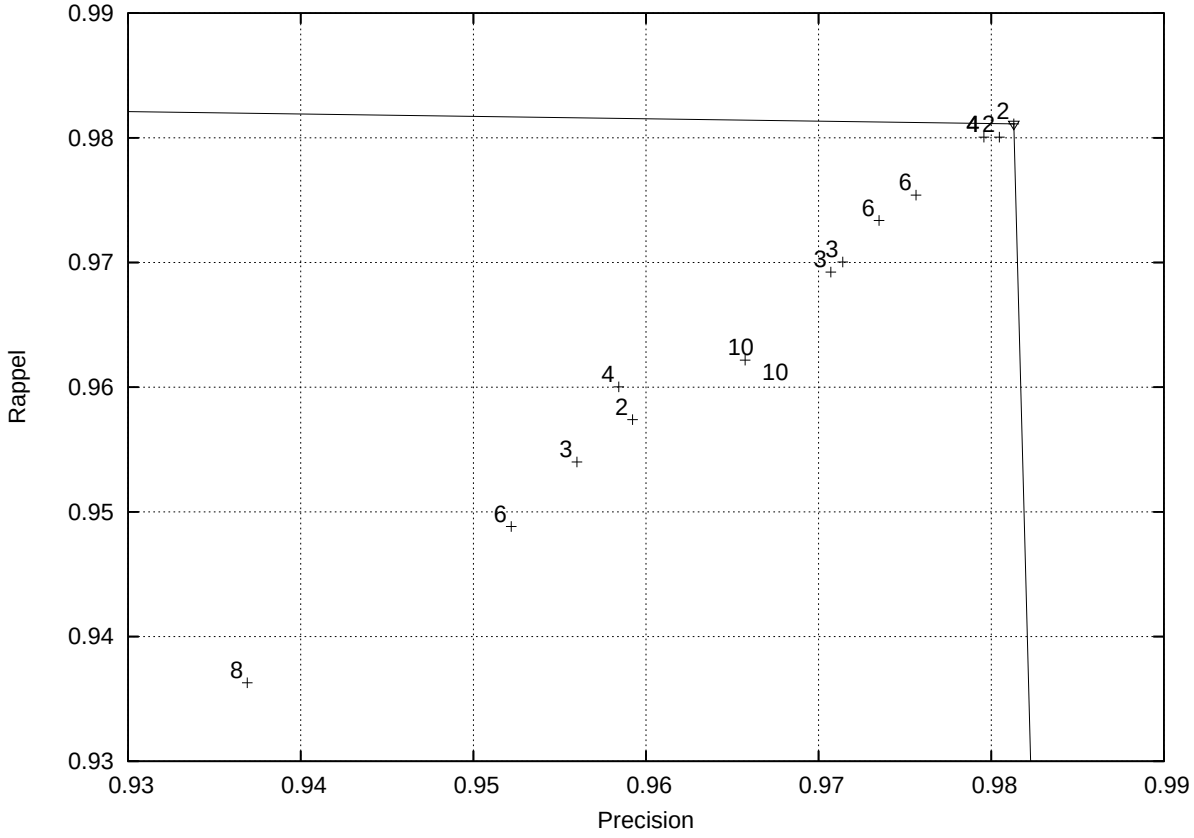

FIG. 5 – Front de Pareto pour la tâche 1, identification du genre.

La figure 5 montre un nuage de points homogène, les résultats des participants pour la tâche 1, identification du genre, sont très bons (pas un seul n'est inférieur à 0,856) au niveau du rappel et de la précision. En tête, les trois équipes ex-aequo, le LIA, le LIA jeunes chercheurs et le LIP6. Nous pouvons observer leur proximité avec le front de Pareto.

Nous relevons par ailleurs que l'ensemble des points sans exception aucune se situe sur une diagonale où rappel et précision sont en parfait équilibre, ce qui demeure assez exceptionnel et innatendu. Associant cette particularité aux très bons résultats obtenus par les participants, nous en déduisons que la tâche d'identification de genre avec seulement deux choix possibles s'est finalement révélée très facile. Les styles de l'encyclopédie Wikipédia et du journal *Le Monde* semblent bien se différencier.

3.1.2 Identification des catégories

Les points représentés sur la figure 6 sont plus disparates pour la tâche d'identification des catégories, mais avec également de bons résultats, la différence entre les deux figures pouvant également s'expliquer par le nombre de critères : 4 catégories à reconnaître contre deux genres distincts. Ici aussi, les équipes en tête sont les mêmes que pour l'identification du genre.

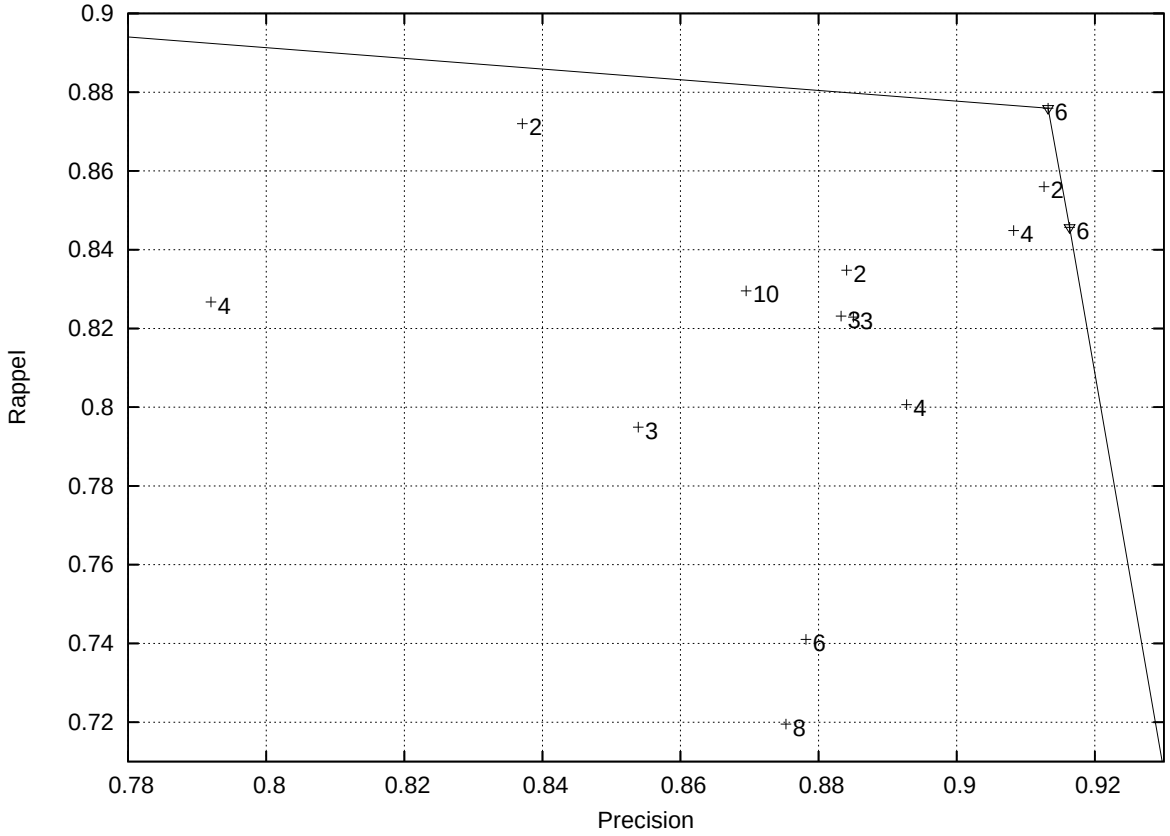

FIG. 6 – Front de Pareto pour la tâche 1, identification des catégories.

Au vu des résultats et contrairement à ce que nous avions imaginé lors de la préparation de ce défi, l'identification des catégories de la tâche 2 n'a pas posé plus de difficultés que celle de la tâche 1, alors que la tâche 2 comprend cinq catégories et que trois d'entre elles sont proches thématiquement (*France, International, Société*). Il apparaît même que la moitié des équipes a mieux réussi l'identification des catégories de la tâche n° 2 que celle de la tâche n° 1. Les résultats portant sur l'identification des catégories restent cependant très proches d'une tâche à l'autre comme le confirment les graphiques 2 et 3.

Il importe cependant de préciser que les soumissions les moins bonnes sont celles où l'identification des catégories de la tâche 2 a retourné de mauvais résultats (par exemple dans le cas des deuxièmes soumissions des équipes réunies autour de F. Rioult et d'E. Charton). Le plus grand nombre de catégories à identifier et la proximité sémantique de trois catégories sur cinq semble donc avoir occasionné quelques difficultés pour ces soumissions, la différence étant moins preignante pour l'identifcation des catégories de la tâche 1 sur ces mêmes soumissions.

4 Les méthodes utilisées par les participants

La confrontation de méthodes s'est avérée une fois de plus très productive, à la fois parce qu'elle montre des accords de performance sur des méthodes qui deviennent classiques, et qu'en même temps elle fait émerger des possibilités d'amélioration par des méthodes de conception plus originale.

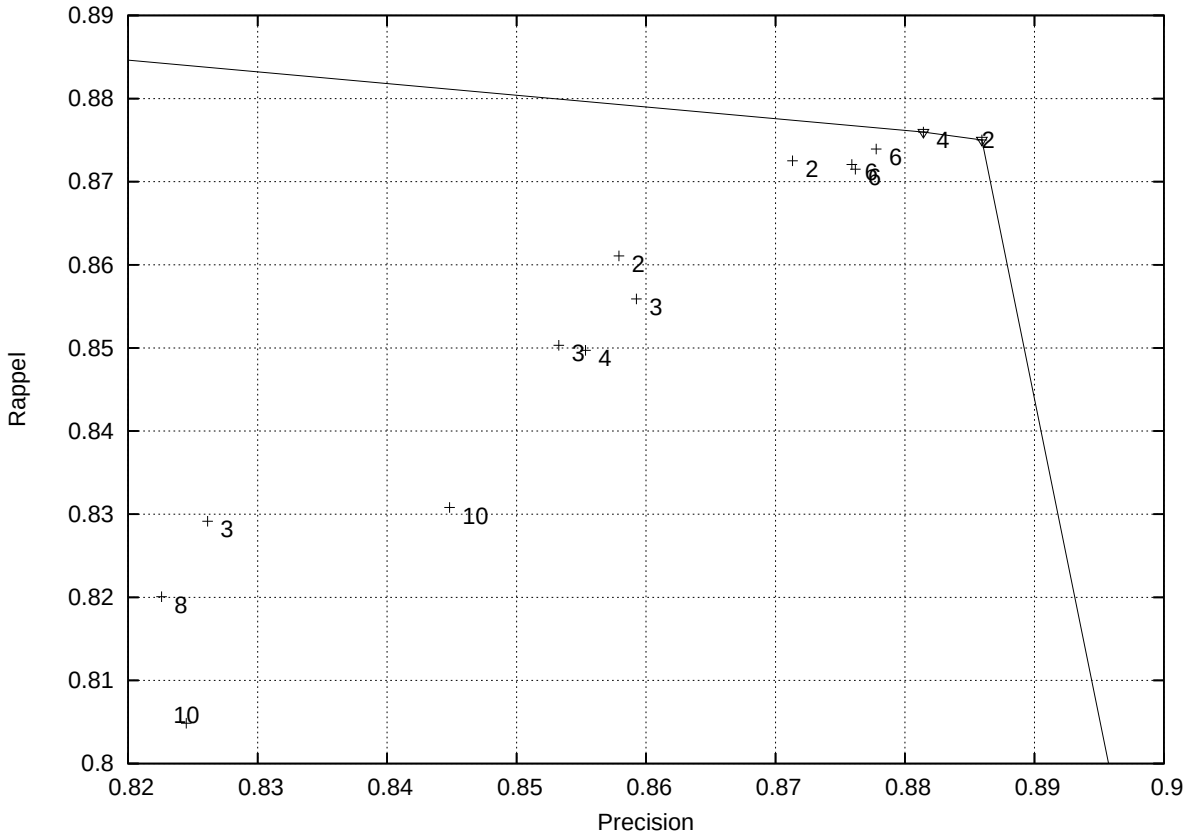

FIG. 7 – Front de Pareto pour la tâche 2, identification des catégories.

Le processus de classification comporte globalement deux étapes, en premier lieu une sélection des descripteurs du corpus, sur lesquels on applique ensuite un algorithme de classification. Plusieurs participants ont introduit une troisième étape de fusion des résultats de différents processus de classification ((Plantié *et al.*, 2008), (Béchet *et al.*, 2008), (Charton *et al.*, 2008)). Les méthodes de fusion améliorent généralement les résultats.

La sélection des descripteurs a donné lieu à une grande variété de méthodes. (Plantié *et al.*, 2008) et (Trinh *et al.*, 2008) ont choisi de ne pas lemmatiser les mots des documents. Chacun effectue ensuite une réduction de l'espace des mots, (Plantié *et al.*, 2008) par le calcul de l'information mutuelle sur chaque dimension, et (Trinh *et al.*, 2008) en calculant le score Okapi de chaque mot. (Béchet *et al.*, 2008) effectue des sélections par regroupement et normalisation de termes suivant des règles apprises automatiquement à partir de la maximisation du critère de discrimination. Des essais intéressants de classification du genre à partir du style et non du contenu thématique ont été proposés, soit à partir de la ponctuation des textes (Béchet *et al.*, 2008) ou des catégories morphosyntaxiques (Cleuziou & Poudat, 2008). Le critère d'impureté de Gini s'avère être un facteur discriminant efficace.

L'algorithme SVM a été plusieurs fois utilisé ((Cleuziou & Poudat, 2008), (Trinh *et al.*, 2008), (Plantié *et al.*, 2008), (Charton *et al.*, 2008)) avec de très bonnes performances. L'utilisation d'un noyau linéaire semble avoir été le meilleur choix. Cette méthode est très performante, mais sa limite est d'être peu explicative sur la discrimination linguistique qu'elle opère entre les classes. Par ailleurs, l'algorithme SVM séparant un corpus de documents en deux classes, des solutions doivent être trouvées pour résoudre le problème multi-classes. (Trinh *et al.*, 2008)

propose une méthode probabiliste efficace pour le choix de la meilleure classe, basée sur la maximisation de la log-vraisemblance conditionnelle. En revanche les classifieurs probabilistes à base de n-grammes de mots offrent l'avantage d'être explicatifs et permettent des observations intéressantes sur les discriminants linguistiques ((Charnois *et al.*, 2008), (Béchet *et al.*, 2008)). Les méthodes de boosting appliquées sur des classifieurs simples permettent d'en améliorer les performances.

Conclusion

Le défi cette année comportait plusieurs enjeux. Tout d'abord une possibilité de bilan sur les méthodes de classification en genre et de classification en thèmes, déjà largement explorées. Et ensuite une exploration de la classification thématique d'un mélange de genres, et de l'impact possible d'une classification en genre sur une classification en thème. Les systèmes rassemblant des méthodes innovantes de sélection des termes discriminants, des algorithmes éprouvés de classification, et des processus de fusion entre résultats, ont été les plus performants ((Béchet *et al.*, 2008), (Charton *et al.*, 2008)), en concurrence avec un système utilisant un SVM amélioré (Trinh *et al.*, 2008).

La classification en genre s'est révélée, pour les corpus choisis, plus facile. Les genres journalistiques et encyclopédiques se séparent bien, même dans des domaines semblables. Les catégories thématiques semblent plus difficiles à séparer, et la connaissance préalable du genre ne paraît produire qu'une amélioration à la marge. Il s'agissait d'un premier essai et d'autres possibilités de corpus sont à explorer, mettant en jeu par exemple des différences en genres spécialiste/néophyte dans des domaines de spécialité.

Références

BÉCHET F., EL-BÈZE M. & TORRES-MORENO J.-M. (2008). En finir avec la confusion des genres pour mieux séparer les thèmes. In *Actes TALN'08*, Avignon.

CHARNOIS T., DOUCET A., MATHET Y. & RIOULT F. (2008). Trois approches du GREYC pour la classification de textes. In *Actes TALN'08*, Avignon.

CHARTON E., CAMELIN N., ACUNA-AGOST R., GOTAB P., LAVALLEY R., KESSLER R. & FERNANDEZ S. (2008). Pré-traitements classiques ou par analyse distributionelle : application aux méthodes de classification automatique déployées pour DEFT08. In *Actes TALN'08*, Avignon.

CLEUZIOU G. & POUDAT C. (2008). Classification de textes en domaines et en genres en combinant morphosyntaxe et lexique. In *Actes TALN'08*, Avignon.

PAROUBEK P., BERTHELIN J.-B., EL AYARI S., GROUIN C., HEITZ T., HURAULT-PLANTET M., JARDINO M., KHALIS Z. & LASTES M. (2007). Résultats de l'édition 2007 du DÉfi Fouille de Textes. In *Actes de l'atelier de clôture du 3ème DÉfi Fouille de Textes*, p. 9–17, Grenoble : Association Française d'Intelligence Artificielle.

PLANTIÉ M., ROCHE M. & DRAY G. (2008). Défi DEFT08 : Classification de textes en genre et en thème : Votons utile ! In *Actes TALN'08*, Avignon.

TRINH A.-P., BUFFONI D. & GALLINARI P. (2008). Classifieur probabiliste avec Support Vector Machine (SVM) et Okapi. In *Actes TALN'08*, Avignon.

Méthodes des participants

En finir avec la confusion des genres pour mieux séparer les thèmes

Frederic Bechet[1], Marc El-Bèze[1] et Juan-Manuel Torres-Moreno[1,2]
[1] Laboratoire Informatique d'Avignon, UAPV
339 chemin des Meinajariès, BP1228, 84911 Avignon Cedex 9, France
{frederic.bechet,marc.elbeze,juan-manuel.torres}@univ-avignon.fr
[2] École Polytechnique de Montréal, Département de génie informatique
CP 6079 Succ. Centre Ville H3C 3A7, Montréal (Québec), Canada.

Résumé. Nous présentons des modèles d'apprentissage probabilistes appliqués à la tâche de classification telle que définie dans le cadre du défi DEFT'08 : la prise en compte des variations en genre et en thème dans un système de classification automatique. Une comparaison entre les résultats en validation et en tests montrent une coïncidence remarquable, et mettent en évidence la robustesse et performances de la fusion que nous proposons. Les résultats que nous obtenons, en termes de précision, rappel et F-score strict sur les corpus de test sont très encourageants.

Abstract. We present a set of probabilistic models applied to binary classification as defined in the DEFT'08 challenge. The challenge consisted a mixture of two differents problems in Natural Language Processing : identification of type and thematic category detection. Machine Learning and Bayes models have been used to classify documents. Applied to the DEFT'08 data test the results in terms of precision, recall and strict F-measure are very promising.

Mots-clés : Méthodes probabilistes, Apprentissage automatique, Classification de textes par leur contenu, défi DEFT.

Keywords: Statistical methods, Machine Learning, Text classification, DEFT challenge.

