Fusion probabiliste appliquée à la détection et classification d'opinions

Juan-Manuel Torres-Moreno[(1,2)], Marc El-Bèze[(1)], Fréderic Béchet[(1)] et Nathalie Camelin[(1)]

[(1)]Laboratoire Informatique d'Avignon – Université d'Avignon et des Pays de Vaucluse
BP 1228, 84911 Avignon Cédex 09 France
[(2)]École Polytechnique de Montréal – Département de génie informatique
H3C3P8 Montréal (Québec) Canada

Résumé – Abstract

Nous présentons des modèles d'apprentissage probabilistes appliqués à la tâche de classification telle que définie dans le cadre du défi DEFT'07 : la classification d'un texte suivant l'opinion qu'il exprime. Pour classer les textes, nous avons utilisé plusieurs classifieurs et une fusion. Une comparaison entre les résultats en validation et en tests montrent une coïncidence remarquable et mettent en évidence la robustesse et performances de l'algorithme de fusion. Les résultats que nous obtenons, en termes de précision, rappel et F-score sur les sous corpus de test nous ont permis de remporter le défi.
We present probabilistic learning models applied to sentiment classification task as defined in the DEFT'08 challenge. In this task, the texts must be classified following theirs opinions. We have used a mix of several classifiers. A comparison between the results and validation tests shows a remarkable coincidence and highlight the robustness and performance of our mixture algorithm. Our results, (precision, recall and F-score) on the test corpus, enabled us to win the challenge.

Mots-clefs – Keywords

Méthodes probabilistes, Apprentissage automatique, Classification de textes par leur contenu, défi DEFT.
Probabilistic methods, Machine learning, Text Classification, DEFT challenge.

1 Introduction

En juillet 2007 dans le cadre de la plate-forme AFIA 2007[1], a été organisé la troisième édition de DEFT (DÉfi Fouille de Textes) (Azé & Roche, 2005; Azé *et al.*, 2006). Cela a été la deuxième participation dans DEFT de l'équipe Traitement Automatique de la Langue Naturelle (TALNE) du Laboratoire Informatique d'Avignon (LIA)[2]. Lors de la première competition en 2005 (El-Bèze *et al.*, 2005), notre équipe avait remporté le défi. À l'époque, le problème était de classer les segments des allocutions de Jacques Chirac et François Mitterrand préalablement mélangées[3]. Le défi DEFT en 2007[4] a été motivé par le besoin de mettre en place des techniques de fouille des textes permettant de classer de textes suivant l'opinion qu'ils expriment. Concrètement, il s'agisait de classer les textes de quatre corpus en langue française selon les opinions qui y sont formulées. La classification d'un corpus en classes pré-déterminées, et son corollaire le profilage de textes, est une problématique importante du domaine de la fouille de textes. Le but d'une classification est d'attribuer une classe à un objet textuel donné, en fonction d'un profil qui sera explicité ou non suivant la méthode de classification utilisée. Les applications sont variées. Elles vont du filtrage de grands corpus (afin de faciliter la recherche d'information ou la veille scientifique et économique) à la classification par le genre de texte pour adapter les traitements linguistiques aux particularités d'un corpus. La tâche proposée par DEFT'07 visait le domaine applicatif de la prise de décision. Attribuer une classe à un texte, c'est aussi lui attribuer une valeur qui peut servir de critère dans un processus de décision. Et en effet, la classification

[1]Association Française pour l'Intelligence Artificielle, `http://afia.lri.fr`
[2]`http://www.lia.univ-avignon.fr`
[3]Pour plus de détails concernant DEFT'05, voir le site `http://www.lri.fr/ia/fdt/DEFT05`
[4]`http://deft07.limsi.fr/`

d'un texte suivant l'opinion qu'il exprime a des implications notamment en étude de marchés. Certaines entreprises veulent désormais pouvoir analyser automatiquement si l'image que leur renvoie la presse est plutôt positive ou plutôt négative. Des centaines de produits sont évalués sur Internet par des professionnels ou des internautes sur des sites dédiés : quel jugement conclusif peut tirer de cette masse d'informations un consommateur, ou bien encore l'entreprise qui fabrique ce produit ? En dehors du marketing, une autre application possible concerne les articles d'une encyclopédie collaborative sur Internet telle que Wikipédia : un article propose-t-il un jugement favorable ou défavorable, ou est-il plutôt neutre suivant en cela un principe fondateur de cette encyclopédie libre ? À priori, un travail de détection et de classification d'opinion paraît très simple. Or, de nombreuses raisons font que le problème est complexe. Facteur aggravant : on ne dispose que de corpus de taille moyenne, déséquilibrés par rapport à leurs classes. Dans cet article nous décrivons les méthodes employées dans le cadre de DEFT'07 qui nous ont permis de remporter le défi. Nous décrivons en section 2 le corpus et la méthode d'évaluation proposée. En section 4 nous présentons les outils de classification de texte utilisées. La représentation de textes ainsi qu'une agglutination et normalisation graphique sont detaillées en section 3. Nos outils de classification sont décrits en section 4. Des expériences et résultats sont rapportés et discutés en section 5, avant de conclure et d'envisager quelques perspectives.

2 Description des corpus

Les organisateurs du défi DEFT'07 ont mis à la disposition des participants quatre corpus hétérogènes :

aVoiraLire. Critiques de films, livres, spectacles et bandes dessinées. Ce corpus comporte 3 460 critiques et les notes qui leur sont associées. Etant donné que beaucoup d'organes de diffusion de critiques de films ou de livres[5] attribuent, en plus du commentaire, une note sous la forme d'une icône. Les organisateurs du défi ont retenu une échelle de 3 niveaux de notes. Ceci donne lieu à 3 classes bien discriminées : 0 (mauvais), 1 (moyen), et 2 (bien).

jeuxvideo. Le corpus de tests de jeux vidéo comprend 4 231 critiques. Chaque critique comporte une analyse des différents aspects du jeu – graphisme, jouabilité, durée, son, scénario, etc. – et une synthèse globale du jugement. Comme pour le corpus précédent, a été retenue une échelle de 3 niveaux de notes, qui donne les 3 classes 0 (mauvais), 1 (moyen), et 2 (bien).

relectures. Relectures d'articles de conférences. Ce corpus comporte 1 484 relectures d'articles scientifiques qui alimentent les décisions de comités de programme de conférences et renvoient des conseils et critiques aux auteurs. L'échelle retenue comporte 3 niveaux de jugement. La classe 0 est attribuée aux relectures qui proposent un rejet de l'article, la classe 1 est attribuée aux relectures qui retrouvent l'acceptation sous condition de modifications majeures ou en séance de posters, et la classe 2 regroupe les acceptations d'articles avec ou sous des modifications mineures. Ce corpus (comme le suivant) a subi un processus préalable d'anonymisation de noms des personnes.

débats. Le corpus des débats parlementaires est composé de 28 832 interventions de députés portant sur des projets de lois examinés par l'Assemblée Nationale. À chaque intervention, est associé le vote de l'intervenant sur la loi discutée. 0 (en faveur) ou 1 (contre).

Les corpus ont été scindés par les organisateurs en deux parties : une partie (environ 60%) des données a été fournie aux participants comme données d'apprentissage afin de mettre au point leurs méthodes, et une autre partie (environ 40%) a été réservée pour les tests proprement dits. Sous peine de disqualification, aucune donnée, en dehors de celles fournies par le comité d'organisation ne pouvait être utilisée. Ceci exclut notamment l'accès aux sites web ou à n'importe quelle autre source d'information. Nous présentons au tableau 1, des statistiques brutes (nombre de textes et nombre de mots) des différents corpus. Des exemples portant sur la structure et les détails des corpus, peuvent être consultés dans le site du défi[6].

