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 les 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 choisies par le Laboratoire Informatique d'Avignon (LIA) pour la participation à la campagne DEFT'08. Trois approches sont décrites ainsi que diverses méthodes de fusion, telles que celles mises en œuvre dans les trois soumissions envoyées pour le défi. Mis à part les méthodes de classification employées, les principales caractéristiques de notre méthodologie sont les suivantes :

- pré-traitement des données basé sur une chaîne d'outils linguistiques de *shallow parsing* ;
- méthodologie de développement et de réglage 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.). Comme aucune méthode générique n'a fait 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 diverses représentations. Cette approche nous permet, en outre, d'estimer facilement les mesures de confiance sur les hypothèses produites lors de l'étiquetage. Plusieurs systèmes ont été développés, utilisant diverses 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 par la fusion de leurs résultats. Nous allons présenter les trois méthodes utilisées, en commençant par détailler le pré-traitement des données.

2.1 Pré-traitement des données

Chaque document traité durant la campagne DEFT correspond à un texte *brut* issu du corpus électronique du journal *Le Monde* ou du site *Wikipedia*. En premier, nous avons appliqué sur ces documents une chaîne de traitement d'étiquetage de surface (*shallow parsing*) développée au LIA[1]. Cette chaîne est composée des modules suivants :
tokeniseur → *étiqueteur morpho-syntaxique* → *lemmatiseur* → *extracteur d'entités-nommées*.

Le lexique utilisé par les différents programmes contient 270K entrées. L'étiqueteur morpho-syntaxique utilise un jeu de 105 étiquettes et implémente une approche probabiliste basée sur les Chaînes de Markov Cachées. L'extracteur d'entités nommées a été développé dans le cadre de la campagne d'évaluation ESTER pour la détection d'entités nommées se trouvant dans les retranscriptions d'émissions radiophoniques d'information. Il repose également sur une approche statistique, à base de Champs Conditionnels Aléatoires (*Conditional Random Fields*), et utilise un jeu de 8 étiquettes : personnes, lieux, organisations, entité géographique/socio-politique, montant, temps, produits, bâtiments.

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 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

[1] Ces outils sont téléchargeables à l'adresse :
`http ://lia.univ-avignon.fr/personnel/fiches-personnelles/bechet/`

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. Ainsi, nous avons obtenu des groupes de 3045 documents pour la tâche 1 et de 4711 documents 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 des 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 chaque terme. Pour répondre à cette attente, nous avons légèrement adapté le critère[3] d'impureté de Gini (cf. formule 0). 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 1) et que nous appelons critère de pureté de Gini $PG(i)$.

$$IG(i) = \sum_{t \neq j} P(j|i)P(t|i) = 1 - \sum_{j=1}^{k} P^2(j|i) \quad (0) \qquad PG(i) = 1 - IG(i) = \sum_{j=1}^{k} P^2(j|i) \quad (1)$$

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.

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

$PG'(i)$ 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 obtenue après avoir regroupé 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

[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.

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 CLA_W et CLA_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. 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 pour DEFT'07. Cette fois, notre objectif est 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 d'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, de cette façon, 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 faisaient plus l'objet d'une relecture par des correcteurs humains, ils étaient néanmoins lus et relus par leurs auteurs et donc plus propres que ceux de Wikipédia. De ce fait, la tentation d'appliquer des outils de correction orthographique (par exemple celui qui remplace le triplement d'une consonne par son doublement, ou supprime son redoublement `dreyff ?uss ? ->`

[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.

`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, quand la correction est faite pour cette tâche, nous avons choisi d'ajouter un terme fictif : `TYPOW`. Les erreurs d'accents sont aussi traitées, comme le montre l'exemple : `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 plus haut, 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 pour E_1LiA, 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 (Bahl *et al.*, 1988) proposent de recourir à la loi de Poisson, loi 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 afin 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 qui mène à la formule 3 déterminant la classe optimale $\hat{c}$ en fonction de la longueur moyenne $\mu(c)$ d'un article dans la classe c, et de $\mu(i, c)$ le nombre moyen de fois où le terme i apparaît dans un article de la classe c.