1 Introduction

Cet article présente les approches suivies au Laboratoire Informatique d'Avignon (LIA) pour la participation à la campagne de test DEFT'08. Trois approches sont présentées ainsi que diverses méthodes de fusion, représentant les trois soumissions envoyées pour le défi. Outre les méthodes de classification employées, les principales caractéristiques de notre méthodologie sont les suivantes :

– pré-traitement des données basée sur une chaîne d'outils linguistiques de *shallow parsing* ;
– méthodologie de développement et de réglages des systèmes utilisant la validation croisée en 5 sous-ensembles du corpus d'apprentissage (*5-fold cross-validation*) ;
– fusion systématique des systèmes développés afin d'augmenter la robustesse de la classification et réduire le risque de sur-apprentissage.

2 Méthodes

Les outils de classification de texte peuvent se différencier par la méthode de classification utilisée et par les éléments choisis afin de représenter l'information textuelle (mots, étiquettes morpho-syntaxique –*Part Of Speech*, POS–, lemmes, stemmes, sac de mots, sac de n-grammes, longueur de phrase, etc.). Parce qu'il n'y a pas de méthode générique ayant donné la preuve de sa supériorité (dans tous les cas de classification d'information textuelle), nous avons décidé d'utiliser une combinaison de différents classifieurs et de différents représentations. Cette approche nous permet, en outre, d'en déduire facilement les mesures de confiance sur les hypothèses produites lors de l'étiquetage. Trois systèmes ont été développés, utilisant plusieurs méthodes de classification et différentes représentations textuelles. Il s'agit d'obtenir des *avis différents* sur l'étiquetage d'un texte. En outre, le but n'est pas d'optimiser le résultat de chaque classifieur indépendamment mais de les utiliser comme des outils dans leur paramétrage par défaut et d'approcher l'optimum pour la fusion de leurs résultats. Nous allons présenter les trois méthodes utilisées, en commençant par détailler le pré-traitement des données effectuée.

2.1 Pré-traitement des données

Chaque document fourni durant la campagne DEFT est constitué de texte *brut* provenant de corpus électronique du journal *Le Monde* ou du site *Wikipedia*. Le premier traitement effectué sur ces documents est d'appliquer une chaîne de traitement d'étiquetage de surface (ou *shallow parsing*) développée au LIA[1]. Cette chaîne est composée des modules suivants :
tokeniseur → *étiqueteur morpho-syntaxique* → *lemmatiseur* → *extracteur d'entité-nommées*.

Le lexique utilisé pour les tokeniseur, étiqueteur et lemmatiseur contient 270K entrées. L'étiqueteur morpho-syntaxique est basée sur un jeu de 105 étiquettes et implémente une approche probabiliste basée sur les Chaînes de Markov Cachées. L'extracteur d'entité nommées a été développé dans la cadre de la campagne d'évaluation ESTER sur la détection d'entité nommées à partir de transcriptions de l'oral d'émissions radiophoniques d'information. Il est également basé sur une approche statistique, à base de Champs Conditionnels Aléatoires (ou Conditional Random Fields), et utilise un jeu de 8 étiquettes : personnes, lieux, organisations, entité géographique/socio-politique, montant, temps, produits, batiments.

2.2 Validation croisée

La méthode suivie pour l'apprentissage et le réglage des paramètres de classifieurs est celle de la validation croisée en 5 sous-ensembles (*5-fold cross validation*). Le principe général de la validation croisée est le suivant :
– Diviser toutes les données D disponibles en k groupes $D = G_1, \ldots, G_k$;
– *Erreur* = 0 ;
– Pour i allant de 1 à k
  – *E_test* = G_i ;
  – *E_train* = $D - G_i$;
  – apprentissage du modèles M sur *E_train* ;

[1] ces outils sont téléchargeables sur la page :
`http ://lia.univ-avignon.fr/personnel/fiches-personnelles/bechet/`

– *Erreur* $+=$ évaluation de M sur *E_test* ;

À la fin de k itérations, *Erreur* contient l'évaluation de la méthode de classification sur l'ensemble des données disponibles. En minimisant cette quantité lors du développement et de la mise au point des différents classifieurs, nous avons l'avantage d'avoir pu tester ces méthodes sur l'ensemble des données disponibles en ayant limité le risque de sur-apprentissage. Pour chaque tâche du défi nous avons segmenté le corpus d'apprentissage en 5 sous-ensembles. Nous avons obtenu des groupes de 3045 documents pour la tâche 1 et 4711 pour la tâche 2.

2.3 Exécution 1 : l'approche *E*$_1$*LiA*

Nous décrivons, dans cette section, les traits essentiels des systèmes qui ont permis de produire ce qui dans le cadre de DEFT'08 porte le nom de *première exécution* du LIA et que nous dénoterons *E*$_1$*LiA* par la suite. Faute de place, nous ne pourrons aborder ici toutes les idées que nous avons testées. La plus importante réside sans doute dans le fait de fusionner les sorties de plusieurs systèmes. Quand ce principe est appliqué sur plusieurs systèmes développés par des personnes différentes[2], il y a plus de chances que l'indépendance de leur conception garantisse de meilleurs résultats, sinon une certaine robustesse. En tous les cas, une bonne façon d'accroître la variété recherchée consiste à employer des méthodes différentes. Si les 3 méthodes retenues pour *E*$_1$*LiA* relèvent d'une modélisation probabiliste, leurs différences suffisent à atteindre la diversité recherchée : chaîne de Markov, loi de Poisson, adaptation d'un *cosine* classique pour y intégrer des facteurs discriminants. Pour chacun de ces modèles, l'estimation des probabilités ou des poids n'est pas effectuée à partir de comptes mais de fractions d'unité rendant compte de la plus ou moins grande capacité des termes à caractériser un genre ou un thème. Ce choix déjà mis en œuvre, lors de notre participation à DEFT'07 (Torres-Moreno *et al.*, 2007) a été affiné pour tirer parti de la capacité d'un terme à réfuter une classe ou de façon plus générale à en réfuter (resp. caractériser) x parmi k.

Facteur Discriminant

À l'instar de ce que nous avions déjà fait, lors du précédent défi, nous avons recherché un facteur permettant de mesurer le pouvoir discriminant de tel ou tel terme. Pour répondre à cette attente, nous avons légèrement adapté le critère[3] d'impureté de Gini (cf. formule 1). Pour renverser le point de vue, nous proposons d'utiliser un critère qui n'est rien d'autre que le complément à 1 du premier (cf. formule 2) et que nous appelons critère de pureté de Gini $PG(i)$.

$$IG(i) = \sum_{t \neq j} P(j|i)P(t|i) = 1 - \sum_{j=1}^{k} P^2(j|i) \qquad PG(i) = 1 - IG(i) = \sum_{j=1}^{k} P^2(j|i) \quad (1)$$

Dans la formule 1, i désigne un terme, j et t l'une ou l'autre des k classes. $PG(i)$ prend ses valeurs entre 1, dans le meilleur des cas et l'inverse du nombre de classes, quand i n'est pas du tout discriminant. Nous avons affiné ce critère en le combinant linéairement avec un critère plus conciliant $PG'(i)$ qui accorde une importance égale aux termes caractérisant 1 ou 2 classes.

$$PG'(i) = \max_{c} \sum_{j=1}^{k-1} P_c^2(j|i) \quad (2)$$

[2]Comme cela a été fait pour les deux autres *exécutions*

[3]Critère employé depuis longtemps comme substitut à l'entropie pour la construction des arbres de décision.

$PG'(i)$ peut être estimé en appliquant la formule 2. De cette façon, on est amené à rechercher une valeur maximale de P_c qui n'est rien d'autre qu'une des $(k-1) * k/2$ distributions que l'on obtient en regroupant 2 des k classes. L'introduction de cette variante a été essentiellement motivée par le fait que nous avons souhaité, pour la tâche 1, voir ce que pouvait donner une détection conjointe du genre et de la catégorie. Dans ce cas, on se retrouve avec un jeu de 8 classes, où il n'est pas absurde de penser qu'il doit y avoir de forts recouvrements entre X_W et X_LM. Un tel lissage peut être généralisé en regroupant x classes parmi k *avec* $(2 < x < k)$. Lorsque $x = k - 1$, c'est le pouvoir de réfuter une classe qui est associé au terme i.

Agglutination et Normalisation

Ce qui est mis en œuvre dans cette phase pourrait être vu comme une simple étape préalable au cours de laquelle sont appliquées des règles de réécritures pour regrouper les mots[4] en unités de base. Un autre ensemble de règles appropriées est mis à contribution pour normaliser les graphies. Or, pour rester indépendant de la langue et de la tâche, nous n'avons pas souhaité demander à des experts de produire ces deux ensembles de règles, ni même recourir à des ressources préexistantes[5], comme nous l'avions fait lors de DEFT07. Cette fois, notre objectif était de faire émerger, de façon automatique, ces règles à partir des textes. Pour ce faire, nous avons choisi de prendre appui sur le contexte, les classes, et une mesure numérique. Deux termes consécutifs ne sont collés que si le pouvoir discriminant[6] de l'agglutination qui en résulte est supérieur à celui de chacun de ces composants, et si la fréquence d'apparition est supérieure à un certain seuil. Le principe est le même pour les règles de réécriture dont la vocation est soit de corriger les éventuelles coquilles, soit de généraliser une expression (par exemple remplacer les noms des mois par une entité abstraite `MOIS`). Nous envisageons de proposer par la suite une modélisation plus élaborée qui apporterait une réponse à la question suivante : comment, au moyen des opérateurs de concaténation et d'alternance, inférer des automates probabilistes à partir d'un corpus étiqueté en termes de classes thématiques ?

Pour donner une idée des unités de base sur lesquelles les modèles ont été entraînés dans le cadre de *E*$_1$*LiA*, nous avons sélectionné un petit nombre d'agglutinations obtenues à l'issue de quelques itérations. Par exemple, en filtrant celles qui contiennent le mot roi dans la tâche 2, la plus longue *il-voter-mort-du-roi* est apparue 27 fois dans la catégorie *FRA*, et uniquement dans cette catégorie. Le pouvoir discriminant maximal qui lui est associé dénote une particularité de l'histoire de France. Pour illustrer notre propos sur les opérations conjointes d'agglutination et de normalisation, nous avons observé que parmi les expressions contenant le mot *lundi* la plus longue *lundi-de*-HEURE-à-HEURE est apparue 17 fois dans le genre LM, et jamais dans W. Une recherche plus poussée dans le modèle fait apparaître clairement qu'un journal (contrairement à Wikipédia) donne les horaires d'ouverture et de fermeture d'un musée ou d'une exposition. On peut, donc, remplacer cette agglutination par une expression régulière plus générale JOUR-*de*-HEURE-*à*-HEURE. À condition bien entendu que le pouvoir discriminant ne soit pas affaibli, il est intéressant d'augmenter ainsi la couverture de chaque unité. À l'issue d'une cinquantaine d'itérations, nous avons ainsi produit automatiquement entre 15 000 et 20 000 règles de réécriture selon les tâches, et entre 25 000 et 70 000 règles d'agglutination. Ces nombres, ainsi que les exemples que nous avons donnés, permettent d'imaginer le temps et la diversité de l'expertise qu'il aurait fallu réunir, si nous avions dû produire manuellement ces règles.

Pour la tâche 3, nous avons fait l'hypothèse que si les articles du journal *le Monde* ne fai-

[4]Il serait plus correct de dire leurs lemmes car nous utilisons les formes lemmatisées par LIA_TAG

[5]Il s'agissait tout simplement de listes d'expressions figées et autres proverbes.

[6]Par exemple, le critère de pureté de Gini tel qu'il est défini en section précédente.

saient plus comme autrefois l'objet d'une relecture par des correcteurs humains, ils étaient néanmoins lus et relus par leurs auteurs et donc plus propres. De ce fait, la tentation d'appliquer des outils de correction orthographique (comme par exemple celui qui remplace le triplement d'une consonne par son doublement, ou supprime son redoublement `dreyff ?uss ? -> dreyfus`) pouvait ici être contre-productive. D'un autre côté, ne pas corriger enlève au modèle la capacité de capturer dans leur intégralité les fréquences des termes associés aux thèmes abordés de façon spécifique dans LM ou W. Pour jouer sur les deux tableaux, nous avons choisi d'ajouter, quand la correction est faite pour cette tâche un terme fictif : `TYPOW`. Les erreurs d'accents sont aussi traitées, comme le montre l'exemple suivant : `bat[ôo]nnets ? -> bâtonnet TYPOW` ainsi que quelques manifestations du phénomène de dysorthographie qui sévit sur le Web et quasiment pas dans LM : `commerica(le ?s ?|ux)-> commercial TYPOW.`

Méthodes

Comme signalé en début de section, nous avons choisi de diversifier les méthodes pour avoir un grand nombre de sorties en vue d'une fusion aussi performante que possible. Nous n'en avons retenu que 3, mais grâce à quelques variantes, nous en avons obtenu environ une dizaine.

Méthode probabiliste ou classifieur n-grammes : la méthode des *n-grammes*, que nous avons employée lors de DEFT'07, s'apparente à une modélisation markovienne. Elle a été appliquée ici à l'identique. Notons que le modèle peut-être vu comme un unigramme sur les unités composites définies à la section 3.2, ou comme un modèle *n-gramme*, n étant variable et déterminé par le critère discriminant défini à la section 1.1. Ces différents points de vue ont donné lieu à la mise en place de trois variantes, qui correspondent à trois réponses apportées à une question cruciale : faut-il prévoir à des fins de lissage une procédure de repli sur les différents composants d'une agglutination ? La première variante notée *Prb* consiste à considérer que l'agglutination a été vue dans sa globalité ainsi que chacun de ses composants. La seconde *Prb77* consiste à ignorer les parties pour privilégier le tout. À l'inverse, la troisième *Prb7* ignore l'agglutination qui est décomposée en chacune de ses parties. Elle diffère d'un simple *unigramme* car, à l'issue de la composition, des réécritures ont pu modifier de façon conséquente le texte d'origine.

Poisson : dans l'optique de doter un système de reconnaissance de la parole d'un pré filtre acoustique (**?**) proposent de recourir à la loi de Poisson particulièrement bien adaptée pour modéliser les événements rares. Nous avons transposé cette méthode pour qu'elle puisse être appliquée sur des termes et non des observations acoustiques et que la liste ordonnée en sortie ne soit pas une liste de mots candidats mais la liste ordonnée des classes. Nous n'avons pas suffisamment de place ici pour dérouler la séquence d'équations qui mène à la formule 3 déterminant la classe optimale $\hat{k}$ en fonction de la longueur moyenne $\mu(k)$ d'un article dans la classe k, et de $\mu(i,k)$ le nombre moyen de fois où le terme i apparaît dans un article de la classe k.

$$\hat{k} = \underset{k}{\textit{ArgMax}} \sum_{i=1}^{T} \log \mu(i,k) - \mu(k) + \log P(k) \tag{3}$$

Si un article testé n'a aucun terme en commun avec ceux qui constituent le corpus d'apprentissage, le troisième terme fait en sorte que la classe la plus probable sera choisie. Il est à noter que, comme pour la méthode précédente, les paramètres des différents modèles sont estimés à partir des fréquences relatives des événements pondérées par le critère $PG(i)$. Ainsi, plus un terme i est discriminant plus il contribuera à la sélection de la classe k à laquelle il est rattaché. Dans le même esprit que pour *Prb*, nous avons, ici aussi, trois variantes : *Poi*, *Poi7* et *Poi77*.

Cosine : les articles du corpus d'apprentissage appartenant à une classe sont considérés comme formant un seul document. À chaque classe, il est ainsi possible d'associer un vecteur dont les

composantes sont des poids autres que le classique TF(i,k).IDF(i). En effet, nous avons pensé qu'il convenait de remplacer de façon totale ou partielle IDF(i) par PG(i). Lors du test, tout nouvel article est "vectorisé" et le calcul de Cosine est effectué pour mesurer sa ressemblance avec chacune des classes. Lors de la fusion ultérieure, cette mesure notée *Cos* (ainsi que ses variantes *Cos7* et *Cos77*) sera interprétée de façon un peu abusive comme une probabilité.

Stratégies

Pour coller au plus près des spécificités de DEFT'08 telles que définies par les organisateurs du défi, nous avons tenté d'apporter une réponse aux deux questions suivantes : 1. l'identification préalable du genre permet-elle d'améliorer le fonctionnement de la classification thématique ? 2. le fait de connaître *a priori* le genre aurait-il permis d'aller encore plus loin ? Pour répondre à la première question, nous avons identifié le genre des notices de la tâche 2, en utilisant les modèles appris sur la tâche 1, autant pour les données de test que d'apprentissage. Les organisateurs du défi ayant mis à notre disposition les références du genre pour les données de test de la tâche 2, il nous a été possible d'estimer la qualité de cet étiquetage préalable. Nous avons sélectionné les six méthodes qui s'étaient montrées être les meilleures lors de la phase d'apprentissage sur la tâche 1. Après avoir fusionné des différentes hypothèses de genre produites par ces méthodes sur la tâche 2, nous avons observé que le F-score de cet étiquetage valait 0,9567. La prise en compte du genre après une identification préalable du genre par les moyens que nous venons de décrire, et en limitant l'emploi de cette stratégie à 3 méthodes (Cos32, Poi32, Prb32) a permis d'améliorer l'identification des catégories de la tâche 2. En effet, le F-score augmente de plus de 1% : on passe de 0,8605 à 0,8719.

La qualité de l'identification du genre pour la tâche 2 se situe plus de 2% en dessous de celle observée sur la tâche 1. S'il est vrai que la différence entre le jeu des catégories de la tâche 2 et celui de la tâche 1 explique en partie ce décalage, on peut y voir l'influence implicite des catégories sur la détection du genre. Par ailleurs, pour donner un début de réponse à la seconde question, nous avons voulu mesurer l'impact sur la catégorisation thématique des erreurs commises sur le genre lors du test. Pour cela, nous avons pris appui sur les références du genre pour sélectionner les modèles thématiques correspondants. Le F-score atteint alors 0,8755 avec cette fois 13897 étiquettes correctes. Cela laisse présumer que si les étiquettes de genre avaient été connues également lors de l'apprentissage, les résultats auraient pu être encore meilleurs.

Discussion

Faute de place, nous n'avons reporté dans cette section que les résultats détaillés concernant la tâche 2. Nous avons tiré de nos observations sur l'apprentissage quelques enseignements qui ont été vérifiés sur le test. Le plus important concerne le choix d'une stratégie hiérarchisée : il est intéressant d'identifier d'abord le plus facile, ici le genre, puis le plus difficile, ici la catégorie. Si le système a connaissance du genre[7], il peut encore mieux identifier le thème. Nous aurions pu remarquer également qu'ici la variante 77 (ni repli ni décomposition) est supérieure aux 2 autres, et que Cosine surpasse les 2 autres méthodes. Nous ne l'avons pas fait, car cela n'est pas toujours vrai sur les 2 autres tâches. Il convient de remarquer enfin que pour E_1*LiA* l'étiquetage en entités nommées n'a pas été utilisé, mais laissé en perspective pour des travaux à venir.