2.1 Évaluation stricte

Le but du défi a consisté à classer chaque texte, issu des quatre corpus, selon l'avis qui y est exprimé. Positif, négatif ou neutre dans le cas où il y a trois classes, pour ou contre dans le cas binaire (corpus de débats parlemen-

[5] Par exemple voir le site `http://www.avoir-alire.com`

[6] `http://deft07.limsi.fr/corpus-desc.php`

Corpus	Textes (A)	Mots (A)	Textes (T)	Mots (T)
aVoiraLire	2 074	490 805	1 386	319 788
jeuxvideo	2 537	1 866 828	1 694	1 223 220
relectures	881	132 083	603	90 979
débats	17 299	2 181 549	11 533	1 383 786

Table 1: Statistiques brutes sur les quatre corpus d'apprentissage (A) et de test (T).

taires). Intuitivement, la tâche de classer les avis d'opinion des articles scientifiques est la plus difficile des quatre car le corpus afférent contient beaucoup moins d'informations que les trois autres, mais d'autres caractéristiques particulières à chaque corpus ont aussi leur importance. Les algorithmes seront évalués sur des corpus de test (T) avec des caractéristiques semblables à celui d'apprentissage (A) (cf. tableau 1), en calculant le *Fscore* des documents bien classés, moyenné sur tous les corpus :

$$Fscore(\beta) = \frac{(\beta^2 + 1) \times \langle Précision \rangle \times \langle Rappel \rangle}{\beta^2 \times \langle Précision \rangle + \langle Rappel \rangle} \tag{1}$$

où la précision moyenne et le rappel moyen sont calculés comme :

$$\langle Précision \rangle = \frac{\sum_{i=1}^{n} Précision_i}{n} \; ; \; \langle Rappel \rangle = \frac{\sum_{i=1}^{n} Rappel_i}{n} \tag{2}$$

Etant donné pour chaque classe i :

$$Précision_i = \frac{\{\text{Nb de documents correctement attribués à la classe } i\}}{\{\text{Nb de documents attribués à la classe } i\}} \tag{3}$$

$$Rappel_i = \frac{\{\text{Nb de documents correctement attribués à la classe } i\}}{\{\text{Nb de documents appartenant à la classe } i\}} \tag{4}$$

D'après les règles du défi, un document est attribué à la classe d'opinion i si : i/ seule la classe i a été attribuée à ce document, sans indice de confiance spécifié ; ii/ la classe i a été attribuée à ce document avec un meilleur indice de confiance que les autres classes (s'il existe un indice de confiance).

2.2 Indice de confiance pondéré

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 classe d'opinion donnée. Le F-score pondéré par l'indice de confiance a été 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 ont été pondérés par l'indice de confiance.

$$Précision_i = \frac{\sum_{\text{AttribuéCorrect}_i=1}^{\text{NbAttribuéCorrect}_i} \text{Indice\_confiance}_{\text{AttribuéCorrect}_i}}{\sum_{\text{Attribué}_i=1}^{\text{NbAttribué}_i} \text{Indice\_confiance}_{\text{Attribué}_i}} \tag{5}$$

$$Rappel_i = \frac{\sum_{\text{AttribuéCorrect}_i=1}^{\text{NbAttribuéCorrect}_i} \text{Indice\_confiance}_{\text{AttribuéCorrect}_i}}{\{\text{Nb de documents correctement attribués à la classe } i\}} \tag{6}$$

avec :

- $\text{NbreAttribuéCorrect}_i$: nombre de documents appartenant effectivement à la classe i et auxquels le système a attribué un indice de confiance non nul pour cette classe.
- NbreAttribué_i : nombre de documents attribués auxquels le système a attribué un indice de confiance non nul pour la classe i.

Dans le cadre de DEFT'07, le calcul du F-score retenu par les organisateurs est ensuite calculé à l'aide des formules (1) et (2), du F-score classique modifié (cette réécriture suppose évidemment que β soit égal à 1 de façon à ne privilégier ni précision ni rappel).

3 Représentations de documents

Un même texte peut être représenté par les différents paramètres qu'il est possible d'en extraire. Les représentations les plus courantes sont les mots, les étiquettes morpho-syntaxiques –*Part Of Speech*, POS– ou les lemmes. La tâche qui nous occupe consiste à retrouver l'*opinion* exprimée dans les textes. En nous inspirant de l'approche typique de l'analyse des opinions (Hatzivassiloglou & McKeown, 1997), nous utilisons un paramètre de représentation supplémentaire, une étiquette nommée *seed*. Un *seed* est un mot susceptible d'exprimer une polarité positive ou négative (Wilson *et al.*, 2005). Notre protocole de construction du lexique de *seeds* consiste en deux étapes. Premièrement, une liste de mots polarisés a été créée manuellement. Elle contient par exemple : *aberrant, compliments, discourtois, embêtement, ...* Afin de généraliser la liste de mots polarisés obtenue, chaque mot a été remplacé par son lemme. Nous obtenons ainsi un premier lexique de 565 *seeds*. Deuxièmement, l'algorithme *BoosTexter* a appris sur les textes représentés en mots. Les mots sélectionnés par ce modèle ont été filtrés manuellement, lemmatisés et ajoutés au lexique. Au final, nous obtenons un lexique d'environ 2 000 *seeds*. En effet, une phrase représentée en *seeds* ne contient alors que les lemmes faisant partie de ce lexique.

3.1 Agglutination et normalisation graphique

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[7] en unités de base. Un autre ensemble de règles appropriées est mis à contribution pour normaliser les graphies. 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. Le recours à une étape de prétraitement comme la lemmatisation est motivé par le taux de flexion élevé de la langue française. Néanmoins, dans le problème qui nous occupe, il s'avère utile de ne pas voir disparaître nombre d'informations comme par exemple certains conditionnels ou subjonctifs. Dans une relecture d'article, la présence de propositions comme " *Il aurait été préférable* " ou " *il eût été préférable* " laisse supposer que l'arbitre n'est pas totalement en faveur de l'acceptation du texte qu'il a relu. Pour ne en être privés, nous avons bridé la lemmatisation pour un petit nombre de cas susceptibles de servir de points d'appui lors de la prise de décision. Pour au moins deux systèmes, les textes lemmatisés ont été soumis à une étape que l'on pourrait qualifier de normalisation graphique. Quelque 30 000 règles écrites pour l'occasion ont permis de réunifier les variantes graphiques (essentiellement des noms propres) et de corriger un grand nombre de coquilles. Il est à noter que certaines de ces fautes d'orthographe ont pu être introduites par l'étape de réaccentuation automatique que nous avons appliquée au préalable sur les quatre corpus. En cas d'ambiguïté, ces récritures sont faites en s'appuyant sur les contextes gauches ou droits (parfois les deux). Par exemple : *Thé-Old-Republic* $\Rightarrow$ *the-Old-Republic*. Ces règles de réécriture avaient aussi pour but de combler certaines lacunes de notre lemmatiseur. Il n'est pas inutile de ramener à leur racine des flexions même peu fréquentes de verbes qui ne se trouvaient pas dans notre dictionnaire (comme *frustrer, gâcher*, ou *gonfler*). Enfin, quelques règles (peu nombreuses) avaient pour mission d'unifier sous une même graphie des variantes sémantiques (par exemple : *tirer-balle-tempe* et *tirer-balle-tête*).