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

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 ont été estimées en s'inspirant du 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 aux spécificités de DEFT'08, 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 95,7. 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 en valeur absolue de plus de 1% : on passe de 86,1 à 87,2.

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 87,6. 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 quelques résultats concernant la tâche 2. Nous avons tiré de nos observations sur l'apprentissage des 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.

2.4 Classification par *Boosting* de classifieurs simples : l'approche E_2*LiA*

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

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

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, 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 aux classifieurs simples de se concentrer sur les exemples les plus problématiques. L'algorithme de boosting produit 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 expériences. 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 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).

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

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. À 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 IIIe et le IIe 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) permettait 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 rencontrés dans la phase de test restent des événements vus lors de l'apprentissage. Enfin, la ponctuation reste relativement stable entre les langues et, sauf exceptions rares, 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 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 des 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) :

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

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

combinée avec une interpolation simple :

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

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 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 réalisée. Les performances en test sont F-score=90,2 (genre) et F-score=81,3 (catégorie) pour la tâche 1 et F-score=84,4 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 de DEFT'08 où les corpus restent relativement distincts (sauf pour la classe Télévision). Nous avons décidé d'incorporer ce modèle dans notre fusion.

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

[10] En fait nous avons gardé tout ce qui n'est pas un mot. Ainsi, nous avons retenu tout symbole (au sens le plus large du terme) et pas uniquement les signes de ponctuation.

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 une forme de 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'apprentissage 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. De ce fait, 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 devait fusionner : au maximum 5 dans nos expériences. Évidemment, 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 les trois 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 optimisée
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 systématiquement sur toutes les tâches.

Méthode	E_1*LiA* (**S1**)	E_2*LiA*	E_3*LiA*	Fusion opt. app. (**S2**)	Fusion opt. test	Fusion **S3**
Tâche_1_CAT	85.2	85.9	81.3	88.3	88.3	85.4
Tâche_1_GENRE	95.8	97.9	90.2	98.1	98.4	98.0
Tâche_2_CAT	86.1	85.2	84.4	87.2	87.9	88.0

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 présentés dans le tableau 2 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. Les trois soumissions (**S1**, **S2** et **S3**) sont précisées. La soumission **S3** correspondant à la fusion entre les 3 systèmes présentés dans cet article et la fusion des systèmes "*jeunes chercheurs*" du LIA. La colonne *Fusion opt. app.* indique les résultats obtenus avec des coefficients de fusion optimisés sur le corpus d'apprentissage. On remarque qu'en optimisant ces coefficients sur le corpus de test (colonne *Fusion opt. test*), les gains sont faibles, ce qui valide la robustesse de notre méthode de fusion.

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 très haut niveau des performances nous amène à faire deux observations. La difficulté de la tâche a été certainement atténuée, dans un cas (le genre) par le petit nombre de classes et aussi par la séparabilité des classes hormis quelques confusions liées à l'étendue des sujets abordés dans les thématiques audiovisuelles (TEL) et sociétales (SOC). Quand on atteint un résultat au dessus de 98% il n'est pas évident de trouver les moyens de l'améliorer. Toutefois, 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.

Remerciements : nous remercions le comité d'organisation de la campagne de DEFT'08 et tout particulièrement Cyril Grouin (LIMSI-LIR) et Martine Hurault-Plantet (LIMSI-LIR).

Références

BAHL L., BAKIS R., DE SOUZA P. & MERCER R. (1988). Obtaining candidate words by polling in a large vocabulary speech recognition system. In *ICASSP-88*, p. 489–492.

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 *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. *J. 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 *TALN 2007-Atelier DEFT'07*, p. 119–133.