[7] Il est légitime de faire l'hypothèse que des informations sur la provenance des textes peuvent être fournies sans grandes difficultés au système.

2.4 Classification par *Boosting* de classifieurs simples : l'approche *E_2LiA*

La deuxième approche utilisée pour le défi DEFT'08 est basée sur la méthode du *Boosting* de classifieurs simples *AdaBoost* proposée par (Freund & Schapire, 1997). L'algorithme AdaBoost a pour but d'améliorer la précision de règles de classification en combinant plusieurs hypothèses dites faibles (hypothèses peu précises). Les algorithmes de boosting travaillent en re-pondérant répétitivement les exemples dans le jeu d'entraînement et en ré-exécutant l'algorithme d'apprentissage précisément sur ces données re-pondérées. Cela permet aux classifieurs simples de se concentrer sur les exemples les plus problématiques. L'algorithme de boosting obtient ainsi un ensemble de classifieurs qui sont ensuite combinés en une seule règle de classification appelée hypothèse finale ou combinée. L'hypothèse finale est un vote pondéré des hypothèses faibles.

Cet algorithme peut être résumé de la manière suivante :
Étant donné :
- Un jeu de données $S : (x_1, y_1), \ldots, (x_m, y_m)$ où, à chaque exemple $x_i \in X$, on associe une étiquette $y_i \in Y = \{-1, +1\}$;
- Une distribution initiale des poids $D_1(i) = 1/m$ uniforme sur ces données ;
- Un apprenant faible (*weak learner*).
- Alors pour chaque tour $t = 1, ..., T$:
  - Entraîner l'apprenant faible sur le jeu de données S avec la distribution D_t ;
  - Obtenir le classifieur $h_t : X \rightarrow \{-1, +1\}$ ainsi que l'erreur : $e_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$
  - Calculer la pondération du tour t : $\alpha_t = \frac{1}{2} \ln(\frac{1-e_t}{e_t})$
  - Mettre à jour la distribution : $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
    avec Z_t un facteur de normalisation permettant à D_{t+1} d'être une distribution.
- En sortie, on obtient un classifieur correspondant au vote pondéré de toutes les classifieurs *faibles*, un par itération : $h_{final} = \sum_{t=1}^{T} \alpha_t h_t(x)$

Nous avons utilisé l'implémentation BoosTexter[8] d'Adaboost dans nos exprériences. Deux types de classifieurs simples (*weak learner*) ont été utilisés :

1. des *n-grammes* de tokens, avec $1 \leq n \leq 3$, soit sur les mots, soit sur les lemmes ;
2. des informations numériques correspondant aux entités nommées : % d'entités de chaque type par rapport à l'ensemble des entités d'un document ; année minimale et maximale trouvée dans les entités *dates* d'un document.

Le but des informations numériques sur les entités nommées est d'une part de caractériser chaque document par rapport à la densité de chaque type d'entités, et d'autre part d'utiliser les années pour caractériser la portée temporelle d'un document. Voici les 20 *n-grammes* de lemmes selectionnés comme étant les plus pertinents pour la classification en catégories de la tâche 1 :`(joueur)(pour_cent)(»)(film)`
`(championnat)(album)(entreprise)(diffuser)`
`(.téléphone,)(olympique) (artiste)(émission)`
`(...)(vainqueur) (milliard)(musique)(champion du monde)`
`(être un peintre)(.,) (économique)`

On voit bien les deux dimensions de la classification d'un document : le thème porté par des mots clés (*joueur*, *championnat*, etc.) et les informations de formatage et de mise en page comme par exemple le symbole "»" qui est présent, dans le corpus de la tâche 1, dans 4311

[8] http ://www.cs.princeton.edu/ schapire/boostexter.html

documents du journal *Le Monde* contre 1449 documents de *Wikipedia*, et dans seulement 82 documents *Télévision* contre 2789 documents *Art*. Enfin des mesures de confiance sont estimés par une fonction de régression logistique appliquée sur les scores de classification, comme proposé dans (E.Schapire *et al.*, 2005).

2.5 *E*$_3$*LiA* : modèle de langage vide de mots :-) *versus* :-(?

A l'oral, la voix monte, descend, observe des temps de pause, des arrêts,... Elle permet à elle-seule de moduler et de cadencer notre discours, de lui donner tout son sens, de mettre en avant certains mots, certaines phrases. A l'écrit, toute cette richesse inhérente à la voix n'est plus[9]. Ainsi, des moyens de se faire comprendre des lecteurs, de traduire les oscillations de timbre et de rythme de la voix ont été inventés. C'est dans l'antiquité (entre le III et le II siècle avant JC) qu'ont été introduits les signes de ponctuation. Ainsi les 10 signes actuels demeurent inchangés depuis le XVIIe siècle, mais de nouveaux tentent régulièrement de s'imposer, notamment avec les nouveaux média de communication comme l'Internet et les SMS. Dans le cas du défi DEFT'08, nous voulions savoir si la ponctuation et rien qu'elle permettrait de discriminer le genre (et probablement la classe) des documents. Dans le cas affirmatif, cela présenterait plusieurs avantages. D'abord les signes de ponctuation font partie d'un sous-ensemble très réduit. Dans l'utilisation des modèles n-grammes, on peut se passer des techniques du lissage ou de *Back-Off* (Manning & Schütze, 2000), car à la différence des mots, la plupart des signes vus dans la phase d'apprentissage restent des évènements vus lors du test. Enfin la ponctuation reste relativement stable entre les langues et, sauf rares exceptions, la plupart des signes sont les mêmes. Nous avons développé un classifieur classique incorporant des techniques élémentaires de n-grammes, mais en utilisant les signes à la place des mots. Ces techniques, descendantes directes de l'approche probabiliste (Manning & Schütze, 2000) appliquées à la classification de texte, ont prouvé leur efficacité dans les défis DEFT précédents (El-Bèze *et al.*, 2005; Torres-Moreno *et al.*, 2007). Pour cela, nous avons réalisé un filtrage de tous les mots des corpus d'apprentissage. Pour la tâche 1, nous avons construit les modèles n-grammes associés au genre $g \in \{W, LM\}$. L'ensemble de signes de ponctuation s du genre *Wikipédia* est $s_W \approx 50$ et du genre *Le Monde*[10] $s_{LM} \approx 80$. Le score du genre $\tilde{g}$ étant donné un document et une séquence de signes s, est ainsi calculé (application du théorème de Bayes) :

$$\tilde{g} = \arg\max_g P(g|s) = \arg\max_g \frac{P(s|g)P(g)}{P(s)} = \arg\max_g P(s|g)P(g) \quad (4)$$

$$\tilde{g} \approx \arg\max_g P(s|g) = \arg\max_g \prod_i P_g(s_i|s_{i-2}, s_{i-1}) \quad (5)$$

combinée avec une interpolation simple :

$$P_g(s_i|s_{i-2}, s_{i-1}) \approx \lambda_2 P_g(s_i|s_{i-1}, s_{i-2}) + \lambda_1 P_g(s_i|s_{i-1}) + \lambda_0 P_g(s_i); \sum \lambda_n = 1 \quad (6)$$

Nous avons limité notre modèle à des 3-grammes et nous l'avons appliqué au corpus de la tâche 1, sans faire d'autres traitements particuliers. Nous avons déterminé le genre, mais le F-score restait modeste. Nous avons enrichi le modèle avec les mots dits fonctionnels (définis par opposition aux mots sémantiquement pleins). Les classes ont été identifiées de façon similaire, en

[9] http ://www.la-ponctuation.com/

[10] En fait nous avons gardé tout ce qui n'est pas un mot. Ainsi des symboles dans un sens large, plutôt qu'uniquement des signes de ponctuation ont été retenus.

s'appuyant sur le genre calculé. Une autre stratégie, la longueur des phrases, a été aussi utilisée. Pour la tâche 2, uniquement l'identification des catégories a été réalisé. Les performances en test sont F-score=90,21 (genre) et F-score=81,30 (catégorie) pour la tâche 1 et F-score=84,41 pour la tâche 2. Malgré ces performances relativement inférieures aux autres approches, nous avons montré qu'il est possible de faire une classification presque sans mots, au moins dans les tâches DEFT'08 ou les corpus restent relativement distincts (sauf pour la classe Télévision). Nous avons décidé d'incorporer ce modèle dans notre fusion.

2.6 Fusion de modèles

La fusion de modèles est une méthode permettant d'augmenter facilement la robustesse des règles de classification en multipliant les points de vue sur le même phénomène, sans chercher pour autant à régler au plus fin chaque méthode. Un réglage trop fin comporte toujours le risque d'effectuer du sur-apprentissage. Nous avons montré lors du défi DEFT'07 qu'utiliser un très grand nombre de classifieurs permettait d'éviter ce phénomène. En effet, nous avons obtenu des résultats remarquablement proches entre les corpus d'apprentissages et ceux du test du défi. Diverses méthodes peuvent être employées pour fusionner des hypothèses de classification : vote simple, vote pondéré, moyenne pondérée des scores de confiance, régression, classifieur de classifieurs, etc. Nous avons choisi, comme lors du défi précédent, de privilégier les méthodes simples, nous avons utilisé la moyenne pondérée des scores de confiance, avec un jeu de coefficients choisi pour minimiser l'erreur sur le corpus d'apprentissage. Nous avons utilisé une méthode exhaustive, ce choix se justifiant par le faible nombre de classifieurs que l'on va fusionner : au maximum 5 dans nos expériences. Evidemment, pour un nombre de classifieurs plus important, le choix d'une méthode approchée s'impose.

3 Résultats et discussion

Nous présentons dans cette section les résultats obtenus sur 3 tâches de classification :
- *Tâche_1_CAT* - Reconnaissance de la catégorie thématique sur le corpus de la tâche 1 parmi les quatre classes : *ART* (Art), *ECO* (Économie), *SPO* (Sports), *TEL* (Télévision)
- *Tâche_1_GENRE* - Reconnaissance du genre sur le corpus de la tâche 1 parmi les deux classes : *W* (Wikipédia), *LM* (Le Monde)
- *Tâche_2_CAT* - Reconnaissance de la catégorie thématique sur le corpus de la tâche 2 parmi les six classes : *FRA* (Politique française), *INT* (International), *LIV* (Littérature), *SCI* (Sciences), *SOC* (Société)

Méthode	E_1*LiA*	E_2*LiA*	E_3*LiA*	Fusion optimale
Tâche_1_CAT	90.1	90.9	81.6	92.3
Tâche_1_GENRE	97.8	98.1	90.8	98.9
Tâche_2_CAT	87.6	84.8	80.4	88.7

TAB. 1 – F-score obtenue par nos 3 méthodes sur le corpus d'apprentissage par la méthode de la validation croisée en 5 sous-ensembles

La table 1 présente les résultats obtenus sur le corpus d'apprentissage, par la méthode de la validation croisée présentée précédemment. Comme on peut le voir, la fusion fait gagner systé-

matiquement sur toutes les tâches.

Méthode	E_1*LiA*	E_2*LiA*	E_3*LiA*	Fusion devt	Fusion optimale
Tâche_1_CAT	85.2	85.9	81.3	88.3	88.3
Tâche_1_GENRE	95.8	97.9	90.2	98.1	98.4
Tâche_2_CAT	86.1	85.2	84.4	87.2	87.9

TAB. 2 – F-score obtenus par les 3 systèmes utilisés lors de la soumission au test

Les résultats obtenus sur le corpus de test du défi DEFT'08 valident nos approches : à l'exception de la classification en thème du corpus 1, pour laquelle on constate une perte de 4 points de F-score, les résultats obtenus entre l'apprentissage et le test sont très proches.

4 Conclusion

Le protocole de réglage par validation croisée ainsi que le choix systématique de fusionner les sorties de systèmes complémentaires s'avère une fois de plus performant et robuste. Le fait de n'avoir pas optimisé de façon spécifique chacun des composants de la fusion laisse une marge que nous espérons substantielle pour améliorer nos modèles.

Références

EL-BÈZE M., TORRES-MORENO J.-M. & BÉCHET F. (2005). Peut-on rendre automatiquement à César ce qui lui appartient ? Application au jeu du Chirand-Mitterrac. In *Actes de la conférence "Traitement Automatique des Langues Naturelles" (TALN 2005) - Atelier DEFT'05*, volume 2, p. 125–134.

E.SCHAPIRE R., ROCHERY M., RAHIM M. & GUPTA N. (2005). Boosting with prior knowledge for call classification. *IEEE*, **13**(1), 174–181.

FREUND Y. & SCHAPIRE R. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, **55**(1), 119–139.

MANNING C. D. & SCHÜTZE H. (2000). *Foundations of Statistical Natural Language Processing*. The MIT Press.

TORRES-MORENO J.-M., EL-BÈZE M., BÉCHET F. & CAMELIN N. (2007). Comment faire pour que l'opinion forgée à la sortie des urnes soit la bonne ? application au défi deft 2007. In *Actes de la conférence "Traitement Automatique des Langues Naturelles" (TALN 2007) - Atelier DEFT'07*, p. 119–133.

Trois approches du GREYC pour la classification de textes

Thierry Charnois Antoine Doucet Yann Mathet François Rioult
GREYC, Université de Caen, CNRS UMR 6072
Bd Maréchal Juin, BP 5186, 14032 Caen Cedex
{charnois, doucet, mathet, frioult}@info.unicaen.fr

Résumé. Cet article présente la participation de l'équipe du GREYC à DEFT'08, en détaillant les différentes approches mises en place ainsi que les résultats obtenus. Plusieurs techniques très différentes ont été étudiées et mises en oeuvre. D'une part, un traitement à base de n-grammes a constitué un classifieur indépendant. D'autre part, deux autres traitements s'appuient sur un classifieur supervisé par règles d'association, qu'ils alimentent chacun avec des indices provenant d'une chaîne de traitements linguistiques pour l'un, et d'extraction de séquences pour l'autre.

Abstract. In this paper, we present the various approaches and corresponding results of the GREYC laboratory to the DEFT'08 challenge. A couple of distinct techniques were experimented with. On the one hand, an n-gram based classifier was tested. On the other hand, two different types of data were fed to a supervised classifier, based on association rules : 1) linguistic markers, and 2) discontinuous word sequences.

Mots-clés : Fouille de texte, classification par n-grammes, classification par règles d'association, TALN, séquences de mots.

Keywords: Text data mining, n-gram classification, association rule-based classification, NLP, word sequences.

1 Introduction

Le thème de cette édition 2008 de DEFT concerne la classification de textes en catégorie et en genre. Pour l'entraînement de la tâche, deux corpus sont disponibles. Le premier est un ensemble d'articles du journal Le Monde et d'articles de Wikipédia avec un double étiquetage : l'un en genre (Le Monde ou Wikipédia) et l'autre en catégorie thématique (sport, art, économie, *etc.*). Le deuxième corpus est également un ensemble d'articles issus des mêmes sources, mais avec un simple étiquetage en catégorie (les catégories de ce corpus étant différentes du premier). Pour le test, deux corpus sans étiquette ont été fournis et la tâche consiste à reconnaître le genre et la catégorie du premier corpus, la catégorie pour le deuxième.

Le laboratoire GREYC présente une équipe à DEFT pour la troisième année consécutive. Pour cette édition 2008, sa composition ainsi que les techniques mises en oeuvre sont en partie issues de la session précédente, et en partie nouvelles :

– Une approche complète à base de n-grammes, conçue et utilisée à l'occasion de DEFT'07, a été reprise et adaptée aux spécificités de DEFT'08,

– Un classifieur supervisé par règles d'association, utilisé depuis DEFT'06, est chaque année « alimenté » par différentes chaînes de traitement adaptées aux tâches spécifiques de chaque défi.

Pour cette session 2008, une chaîne de traitements linguistiques basée sur la plate-forme Linguastream a été conçue pour créer des indices concernant la classification en genres du corpus 1, et un autre traitement par extraction de séquences répondant ainsi à la classification en catégories des corpus 1 et 2.

Nous présentons dans les parties qui suivent chacune des trois approches, et terminerons pas une analyse comparative de nos résultats avant de conclure.

2 Un classifieur à base de n-grammes

Un classifieur autonome à base de n-grammes a été développé au GREYC en 2007 à l'occasion du défi DEFT'07, il est donc utilisé pour la deuxième année consécutive. Il a été conçu nativement pour gérer autant de catégories qu'on le souhaite, et a pu être adapté relativement facilement à la session DEFT'08 dans ses grands principes, même si un certain nombre d'aménagements techniques ont dû être effectués. Le présent exposé étant concis, nous invitons le lecteur désireux de connaître le détail de ce traitement à se référer aux actes de DEFT'07 (Vernier *et al.*, 2007).

2.1 Principe

La technique des n-grammes consiste à observer les collocations contiguës sur une fenêtre de n tokens consécutifs d'un flux, et à essayer de tirer de ces observations des régularités relatives à un aspect particulier de ce flux (Stubbs & Barth, 2003). Par exemple, certains n-grammes seront caractéristiques de tel type de corpus car très récurrents dans ce dernier, et beaucoup plus rares ailleurs. Pour illustrer de façon très simplifiée l'hypothèse de cette approche dans DEFT'08, nous espérons trouver des n-grammes très caractéristiques d'une catégorie par rapport aux autres catégories. Par exemple « en troisième division » ou « un match difficile » sont plutôt caractéristiques de la catégorie SPORT, tandis que « présentateur du JT », « une émission débridée » seront quant à eux plutôt caractéristiques de la catégorie TELEVISION. Ainsi, lors de l'analyse d'un texte du corpus de test, si l'on tombe sur le tri-gramme « présentateur du JT », nous serons tentés de ranger ce texte en catégorie TELEVISION. Bien sûr, deux difficultés apparaissent : il se peut qu'au cours de l'apprentissage, un même n-gramme soit présent dans des textes issus de différentes catégories ; et il est fréquent que lors du test, nous trouvions au sein d'un même texte des n-grammes issus de différentes catégories, rendant le choix plus difficile. L'idée que nous mettons en œuvre pour pallier ces difficultés est de deux ordres : ne retenir pour chaque catégorie que les n-grammes les plus discriminants, c'est-à-dire ceux étant le moins susceptibles d'apparaître dans des textes appartenant à d'autres catégories ; pondérer les n-grammes, c'est-à-dire associer à chacun un poids d'autant plus important qu'il apparaît fréquemment dans sa catégorie cible relativement aux autres catégories. Puis, pour chacune des catégories, sommer les poids de tous les n-grammes (de cette catégorie) trouvés dans le texte testé. De la sorte, nous obtenons une note globale pour chaque catégorie, que nous pouvons mettre en balance avec les notes globales obtenues pour les autres catégories.

2.2 Apprentissage

La collecte des n-grammes est effectuée pour chaque catégorie (ou genre). Nous extrayons dans ce but la collection des n-grammes dans chaque catégorie, puis calculons les n-grammes émergents par comparaison des n-grammes d'une catégorie à ceux des autres catégories :
– si un n-gramme est absent des autres catégories, on lui attribue un poids infini ;
– s'il est présent dans les autres catégories, on lui associe un poids égal au rapport entre sa fréquence relative dans la catégorie à caractériser et sa fréquence dans les autres.
Le n-gramme sera finalement utilisé dans le classifieur si son poids est supérieur à un certain seuil paramétrable.