Les différents exemples donnés ci-dessus font apparaître des regroupements sous la forme d'expressions plus ou moins figées[8]. Celles-ci ont été constituées par application de règles régulières portant sur des couples de mots. Pour leur plus grande partie, les 30 000 règles que nous avons utilisées proviennent d'un simple calcul de collocation effectué selon la méthode du rapport de vraisemblance (Mani & Maybury, 1999). Une autre part non négligeable est issue de listes d'expressions disponibles sur la toile[9]. Nous y avons ajouté également des proverbes (comme *tirer-son-épingle-jeu, mettre-feu-poudre*) extraits de listes se trouvant sur des sites web[10]. Mais nous sommes conscients que même si nous avons tenté de contrôler au maximum ces ajouts, des expressions comme "*les pieds sur terre* " ou "*un pied à terre* " ont pu être fondues à tort dans une même graphie **pied-terre**. Enfin d'autres expressions proches des slogans martelés lors de campagnes électorales de 2007, (comme *travailler-plus-pour-gagner-plus* ou *ordre-juste*) nous ont été fournies, à l'époque, par une actualité plus brûlante. Pour DEFT'08 nous avons changé la forme de cette agglutination/normalisation (Béchet *et al.*, 2008). L'objectif était de faire émerger, de façon automatique, ces règles à partir des textes.[11]

[7] Il serait plus correct de dire leurs lemmes car nous utilisons les formes lemmatisées par *LIA_TAGG*

[8] Pour l'identification de plusieurs noms propres (noms de jeux vidéo et vedettes du show-bizz) les étudiants et les enfants de l'un des co-auteurs de cet article ont été mis à contribution. Qu'ils en soient ici remerciés.

[9] Comme celle qui se trouve à l'adresse `http://www.linternaute.com/expression/recherche`

[10] Comme `http://www.proverbes.free.fr/rechprov.php`

[11] 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 (par exemple, le critère de pureté de Gini) 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

4 Outils de classification

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 (mot, étiquette 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 toutes les tâches de classification d'information textuelle), nous avons décidé d'utiliser une combinaison de différents classifieurs et de différents éléments de texte. Cette approche nous permet, en outre, d'en déduire facilement les mesures de confiance sur les hypothèses produites lors de l'étiquetage. Neuf systèmes de décision ont été implantés en utilisant les classifieurs présentés ci-bas et les différentes représentations présentées dans la section 3. Ainsi, 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. Parce que ces outils sont basés sur des algorithmes de classification différents avec des formats d'entrée différents, ils n'utilisent pas les mêmes éléments d'information afin de caractériser un concept. Une combinaison de plusieurs classifieurs utilisant différentes sources d'information en entrée peut permettre d'obtenir des résultats plus fiables, évaluée par des mesures de confiance basées sur les scores donnés par les classifieurs. Nous ferons ensuite une présentation brève des classifieurs utilisés.

4.1 LIA_SCT

LIA_SCT (Béchet *et al.*, 2000) est un classifieur basé sur les arbres de décisions sémantiques (*SCT-Semantic Classification Tree* (Kuhn & De Mori, 1995)). Il suit le principe d'un arbre de décision: à chaque nœud de l'arbre une question est posée qui subdivise l'ensemble de classification dans les nœuds fils jusqu'à la répartition finale de tous les éléments dans les feuilles de l'arbre. La nouveauté des SCT réside dans la construction des questions qui se fait à partir d'un ensemble d'expressions régulières basées sur une séquence de composants. Leur ordre dans le vecteur d'entrée a donc une importance. De plus, chaque composant peut se définir suivant différents niveaux d'abstraction (mots et POS par exemple) et d'autres paramètres plus globaux peuvent également intégrer le vecteur (nombre de mots du document par exemple). Lorsque l'arbre est construit, il prend des décisions sur la base de règles de classification statistique apprises sur ces expressions régulières. Lorsqu'un texte est classé dans une feuille, il est alors associé aux hypothèses conceptuelles de cette feuille selon leur probabilité. Dans LIA_SCT les textes sont représentés en lemmes.

4.2 BoosTexter

BoosTexter (Schapire & Singer, 2000) est un classifieur à large marge basé sur l'algorithme de boosting : *Adaboost* (Freund & Schapire, 1996). Le but de cet algorithme est d'améliorer la précision des règles de classification en combinant plusieurs hypothèses dites *faibles* ou peu précises. Une hypothèse faible est obtenue à chaque itération de l'algorithme de boosting qui travaille en re-pondérant de façon répétitive les exemples dans le jeu d'entraînement et en ré-exécutant l'algorithme d'apprentissage précisément sur ces données re-pondérées. Cela permet au système d'apprentissage faible de se concentrer sur les exemples les plus compliqués (ou problématiques). L'algorithme de *boosting* obtient ainsi un ensemble d'hypothèses faibles qui sont ensuite combinées en une seule règle de classification qui est un vote pondéré des hypothèses faibles et qui permet d'obtenir un score final pour chaque constituant de la liste des concepts. Les composants du vecteur d'entrée sont passés selon la technique du sac de mots (l'ordre des mots est irrelevant) et les éléments choisis par les classifieurs simples sont alors des n-grammes sur ces composants. Quatre de nos systèmes utilisent le classifieur *BoosTexter* :

- Système LIA_BOOST_BASELINE : la représentation d'un document se fait en mots. BoosTexter est appliqué en mode 3-grammes ;
- Système LIA_BOOST_BASESEED : chaque document est représenté en seeds, chaque seed est pondéré par son nombre d'occurrences, en mode uni-gramme ;

vocation est soit de corriger d'éventuelles coquilles, soit de généraliser une expression (par exemple remplacer les noms des mois par une entité abstraite `MOIS`). Nous avons proposé en DEFT'08 une modélisation plus élaborée qui apporte une réponse à la question : comment, au moyen des opérateurs de concaténation et d'alternance, inférer des automates probabilistes à partir d'un corpus étiqueté ? À l'issue d'une cinquantaine d'itérations nous avons produit automatiquement entre 15 000 et 20 000 règles de réécriture et entre 25 000 et 70 000 règles d'agglutination. Ces nombres permettent d'imaginer le temps et l'expertise nécessaires si nous avions dû produire manuellement ces règles.

- Système LIA_BOOST_SEED : chaque document est représenté par les mots et également par les *seeds* toujours pondérés par leur nombre d'occurrences, en mode uni-grammes ;
- Système LIA_BOOST_CHUNK : L'outil *LIA-TAGG*[12] est utilisé pour découper le document en un ensemble de syntagmes lemmatisés. Chaque syntagme contenant un *seed* ainsi que le syntagme précédent et suivant sont retenus comme représentation. Les autres syntagmes sont rejetés de la représentation du document. *BoosTexter* est appliqué en mode 3-grammes sur cette représentation.

4.3 SVM Nath_Torch

SVMTorch (Collobert et al.*, 2002)* est un classifieur basé sur les machines à support vectoriel (*Support Vector Machines –SVM–*) proposées par Vapnik (Vapnik, 1982; Vapnik, 1995). Les *SVM* permettent de construire un classifieur à valeurs réelles qui découpe le problème de classification en deux sous-problèmes : transformation non-linéaire des entrées et choix d'une séparation linéaire *optimale*. Les données sont d'abord projetées dans un espace de grande dimension où elles sont linéairement séparables selon une transformation basée sur un noyau linéaire, polynomial ou gaussien. Puis dans cet espace transformé, les classes sont séparées par des classifieurs linéaires qui déterminent un hyperplan séparant correctement toutes les données et maximisant la *marge*, la distance du point le plus proche à l'hyperplan. Elles offrent, en particulier, une bonne approximation du principe de minimisation du risque structurel (*i.e*: trouver une hypothèse h pour laquelle la probabilité que h soit fausse sur un exemple non-vu et extrait aléatoirement du corpus de test soit minimale). Dans nos expériences, la technique la plus simple du sac de mots est utilisée: un document est représenté comme un vecteur dont chaque composante correspond à une entrée du lexique de l'application et chaque composante a pour valeur le nombre d'occurrences de l'entrée lexicale correspondant dans le texte. Le système (LIA_NATH_TORCH) est obtenu avec *SVMTorch*. Le vecteur d'entrée est représenté par le lexique des *seeds*.

4.4 Timble

Timble (Daelemans et al.*, 2004)* est un classifieur implémentant plusieurs techniques de *Memory-Based Learning –MBL–*. Ces techniques, descendantes directes de l'approche classique des k-plus-proches-voisins (*K Nearest Neighbor k-NN*) appliquée à la classification, ont prouvé leur efficacité dans un large nombre de tâches de traitement du langage naturel. Le paramétrage par défaut de *TiMBL* est un algorithme *MLB* qui construit une base de données d'*instances de base* lors de la phase d'entraînement. Comme pour *SVM-Torch*, une instance est un vecteur de taille fixe dont les composantes sont les entrées du lexique ayant pour valeur le nombre d'occurrences dans le document. À cela s'ajoute une composante indiquant quelle est la classe à associer à ce vecteur de paires { caractéristique-valeur }. Lorsque la base de données est construite, une nouvelle instance est classée par comparaison avec toutes les instances existantes dans la base, en calculant la distance de celle-ci par rapport à chaque instance en mémoire. Par défaut, *TiMBL* résout l'algorithme *1-NN* avec la métrique *Overlap Metric* qui compte simplement le nombre de composantes ayant une valeur différente dans chacun des 2 vecteurs comparés. Cette métrique est améliorée par l'*Information Gain –IG–* introduit par (Quinlan, 1986; Quinlan, 1993) qui permet de mesurer la pertinence de chaque composante du vecteur. Le système LIA_TIMBLE est formé de l'outil *TiMBL* appliqué sur les *seeds*.

4.5 Modélisation probabiliste uni-lemme et familles de mots

Nous avons voulu simplifier au maximum un classifieur et savoir si les modèles n-grammes avec $n > 1$ apportent vraimment des éléments discriminants. Nous avons décidé d'implanter un classifieur incorporant des techniques élementaires sur les n-lemmes. Ces techniques, descendantes directes de l'approche probabiliste (Mani & Maybury, 1999) appliquées à la classification de texte, ont prouvé leur efficacité dans le défi précédent (El-Bèze *et al.*, 2005).

Les textes ont été filtrés légèrement (afin de garder notamment des petites tournures comme la voix passive, les formes interrogatives ou exclamatives), un processus d'agregation de mots composés, puis regroupés dans des mots de la même famille (via un dictionnaire d'environ 300 000 formes). Ce processus comporte un regroupement et lemmatisation particulièrs. Ainsi, des mots tels que : *chantaient*, *chant*, *chantons*, et même *chanteurs* et *chanteuses*

[12] `http://lia.univ-avignon.fr/fileadmin/documents/Users/Intranet/chercheurs/bechet/download_fred.html`

seront ramenés au lemme **chanter**, ce qui diffère d'une lemmatisation classique. Nous avons limité notre modèle à $n = 1$, soit des uni-lemmes, ce qui nous évite de calculer beaucoup de coefficients de lissage. Nous avons transformé donc chaque document en un sac d'uni-lemmes. Puis nous avons calculé la classe d'appartenance d'un document comme :

$$P_t(w) \approx \prod_i \lambda_1 P_t(w_i) + \lambda_0 U_0 \tag{7}$$

Nous avons appliqué ce modèle d'uni-lemmes à tous les corpus, sans faire d'autres traitements particulières.

4.6 Modélisation sélon la théorie de l'information

Nous avons envisagé ici de recourir à une modélisation somme toute classique en théorie de l'information, tout en cherchant à y intégrer quelques unes des spécificités du problème. La formulation que nous avons retenue initialement se rapproche de celle que nous avions employée lors d'un précédent DEFT (El-Bèze *et al.*, 2005).

$$\widetilde{t} = Arg_t \max P(t) \times P(w|t) = Arg_t \max P(t) \times P_t(w) \tag{8}$$

L'étiquette t pouvant prendre ses valeurs dans un ensemble de cardinal réduit à 2 ou 3 éléments [0-1] ou [0-2], a priori le problème pourrait paraître simple, et la quantité des données fournies suffisante pour bien apprendre les modèles. Même si le vocabulaire propre aux différents corpus n'est pas si grand (entre 9ă000 mots différents pour le plus petit corpus et 50ă000 pour le plus grand), il reste que certaines entrées sont assez peu représentées. Aussi dans la lignée de ce qui se fait habituellement pour calculer la valeur du second terme de l'équation 8 nous avons opté pour un lissage de modèles n-lemmes (n allant de 0 à 3).

$$P_t(w) \approx \prod_i \lambda_3 P_t(w_i|w_{i-2}w_{i-1}) + \lambda_2 P_t(w_i|w_{i-1}) + \lambda_1 P_t(w_i) + \lambda_0 U_0 \tag{9}$$

L'originalité de la modélisation que nous nous sommes proposés d'employer dans le cadre de DEFT'07 réside essentiellement dans les aspects discriminants du modèle. Par manque de place, il ne nous est pas possible de détailler ici les différentes caractéristiques de cette nouvelle approche. Cela sera fait lors d'une publication ultérieure. Mais nous pouvons en dire au moins quelques mots. Lors de l'apprentissage, les comptes des n-lemmes sont rééchelonnés en proportion de leur pouvoir discriminant. Ce dernier est estimé selon un point de vue complémentaire au critère d'impureté de Gini selon la formule suivante.

$$G(w,h) \approx \sum_i P_t^2(t|w,h) \tag{10}$$

Les entrées w et leurs contextes gauches h qui ne sont apparus qu'avec une étiquette donnée t et pas une autre, ont un pouvoir discriminant égal à 1. Ce critère a été lissé avec un sous-critère G' permettant de favoriser (certes dans une moindre mesure que G) les couples (w,h) qui n'apparaissent que dans 2 étiquettes sur 3. Notons tout d'abord que l'emploi de tels critères discriminants est une façon de pallier le fait que l'apprentissage par recherche d'un maximum de vraisemblance ne correspond pas vraiment aux données du problème. Deuxièmement, il est aisé de comprendre combien un regroupement massif des entrées lexicales par le biais des collocations (cf. section 3) peut avoir un effet déterminant sur le nombre des événements à coefficient discriminant élevé. Ces deux remarques visent à souligner que sur ce point particulier le fameux croissement entre methode symbolique et numérique a son mot à dire. En dernier lieu, nous avons aussi adapté le calcul du premier terme $P(t)$ de l'équation 8 en combinant la fréquence relative de l'étiquette t avec la probabilité de cette même étiquette sachant la longueur du texte traité. Pour cela, nous avons eu recours à la loi Normale.