2.3 Classification : choisir un genre ou une catégorie

Assigner une catégorie à un texte du corpus de test consiste à parcourir le texte au moyen d'une fenêtre de longueur n, et à recommander une catégorie pour laquelle un n-gramme découvert est émergent. Nous obtenons pour le texte de test autant de sommes de poids qu'il y a de catégories. Ces sommes correspondent à l'indice de confiance que l'on peut accorder à chaque catégorie. Nous pouvons alors assigner comme catégorie celle obtenant la somme de poids la plus élevée. Par ailleurs, notre application permet de combiner différents traitements à l'envi, par exemple bi-grammes et tri-grammes, selon que le corpus se prête mieux à telle ou telle combinaison.

2.4 Application aux différentes tâches

2.4.1 Méthode générale et ajustements

Chacun des deux corpus d'apprentissage fournis a tout d'abord été coupé en deux parties égales, constituant pour notre phase d'apprentissage un corpus d'apprentissage (par ex. la première moitié) et un corpus de test (par ex. la seconde moitié). De la sorte, nous avons une vision non biaisée des performances des traitements effectués, ce qui est indispensable pour réaliser les paramétrages les plus adéquats. Les apprentissages et tests ainsi réalisés, nous avons observé les résultats et constaté que certaines catégories étaient sous ou sur représentées, selon les rapports entre les rappels obtenus par chacune. C'est alors que nous appliquons des coefficients d'ajustement, afin que d'une part une catégorie sous représentée soit privilégiée (il suffit pour cela de multiplier les valeurs fournies par leurs n-grammes par un coefficient supérieur à 1), et d'autre part qu'une catégorie sur-représentée soit défavorisée (application d'un coefficient inférieur à 1). À l'issue de ces réglages, réalisés empiriquement par essais successifs (une automatisation nous aurait permis d'affiner nettement le réglage obtenu), nous obtenons une répartition en catégories beaucoup plus homogène en terme de rappel, et comme escompté, une amélioration du F-score.

Utilisée pour l'exécution 1 du défi, cette méthode donne un F-score de 96.4% sur le genre, 84.9% (tâche 1) et 83.7% (tâche 2) pour les catégories.

2.4.2 Test d'une hypothèse de DEFT'08 : la connaissance du genre aide-t-elle à mieux trouver la catégorie ?

La partie « GENRE » de la tâche 1 donnant lieu à des résultats extrêmement élevés (supérieurs à 96%), il est tentant d'essayer d'en tirer parti pour améliorer les scores sur la tâche 2. Le principe testé est le suivant :

1. appliquer le traitement en GENRE au corpus 2. On trouve alors pour chacun des textes, avec un degré de confiance supérieur à 96%, le genre de ce dernier.
2. diviser ainsi le corpus d'apprentissage en deux sous-corpus, l'un contenant les textes jugés appartenir au genre « Wikipedia », l'autre ceux jugés appartenir au genre « Le Monde ».
3. appliquer alors, de façon séparée, l'apprentissage en n-grammes sur les deux sous-corpus.
4. procéder de la même façon pour la phase de test.

Les résultats sont finalement légèrement inférieurs au traitement classique, ce qui ne permet pas de répondre positivement à la question posée. En fait, la réponse doit être plus nuancée :

1. le fait de particulariser le traitement selon le genre donne vraisemblablement des n-grammes plus précis que dans le cas général.
2. mais ceci est contrecarré par le fait que les deux demi-corpus d'apprentissage correspondant sont chacun, évidemment, deux fois plus petits que le corpus originel.

Il est donc probable qu'avec des corpus de plus en plus grands, compte tenu du fait que le gain obtenu par une augmentation de la taille des corpus est forcément limité asymptotiquement, nous finirions par effacer l'inconvénient mentionné en 2), et mettre en avant de façon enfin positive le gain obtenu en 1).

3 Approche TALN

L'approche TALN s'est focalisée sur la classification en genre (corpus 1). Elle vise à combiner analyse linguistique et apprentissage automatique. Nous reprenons, en les adaptant à la tâche fixée, les principes généraux déjà présentés lors des éditions 2006 et 2007 du défi DEFT (Widlöcher *et al.*, 2006) et (Vernier *et al.*, 2007), principalement :
– une phase de modélisation pour dégager des critères linguistiques génériques et pertinents pour la classification ;
– la réalisation d'une chaîne de traitements pour repérer ces indices et les marquer dans les corpus ;
– enfin, la mise au point d'un classifieur à partir du marquage textuel des indices (détaillé en section 5).

3.1 Modélisation linguistique

Deux genres – deux styles

Ce travail s'appuie sur une observation minutieuse du corpus et de sa nature pour procéder à une catégorisation linguistique des deux genres à discriminer. Elle repose sur l'hypothèse sous-jacente suivante : les deux genres sont révélateurs de deux styles, l'un journalistique (Le

Monde), l'autre encyclopédique (Wikipedia), qui vont utiliser leurs propres marques linguistiques. Le premier est plus expositif ou narratif, incluant des citations, et induit l'usage d'une palette assez large de formes langagières (temps verbaux variés, interrogation, négation, citation...).

À l'opposé, les textes de Wikipédia, par nature encyclopédique, nous semblent relever du style définitoire plus spécifique, que souligne l'usage fréquent de marques méta-linguistiques comme « être un », « désigne », « définit », *etc*. (*cf*. (Chaurand & Mazière, 1990) sur la notion d'acte définitoire).

Lors du traitement de ces marques, seules celles apparaissant en tête du texte, c'est-à-dire **en première phrase**, ont été considérées. Cette contrainte répond à l'hypothèse selon laquelle cette position joue un rôle priviligié dans l'organisation discursive et en particulier nous pensons que les marques discriminantes en matière de genre textuel sont celles qui sont situées dans cette position.

Nous passons maintenant en revue les différents indices retenus comme propriétés discriminantes et caractéristiques des deux genres.

Indices Wikipédia Ce type d'indice est en nombre restreint. Il concerne les verbes induisant un mode définitoire : forme verbale « être » suivie d'un déterminant, « désigner », « définir », « signifier », etc. Nous y avons ajouté les marques exprimant une naissance ou un décès : « né le »... Au moins l'un des deux indices apparaît[1] dans 6107 articles de Wikipédia contre 595 articles du Monde.

Indices pour Le Monde Les indices que nous considérons comme caractéristiques de ce genre sont relativement moins nombreux en terme d'occurrences, mais plus divers :
- les formes énonciatives (pronoms personnels des 1ère et 2ème personne, marques de citation) indiquant la présence d'un locuteur, cas typique de l'interview. L'une des formes est présente dans 1957 articles du Monde (respectivement 148 articles de Wikipédia) ;
- les temps verbaux (passé, futur, conditionnel présent) significatifs de la narration (versus le présent atemporel de la définition) : 4928 articles du Monde (resp 1035) ;
- les formes marquant une question, une exclamation ou une négation sont présentes dans 1276 articles du Monde (resp 158) ;
- les marques anaphoriques 1223 articles du Monde (resp 301) ;
- les formes de type « c'est un » et les formes impersonnelles (« il y a », « il est difficile de », *etc*.) : 687 (resp 126).

3.2 Réalisation informatique

Le repérage et le marquage des indices linguistiques a été réalisé à l'aide de la plate-forme LinguaStram (Bilhaut & Widlöcher, 2006) dédiée au TALN. Une chaîne de traitements séquentiels a été mise au point. Un premier traitement extrait la première phrase de chaque texte du corpus. Puis un composant morpho-syntaxique se charge de donner pour chaque mot sa catégorie syntaxique et sa forme lemmatisée[2]. Le coeur du traitement se compose de grammaires DCG (une

[1] Le calcul a été opéré sur le corpus d'apprentissage et sur la première phrase de chaque texte.

[2] nous utilisons ici l'outil bien connu TreeTagger

par indice) qui opèrent le marquage des indices linguistiques recherchés. En bout de chaîne, un dernier composant produit une matrice dans laquelle chaque ligne correspond à un texte du corpus et chaque colonne à un attribut étiqueté par un indice. La valeur de cet indice est le nombre d'occurrences de l'objet ou 0 si l'indice est absent de la première phrase.

3.3 Classification

Les règles de classification sont produites automatiquement à partir de la matrice (voir section 5). Le traitement sur la première phrase donne un F-score de 86%. On observe cependant dans la matrice un nombre élevé de lignes contenant une forte proportion de valeurs nulles (par exemple 15% des lignes n'ont qu'un attribut non nul). La prise en compte du texte dans sa totalité mérite d'être expérimentée. Cela nécessite de distinguer les critères à n'appliquer que sur la tête du texte d'une part, et d'autre part ceux à appliquer sur tout le texte. Par exemple, la forme « être + déterminant » a, comme on l'a vu, un nombre d'occurrences très faible au sein de la première phrase du Monde (relativement aux textes de Wikipédia). La probabilité d'apparition de cette forme est sans doute beaucoup plus importante dans les phrases suivantes ; le marquage de cet indice dans tout le texte ferait donc perdre à ce critère son pouvoir discrimant.

4 Utilisation de séquences discontinues de mots

Motivés par de précédentes expériences en recherche d'information (Doucet, 2004), nous avons voulu tester l'efficacité de l'utilisation de descripteurs séquentiels dans le cadre de la classification textuelle supervisée.

4.1 Motivation

La majorité des méthodes d'exploitation de contenus textuels adoptent un modèle de type « sac de mots », considérant implicitement les occurrences de mots comme des faits indépendants.

Si ces méthodes permettent d'obtenir de bons résultats, il semble toutefois raisonnable de penser que l'intégration d'une information supplémentaire, telle que la prise en compte des unités lexicales complexes, doit permettre d'améliorer la performance des systèmes de classification.

L'un des défauts d'une telle représentation documentaire est qu'elle ne tient pas compte des positions relatives des mots dans le document, ce qui semble intuitivement anormal, étant donné que des mots sont plus probablement liés s'ils apparaissent côte à côte plutôt qu'au début et à la fin d'un livre. En outre, l'occurrence simultanée de plusieurs mots induit souvent un sens différent de « l'addition » de la signification de ces mots pris individuellement. Par exemple, si l'on dit d'une personne qu'elle a « la main verte », on ne parle pas réellement de la couleur de sa main ; cela signifie que cette personne est douée pour le jardinage. Les expressions métaphoriques sont source de nombreux exemples de ce type (par exemple « avoir un chat dans la gorge »).

De nombreuses méthodes existent pour extraire des unités lexicales complexes. Elles reposent sur des critères statistiques, sur des critères syntaxiques, ou bien encore sur ces deux types de critères à la fois (application de méthodes statistiques après un filtrage syntaxique, par exemple).

Un défaut des méthodes statistiques pures est que, pour des raisons de complexité calculatoire, il est impossible en pratique de calculer une mesure d'association pour tous les ensembles de mots pouvant éventuellement former une unité lexicale.

Ainsi, les chercheurs ont toujours placé un certain nombre de restrictions lors de l'extraction d'unités multi-mots, en n'appliquant de mesures d'association qu'aux ensembles de mots respectant certaines contraintes, comme par exemple une longueur maximale, ou des positions d'occurrence rigides (positions relatives fixes ou restreintes par un nombre maximum de mots autorisés entre deux mots d'une unité lexicale).

La restriction la plus courante est d'imposer l'adjacence des termes (n-grammes), ce qui implique une perte d'information considérable. Par exemple si le mot « et » intervient entre deux autres mots, ils sont très certainement liés mais cela ne peut être pris en considération.

4.2 Séquences Fréquentes Maximales

Pour permettre l'extraction de séquences de mots sans restriction sur la longueur des séquences, ni sur la distance séparant leurs composants, nous proposons l'utilisation de séquences fréquentes maximales (SFM) (Ahonen-Myka & Doucet, 2005)).

Définition. Une séquence est dite fréquente si elle apparaît dans un nombre de phrases supérieur à un seuil de fréquence donné. Elle est maximale si on ne peut y insérer aucun autre mot sans pour autant faire descendre sa fréquence sous ce seuil.

Exemple. L'utilisation des SFMs permet d'appréhender le fait que la séquence « président Bush » apparaît dans chacun des 2 fragments textuels suivants, ce qui ne serait notamment pas le cas avec une méthode nécessitant des contraintes d'adjacence :

```
...Le président des Etats Unis George Bush...
...Président George W. Bush...
```

4.3 Application à DEFT'08

Apprentissage. À l'aide de l'étiquetage fourni dans les données d'apprentissage, nous avons construit une (sous-)collection de documents correspondant à chaque genre et chaque catégorie. Nous avons alors lancé l'extraction des SFMs dans chacune de ces collections, obtenant ainsi un ensemble de SFMs représentatives du genre et/ou de la catégorie correspondante. Afin de faciliter les comparaisons entre SFMs, nous avons finalement décomposé chaque SFM en chacune des paires de mots qui les composent. Nous avons alors utilisé le classifieur présenté en Section 5 afin d'extraire les règles à appliquer au corpus de test.

Test. Afin d'extraire nos séquences dans des proportions comparables, nous avons formé plusieurs sous-collections disjointes à l'aide de l'algorithme de clustering $k-means$. L'extraction des SFMs a alors été conduite parallèllement dans chacune des sous-collections. Cette approche

« diviser pour mieux régner » permet une extraction de séquences plus rapide et plus exhaustive (Doucet & Ahonen-Myka, 2006). Après normalisation, chaque document du corpus de test était associé à un ensemble de paires ordonnées issues de l'apprentissage. Les règles du classifieur fournissent la décision finale.

5 Classification surpervisée

Les données étiquettées par les approches TALN et extraction de séquences décrivent les textes du corpus à l'aide de descripteurs. Un descripteur particulier désigne l'appartenance à la classe. Pour répondre au défi, nous avons calculé sur ces données supervisées un classifieur à base de règles d'association, entraîné par 10-cross-validation.

5.1 Classification surpervisée à base d'association

Une règle de classification est une règle d'association (Agrawal *et al.*, 1996) concluant sur un attribut de classe. Si de telles règles peuvent être découvertes dans les données, l'intuition indique qu'elles peuvent aider à classer les textes supportant les descripteurs de la prémisse de la règle. CMAR (Li *et al.*, 2001) (Classification based on Multiple class-Association Rules) est une méthode populaire de classification à base de règles d'association. Les règles sont mesurées par un indice de corrélation fourni par un χ^2 normalisé. La redondance des règles est évitée en ne conservant que celles qui sont à prémisse minimale. Un nouvel exemple sera classé à l'issue d'un vote réalisé par toutes les règles qui s'appliquent, selon leur pondération.

Pour le défi, nous avons utilisé notre adaptation de CMAR, qui consiste à généraliser la forme des règles de classification en autorisant la présence de négation en prémisse (pour caractériser des objets contenant un motif mais en excluant un autre) ou en conclusion (pour des objets excluant une classe) (Rioult *et al.*, 2008).

5.2 Application aux données du défi

Les données de la méthode TALN (*cf.* section 3) ont été utilisées pour effectuer la classification en genre (Le Monde / Wikipedia) et les séquences (*cf.* section 4) ont classé les textes en catégories.

5.2.1 Données TALN - genre

Quelques expériences rapides ont permis de constater que la seule règle `être + déterminant` $\rightarrow$ `Wikipedia`, de confiance 73% (présente 4703 fois dans les 6398 textes de classe Wikipedia et uniquement 446 fois dans les 88825 textes du Monde), permet d'obtenir sur l'échantillon d'apprentissage un F-score moyen de 85.2%. Le F-score théorique d'un classifieur utilisant uniquement cette règle est de 86%, qui est le résultat obtenu sur les données de test (exécution 2).

Nous avons tenté d'utiliser des règles moins fréquentes, mais les performances étaient moins bonnes. Même en expérimentant des règles justifiées d'un point de vue de la sémantique, nous n'avons pu améliorer les performances de ce simple indice.

5.2.2 Données séquences - catégorie

Les données de séquences fréquentes extraites répertorient 11156 paires de mots et sont très volumineuses. Après un filtrage des plus fréquentes (présentes dans plus de 500 textes), nous obtenons une matrice de 15215 textes et 10414 descripteurs.

Les méthodes d'extraction de connaissance à base de motifs, telles que les règles d'associations utilisées ici pour calculer un classifieur, sont très coûteuses en temps de calcul. Dans le pire des cas, elles sont polynomiales en nombre d'objets et exponentielles en nombre d'attributs. La taille de la matrice à traiter ici est très pénalisante car elle contient beaucoup d'attributs.

Afin d'effectuer les calculs dans le temps imparti, nous avons dû restreindre les règles à des prémisses contenant une unique séquence. Malgré cette simplification, le F-score vérifié par 10-cross-validation était très bon : entre 92 et 93%.

Hélas, les séquences ont été obtenues sur l'intégralité de l'échantillon d'apprentissage, et nos expérimentations ont été fortement biaisées. D'ailleurs, les performances obtenues lors du défi sont très médiocres (67% pour la tâche 1, et 32% pour la tâche 2). Ces résultats ne remettent cependant pas en cause la qualité des séquences calculées ni la méthode de décision utilisée. Si le calcul des séquences avait été intégré au processus de validation croisée, le classifieur aurait révélé de nettement moins bonnes performances et nous aurions alors cherché à les améliorer. Nous proposons, lors de la réunion des participants, de donner des scores plus représentatifs du potentiel de notre méthode.

6 Analyse - perspectives

La table 1 indique les F-scores obtenus à l'aide de nos différentes méthodes. L'exécution 1 correspond à la méthode n-grammes (section 2), l'exécution 2 correspond à la méthode TALN et extraction de séquences, puis décision par classification par règles d'association (sections 3 à 5). L'exécution 3 utilise la méthode n-grammes avec un paramétrage différent pour les tâches 1 genre et 2 catégorie, et la méthode séquences pour les catégories de la tâche 1.

	exécution 1 (n-grammes)	exécution 2 TALN + séquences	exécution 3 mix 3 méthodes	moyenne participants
tâche 1 genre	96.4	85.6	96.4	95.9
tâche 1 catégorie	84.9	67.2	67.2	82.6
tâche 2 catégorie	83.7	32.8	81.5	81.1

TAB. 1 – Récapitulatif des F-scores (en %) obtenus avec les différentes méthodes.

Une analyse des résultats montre que la méthode n-grammes donne les meilleurs résultats. Ces résultats se situent dans la moyenne de l'ensemble des participants à DEFT. Par ailleurs, ils restent homogènes sur les trois tâches, contrairement à l'an dernier. La méthode des n-grammes tire ainsi parti de la taille importante des corpus nécessaire à l'établissement de bonnes performances. En revanche, la connaissance du genre ne permet pas d'améliorer les résultats pour les raisons évoquées en section 2.4.2. Enfin, il est à noter que cette approche est peu coûteuse en ressources machine et ne demande que quelques minutes pour l'apprentissage et le test.