5 Résultats et discussion

5.1 Validation croisée

Afin de tester nos méthodes et de règler leurs paramètres, nous avons scindé l'ensemble d'apprentissage (A) de chaque corpus en cinq sous-ensembles approximativement de la même taille (en nombre de textes à traiter). 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 modèle M sur *E_train*;
  - *erreur* $+=$ évaluation de M sur *E_test*;

À l'issue 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, l'avantage nous est donné d'avoir testé ces méthodes sur l'ensemble des données disponibles, tout 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 allons présenter nos résultats en deux items : d'abord ceux obtenus sur les ensembles de devéloppement (D) et de validation (V) où nous avons paramétré nos systèmes, et ensuite les résultats sur les données de test (T) en appliquant les algorithmes.

5.2 Évaluation sur les corpus de devéloppement (D) et de validation (V)

Le decoupage des corpus en cinq sous-ensembles de devéloppement (D) est le fruit d'un tirage aléatoire. Ce découpage permet, selon nous, d'éviter de régler les algorithmes sur un seul ensemble d'apprentissage (et un autre seul de test), ce qui pourrait conduire à deux travers, le biais expérimental et/ou le phenomène de surapprentissage. Nous présentons aux tableaux 2 (**aVoiraLire**), 3 (**jeuxvideo**), 4 (**relectures**) et 5 (**débats**) des statistiques des sous-ensembles de devéloppement (T) et de validation (V) en fonction de leurs classes pour chacun des corpus.

Corpus **aVoiraLire**							
	Total	Classe 0		Classe 1		Classe 2	
Ensembles (D)	Textes	Textes	%	Textes	%	Textes	%
1	1 660	231	13,92	486	29,27	943	56,81
2	1 659	249	15,01	494	29,78	916	55,21
3	1 659	244	14,71	498	30,02	917	55,27
4	1 659	248	14,95	490	29,54	921	55,51
5	1 659	264	15,91	492	29,66	903	54,43
Ensembles (V)	textes	Textes	%	Textes	%	Textes	%
1	414	45	10,84	123	29,64	247	59,52
2	415	61	14,70	123	30,12	229	55,18
3	415	65	15,66	117	28,19	233	56,14
4	415	60	14,46	121	29,16	234	56,39
5	415	78	18,84	129	31,16	207	50,00

Table 2: Statistiques par classe sur les ensembles de devéloppement (D) et de validation (V), **aVoiraLire**.

Sur la figure 1, nous montrons le F-score du système de fusion sur les quatre corpus (V). L'apprentissage a été réalisé sur les ensembles de devéloppement et le F-score a été calculé sur les cinq ensembles de validation (V). On peut constater que le corpus de relectures d'articles scientifiques est le plus difficile à traiter. En effet, ce corpus comporte le plus petit nombre de textes (environ 704 en devéloppement et 177 en validation). Il est aussi très dur à classer étant donnée des particularités propres à ce corpus que nos avons détecté : les arbitres corrigent souvent le texte des articles à la volée (directement dans leurs commentaires), ce qui est une introduction de bruit. Nous y reviendrons lors de la discussion de nos résultats.

5.3 Évaluation sur les corpus de test

Nous avons défini l'ensemble d'apprentissage $\{A_j\} = \{D_j\} \cup \{V_j\}$; $j =$**{aVoiraLire**, **jeuxvideo**, **relectures**, **débats**}. Le tableau 7 montre les statistiques par classe pour les quatre corpus de test (T) et d'apprentissage (A). On peut constater que la distribution des données en apprentissage et en test est très homogène, ce qui en principe, facilite la tâche de n'importe quel classifieur.

Corpus **jeuxvideo**							
	Total	Classe 0		Classe 1		Classe 2	
Ensembles (D)	textes	Textes	%	Textes	%	Textes	%
1	2 032	412	20,28	917	45,13	703	34,59
2	2 029	395	19,47	905	44,60	729	35,93
3	2 029	350	17,25	951	46,87	728	35,88
4	2 029	467	23,02	946	46,62	616	30,36
5	2 029	364	17,94	945	46,57	720	35,48
Ensembles (V)	textes	Textes	%	Textes	%	Textes	%
1	505	133	26,18	221	43,50	154	30,31
2	508	30	5,91	220	43,31	258	50,79
3	508	147	28,94	215	42,32	146	28,74
4	508	102	20,08	261	51,38	145	28,54
5	508	85	16,83	249	49,31	171	33,86

Table 3: Statistiques par classe sur les ensembles de devéloppement (D) et de validation (V), **jeuxvideo**.

Corpus **relectures**							
	Total	Classe 0		Classe 1		Classe 2	
Ensembles (D)	textes	Textes	%	Textes	%	Textes	%
1	708	151	21,33	262	37,01	295	41,67
2	704	179	25,43	208	29,55	317	45,03
3	704	184	26,14	200	28,41	320	45,46
4	704	206	29,26	214	30,40	284	40,34
5	704	188	26,70	228	32,39	288	40,91
Ensembles (V)	textes	Textes	%	Textes	%	Textes	%
1	173	39	22,03	50	28,25	88	49,72
2	177	21	11,86	64	36,16	92	51,98
3	177	43	24,29	78	44,07	56	31,64
4	177	48	27,12	70	39,55	59	33,33
5	177	76	43,93	16	9,25	81	46,82

Table 4: Statistiques par classe sur les ensembles de devéloppement (D) et de validation (V), **relectures**.

La figure 2 montre les performances en F-score de chacun de nos classifieurs, ainsi que leurs moyennes sur les quatre ensembles de test. On constate que les classifieurs LIA_TIMBLE et LIA_SCT ont les perfomances les plus basses.

La figure 3 illustre les performances en F-score d'une fusion *incrémentale* des méthodes ajoutées. Cependant, l'ordre affiché n'a strictement aucun impact dans la fusion finale : il a été choisi uniquement pour mieux illustrer les résultats. On peut voir que nos résultats se placent bien au dessus de la moyenne des équipes participantes dans le défi DEFT'07, tous corpus confondus.

Sur la figure 4 nous montrons une comparaison du F-score de l'ensemble de validation (V) vs. celui de test (T), sur les quatre corpus. On peut constater la remarquable coïncidence entre les deux, ce qui signifie que notre stratégie d'apprentissage et de validation sur cinq sous-ensembles et de fusion de plusieurs classifieurs a bien fonctionné.

5.4 Discussion

Nous avons constaté que l'utilisation de collocations et réécriture (cf. section 3.1) permet d'augmenter les perfomances des méthodes. Par exemple, avec la méthode probabiliste à base d'uni-lemmes sur le corpus de validation nous sommes passés de 1 285 à 1 310 bien classés (F=57,41 → 58,89) dans le corpus **aVoiraLire**, de 1801 à 1916 (F=70,77 → 75,15) en **jeuxvideo**, de 445 à 455 (F=48,36 → 49,53) en **relectures** et de 10 364 à 11 893 (F=62,21 → 67,12) en **débats**. Dans le corpus de test les gains sont aussi non négligeables. Nous sommes passés en **aVoirAlire** de 863 à 860 (F=57,40 → 56.32); de 7530 à 7635 (F=65,82 → 66,88) en **débats** ; de 1169 à 1205 (F=69,51 → 71,48) en **jeuxvideo** et de 317 à 313 (F=51,81 → 52,04) en **relectures**. Ceci confirme l'hypothèse

Corpus **débats**					
	Total	Classe 0		Classe 1	
Ensembles (D)	textes	Textes	%	Textes	%
1	13 840	7 893	57,03	5 947	42,97
2	13 839	8 525	61,60	5 314	38,40
3	13 839	8 710	62,94	5 129	37,06
4	13 839	7 587	54,82	4 890	35,33
5	13 839	7 913	57,18	5 926	42,82
Ensembles (V)	textes	Textes	%	Textes	%
1	3 459	2 487	71,88	973	28,12
2	3 460	1 841	53,21	1 619	46,79
3	3 460	1 690	48,84	1 770	51,16
4	3 460	1 875	58,39	1 585	56,54
5	3 460	2 507	72,48	952	27,52

Table 5: Statistiques par classe sur les ensembles de devéloppement (D) et de validation (V), **débats**.

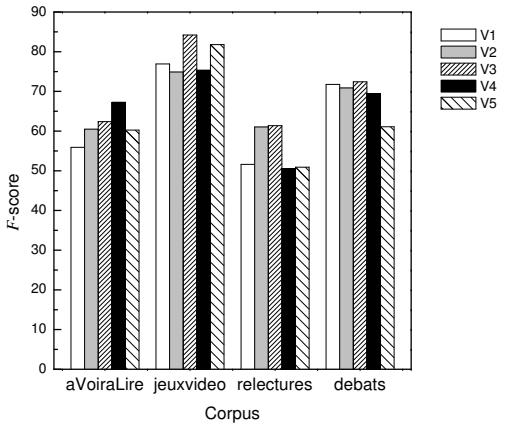

Figure 1: F-score obtenu par l'algorithme de fusion sur les cinq ensembles de validation (V). Nous affichons des résultats regroupés par corpus.

que la réécriture aide à mieux capturer la polarité des avis.

Nous avons realisé une analyse *post-mortem* de nos résultats. Nous présentons ci-bas, quelques exemples de notices qui ont été mal classés par nos systèmes. Nous avons delibérement gardé les notices dans leur état : majuscules mal placés et même avec les fautes d'ortographe ou de grammaire. En particulier, nous avons décidé de montrer majoritairement, des avis d'opinion venant du corpus de relectures d'articles scientifiques, corpus qui avait posé plus de difficultés aux algorithmes (F-score plus faible) que les autres. Par exemple, considérez la notice 3:36 (**relectures**) :

> ***3:36 relectures***
> *L'idée d'appliquer les méthodes de classification pour définir des classes homogènes de pages web est assez originale par contre, la méthodologie appliquée est classique. Je recommande donc un « weak accept » pour cet article.*

Nos systèmes l'ont classé 1 (accepté avec des modifications majeures), et après une lecture directe, on pourrait effectivement en déduire que la classe est 1 alors que la référence est 2 (accepté).

Notice 3:2 (**relectures**). L'article a été accepté mais notre système le classe comme rejeté. Il comporte beaucoup d'expressions négatives comme : " *parties de l'article me paraissent déséquilibrées* ", " *Le travail me paraît inachevé*

Corpus	Précision	Rappel	F-score	Correctes	Total
aVoiraLire (V)	0,6419	0,5678	0,6026	1 385	2 074
jeuxvideo (V)	**0,8005**	**0,7730**	**0,7865**	2 005	2 537
relectures (V)	0,5586	0,5452	0,5518	510	881
débats (V)	0,7265	0,7079	0,7171	12 761	17 299

Table 6: Précision, Rappel et F-score obtenus par notre méthode de fusion, sur les corpus de validation (V).

Corpus de test	Total textes	Classe 0 Textes	Classe 0 %	Classe 1 Textes	Classe 1 %	Classe 2 Textes	Classe 2 %
aVoiraLire (T)	1 386	207	14,94	411	29,65	768	55,41
jeuxvideo (T)	1 694	332	19,60	779	45,99	583	34,42
relectures (T)	603	157	26,04	190	31,51	256	42,45
débats (T)	11 533	6 572	56,98	4 961	43,02	⊘	⊘
Corpus d'apprentissage	Total textes	Classe 0 Textes	Classe 0 %	Classe 1 Textes	Classe 1 %	Classe 2 Textes	Classe 2 %
aVoiraLire (A)	2 074	309	14,90	615	29,65	1150	55,45
jeuxvideo (A)	2 537	497	19,59	1166	45,96	874	34,45
relectures (A)	881	227	25,77	278	31,55	376	42,68
débats (A)	17 299	10 400	60,12	6 899	39,88	⊘	⊘

Table 7: Statistiques par classe sur les quatre corpus d'apprentissage (A) et de test (T).

", " *la nouvelle méthode proposée pose des problèmes complexes ... qui ne sont pas traités dans ce papier* ", cependant il a été accepté.

> ***3:2 relectures***
> ***Les différentes parties de l'article me paraissent déséquilibrées****. Les auteurs présentent d'abord un état de l'art dans le domaine de la visualisation des connaissances dans les systèmes de gestion de connaissances. Ils décrivent ensuite le serveur <anonyme /> et sa représentation des connaissances sous forme d'arbre en section 3 et une partie de la section 4. L'approche proposée par les auteurs (représentation par graphes n'est présentée qu'en 4.2 sur moins d'une page).* ***Les problèmes posés par cette méthode sont survolés par les auteurs****, ils font référence aux différents papiers traitant de ces problèmes et* ***n'exposent pas du tout*** *les heuristiques choisies dans leurs approches.* ***Le travail me paraît inachevé****, et* ***la nouvelle méthode proposée pose des problèmes complexes*** *au niveau de la construction de ce graphe* ***qui ne sont pas traités dans ce papier****.*

Notice 3:6 (**relectures**). L'article a été accepté, alors que notre système le classe comme rejeté. L'article arbitré est peut-être trop court, mais la relecture qui le concerne, elle l'est aussi :

> ***3.6 relectures***
> *Article trop court pour pouvoir être jugé. Je suggère de le mettre en POster si cela est prévu.*

Pour la notice 3:9 (**relectures**) on décèle le même problème : l'article est accepté alors que le système le rejette. Constatons que l'arbitre focalise uniquement sur des remarques de forme :

> ***3:9 relectures***
> *Question : comment est construit le réseau bayesien ? Un peu bref ici... Remarques de forme : page 2, 4ème ligne, " comprend " 5ème ligne : "annotées" ou "annoté" page 3 : revoir la phrase confuse précédant le tableau dernière ligne, répétition de "permet" page 5 : 7ème ligne accorder "diagnostiqué" et "visé" avec "états" ou avec "connaissances"*

Pour le texte de la notice 3:567 (**relectures**), l'article en question est rejeté alors que le système l'accepte. Phrases encourageantes au début finisent par être mitigées. Beaucoup d'expressions positives (" *bien organisé* ", " *facile à suivre* ", " *bibliographie est plutôt complète* ", "*solution proposée et interessante et originale* ", " *La soumission*

Corpus	Précision	Rappel	F-score	Correctes	Total
aVoiraLire (T)	0,6540	0,5590	0,6028	931	1 386
jeuxvideo (T)	**0,8114**	**0,7555**	**0,7824**	1 333	1 694
relectures (T)	0,5689	0,5565	0,5626	353	603
débats (T)	0,7307	0,7096	0,7200	8 403	11 533

Table 8: Précision, Rappel et F-score obtenus par notre méthode de fusion, sur les corpus de test (T).