Les résultats obtenus par la méthode TALN sont honorables et valident l'intérêt de cette approche tant pour confirmer des hypothèses linguistiques, que pour son originalité. Originalité

qui tient à son aspect « sémantique » dans la mesure où l'interprétation est privilégiée à la forme. Si le choix des critères est lié à la tâche, les critères sont en eux-mêmes génériques et indépendants du corpus et de la tâche. Une amélioration possible et intéressante consisterait en une démarche plus interactive avec l'apprentissage. En effet, le choix des critères est effectué manuellement et pour leur pertinence linguistique : une phase préliminaire qui étudierait et analyserait les n-grammes et / ou les segments discontinus révélateurs d'une classe pourrait faire émerger des critères supplémentaires.

Les performances de la méthode utilisant les séquences maximales de mots sont décevantes. Ainsi que nous l'avons évoqué à la section 5.2.2, l'apprentissage a été effectué dans de mauvaises conditions, ce qui a provoqué un biais important pour les performances attendues. Nous souhaitons retravailler sur cette méthode afin d'obtenir des résultats plus représentatifs.

Références

AGRAWAL R., MANNILA H., SRIKANT R., TOIVONEN H. & VERKAMO A. (1996). Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, p. 307–328.

AHONEN-MYKA H. & DOUCET A. (2005). Data mining meets collocations discovery. In *Inquiries into Words, Constraints and Contexts, Festschrift for Kimmo Koskenniemi*, p. 194–203 : CSLI Studies in Computational Linguistics. .

BILHAUT F. & WIDLÖCHER A. (2006). LinguaStream : An Integrated Environment for Computational Linguistics Experimentation. In *11th Conference of the European Chapter of the Association of Computational Linguistics (EACL'06)*, p. 95–98.

CHAURAND J. & MAZIÈRE F. (1990). *La définition*. Larousse, collection Langue et Langage.

DOUCET A. (2004). Utilisation de séquences fréquentes maximales en recherche d'information. In *Proceedings of the 7th International Conference on the Statistical Analysis of Textual Data (JADT-2004)*, p. 334–345, Louvain-La-Neuve, Belgium : JADT-2004.

DOUCET A. & AHONEN-MYKA H. (2006). Fast extraction of discontiguous sequences in text : a new approach based on maximal frequent sequences. In *Proceedings of IS-LTC 2006, Information Society - Language Technologies Conference*, p. 186–191, Ljubljana, Slovenia.

LI W., HAN J. & PEI J. (2001). Cmar : Accurate and efficient classification based on multiple class-association rules. In *IEEE International Conference on Data Mining (ICDM'01), San Jose, USA*.

RIOULT F., ZANUTTINI B. & CRÉMILLEUX B. (2008). Apport de la négation pour la classification supervisée à l'aide d'associations. In *Conférence francophone sur l'apprentissage automatique*.

STUBBS M. & BARTH I. (2003). Using recurrent phrases as text-type discriminators : A quantitative method and some findings. *Functions of Language*, **10**, 61–104(44).

VERNIER M., MATHET Y., RIOULT F., CHARNOIS T., FERRARI S. & LEGALLOIS D. (2007). Classification de textes d'opinions : une approche mixte n-grammes et sémantique. In *3ème DÉfi Fouille de Textes (DEFT'07) associé à la plateforme AFIA'07*, p. 99–109.

WIDLÖCHER A., BILHAUT F., HERNANDEZ N., RIOULT F., CHARNOIS T., FERRARI S. & ENJALBERT P. (2006). Une approche hybride de la segmentation thématique : collaboration du traitement automatique des langues et de la fouille de texte. In *Deuxième DÉfi de Fouille de Textes (DEFT'06), Semaine du Document Numérique (SDN'2006)*.

Pré-traitements classiques ou par analyse distributionnelle : application aux méthodes de classification automatique déployées pour DEFT08

Eric Charton[1] Nathalie Camelin[1] Rodrigo Acuna-Agost[1] Pierre Gotab[1]
Remi Lavalley[1] Remy Kessler[1] Silvia Fernandez[1]
(1) LIA - Université d'Avignon, BP1228 84911 Avignon cedex 09 France
{eric.charton, nathalie.camelin, rodrigo.acuna-agost, remy.kessler, silvia.fernandez}@univ-avignon.fr
{pierre.gotab, remi.lavalley}@etd.univ-avignon.fr

Résumé. Nous décrivons dans cet article un ensemble de méthodes de classification automatique mises en œuvre dans le cadre de la campagne DEFT08. Notre approche a d'abord consisté à évaluer les performances des systèmes état de l'art de l'apprentissage automatique (classifieurs SVM, AdaBoost, probabilistes et cosine). Nous avons ensuite cherché à améliorer les performances de ces classifieurs en élaborant une méthode de construction de classes discriminantes par analyse distributionnelle. Cette méthode de normalisation des classes, appliquée aux classifieurs SVM et probabilistes, a amélioré significativement nos résultats. Deux méthodes de fusion des divers résultats obtenus par les classifieurs ont également été testées.

Abstract. In this paper we describe a set of automatic classification methods applied to the DEFT08 campaign. First, we evaluated and compared some of state-of-the-art classifiers like SVM, AdaBoost, probabilistic-based classifiers, and cosine-based classifiers. Subsequently, we developed a method to normalize classes using a distributional analysis of the text with the aim of improving the performance. Lastly, some additional results were obtained by two merging methods that showed to increase the scores of the individual classifiers

Mots-clés : Méthodes de Classification Automatique, SVM, AdaBoost, Classifieur Bayésien Naïf, Analyse distributionnelle .

Keywords: Automatic Classification Methods, SVM, AdaBoost, Naïve Bayesian Classifier, Distributionnal analysis.

1 Introduction

La campagne DEFT 2008 (DÉfi de Fouilles de Texte) a pour sujet la classification en genre et en thème de textes. Au-delà de la reconnaissance de la catégorie du document, la reconnaissance de son genre est utile pour guider l'utilisation ultérieure qui peut en être faite (*e.g.* orientation de courriels, veille scientifique, ...). Mais reconnaître à la fois le thème et le genre d'un document relève-t-il d'une problématique particulière ?

Le défi propose deux tâches distinctes qui permettent de réfléchir à ce problème. L'une confronte classification en genre et en thème tandis que l'autre se focalise uniquement sur la classification en thème.

2 Analyse des corpus

2.1 Corpus relatif à la tâche 1 : Classification en genre et en thème

Ce corpus, noté *CORPUS1*, est étiqueté en thème selon quatre classes : économie (*ECO*), art (*ART*), télévision (*TEL*) et sport (*SPO*). Chaque document est également annoté selon qu'il provient du journal Le Monde (*LM*) ou de Wikipedia (*W*). La répartition des volumes de documents pour chaque classe dans le corpus d'apprentissage (*APP1*) et d'évaluation (*EVAL1*) est indiquée dans le tableau 1.

CORPUS1	*ECO*	*TEL*	*ART*	*SPO*	*LM*	*W*	Échantillons
APP1	30.41%	8.88%	37.88%	22.82%	57.97%	42.02%	15223
EVAL1	29.11%	12,75%	36.27%	21.84%	53.63%	46.36%	10596

TABLE 1 – Répartition des volumes de documents de *CORPUS1* pour chaque classe

On observe sur le corpus d'apprentissage de cette tâche, l'existence de classes dont la répartition est très disparate. Ce déséquilibre est conjugué à des imbrications inter-classes importantes [1]. L'imbrication est particulièrement marquée dans le cas des classes *TEL* et *ART*. La confusion entre ces deux classes est conjuguée à une faible représentation de *TEL* par rapport à *ART* (respectivement 8.88% et 37.88% des documents).

2.2 Corpus relatif à la tâche 2 : Classification en thème

Le corpus de la tâche 2, noté *CORPUS2*, est composé de documents également issus des deux sources Le Monde et Wikipédia mais est annoté uniquement en thèmes selon cinq classes : société (*SOC*), actualité ou information française (*FRA*), ou internationale (*INT*), sciences (*SCI*) et littérature (classe *LIV*). La répartition des classes dans *CORPUS2* est donnée dans le tableau 2 pour les corpus d'apprentissage (*APP2*) et d'évaluation (*EVAL2*).

CORPUS2	*SOC*	*FRA*	*INT*	*LIV*	*SCI*	Échantillons
APP2	16.04%	14.12%	22.52%	19.43%	27.87%	23550
EVAL2	16.03%	14.12%	22.53%	19,42%	27.87%	15693

TABLE 2 – Répartition des volumes de documents de *CORPUS2* pour chaque classe

Il apparaît que contrairement à *CORPUS1*, *CORPUS2* repose sur une répartition des classes plus homogène. On note, à nouveau, une légère sous-représentation d'une classe (*FRA*) associée à une sur-représentation d'une autre (*SCI*). Les trois classes *FRA*, *SCI* et *SOC* ont par ailleurs des intersections fortes, pouvant conduire à une faiblesse de rappel sur les classes *FRA* et *SOC* et un manque de précision très marqué sur *SOC*.

1. Les distances entre classes ont été mesurées par similarité cosine

3 Présentation théorique des différentes méthodes de classification automatique appliquées à DEFT08

Les Machines à supports vectoriels Les machines à supports vectoriels (Support Vector Machine - *SVM*) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression (Vapnik, 1995). La première idée clé est la notion de marge maximale, *i.e* la distance entre la frontière de séparation et les échantillons les plus proches est maximisée. L'autre idée maîtresse des SVM est de transformer, grâce à une fonction noyau, l'espace de représentation des données d'entrées en un espace de plus grande dimension, dans lequel il est probable qu'il existe un séparateur linéaire. Les deux systèmes implémentés ont été développés à partir de l'outil LIBSVM (Fan et al., 2005).

Le boosting Le principe du boosting réside dans la combinaison de plusieurs hypothèses dites *faibles* afin d'améliorer la précision des règles de classification. Chaque hypothèse faible est obtenue par itérations successives de l'algorithme de boosting. À chacune des exécutions de l'algorithme, une nouvelle distribution de probabilité *a priori* sur les exemples d'apprentissage est calculée en fonction des résultats de l'algorithme à l'exécution précédente. Introduit par (Freund & Schapire, 1995), l'algorithme standard s'appelle AdaBoost (adaptive boosting). Deux implémentations particulières de l'algorithme Adaboost ont été utilisées : BoosTexter et icsiboost.

Classification Bayésienne Naïve Le classifieur Bayésien Naïf est un classifieur probabiliste simple appliquant le théorème de Bayes. La formule générique de ce classifieur est la suivante : la probabilité qu'un document D contenant des mots m appartienne à une classe k est égale à :

$$P(k|D) = \frac{p(D|k)\, p(k)}{p(D)} \tag{1}$$

En règle générale, les classifieurs Bayésiens Naïfs appliquent la méthode de l'estimateur de maximum de vraisemblance pour décider de l'attribution d'une classe, comme suit $P(k|D) = p(m_1, m_2, ..., m_n|k) = \prod_{i=1}^{n} p(m_i|k)$. Un des avantages du classifieur Bayésien Naïf réside dans sa capacité d'estimation des paramètres avec peu de données d'apprentissage.

Mesure de similarité Cosine La mesure de similarité par la formule Cosine permet de calculer le cosinus de l'angle entre un article et un vecteur représentant de chacune des classes (Salton & McGill, 1983). Plus celui-ci est élevé, plus l'article est proche de la classe correspondante. La formule classique a été adaptée à la tâche de classification en se basant sur le critère d'impureté de Gini $Gin(i) = \sum (P_k)^2$ afin d'en augmenter le pouvoir discriminant :

$$s(n,k) = \frac{\sum (W_{in} \times Wik \times Gin(i))}{\sqrt{\sum W_{ik}^2}} \tag{2}$$

où W_{ik} représente le poids du mot *i* dans le modèle d'apprentissage de la classe *k* et W_{in} le poids du mot *i* dans l'article *n*.

Énergie textuelle L'énergie textuelle correspond au modèle magnétique d'Ising déjà utilisé avec succès dans des tâches de traitement de la langue naturelle (Fernandez et al., 2008). Dans ce modèle, k sous-ensembles correspondant aux documents d'apprentissage pour chaque classe sont considérés comme k matériaux différents. Chaque matériau est composé de n atomes (des mots différents) qui interagissent différemment pour chaque classe. Ces interactions entre mots sont données par la matrice $J_k = \{j_{i,j}\}$ où chaque élément $j_{i,j} = \sum_k tf_i^k tf_j^k$ correspond à la co-occurrence de ces mots dans le sous-ensemble k. Les k matrices $J_1, J_2, ..., J_k$ seront utilisées pour classer les documents de test Dt. Pour chaque Dt, l'énergie des mots est calculée selon :

$$E_k = S \times J_k \times S^T \tag{3}$$

où S est la représentation vectorielle du document à classer. Plus la valeur de E_k est élevée, plus il est probable que ce document soit un échantillon du matériau k.

4 Implémentation des différents systèmes de classification

4.1 Protocole expérimental

Pour chaque tâche, la méthode de validation des classifieurs est une validation croisée classique sur N partitions du corpus d'apprentissage. Chacun des corpus d'apprentissage de *APP1* et *APP2* a été divisé en N parties égales qui constituent N sous-corpus. Nous utilisons $N-1$ sous-corpus en tant que données d'apprentissage et le Nième corpus est réservé au test. N jeux de tests *tournants* sont ainsi créés, le score final d'une classification étant ensuite calculé par la moyenne de tous les scores obtenus sur chacun des N corpus de test.

4.2 Méthodes de pré-traitements de texte

Les méthodes de classification (classifieurs) ont été appliquées avec des classes de mots issues des corpus d'apprentissage, au besoin normalisés selon plusieurs techniques dont l'efficacité est attestée par l'état de l'art (*e.g.* : étiquetage morpho-syntaxique, lemmatisation, filtrage des textes, n-grammes). Nous avons complété ces techniques par une méthode de filtrage des mots d'après leurs distributions dans les classes.

4.2.1 Pré-traitements classiques

Dans un premier temps, le texte à classer a été filtré par plusieurs opérations : suppression de toutes marques typographiques, minusculisation.

Dans un second temps, plusieurs représentations de la phrase ont été proposées :
– ensemble des mots ;
– représentation morpho-syntaxique : chaque mot de la phrase est représenté par sa forme morpho-syntaxique (Part Of Speech - *POS*) ;
– représentation canonique : chaque mot de la phrase est représenté par son lemme.

Ces deux dernières représentations différentes pour un mot sont obtenues grâce à l'utilitaire LIA-TAGG [2]. Chaque participant de l'équipe a fait ses propres choix quant à la représentation

[2] 2. `http://www.lia.univ-avignon.fr/chercheurs/bechet/download\_fred.html`

du texte qui sont indiqués dans chacune des descriptions des systèmes.

Diverses techniques d'agglutination de groupes de mots par n-grammes ont également été proposées. Nous avons retenu pour la majorité des classifieurs des classes de mots composées de tri-grammes qui permettent une amélioration des performances.

Un antidico [3] contenant notamment : un ensemble de mots fonctionnels (*e.g.* : être, avoir, pouvoir, falloir) ; des expressions courantes (*e.g.* : c'est-à-dire, chacun de) ; des chiffres (numériques et/ou textuels) ; des symboles comme <$>, <#>, <*>, a également été utilisé dans certains systèmes pour filtrer le texte.

4.2.2 L'hypothèse distributionnelle et les distributions de Zipf-Mandelbrot

Nous introduisons dans cette tâche une méthode de normalisation de textes originale, exploitant les propriétés de la distribution des mots dans un corpus pour élaborer des classes de détection. Des auteurs ont déjà souligné les performances obtenues dans une tâche de classification de données textuelles, sur des classifieurs SVM associés à des classes de mots extraites d'après des distributions de Zipf-Mandelbrot (Leopold & Kindermann, 2002).

La loi de Zipf-Mandelbrot est une distribution de probabilité discrète. Elle est connue également sous le nom de loi de Pareto-Zipf (Zipf, 1949), la loi de Pareto étant la forme continue de la loi de Zipf. La loi de Zipf prédit que si dans un texte de longueur N on range les mots dans l'ordre de leur fréquence d'apparition, alors la fréquence $f(r)$ du mot de rang r est approximativement de forme : $f(r) = \frac{K}{r}$

Considérons des objets textuels O, mot ou un groupe de mots (n-grammes) représentant les occurrences rencontrées dans le corpus. Si nous postulons que les fréquences des O dans un corpus suivent approximativement des distributions de Zipf-Mandelbrot, nous pouvons établir une table de fréquences des O pour un corpus de textes tels que ceux présentés dans ce défi. Dans cette table, nous trouverons en premier lieu dans les plus hauts rangs, des mots outils (ou des objets O contenant des mots outils). Dans le cadre applicatif de DEFT08, les mots de plus haute fréquence dans *CORPUS1* et *CORPUS2* sont sans surprise les mots outils : *de, la, le et, des, les, en, du, un, est, dans* .... On observe à la suite de cette liste de mots outils, l'apparition de premiers mots plus caractéristiques d'une classe (*e.g.* sur *APP1* : "france" en 32 ème rang dans la classe *TEL*, "équipe" au rang 39 de *SPO*, "film" au 45 ème rang de *ART*).

Il est déduit de ces observations qu'il est possible, en exploitant plusieurs distributions de Zipf-Mandelbrot modélisant la distribution de mots de l'intégralité du corpus et celle de chaque sous corpus, de supprimer automatiquement les mots de probabilité proche (qui correspondent aux mots outils ou peu discriminants) et de ne conserver que les mots discriminants pour une classe donnée.

Cette méthode est schématisée par la figure 1. Sur la gauche, un exemple de mot outils dont la probabilité d'apparition dans chaque sous corpus est très proche de celle observée dans la totalité du corpus : ces mots sont supprimés dans toutes les classes de détection. Au centre, un exemple de mot discriminant ("équipe" de la classe SPO) dont la probabilité d'apparition est très supérieure à celle observée par la distribution de Zipf modélisée d'après l'ensemble du corpus. On choisira ici de conserver ce mot uniquement dans la classe de détection SPO. On utilisera pour procéder à la sélection un intervalle de confiance ou le critère d'impureté de Gini.