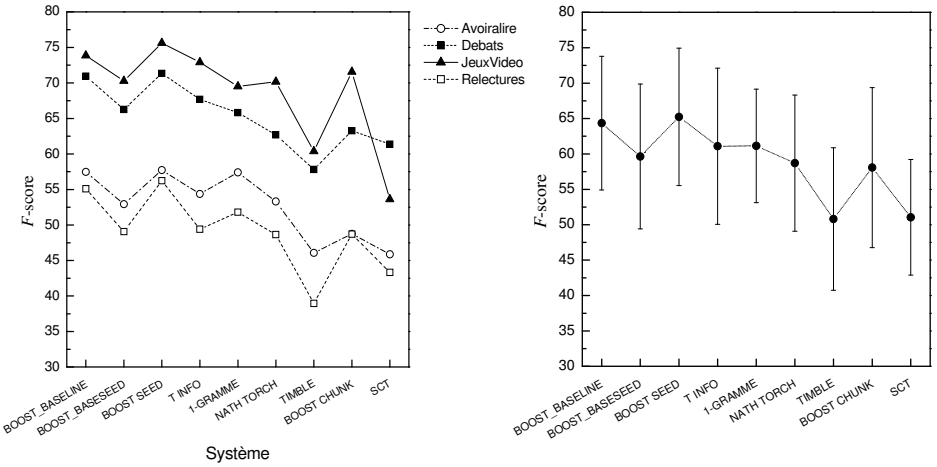

Figure 2: F-score de chacune des méthodes sur les quatre corpus de test, ainsi que leurs moyennes.

d'une nouvelle version ... sera intéressante ") n'arrivent pas a renverser le rejet.

> ***3:567 relectures***
> *Commentaire : L'article est* ***plutôt bien organisé*** *(malgré de trop nombreux chapitres), le cheminement de* ***la logique est facile à suivre****. Cependant il y a de trop nombreuses fautes de français ainsi que d'anglais dans le résumé.* ***La bibliographie est plutôt complète****.* ***La solution proposée et interessante et originale****, cependant des notions semblent mal maîtrisées. Ainsi dans la section 8, la phrase «Cette convergence ne vient pas des algorithmes génétiques de manière intrinsèque, mais de l'astuce algorithmique visant à conserver systématiquement le meilleur individu dans la population » démontre une incompréhension du fonctionnement même d'un algorithme génétique. La soumission d'une nouvelle version modifiée de cet article, présentant également les premiers résultats obtenus avec le prototype à venir* ***sera intéressante pour la communauté****.*
> *Références : Originalité : Importance : Exactitude : Rédaction :*

Pour finir, étudions une notice du corpus de films, livres et spectacles : le texte 1:10 du corpus **aVoiraLire**. Malgré des expressions avec une certaine charge positive, telles que : "*...est un événement* ", "*Agréable surprise* " ou encore "*l'image d'une cohérence artistique retrouvée* ", la notice reste difficile à classer. Évidemment notre système se trompe. Mettons à notre tour le lecteur au défi de trouver la véritable classe[13].

[13] Si vous êtiez tenté de le mettre en classe 2 (bien), sachez que la classe véritable est la 1 (moyen).

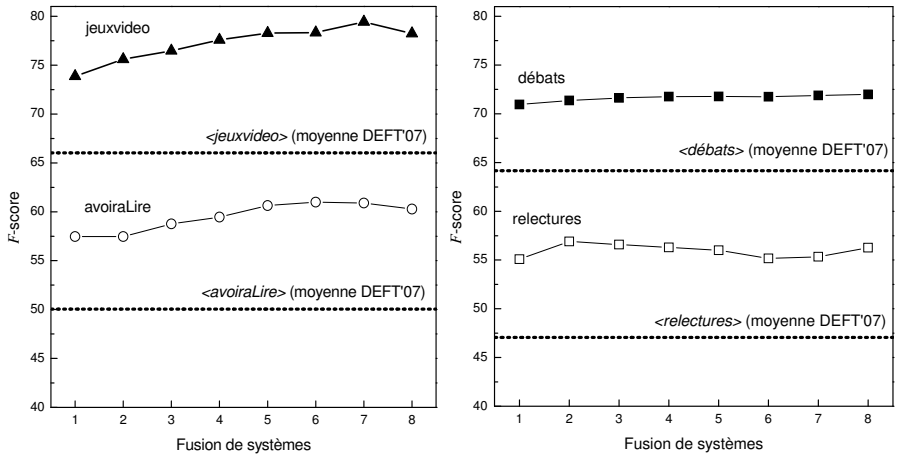

Figure 3: F-score de la fusion suivant nos neuf méthodes ajoutées. 1: BOOST_BASELINE ; 2: (1) ∪ BOOST_BASESEED ∪ BOOST_SEED ; 3: (2) ∪ Théorie information ; 4: (3) ∪ 1-gramme ; 5: (4) ∪ NATH_TORCH ; 6: (5) ∪ TIMBLE ; 7: (6) ∪ BOOST_CHUNK ; 8: (7) ∪ SCT

aVoiraLire 1:10
Depuis trente-six ans, chaque nouvelle production de David Bowie est un événement. Heathen , ne fait pas exception à cette règle. On reconnaît instantanément la patte de son vieux compère Tony Visconti. La voix de Bowie est mise en avant. Agréable surprise, surtout qu'elle n'a rien perdu depuis ses débuts. Là, commence le voyage. Ambiance, mélange dosé des instruments. Dès l'ouverture de l'album avec Sunday , un sentiment étrange nous envahit. Comme si Bowie venait de rentrer d'un voyage expérimental au coeur même de la musique. Retour aux sources. L'ensemble du disque est rythmé par cette pulsation dont le duo a le secret. Le tout saupoudré de quelques pincées d'électronique. Le groupe est réduit au minimum. Outre Bowie en chef d'orchestre et Visconti, David Torn ponctue les compositions de ses guitares aventureuses et Matt Chamberlain apporte de l'âme à la rythmique. Un quatuor à cordes fait une apparition, comme Pete Townshend (The Who) ou Dave Grohl (ex-batteur de Nirvana). Avec trois reprises réarrangées et neuf compositions originales, le 25e album de Bowie est à l'image d'une cohérence artistique retrouvée.

6 Conclusion et perspectives

La classification de documents textuels en fonction des tendances d'opinion qu'ils expriment reste une tâche très difficile, même pour une personne. La lecture directe ne suffit pas toujours pour se forger un avis et privilégier une classification pa rapport à une autre. Nous avons décidé d'utiliser des approches de représentation numériques et probabilistes, afin de rester aussi indépendant que possible des sujets traités. Nos méthodes ont fait leur preuve. Nous avons confirmé l'hypothèse que la réécriture (normalisation graphique) et l'agglutination par (collocations) aident à capturer le sens des avis. Ceci se traduit par un gain important de performances. Probablement la méthode de normalisation et d'agglutination utilisée en DEFT'08 aurait eu un impact favorable pour les performances. Cela reste un sujet à étudier. Nous avons présenté une stratégie de fusion de méthodes assez simple. Celle ci s'est avérée robuste et performante. Une stratégie similaire a été utilisée lors du défi DEFT'08. Dans ce dernier cas, le défi comportait la prise en compte des variations en genre et en thème dans un système de classification automatique[14]

[14]Des corpus électronique issus du journal *Le Monde* ou du site *Wikipedia*. Les tâches ont été les suivantes : *Tâche 1 Catégorie* - Reconnaissance de la catégorie thématique parmi les quatre classes : *ART* (Art), *ECO* (Économie), *SPO* (Sports), *TEL* (Télévision) ; *Tâche 1 Genre* - Reconnaissance du genre parmi les deux classes : *W* (Wikipédia), *LM* (Le Monde) ; *Tâche 2 Catégorie* - Reconnaissance de la catégorie

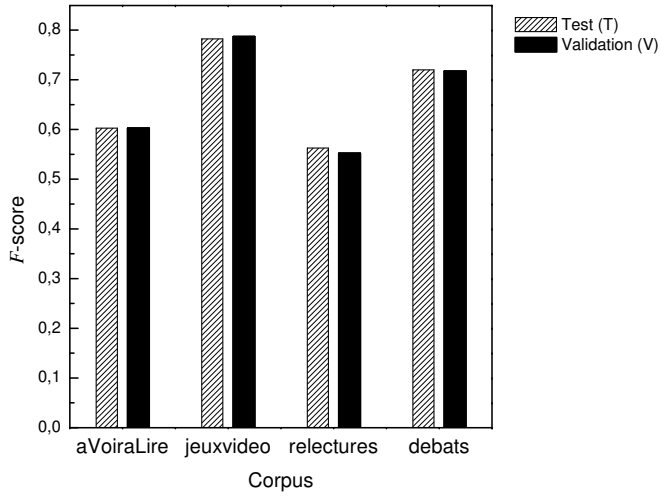

Figure 4: Comparaison du F-score de l'ensemble de validation (V) vs. celui de test (T) pour chacun des corpus, obtenu par notre système de fusion.

Cette stratégie nous a permis de remporter également le défi DEFT'08, en ex-equo avec deux autres équipes. Nos F-scores sont au-dessus des moyennes sur les quatre corpus de test, notamment sur celui de **jeuxvideo**. La stratégie de fusion a montré des résultats supérieurs à n'importe laquelle des méthodes individuelles. La dégradation en précision et rappel reste faible, même si nous n'avons pas écarté de la fusion des méthodes peut-être moins adaptées à cette problématique. La fusion est donc une façon robuste de combiner plusieurs classifieurs. Il faut souligner la remarquable équivalence entre les résultats obtenus lors de l'apprentissage et la prédiction sur les ensembles de test : à un point près de différence. Le module de fusion n'a pas vraiment été optimisé, car un même poids a été attribué au vote de chaque système. En effet, 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. En DEFT'07 et 08 nous avons choisi de privilégier les méthodes simples (Béchet *et al.*, 2008). Ceci peut ouvrir facilement la voie à une possible amélioration. D'abord, on aurait pu utiliser utiliser une méthode exhaustive de recherche de poids. De ce fait, on aurait pu utiliser, par exemple, la moyenne pondérée des scores de confiance, avec un jeu de coefficients choisi pour minimiser l'erreur sur le corpus d'apprentissage. Une autre façon de trouver les poids, est au moyen des techniques d'apprentissage. Des techniques d'optimisation ou probabilistes pourraitent être intégrées afin de regler automatiquement les paramètres de pondération des juges, tel qu'on a fait lors de la campagne DEFT'05 (El-Bèze *et al.*, 2005) avec un perceptron optimale (Torres-Moreno *et al.*, 2002; Gordon & Berchier, 1993). Il faut dire qu'un système modulaire comme le notre, peut accepter l'ajout d'autres méthodes de classification (comme celle que nous avons présenté en DEFT'08 qui classe les textes en utilisant uniquement les signes de ponctuation et les mots vides de signification) dans le but d'accroitre les performances. Il restent donc des voies à explorer. En ce qui concerne l'analyse detaillée des resultats, le corpus de **relectures** des articles scientifiques reste de loin le plus difficile à traiter. Nous avions déjà avancé l'hypothèse qu'en raison du faible nombre de notices, il serait difficile à classer. Il y a d'autres facteurs qui interviennent également. Les relectures souvent comportent, dans le corps du texte, des corrections adressées aux auteurs. Ceci vient bruiter nos classifieurs. Les relectures sont parfois trop courtes, ou bien elles ont été redigées par des arbitres non francophones (encore une autre source de bruit) ou bien elles contienent beaucoup d'anglicismes (*weak acceptation, boosting, support vector,...*). Un autre facteur, peut-être plus subtil : un article peut être lu par plusieurs arbitres (deux, trois voire plus) qui émettent des avis opposés. Dans une situation où les arbitres A et B acceptent l'article et un troisième C le refuse, normalement l'article doit être accepté. Donc, dans le corpus **relectures**, l'avis de C sera asimilé à la classe acceptée, et cela malgré son avis négatif.

thématique parmi les six classes : *FRA* (Politique française), *INT* (International), *LIV* (Littérature), *SCI* (Sciences), *SOC* (Société).

Remerciements

Nous remercions Martine Hurault-Plantet et Cyril Grouin (LIMSI-LIR) ainsi que le comité d'organisation de DEFT'09.

Références

Azé J., Heitz T., Mela A., Mezaour A.-D., Peinl P. et Roche M. (2006). Préparation de DEFT'06 (DÉfi Fouille de Textes). In *Proc. of Atelier DEFT'06*, volume 2.

Azé J. et Roche M. (2005). Présentation de l'atelier DEFT'05. In *Proc. of TALN 2005 - Atelier DEFT'05*, volume 2, p. 99–111.

Béchet F., El-Bèze M. et Torres-Moreno J.-M. (2008). En finir avec la confusion des genres pour mieux séparer les thèmes. In *DEFT'08*, p. 161–170.

Béchet F., Nasr A. et Genet F. (2000). Tagging unknown proper names using decision trees. In *38th Annual Meeting of the Association for Computational Linguistics, Hong-Kong, China*, p. 77–84.

Collobert R., Bengio S. et Mariéthoz J. (2002). Torch: a modular machine learning software library. In *Technical Report IDIAP-RR02-46, IDIAP*.

Daelemans W., Zavrel J., van der Sloot K. et van den Bosch A. (2004). Timbl: Tilburg memory based learner, version 5.1, reference guide. *ILK Research Group Technical Report Series*, p. 04–02.

El-Bèze M., Torres-Moreno J.-M. et Béchet F. (2005). Peut-on rendre automatiquement à César ce qui lui appartient ? Application au jeu du Chirand-Mitterrac. In *TALN 2005 - Atelier DEFT'05*, volume 2, p. 125–134.

Freund Y. et Schapire R. E. (1996). Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, p. 148–156.

Gordon M. et Berchier D. (1993). Minimerror: A perceptron learning rule that finds the optimal weights. In M. Verleysen, Ed., *ESANN*, p. 105–110, Brussels: D facto.

Hatzivassiloglou V. et McKeown K. R. (1997). Predicting the semantic orientation of adjectives. In *European chapter of the ACL*, p. 174–181, Morristown, NJ, USA: ACL.

Kuhn R. et De Mori R. (1995). The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(5), 449–460.

Mani I. et Maybury M. T. (1999). *Advances in Automatic Text Summarization*. MIT Press.

Quinlan J. (1986). Induction of decision trees. *Machine Learning*, **1**(1), 81–106.

Quinlan J. (1993). *C4. 5: Programs for Machine Learning*. Morgan Kaufmann.

Schapire R. E. et Singer Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning*, **39**, 135–168.

Torres-Moreno J.-M., Aguilar J. et Gordon M. (2002). Finding the number minimum of errors in N-dimensional parity problem with a linear perceptron. *Neural Processing Letters*, **1**, 201–210.

Vapnik V. N. (1982). *Estimation of Dependences Based on Empirical Data*. New York, USA: Springer-Verlag Inc.

Vapnik V. N. (1995). *The nature of statistical learning theory*. New York, USA: Springer-Verlag Inc.

Wilson T., Wiebe J. et Hoffmann P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*, p. 347–354, Vancouver, Canada.