3. http://www.up.univ-mrs.fr/~veronis/data/antidico.txt

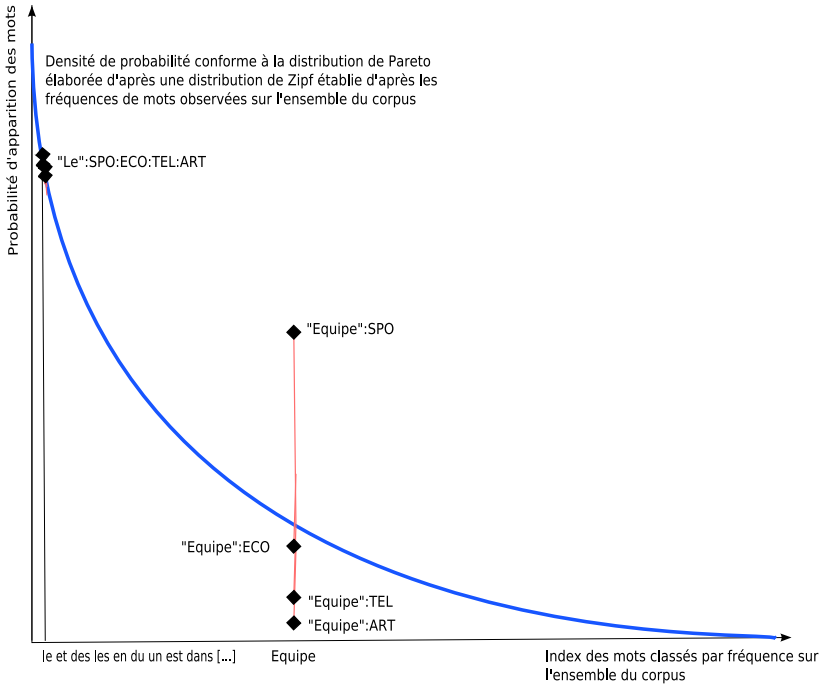

FIGURE 1 – Représentation schématique d'une distribution de Zipf et des variations des probabilités d'apparition des mots selon les classes

4.3 Présentation des systèmes

Sept systèmes ont été implémentés dans le cadre de ce défi. Nous les décrivons ci-dessous.

SVM_Baseline Le classifieur SVM_baseline est conçu comme suit : chaque document est transformé en vecteur de mots préalablement filtrés selon l'antidico (*c.f.* sous-section 4.2.1). Les verbes fléchis sont ramenés à leur racine, ainsi que les mots pluriels et/ou féminins au masculin singulier. Les vecteurs sont ensuite soumis au classifieur pour la phase d'apprentissage des SVM. Le noyau choisi a été le noyau linéaire, celui-ci s'étant avéré le plus performant, ceci s'expliquant par la taille importante de notre lexique.

SVM-Extended Ce système SVM utilise les distributions de mots préalablement filtrés. Le filtrage consiste à ne conserver que les mots sous une forme majuscule. En premier lieu, deux séries d'index des n-grammes et de leur fréquence sont générées : un premier index pour l'ensemble du corpus et une seconde série d'index correspondant à chaque classe. Dans un second temps, tous les n-grammes apparaissant dans plusieurs catégories avec une probabilité proche (l'intervalle de variabilité est réglable) sont supprimés des index. Cette phase permet de supprimer automatiquement les mots outils et non discriminants. Selon le même procédé, ne sont conservés dans les index de chaque catégorie que les n-grammes caractéristiques de cette catégorie. Il est ensuite procédé à la construction des vecteurs caractéristiques de chaque classe. Dans le cadre applicatif décrit ici, les mots d'occurrence 1 ont été supprimés, ces derniers augmentant considérablement le temps de calcul du SVM, sans améliorer notablement ses performances.

Naïve_Bayes_extended Le classifieur Bayésien Naïf utilise k classes de n-grammes, correspondant aux k thèmes. Le contenu des classes est normalisé selon ce qui a été décrit dans la sous-section 4.2.2. Les n-grammes dont la probabilité est identique dans les k classes sont supprimés (mots outils ou non discriminants). Un intervalle de variabilité α est utilisé pour choisir les n-grammes considérés comme discriminants pour une classe donnée. Le n-gramme qui sort de l'intervalle de variabilité est versé dans la classe de détection où il est le plus prégnant et supprimé dans les autres classes. Aucune limite sur le nombre d'occurrences d'un n-gramme justifiant de le maintenir dans une classe de détection n'a été définie. Les classes sont donc de grande taille. Les documents sont ensuite attribués à une classe par un test de maximum de vraisemblance entre les n-grammes contenus dans un document et ceux contenus dans les k classes.

BoosTexter Dans le cadre de cette implémentation de *BoosTexter*[4], le classifieur faible est un arbre de décision binaire à un niveau qui teste la présence/l'absence d'un n-gramme dans la phrase et en déduit sa classification, l'hypothèse faible. Les éléments choisis par les classifieurs faibles sont des n-grammes avec $1 \leq n \leq N$ et $N = 3$ déterminé par l'utilisateur. Le nombre de tours d'itération de l'algorithme est de $T = 1500$, empiriquement un bon compromis entre performances et temps de calcul. La phrase est représentée par sa suite de lemmes (*c.f.* sous-section 4.2.1) qui permet une légère augmentation des performances par rapport à l'utilisation des mots.

icsiboost *icsiboost* est une version open-source de BoosTexter développée par le laboratoire ICSI[5]. Le paramètre $N = 3$ a été choisi. Concernant le paramètre T définissant le nombre de tours de l'algorithme, les expériences ont montré qu'un optimum était atteint aux alentours des 2500 tours. Les composants choisis pour représenter la phrase sont à la fois les lemmes et les POS (*c.f.* sous-section 4.2.1). L'utilisation de cette représentation par rapport à une représentation en mots permet une augmentation des performances.

Cosine_Discriminant L'équation 2 a été implémentée. Les unigrammes de lemmes sont considérés mais sans minusculisation préalable. De plus, plusieurs filtres sont appliqués : les signes de ponctuations et les déterminants sont ôtés ainsi que l'ensemble des mots de l'antidico (*c.f.* sous-section 4.2.1).

Enertex Enertex est le classifieur basé sur le concept d'énergie textuelle présenté dans la section 3. Il a été appliqué sur le texte après le même pré-traitement que celui indiqué dans le système SVM-Extended.

4. `http://www.research.att.com/sw/download/`
5. `http://www.icsi.berkeley.edu/`

4.4 Stratégies de fusion

4.4.1 Stratégie de fusion par vote ternaire majoritaire

La fusion par vote majoritaire que nous avons déployée est triviale : elle consiste à confronter les propositions des trois meilleurs classifieurs (SVM_extended, N_Bayes_extended, icsiboost) pour chaque document. Si une majorité l'emporte (2/3 ou 3/3), la classe majoritaire est choisie, dans le cas contraire, la stratégie de fusion se replie sur le système le plus performant (en l'occurrence SVM_extended sur tâche1-catégorie et tâche2-catégorie et icsiboost sur tâche1-genre). Les Fscores obtenus sur APP1 et APP2 sont présentés dans le tableau 3.

systèmes	tâche1-genre	tâche1-catégorie	tâche2-catégorie
SVM_extended	0.9594	0.9150	0.8445
N_Bayes_extended	0.9353	0.8629	0.8316
icsiboost	0.9858	0.9051	0.8409
Fusion ternaire	0.9870	0.9192	0.8676

TABLE 3 – Fscores obtenus par la stratégie de fusion par vote ternaire majoritaire

4.4.2 Stratégie de fusion probabiliste avec BoosTexter

Cette stratégie de fusion permet de prendre en compte le résultat fourni par les systèmes SVM_extended, N_Bayes_extended, icsiboost et BoosTexter. Le principe d'utilisation de BoosTexter est de considérer que la phrase à classer est caractérisée non plus par son ensemble de mots, lemmes ou POS, mais par l'ensemble des résultats obtenus par les classifieurs pour chacune des classes cibles. Par exemple, la phrase 1 :1 est représentée, pour la tâche1-catégorie par :

SPO_{N_Bayes} 0.98140 ECO_{N_Bayes} 0.00431 TEL_{N_Bayes} 0.00984 ART_{N_Bayes} 0.00444, ART_{BoosT} 0.4051887 ECO_{BoosT} 0.406487 SPO_{BoosT} 0.7468423 TEL_{BoosT} 0.3704997, ART_{isci} 0.3422 ECO_{isci} 0.2983 SPO_{isci} 0.7131 TEL_{isci} 0.3311, SPO_{SVM} 1.00

C'est une manière naïve et simple à implémenter qui permet de choisir automatiquement l'étiquette finale en fonction des résultats de chacun des classifieurs. Les Fscores obtenus sur les corpus APP1 et APP2 sont présentés dans le tableau 4.

systèmes	tâche1-genre	tâche1-catégorie	tâche2-catégorie
SVM_extended	0.9594	0.9150	0.8445
N_Bayes_extended	0.9353	0.8629	0.8316
icsiboost	0.9858	0.9051	0.8409
BoosTexter	0.9804	0.8965	0.8316
Fusion probabiliste	0.9880	0.9292	0.8599

TABLE 4 – Fscores obtenus par la stratégie de fusion probabiliste avec BoosTexter

Fscores APP	tâche1-catégorie	tâche1-genre	tâche2-catégorie
SVM_baseline	0.8391	0.93	0.78
SVM_extended	**0.9150**	0.9594	**0.8445**
N_Bayes_extended	0.8629	0.9353	0.8271
BoosTexter	0.8958	**0.9869**	0.8316
icsiboost	0.9051	**0.9858**	**0.8409**
Cosine_Discriminant	0.8508	0.9222	0.8244
Enertex	0.8328	0.8390	0.7561

TABLE 5 – Performances des différents systèmes lors de la phase d'apprentissage

5 Expériences et résultats

5.1 Phase d'apprentissage

L'ensemble des Fscores obtenus par chaque système sur les trois tâches, lors de la validation croisée de la phase d'apprentissage, est présenté dans le tableau 5.

5.2 Mode évaluation

Exécution 1 : Fusion par vote ternaire Les résultats par fusion par vote ternaire (TAB.6) sont les meilleurs obtenus sur les trois exécutions. On observe un maintien des performances sur tâche1-genre et tâche2-catégorie et une baisse de performance assez marquée sur la tâche1-catégorie, explicable par la différence de répartition des classes entre les corpus APP1 et EVAL1.

Exécution 1	Précision	Rappel	Fscore
tâche1-genre	0.9795	0.9800	0.9798
tâche1-catégorie	0.9082	0.8448	0.8754
tâche2-catégorie	0.8814	0.8759	0.8786

TABLE 6 – Évaluation : Performances de l'exécution1

Exécution 2 : Fusion probabiliste Cette soumission a étrangement obtenu une mauvaise performance sur la tâche2-catégorie (TAB.7). Nous réaliserons prochainement un dépouillement de ces résultats pour en éclaircir les causes.

Exécution 2	Précision	Rappel	Fscore
tâche1-genre	0.9584	0.9600	0.9592
tâche1-catégorie	0.7919	0.8267	0.8089
tâche2-catégorie	0.8124	0.558393	0.6618

TABLE 7 – Évaluation : Performances de l'exécution2

Exécution 3	Précision	Rappel	Fscore
tâche1-genre	0.9795	0.9800	0.9798
tâche1-catégorie	0.8972	0.8006	0.8442
tâche2-catégorie	0.8553	0.8497	0.8525

TABLE 8 – Évaluation : Performances de l'exécution3

Exécution 3 : les meilleurs systèmes La tâche1-catégorie est réalisée par SVM_Extended, les deux autres tâches tâche1-genre et tâche2-catégorie sont réalisées avec icsiboost. Les résultats obtenus (TAB.8) confirment les bonnes performances des classifieurs SVM_Extended et icsiboost obtenues lors de la phase d'apprentissage. On note néanmoins une baisse de performances sur tâche1-catégorie.

6 Conclusions et perspectives

Nous avons appliqué plusieurs méthodes diversifiées de classification automatique à la problématique posée par DEFT08. Ces méthodes, prises séparément, présentent des performances proches. Nous avons enrichi ces méthodes par une fusion de leurs résultats. La plus performante étant celle par vote ternaire. Notre système complet est simple à régler, performant et exploite une méthode de préparation de classe originale à base d'analyse distributionnelle. Notre postulat de départ consistait à envisager la classification en genre et thème comme deux tâches indépendantes. Cette idée initiale s'est trouvée confortée par les résultats obtenus lors des expériences. Nous déduisons de ces résultats que les grandes différences de forme et de méthode de rédaction, utilisées pour le journal Le Monde et l'encyclopédie Wikipédia, sont suffisantes pour entraîner des classifieurs *état de l'art* et donner aux classes qu'ils utilisent un caractère suffisamment discriminant. Nous considérons que les méthodes présentées ici possèdent encore un bon potentiel d'amélioration que nous allons nous attacher à étudier dans les mois qui vont suivre.

Références

FAN R.-E., CHEN P.-H. & LIN C.-J. (2005). Towards a Hybrid Abstract Generation System, Working set selection using the second order information for training SVM. In NIPS 2005.

FERNANDEZ S., SANJUAN E. & TORRES-MORENO J. M. (2008). Enertex : un système basé sur l'énergie textuelle. In TALN2008.

FREUND Y. & SCHAPIRE R. E. (1995). A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting. In EuroCOLT : Springer.

LEOPOLD E. & KINDERMANN J. (2002). Text categorization with support vector machines. how to represent texts in input space. Machine Learning.

SALTON G. & MCGILL M. (1983). Introduction to modern information retrieval. Computer Science Series McGraw Hill Publishing Company.

VAPNIK V. N. (1995). The nature of statistical learning theory. Springer-Verlag New York, Inc.

ZIPF G. K. (1949). Human behavior and the principle of least-effort. Addison-Wesley.

Classification de textes en domaines et en genres en combinant morphosyntaxe et lexique

Cleuziou, Guillaume [(1)], Poudat, Céline [(2)]

(1) LIFO – Université d'Orléans, B.P. 6759 F-45067 Orléans Cedex 2
guillaume.cleuziou@univ-orleans.fr
(2) ERTIM – INALCO, 2, rue de Lille, 75343 Paris Cedex 07
cpoudat@gmail.com

Résumé Nous présentons dans cet article le bilan de notre participation au 4e DÉfi Fouille de Textes 2008. L'étude porte sur la problématique de la classification textuelle en domaines et en genres qui représente un enjeu pour la Recherche d'Information (RI). Sa mise en œuvre nécessite notamment la sélection d'un ensemble de descripteurs adéquats. On considère généralement que les domaines sont corrélés au niveau du contenu (mots, termes, etc.) tandis que les genres sont discriminés au niveau morphosyntaxique. Malgré les bons résultats obtenus par ces choix méthodologiques, peu de travaux ont cherché à mesurer l'impact et la complémentarité des deux niveaux de description pour la classification. Le cadre pratique de ce défi permettra de compléter les premiers postulats formulés sur ce travail.

Abstract In this paper we present our contribution to the 4th *DÉfi Fouille de Textes* 2008. The challenge deals with topic and genre texts categorization which is of real interest for Information Retrieval researches. This task requires notably to select appropriate descriptors . In most categorization works, topics (or domains) are generally correlated to the content level (words, terms, bag of words, etc.) and genres to the morphosyntactic one. However, few studies have assessed the impact and the complementarity of the two description levels on genre and domain categorization. The practical framework of the DEFT challenge allows to complement the very first postulates we expressed on this research topic.

Mots-clés : Recherche d'Information, genre, domaine, classification, lexique, morphosyntaxe.

Keywords: Information Retrieval, genre, domain, classification, lexicon, morphosyntax.

1 Introduction

Les classifications textuelles en domaines et en genres représentent un enjeu pour la Recherche d'Information (RI), et leur mise en œuvre nécessite la sélection d'un ensemble de descripteurs adéquats. Dans les faits, domaines et genres sont généralement associés à des niveaux linguistiques différents. Quand il s'agit de classification thématique ou domaniale, les textes sont souvent réduits à l'état de "sacs de mots". Chaque document est alors décrit par le vocabulaire présent dans le corpus. Étant donné la taille de ce vocabulaire, une étape de réduction de l'espace de description est indispensable : sélection d'attributs par des mesures d'intérêt (mesure d'Information Mutuelle, Gain d'Information et mesure du chi2, etc.), reparamétrage de l'espace (LSI, pLSI) ou regroupement d'attributs. Ces formalismes d'indexation permettent d'obtenir des classifieurs performants, atteignant jusqu'à 90% de précision sur grands corpus (Hofmann, 1999, Dhillon et al., 2003). De la même manière, les classifications en genres à partir d'un jeu de variables morphosyntaxiques robuste sont à même d'obtenir de très bons résultats en matières de validation de typologies textuelles (Karlgren et Cutting, 1994, Kessler et al., 1997, Malrieu et Rastier, 2001).

Nous avons toutefois pu constater que la plupart des travaux recensés effectuent de la classification domaniale sur corpus génériquement homogènes (e.g. Reuters ou Newsgroup), et de la classification générique sur corpus discursivement[1] hétérogènes, ce qui augmente le pouvoir classificatoire des variables employées mais limite l'utilisation conjointe et l'évaluation de la portée des deux niveaux descriptifs. DEFT'08 propose ainsi un cadre et un corpus de travail audacieux qui cherche à dépasser cette opposition, bien que les deux « genres » contrastés relèvent de deux discours différents : le discours encyclopédique et le discours journalistique.

Nos travaux précédents (Poudat, Cleuziou, 2003, Poudat et al., 2006 et Cleuziou, Poudat, 2007) se sont attachés à la classification textuelle en domaines et genres au sein du discours scientifique. Dans cette perspective, nous avions utilisé et combiné deux types de traits : (i) des descripteurs lexicaux simples (substantifs les plus représentés) et (ii) un système de catégorisation morphosyntaxique adapté aux caractéristiques les plus saillantes des textes scientifiques (abréviations, connecteurs, modaux, indices de structuration des textes, etc.). Ce dernier jeu de descripteurs s'était avéré particulièrement discriminant lors des tâches de classification domaniale. Nous avions de plus eu recours à deux techniques d'apprentissage supervisée que sont les séparateurs à vaste marge (SVM) et les arbres de décision. La recherche d'une précision maximale justifie le choix de l'approche SVM puisque cette technique est actuellement la plus performante pour la tâche considérée (Dumais *et al.*, 1998); afin de mieux appréhender l'articulation des deux types de traits nous avions en parallèle étudié le résultat d'une méthode de classification de type arbre de décision qui présente l'avantage de fournir une explication du modèle appris.

Bien qu'il diffère substantiellement de nos corpus d'étude, nous avons tenté dans la mesure du possible d'adapter la méthodologie développée au corpus DEFT'08, à plusieurs exceptions près : (i) les délais du défi ne nous permettant malheureusement pas de réadapter de manière pertinente le système de descripteurs développé, nous avons recouru au système d'étiquetage

[1] Discours littéraire, juridique, scientifique, journalistique, etc. Les types de discours sont reliés à des pratiques sociales distinctes et organisent en leur sein les typologies génériques et domaniales. Le discours juridique inclut ainsi les genres de l'arrêt, du décret, de la loi, etc.

plus général du TreeTagger[2]. Nous proposons néanmoins une expérimentation additionnelle dans le présent article en utilisant le jeu de descripteurs précédent, et en comparant les résultats à ceux obtenus avec le TreeTagger ; (ii) lors de nos expériences précédentes, nous avions recouru aux substantifs les plus représentés, étant donné leur statut potentiel de *concepts* dans les textes, et par conséquent d'objets interprétables. Nous avions en effet mis l'accent sur l'intérêt d'une description raisonnée et interprétable des regroupements obtenus dans le processus de classification. L'objectif de DEFT'08 étant tout autre, et visant d'abord à obtenir le meilleur pourcentage de classification possible, ce sont les classes qui se sont avérées les plus efficaces que nous avons retenues, après quelques tests; (iii) dans cette même optique de recherche de performance nous avons enfin été naturellement conduit à privilégier un classifieur de type SVM.

Voici les différentes étapes de la méthodologie utilisée :

1. Etiquetage du corpus avec TreeTagger
2. Construction des dictionnaires de classification
    - extraction des descripteurs lexicaux (parties pleines du discours)
    - extraction des descripteurs morphosyntaxiques (construction de classes)
3. Apprentissage d'un modèle de séparation des classes au moyen d'un SVM linéaire

Le présent article reprend les différentes étapes de ce processus : après avoir documenté les descripteurs mobilisés par la tâche d'apprentissage (section 2), nous décrirons la méthode d'apprentissage SVM utilisée (section 3). La section 4 reprend les résultats obtenus, et nous présentons enfin les conclusions de notre expérience dans la section 5.

2 Descripteurs mobilisés

Nous avons mobilisé deux types de descripteurs : deux ensembles de descripteurs morphosyntaxiques, et un ensemble de descripteurs lexicaux simples. Ces ensembles de descripteurs que nous décrivons dans la suite seront fusionnés (additionnés) afin de constituer un ensemble plus grand de descripteurs utilisé pour la classification.

2.1 Descripteurs morphosyntaxiques – système d'étiquetage TreeTagger

Le corpus a d'abord été étiqueté à l'aide du TreeTagger, avec le jeu d'étiquettes morphosyntaxiques proposé par Achim Stein pour le français[3]. Il s'agit d'un système d'étiquetage robuste (33 catégories), qui permet au TreeTagger français d'atteindre un score de précision élevé (environ 92%). Cette robustesse entraîne naturellement des choix d'étiquetage discutables sur le plan linguistique : ainsi, les auxiliaires ne sont pas reconnus et la catégorie 'pronoms démonstratifs' inclut également les déterminants démonstratifs. A

[2] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[3] http://www.ims.uni-stuttgart.de/~schmid/french-tagset.html

fortiori, on sait que la ponctuation est discriminante en matière de classification, mais TreeTagger n'identifie précisément que les guillemets, les autres ponctuations étant indifféremment labellisées PUN. Malgré ces réserves, c'est à partir de cette annotation que nous avons construit nos dictionnaires d'apprentissage.

28 descripteurs ont au final été conservés (tableau 1) :

Distribution générale des lemmes	*Distribution des verbes*	*Distribution des pronoms*
% de SYM ou ABR dans le document	% de verbes VER:cond parmi les verbes du document	% de pronoms de type PRO (simple) parmi l'ensemble des pronoms du document
% de ADV ou KON	% de VER:futu	% de PRO:DEM
% de ADJ	% de VER:impe	% de PRO:IND
% de PRO	% de VER:impf	% de PRO:PER
% de VER	% de VER:infi	% de PRO:POS
% de DET	% de VER:pper	% de PRO:REL
% de NAM ou NOM	% de VER:ppre	
% de PRP	% de VER:pres	
% de PUN	% de VER:simp	
% de INT	% de VER:subi	
% de NUM	% de VER:subp	

Tableau 1 : Catégories morphosyntaxiques conservées

On obtient donc pour chaque document un ensemble de 28 descripteurs morphosyntaxiques : les 10 premiers traits rendent compte de la distribution des étiquettes rencontrées dans un document sur les 10 classes (1ère colonne du tableau 1); les 10 traits suivants précisent cette fois la distribution au sein de la classe d'étiquettes correspondant aux verbes; enfin les traits suivants précisent la classe des pronoms.

2.2 Descripteurs lexicaux

En ce qui concerne les descripteurs lexicaux, nous avons étudiés plusieurs sélections et plusieurs indexations afin de retenir la solution la plus performante en terme de classification par SVM sur les corpus d'entraînement (évaluations par validations croisées sur un sous ensemble de 1000 documents). La sélection des étiquettes NOM, ADJ, VER, NAM, ABR, SYM, INT, PRO et ADV s'est avérée être la plus performante aussi bien pour discriminer les domaines que pour discriminer les genres. Nous avons alors retenus tous les lemmes correspondant à ces étiquettes sans sélection sur leur nombre d'occurrences dans le corpus d'entraînement.

Concernant l'indexation nous avons comparé une indexation de type fréquence (où chaque document est représenté par le vecteur des fréquences des mots du vocabulaire apparaissant dans le document) avec une indexation pondérée par le tfxidf. De manière assez inattendue nous avons observé une influence négligeable du mode d'indexation, avec de plus un avantage

pour le mode le plus simple (fréquences) lorsqu'une différence de performance était observée. Ce dernier mode d'indexation a donc été retenu pour nos expérimentations finales.

Par la méthode retenue, 31389 descripteurs lexicaux ont alors été retenus pour la tâche 1 et 36935 pour la tâche 2.

2.3 Descripteurs morphosyntaxiques – système d'étiquetage spécifique

Le système d'annotation additionnel que nous avons mobilisé[4] comprend 129 étiquettes[5] au total. Bien qu'il soit originellement adapté aux spécificités du discours scientifique, il prend en compte les recommandations EAGLES, et résulte de l'examen attentif de plusieurs systèmes d'étiquetage (TreeTagger, WinBrill pour le français, développé par l'Inalf, Cordial, etc.). En ce sens, il est plus systématisé et plus complet que celui de TreeTagger et sa granularité est plus élevée.

Deux types d'observations sont ainsi fédérés : un ensemble de catégories morphosyntaxiques générales, ou « de langue », incluant les grandes parties du discours et leurs attributs traditionnels (nombre, temps et modes verbaux, etc.), un ensemble de variables spécifiques et supposées caractéristiques du discours scientifique (distinction des IL anaphorique/impersonnel, des connecteurs généralement étiquetés comme adverbes, annotation des indices de structuration de type *1.1.2.*, des éléments de langue étrangère, symboles etc.). Le système employé inclut donc différents niveaux d'observation linguistique, dans la mesure il combine des variables morphosyntaxiques et sémantiques.

3 Classification par SVM

Les SVM, en anglais *Support Vector Machine* (Machines à Points de Support ou encore Séparateurs à Vaste Marge) sont reconnus pour leurs performances inégalées dans l'application à la classification de textes (Dumais *et al.*, 1998). De manière simplifiée, cette méthode consiste à rechercher un hyperplan séparateur pour deux classes données de manière à maximiser la marge entre les exemples de chacune des deux classes. Les SVM présentent de plus l'intérêt de formaliser le problème d'optimisation à partir seulement des produits scalaires entre objets et ainsi de se prêter à l'utilisation d'un noyau. Cette dernière technique permet de plonger les objets dans un espace de dimension éventuellement plus grande favorisant ainsi la possibilité de trouver un bon séparateur.

Dans la tâche qui nous intéresse ici, les objets (documents) étant décrits dans un espace de dimension très importante (plusieurs dizaines de milliers) il est généralement inutile de chercher à l'augmenter. Nous avons d'abord confirmé empiriquement ce postula sur les tâches du défi pour choisir d'utilisé la version linéaire de la librairie LIBSVM[6].

[4] Précisément documenté sur http://www.revue-texto.net/Corpus/Publications/Poudat/Chapitre_2.pdf

[5] 163 si l'on inclut les étiquettes positionnelles de type [PREPOSITION :1st] / [PREPOSITION :2nd].

[6] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

4 Résultats obtenus

Le tableau ci-dessous montre les résultats que nous avons obtenus avec notre approche à partir de l'annotation TreeTagger. De manière non surprenante, les équipes obtiennent les meilleurs résultats lorsqu'il s'agit de classifier les deux discours journalistique et encyclopédique (96%), tandis que les résultats sont plus mitigés en matière de classification thématique, Le Monde et Wikipédia étant généralistes par nature (81,1% en classification domaniale seule) :

Tâche	**F-score obtenu par notre équipe**	**F-score moyen des équipes**
Tâche 1 classification en genre	94%	**96%**
Tâche 1 classification en domaine	79%	**83%**
Tâche 2 classification en domaine	**82%**	81%

Tableau 2 : Résultats obtenus

Nos résultats s'avèrent en-deçà de la moyenne des équipes lorsqu'il s'agit de classer les genres (écarts de 2,3% et 3,6 % en dessous de la moyenne pour les tâches de classification en genres, et la tâche 1 de classification en domaine, qui inclut précisément l'information de genre).

Ces résultats peuvent surprendre, puisque notre hypothèse de départ consistait précisément à affirmer que le niveau morphosyntaxique avait un impact discriminatoire élevé en matière de classification en genres et discours. Rappelons toutefois (i) que la tâche de DEFT'08 est ici particulière, puisqu'il s'agit de discriminer deux genres, ou plutôt deux discours entre eux, alors que les études existantes prennent en considération un nombre plus important de genres et de discours, ce qui augmente la portée discriminatoire des observations ; (ii) que Wikipédia et Le Monde multiplient les domaines et les thèmes, ce qui laisse supposer l'existence de pratiques discursives internes aux domaines de spécialité, voire de sous-genres domaniaux. Soulignons également que Wikipédia est un genre plus émergent qu'établi, qui mime les discours et les genres existants, ce qui rend sa caractérisation fort complexe.

C'est peut-être en raison de ce dernier constat que nous obtenons finalement des résultats légèrement supérieurs à la moyenne en matière de classification domaniale, observation validée par la seconde expérimentation, qui modifie peu les résultats obtenus. Le Tableau 3 présente les résultats complémentaires obtenus en utilisant la description morphosyntaxique spécifique (cf. section 2.3) pour des raisons de temps les valeurs reportées correspondent à des estimations de précisions (et non de F-Score) obtenues par validations croisées sur des sous-ensembles de 1000 documents pour chaque tâche.

	Description simple			**Description spécifique**		
	Lexique seul	Morphosyn-taxe seule	Mixte	Lexique seul	Morphosyn-taxe seule	Mixte
Tâche 1 Genre	92%	74%	92%	92%	84%	93%
Tâche 1 domaine	80%	45%	82%	80%	54%	81%
Tâche 2 domaine	73%	36%	72%	74%	51%	75%

Tableau 3 : Résultats obtenus (en incluant notre expérience additionnelle)

Sur cette étude complémentaire, on note une amélioration significative du potentiel classificatoire de la description morphosyntaxique seule en utilisant la description spécifique plutôt que la description simple; cependant la description mixte ne semble pas tirer pas profit de cette amélioration. L'observation précédente suggère une analyse plus technique du phénomène : le classifieur SVM considère l'ensemble des descripteurs dans sa globalité et la sur-représentation des descripteurs lexicaux (plus de 30,000) par rapport aux descripteurs morphosyntaxique (une centaine au plus) revient quasiment à négliger cette dernière description. En revanche l'utilisation réaliste d'un classifieur suggèrerait de limiter la taille du vocabulaire (ou à produire un nombre limité de nouveaux traits) de manière à accélérer le traitement d'un nouveau document; on se ramènera alors à un meilleur équilibre entre les deux ensembles de descripteurs. Enfin si on souhaite proposer un classifieur intelligible on cherchera par exemple à produire des règles de décision (ce que ne permet pas l'approche SVM) et on utilisera par exemple un arbre de décision qui cette fois considère chaque descripteur indépendamment et recherche une combinaison de quelques descripteurs permettant de discriminer les classes : dans ce cas nous avons déjà pu observer un intérêt certain à augmenter l'offre de descripteurs en combinant lexique et morphosyntaxe (Cleuziou, Poudat, 2007).

5 Conclusion

Nous avons cherché à évaluer de manière expérimentale l'incidence des niveaux morphosyntaxique et lexical sur la classification en domaines et en genres dans le cadre pratique offert par le défi fouille de textes 2008.

Dans cette perspective, un ensemble de descripteurs morphosyntaxiques adapté aux caractéristiques du discours a été utilisé en parallèle d'un lexique extrait de manière traditionnelle à partir des corpus d'entraînement fournis. Le classifieur SVM a de plus été retenu pour ses performances reconnues pour la tâche considérée.

Les résultats que nous avons pu obtenir se situent dans la moyenne des performances de l'ensemble des équipes participantes. Pourtant d'une part l'originalité apportée par la description morphosyntaxique a peu influencé les modèles de classification appris et d'autre part l'approche globale utilisée reste relativement naïve (pas de sélection ni de pondération des descripteurs lexicaux). La première remarque s'explique par la nature même des corpus considérés et surtout plus techniquement par l'approche SVM utilisée. En revanche le fait qu'une approche simpliste nous permette d'obtenir des résultats tout à fait honorables tend à montrer que dans le cas de corpus volumineux les méthodologies d'indexation plus fines n'ont pas lieu d'être et s'inscrive plutôt dans un cadre de compensation d'information lorsque les corpus d'entraînement sont de taille plus restreinte.

Références

CLEUZIOU G., POUDAT C. (2007). On the impact of lexical and linguistic features in Genre- and Domain-Based Categorization. Actes de *CICLING-2007*. Lecture Notes in Computer Science, 599-610.

DHILLON I. S., MALLELA S., KUMAR R. (2003). A divisive information theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Researches*, vol. 3, 2003, MIT Press, 1265-1287.

DUMAIS S., PLATT J., HECKERMAN D., SAHAMI M. (1998). Inductive learning algorithms and representations for text categorization. Actes de CIKM ’98, ACM Press, 148-155.

HOFMANN T. (1999). Probabilistic Latent Semantic Indexing. Actes de *22nd Annual ACM Conference on Research and Development in Information Retrieval*, 50-57.

KARLGREN J., CUTTING D. (1994). Recognizing text genres with simple metrics using discriminant analysis. Actes de *COLING 94*, 1071-1075.

KESSLER B., NUNBERG G., SCHÜLTZE H. (1997). Automatic detection of text genre. Actes de *EACL’97*, 32-38.

MALRIEU D., RASTIER F. (2001). Genres et variations morphosyntaxiques. *Traitement Automatique des langues*, vol. 42, 2001/2, Hermes, Editions Lavoisier, 548-577.

POUDAT C., CLEUZIOU G. (2003). Genre and Domain processing in an Information Retrieval perspective. Actes de *3rd International Conference on Web Engineering, Lecture Notes in Computer Science*, 399-402.

POUDAT C., CLEUZIOU G., CLAVIER V. (2006). Catégorisation de textes en domaines et genres : complémentarité des indexations. *Document numérique* vol. 9 2006/1, Hermes, Editions Lavoisier, 61-76.

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.

Classifieur probabiliste avec Support Vector Machine (SVM) et Okapi

Anh-Phuc TRINH (1), David BUFFONI (2), Patrick Gallinari (3)

(1)(2) (3) Laboratoire d'Informatique de Paris 6
Anh-Phuc.Trinh@lip6.fr, David.Buffoni@lip6.fr,
Patrick.Gallinari@lip6.fr
(Site Passy Kennedy, 104 av du Président Kennedy, 75016 Paris)

Résumé Ce papier présente le travail réalisé par l'équipe des jeunes chercheurs du LIP6 pour le 4ème DÉfi Fouille de Textes (DEFT'08). Cette année, le défi était de classifier les documents de 2 corpus différents en prenant en compte les variations en genre et en thème. Cet article présente un modèle de classification automatique sous la forme de SVMs estimant les probabilités *a posteriori* des classes pour chaque document.

Abstract This paper describes the participation of LIP6 at DEFT'08. We shall present a method based on SVM to realize this task. We propose a method which estimates posterior probabilities for each document.

Mots-clés : classification automatique de texte, apprentissage machine probabiliste, machine à vecteurs support (SVM), recherche d'information
Keywords: text classification, SVM, information retrieval, probability estimation

1 Introduction

Le défi DEFT 08 [DEF 08] consiste cette année, à classifier thématiquement un ensemble de documents provenant de deux corpus de genre différents (encyclopédique et journalistique). De cette manière, on cherche à trouver les améliorations possibles d'un système de classification thématique en prenant en compte le genre. Ceci a amené, les organisateurs à décomposer ce défi en deux tâches distinctes. La première consiste, à l'aide d'un classifieur automatique, à reconnaître le genre et la catégorie thématique de chaque document appartenant à l'un des deux corpus précédents. La deuxième quant à elle, à pour but de trouver la catégorie thématique de chaque document, indépendamment du corpus.

Nous avons mis en œuvre une méthode statistique essayant de résoudre le plus efficacement possible ces deux tâches. Pour cela, nous nous sommes inspirés du modèle Okapi, modèle de référence en Recherche d'Information, pour représenter les textes sous la forme de vecteurs de scores creux.
Ensuite, nous proposons une nouvelle méthode permettant d'estimer les sorties des classifieurs SVM sous la forme de probabilités, basée sur la maximisation de la log-vraisemblance conditionnelle.

Nous exposerons, tout d'abord, les prétraitements effectués sur les corpus (section 2), ce qui nous mènera, ensuite, à la présentation du modèle utilisé (section 3). Par la suite, nous ferons la synthèse des expériences réalisées avec leurs résultats, (section 4) ce qui nous permettra, enfin, de conclure notre travail.

2 Prétraitement

Pour mettre en œuvre notre modèle, nous avons dû faire certaines hypothèses. Tout d'abord, pour chaque tâche, nous considérons l'ensemble des documents comme des sacs de mots. Les tailles de ces deux ensembles sont données dans le tableau 1.

Tâche	Taille de sac de mots
1	167545
2	219117

Tableau 1 : Taille des sacs de mots par tâche

On peut, à partir de ces statistiques, émettre deux hypothèses distinctes afin d'appliquer notre modèle. La première, la moins restrictive, serait de prendre en compte tous les mots lors de la phase d'apprentissage. Quant à la seconde, afin de réduire les tailles des sacs de mots, nous avons décidé de retirer les mots les plus fréquents. Pour ce faire, nous avons téléchargé une liste de mots les plus fréquents de la langue française, disponible sur site [EDU].

Codages textuels

Nous transformons, à présent, chaque sac de mots en un espace vectoriel, avec en indice les mots associés à un score. Nous présentons ci-après, différentes façons de calculer ces scores :

- Binaire : le vecteur associé au sac de mots est sous la forme binaire ce qui correspond à l'apparition (score = 1) ou à l'absence (score = 0) du mot dans le texte. Cette vision, certes naïve, nous servira de témoin pour les expériences.

- Tf : le vecteur associé contient les fréquences d'apparitions des mots dans le texte.

- Tf-Idf : on remplit le vecteur par les scores Tf-Idf [SAL 88] des mots. Ce type de codage a donné de bons résultats par le passé et est généralement utilisé dans les méthodes dites statistiques.

- Okapi (k, b) : nous attribuons un score Okapi aux mots. Le modèle probabiliste Okapi BM25 [OKP 05] a fait ses preuves en Recherche d'Information en étant plus performant qu'un modèle Tf-Idf. Nous avons, donc, intégré Okapi afin de voir si les performances intrinsèques, meilleures que les autres codages, subsistaient après l'utilisation du classifieur.

$$Score(mot_i, texte) = \frac{(k+1)tf_i}{\left(k\left((1-b)+b\frac{dl}{avdl}\right)+tf_i\right)} \times \log\left(\frac{N-df_i+0.5}{df_i+0.5}\right)$$

où dl est la longueur du document, $avdl$ la longueur moyenne des documents dans tous les corpus, tf la fréquence d'apparition du mot_i dans le document, N le nombre total de documents dans les corpus et df_i le nombre de documents contenant le mot_i.

Notre travail est de fixer les deux paramètres k et b, servant à donner de l'importance à la fréquence d'un mot dans le texte (paramètre k) et à la taille du texte par rapport à la moyenne (paramètre b).

Espace d'étiquettes (8 pour la tâche 1, 5 pour la tâche 2)

Nous avons étiqueté les catégories thématiques suivant le genre, de la forme suivante :

Pour la tâche 1 :

	ART	**ECO**	**SPO**	**TEL**
W	1	2	3	4
LM	5	6	7	8

Pour tâche 2 :

FRA	**INT**	**LIV**	**SCI**	**SOC**
1	2	3	4	5

3 Classifieur probabiliste

Dans cette section, nous décrivons notre modèle de classification multi-classes à l'aide de SVMs retournant les probabilités d'appartenance aux classes.

Définition 1 : (Calcul de la probabilité *a posteriori* pour la classification multi-classes)

Soit E = {($\mathbf{x_1}$, y_1), ($\mathbf{x_2}$, y_2), …, ($\mathbf{x_m}$, y_m)} un ensemble d'apprentissage de m couples d'exemples. On suppose que chaque exemple $\boldsymbol{x_i} \in \Re^n$ et son étiquette associée, y_i, est un entier de l'ensemble Y = {1, 2, 3, …, k} où k correspond au nombre de classes. La probabilité *a posteriori* de classification multi-classes (k>2) est une probabilité conditionnelle, sachant l'exemple $\boldsymbol{x}$, d'étiquette y :

$$P(y/x) = p_i \ avec \sum_{i=1}^{k} p_i = 1 \qquad (1)$$

Un nouveau problème se pose à nous dans le cadre d'une classification multi-classes car les SVMs sont, en règle générale, des classifieurs binaires. Nous pouvons alors appliquer deux sortes de stratégies afin de faire de la classification multi-classes, soit, pour la première, «une classe contre une autre», soit, pour la deuxième, «une contre toutes» (cf. Figure 1). Selon ces deux stratégies, notre problème s'est décomposé en plusieurs sous-problèmes de classification binaire ce qui nous a amené à diviser l'ensemble E en k sous-ensembles différents. Pour la première stratégie, on compare $\binom{2}{k}$ fois, un sous-ensemble *Ei* par rapport à un autre sous-ensemble *Ej*, avec $i \neq j$. Quant à la deuxième stratégie, on compare k fois, un sous-ensemble *Ei* par rapport à $\bigcup_{j \in Y, j \neq i} E_j$. La solution adoptée dans notre travail, a été d'appliquer la première stratégie [HSU 02].

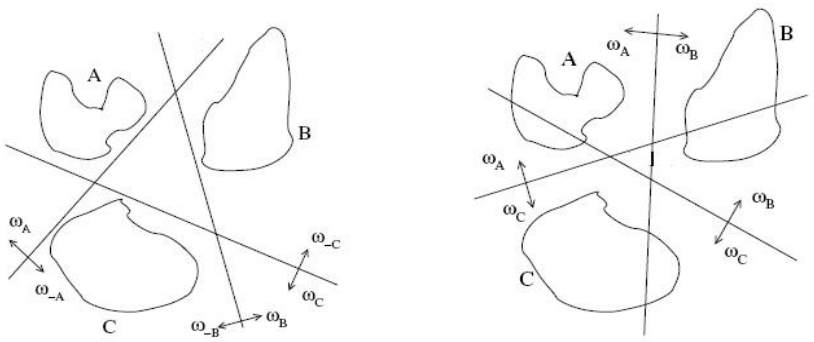

Figure 1 : Deux stratégies de classification multi-classes. A gauche, la stratégie « une contre toutes », à droite, la stratégie, « une contre une ».

Définition 2 : (Conversion de la valeur de sortie d'une classification binaire en une valeur de sortie pour une classification multi-classes)

Le classifieur SVM renvoie une valeur +1 ou -1 suivant qu'un exemple $\boldsymbol{x}$ appartienne à la première ou à la deuxième classe. Cependant, la sortie donnée par le SVM n'est applicable

qu'à un problème de classification binaire. Pour y remédier, nous avons défini une fonction $cl_{ij}(x)$ avec i≠j, renvoyant pour un exemple $\boldsymbol{x}$, le numéro de sa classe (et non +1 ou -1). En d'autres termes :

$$cl_{ij}(\mathbf{x}) = \begin{cases} i & si\ f_{ij}(\mathbf{x}) = -1 \\ j & si\ f_{ij}(\mathbf{x}) = +1 \end{cases} \qquad (2)$$

où *i* et *j* sont les indices des classes, et *f* la fonction de sortie du SVM.

3.1 Sortie probabiliste lors d'une classification binaire

D'après les définitions précédentes, les SVMs retournent des valeurs entières pour chaque exemple $\boldsymbol{x}$. Cependant, on souhaiterait avoir, à la place de ces valeurs, les probabilités d'appartenance de $\boldsymbol{x}$ aux différentes classes. Pour ce faire, dans le cadre de la classification binaire, [PLA 00] et [HER 07] proposent l'utilisation d'une fonction sigmoïde permettant de créer une sortie probabiliste :

$$P(y = i / y = i\ ou\ j, \mathbf{x}) = r_{ij} = \frac{1}{(1 + \exp(A \times f_{ij}(\mathbf{x}) + B))} \qquad (3)$$

où A et B sont estimés en maximisant la log-vraisemblance conditionnelle sur l'ensemble d'apprentissage $E_{i,j}$.

3.2 Sortie probabiliste lors d'une classification multi-classes

L'utilisation de l'équation (3) n'est pas adaptée pour de la classification multi-classes. Pour estimer les probabilités conditionnelles P(y = i | $\mathbf{x}$) = p_i, à partir des classifieurs locaux $cl_{ij}(\boldsymbol{x})$, plusieurs solutions existent, un bref aperçu, est donné dans l'état de l'art ci-dessous.

3.2.1 Etat de l'art

- L'idée la plus simple [KER 90] pour construire la probabilité *a posteriori* (1) à partir des classifieurs locaux $cl_{ij}(\boldsymbol{x})$ est d'utiliser la règle de vote. La probabilité P(y = i | $\boldsymbol{x}$) est égale au nombre moyen de votes pour la classe *i*. Soit la fonction prédicat V($\boldsymbol{x}$) = 1 si $\boldsymbol{x}$ est bien classé, et zéro sinon.

$$p_i = \frac{2}{k(k-1)} \sum_{j \in Y : j \neq i} V(cl_{ij}(\mathbf{x}) = i), \quad i = 1,2,...,k \qquad (4)$$

- Une idée proposée par [HAS 98] est de minimiser la distance de Kullback-Leiber (KL) entre les probabilités de sorties des classifieurs r_{ij} (équation (3)) et $\mu_{ij} = \frac{p_i}{(p_i + p_j)}$ ce qui donne le problème d'optimisation suivant :

$$\min_{\mathbf{p}} KL(\mathbf{p}) = \sum_{i \neq j} |E_{i,j}| \left(r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}} \right) \qquad (5)$$

$$s.c. \sum_{j \in Y: j \neq i} \left| E_{i,j} \right| \mu_{ij} = \sum_{j \in Y: j \neq i} \left| E_{i,j} \right| r_{ij}, \quad \sum_{i=1}^{k} p_i = 1,\ p_i > 0,\ i = 1,2,\ldots,k \qquad (6)$$

- [WU 04] ont suggéré, pour calculer la probabilité *a posteriori* p, de minimiser l'erreur quadratique provenant de l'égalité $r_{ij}.p_j = r_{ji}.p_i$. Pour résoudre ce problème d'optimisation, ils introduisent les multiplicateurs de Lagrange à (7) et à (8).

$$\min_{\mathbf{p}} \frac{1}{2} \sum_{i=1}^{k} \sum_{j \in Y: j \neq i} \left(r_{ji} p_i - r_{ij} p_j \right)^2 \qquad (7)$$

$$s.c. \sum_{i=1}^{k} p_i = 1,\ p_i \geq 0, \quad i = 1,2,3,\ldots k. \qquad (8)$$

3.2.2 Notre modèle

À partir du problème de minimisation de la distance de KL (5), nous proposons de créer un modèle avec une sortie probabiliste qui maximiserait la log-vraisemblance conditionnelle $l(y/\mathbf{x})$.

Par exemple, nous avons un problème de la classification multi-classes, avec $k = 3$. Supposons que l'on a la configuration des décisions $c(f_{ij}(\mathbf{x}))$, formée par l'ensemble de classifieurs locaux :

$$c(f_{ij}(\mathbf{x})) = \{ f_{12}(\mathbf{x}) = -1,\ f_{13}(\mathbf{x}) = +1,\ f_{23}(\mathbf{x}) = -1 \}$$

L'étiquette associée à $\boldsymbol{x}$ est y = 2. Les fonctions caractéristiques actives Θ sont donc :

$$\Theta_2(\mathbf{x}, y) = \begin{cases} 1 \; si \; f_{12}(\mathbf{x}) = -1 \; et \; y = 2 \\ 0 \; autrement \end{cases} ;\ \Theta_{11}(\mathbf{x}, y) = \begin{cases} 1 \; si \; f_{13}(\mathbf{x}) = +1 \; et \; y = 2 \\ 0 \; autrement \end{cases} ;$$

$$\Theta_{14}(\mathbf{x}, y) = \begin{cases} 1 \; si \; f_{23}(\mathbf{x}) = -1 \; et \; y = 2 \\ 0 \; autrement \end{cases}$$

On associe à chaque fonction caractéristique, un paramètre λ_i, ce qui nous donne un vecteur de paramètres $\boldsymbol{\lambda} \in \Re^d$. La probabilité conditionnelle $p(y/\mathbf{x}, \lambda)$ est alors définie sur l'ensemble de fonctions caractéristiques Θ :

$$p(y/\mathbf{x}, \lambda) = \frac{\exp\left(\sum_{l=1}^{d} \lambda_l \times \Theta_l(\mathbf{x}, y) \right)}{Z(\mathbf{x}, \lambda)} \qquad (9)$$

Avec la fonction auxiliaire $Z(\mathbf{x},\lambda)$:

$$Z(\mathbf{x},\lambda)=\sum_{y=1}^{k}\exp\left(\sum_{l=1}^{d}\lambda_l\times\Theta_l(\mathbf{x},y)\right) \qquad (10)$$

A présent, nous pouvons alors maximiser la log-vraisemblance, comme suit :

$$\min_{\lambda} l(\lambda/E)=-\sum_{j=1}^{m}\left[\sum_{l=1}^{d}\lambda_l\times\Theta_l(\mathbf{x}_j,y_j)+\log\frac{1}{Z(\mathbf{x}_j,\lambda)}\right] \qquad (11)$$

Nous initialisons les paramètres λ_l du modèle à 1.0 pour $l=1,2,...d$. De plus, nous considérons comme critère de convergence, la norme du gradient $||l(\lambda/E)|| < 0{,}008$.

4 Expériences réalisées

Pour toutes les expériences, nous avons décomposé les corpus mis à notre disposition, en deux sous-ensembles, 75% du corpus total pour l'apprentissage et les 25% restant pour le test. Concernant l'algorithme SVM, nous avons utilisé le logiciel libSVM [SVM 01].

Tout d'abord, nous avons comparé les différentes performances des solveurs SVMs suivant les différents prétraitements que l'on a vus en section 2. Dans un premier temps, nous avons testé l'utilité d'une liste générale de mots fréquents ajoutée à différents types de codage textuels. La synthèse des résultats peut être retrouvée dans le tableau 2. Par la suite, nous nous sommes intéressés au codage textuels (Binaire, Tf-Idf, Okapi…) afin de savoir celui qui discriminait le mieux les deux corpus. Nous nous sommes servis d'un modèle Okapi avec les paramètres par défaut, à savoir k = 1.2 et b = 0.75. En parallèle, nous avons cherché, pour les SVMs, le noyau offrant les meilleures performances entre les noyaux de type linéaires, polynomiaux et radial, les résultats sont regroupés dans le tableau 3.

Enfin comme dernière expérience, nous avons essayé, expérimentalement, de trouver la meilleure valeur de k et de b pour le modèle Okapi. Nous avons fait varier le paramètre k par pas de 0.5 allant de 1.0 à 10 et le paramètre b par pas de 0.1 de 0.1 à 1.0. On peut se référer au tableau 4, tableau résumé de l'original.

Expériences		Sans liste		Avec liste	
Code	Noyau	Tache 1	Tache 2	Tache 1	Tache 2
Binary	Linéaire	**89.9606**	85.4765	84.7832	**84.6102**
	Poly	**19.6583**	**27.7731**	18.3968	27.6372
	Radial	**19.6583**	**27.7731**	18.3968	27.6372
Tf	Linéaire	**89.3298**	84.5082	83.2589	**84.5932**
	Poly	**20.3154**	**28.2827**	19.3693	27.79
	Radial	**38.9488**	**41.2944**	27.0959	32.2066

Tf-Idf	Linéaire	**89.0145**	**86.5976**	83.1012	86.4957
	Poly	18.7648	27.807	**20.2628**	**28.2996**
	Radial	**48.7516**	**56.4294**	44.9671	53.151

Tableau 2 : les valeurs sont sous la forme d'accuracy donnée par le SVM avec voting rule. En gras, les meilleures performances, pour un type de codage, un noyau et une tâche. On cherche à savoir s'il faut ou non utiliser d'une liste de mots fréquents.

Expériences sans liste		Noyau		
Code	Tache	Linéaire	Poly	Radial
Binary	1	**90.199**	22.5448	68.4096
	2	**84.9682**	28.8769	70.7686
Tf	1	**89.2071**	26.9264	58.1817
	2	**84.9851**	27.9745	51.8259
Tf-Idf	1	**90.199**	29.7839	80.8382
	2	**86.9766**	32.5393	78.2972
Okapi k=1.2 b=0.75	1	**91.1895**	26.7162	81.8958
	2	**87.8514**	33.9618	81.0658

Tableau 3 : on cherche à déterminer le meilleur noyau et parallèlement le meilleur codage textuel sans utiliser de liste de mots fréquents.

Expériences sans liste et avec noyau linéaire		b		
Okapi		0.7	0.8	0.9
k	2.0	90.7254	91.488	90.276
	2.5	91.9619	**92.5138**	90.7773
	3.0	89.9106	91.724	90.4568

Tableau 4 : on cherche à trouver les valeurs de k et de b pour le codage Okapi donnant les meilleures performances.

Au vu de ces résultats préliminaires, nous avons décidé de réaliser un modèle SVM, utilisant un noyau linéaire et intégrant un codage textuel sous la forme d'un score Okapi avec comme paramètres k = 2.5 et b = 0.8.

Nous avons ensuite, soumis à DEFT 08 trois modèles différents. Le premier implémentant un algorithme simple utilisant la règle de vote (équation 4), le deuxième, notre modèle maximisant la log-vraisemblance (équation 11) et enfin le dernier, un classifieur proposé par Wu (équations (7) et (8)). Les résultats de nos trois soumissions sont à retrouver dans le tableau 5.

Tâche / Soumissions		Règle de vote (1)	Log-vraisemblance (2)	Wu (3)	Moyennes
Tâche 1 Genre	F-score Strict	0.9505	0.9734	**0.9755**	0.9596
	F-score Pondéré	0.6249	**0.9728**	0.9584	
Tâche 1 Catégorie	F-score Strict	0.8038	0.8796	**0.8942**	0.8258
	F-score Pondéré	0.3887	**0.8781**	0.8572	
Tâche 2 Catégorie	F-score Strict	0.8740	0.8738	**0.8758**	0.8105
	F-score Pondéré	0.3930	**0.8735**	0.8167	

Tableau 5 : des résultats pour DEFT 08

5 Conclusions

Nous avons présenté une nouvelle idée afin de convertir les sorties entières d'un solveur SVM en une sortie probabiliste pour un problème de classification multi-classes. Nous avons obtenu de bons résultats pour l'ensemble de ce défi. En effet, deux de nos soumissions sont meilleures que les performances moyennes lors de DEFT'08.

Nous constatons que le modèle de Wu est légèrement meilleur que le notre suivant la mesure F-score strict, mais la tendance s'inverse lors du choix d'un F-score pondéré. On ne peut pas donner d'explications précises car ces résultats dépendent fortement des données. Néanmoins, on peut supposer que le fait de créer r_{ij} par validation croisée (car valeur locale), permet à la méthode de Wu d'être plus précise que la notre. Cependant, elle est beaucoup plus coûteuse en temps.

En conclusion et en guise de perspective, nous pourrions appliquer cette idée pour un problème plus complexe comme c'est le cas avec des documents semi-structurés, où nous devons en plus, du texte, tenir compte de la structure.

Remerciements

- Nous remercions sincèrement le comité de DEFT'08 pour l'organisation et les explications qui nous ont été fournies au cours de ce défi.

Références

[DEF 08] http://deft08.limsi.fr/

[SAL 88] Salton, G., Buckley, C. (1988). Term-weighting approaches in automatic text retrieval, *Information Processing & Management 24*,513-523.

[EDU] http://eduscol.education.fr/D0102/liste-mots-frequents.htm

[OKP 05] Robertson S. (2005). How Okapi came to TREC, *Voorhees and Harman*

[KER 90] Knerr S., Personnaz L., Dreyfus G. (1990). Single-layer training revisited: a stepwise procedure for building and training a neural network, *Neurocomputing: Algorithm, Architectures and Applications*. J. Fogelman Springer-Verlag.

[HAS 98] Hastie, T., Tibshirani, R. (1998). Classification by Pairwise Coupling, *Advances in Neural Information Processing Systems 10*. M. I. Jordan, M. J. Kearns, S. A. Solla.

[PLA 00] Platt JC. (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans, 61-74.

[MUL 01] Müller K.-R., Mika S., Rätsch G., Tsuda K., Schölkopf B. (2001). An Introduction to Kernel-Based Learning Algorithms, *IEEE Transactions on Neural Networks 2*, 181-201

[LIN 02] Hsu CW, Lin CJ (2002). A comparison of methods for multi-class support vector machines, *IEEE Transactions on Neural Networks*, 13:415{425}

[WU 04] Wu TF, Lin CJ, Wenig RC (2004). Probability Estimates for Multi-class Classification by Pairwise Coupling, *Journal of Machine Learning Research*, 975-1005

[HER 07] Hérault R, Grandvalet Y, (2007). Sparse probabilistic classifiers. Actes d'*ICML 2007*, 337-344

[TRI 07] Trinh AP, (2007). Classification de texte et estimation probabiliste par Machine à Vecteurs de Support, Actes de *l'atelier du 3ème DEFT*, 71-85.

[SVM 01] Chang CC, Lin CJ (2001). LIBSVM : a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Index des auteurs

Index des mots-clés