Système du LIA pour la campagne DEFT'10 : datation et localisation d'articles de presse francophones

Stanislas Oger[1] Mickael Rouvier[1] Nathalie Camelin[1] Rémy Kessler[1]
Fabrice Lefèvre[1] Juan-Manuel Torres-Moreno[1,2]
(1) Laboratoire Informatique d'Avignon, BP 91228, F-84911 Avignon, France
(2) École Polytechnique de Montréal, CP 6079, Montréal (Québec) H3C3A7 Canada
{nathalie.camelin, remy.kessler, stanislas.oger, mickael.rouvier, fabrice.lefevre,
juan-manuel.torres}@univ-avignon.fr

Résumé. Nous présentons dans cet article les systèmes développés au LIA pour la campagne d'évaluation DEFT'10. La campagne comporte deux tâches (ou *pistes*) distinctes : la première consiste à identifier la décennie de publication d'articles francophones entre 1800 et 1940 et la seconde à identifier le pays (France ou Quebec) et le journal dans lequel a été publié l'article parmi 4. Plusieurs systèmes basés sur des modèles probabilistes ont été développés pour chacune des tâches. Puis ces systèmes ont été fusionnés pour fournir une distribution globale sur l'ensemble des hypothèses, permettant une décision globale. La bonne robustesse des systèmes individuels et de leur fusion entre le corpus d'apprentissage et de test nous a permis d'obtenir de bons résultats, b ien que très contrastés selon les tâches.

Abstract. This paper describes the systems developed at LIA for the DEFT'10 evaluation campaign. This campaign includes two different tasks : (a) identifying the decade of publication of French-written articles, and (b) identifying the country where the articles were published and the journal. Several systems, all based on probabilistic models, were developed. A final fusion step provides an overall distribution on the hypotheses, allowing a better final decision. The good robustness of the individual systems and the fusion system between the training and testing corpora allowed us to obtain good results, although well contrasted over the various tasks.

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

Keywords: Stocahstic approaches, Machine learning, Text classification, DEFT challenge.

1 Introduction

La sixième édition de la campagne d'évaluation DEFT (DÉfi Fouille de Textes) a eu lieu au printemps 2010. Cette année encore un défi original en fouille de textes a été proposé à la communauté francophone avec pour objet deux tâches distinctes : les variations diachroniques et l'origine géographique de corpus de presse francophones.

La thématique Langage du Laboratoire d'Informatique d'Avignon (LIA) [1] a relevé ce défi pour la quatrième fois. Six participants se sont mobilisés pour l'occasion, avec pour la moitié d'entre eux une première participation. Lors des éditions précédentes, le LIA a toujours participé avec succès à ce défi en proposant différents systèmes basés sur des méthodes statistiques ainsi que des méthodes de fusion s'appuyant sur ces systèmes (El-Bèze *et al.*, 2005; Torres-Moreno *et al.*, 2007; El-Bèze *et al.*, 2007; Béchet *et al.*, 2008; Charton *et al.*, 2008; Torres-Moreno *et al.*, 2009).

Le défi actuel implique de classer des articles de presse francophone d'une part selon la période à laquelle ils ont été écrits et d'autre part selon le pays dans lequel ils ont été publiés. Ces deux tâches semblent au premier abord assez simples. En effet, chaque article suit un style particulier. Ce style dépend de l'auteur, certes mais cet auteur évolue à une période donnée et dans une localisation donnée. Chacun de ces paramètres induit d'une part des choix lexicaux spécifiques mais également des tournures stylistiques particulières. C'est ce style qu'il est ici question de définir. Plusieurs mots définissent une localisation plutôt qu'une autre (*e.g.* aiguisoir en fran cais québécois pour taille-crayon) mais qu'en est-il de la distinction de journaux d'un même pays ? En ce qui concerne les humains, il semble raisonnable de penser que des experts littéraires seraient capables de faire le distinguo. Qu'en est-il des méthodes d'apprentissage automatique ? D'autre part, tout francophone èducè est capable de distinguer les styles littéraires marquès (comme entre Victor Hugo et Molière, par exemple). Toutefois cette assertion est beaucoup plus sujette a caution dans le cas de deux grands écrivains de la même époque, ou d'époques proches. Cela nécessiterait certainement des compétences trés particulieres. La tâche de classification n'est donc pas aisée du tout. D'une part, les classes proposées ne sont pas si évidemment dissemblables (*e.g.* 1810 et 1820 ou « Le devoir » et « La presse »). Et d'autre part, le manque de corpus est encore et toujours un problème. En effet, les méthodes que nous proposons sont majoritairement des méthodes discriminantes basées sur des apprentissages supervisés et donc le manque de corpus annoté reste problématique. Les applications d'une telle tâche sont diverses. Elles vont du filtrage de grands corpus pour faciliter la recherche d'information ou la veille scientifique et économique jusqu'à la classification par le type de texte pour adapter les traitements linguistiques aux particularités d'un corpus.

Nous décrivons dans cet article les techniques et méthodes automatiques utilisées pour relever ce défi. La section 2 présente les tâches et les corpus de DEFT'2010. La section 3 décrit chacun des systèmes initiaux développés par les participants du LIA. Ces systèmes initiaux (niveau 1) sont ensuite utilisés par des systèmes de second niveau selon différents principes de fusion, définis dans la section 4. L'evaluation et les résultats des expériences sont rapportés et discutés en sections 5 et 6.

2 Présentation des tâches et des corpus

2.1 Variations diachroniques

Cette piste, relative à la variation diachronique, concerne l'identification de la décennie de publication d'extraits d'articles de presse français. La période couverte va de 1800 à 1944. Il s'agit donc d'une classification de texte de type uni-label sur un ensemble de 14 étiquettes. Une difficulté posée par cette classification concerne l'aspect continu (et non discret) des étiquettes : un texte écrit le 31 décembre 1849 n'aura pas la même étiquette que celui écrit le lendemain. Donc s'il peut sembler à peu prés facile, pour

1. http://www.lia.univ-avignon.fr

un humain, de classer des articles dans deux classes représentant des dates suffisamment éloignées, ne serait-ce que par le sujet évoqué. En revanche, quels sont les indices permettant de définir si un article a été écrit plutôt en 1949 qu'en 1950 ?

Le corpus d'apprentissage se compose de 3594 extraits de longueur 300 mots d'articles de quatre titres de journaux différents : « Le journal de l'empire », « Le Journal des Débats politiques et littéraires », « La Croix » et « Le Journal des Débats ». 15 décennies sont à identifier : 1800, 1810, 1820,...1940. Si un article a été écrit en 1809 alors il appartient à la décennie 1800, s'il est écrit en 1810 alors il appartient à la décennie 1810. Le tableau 1 contient le nombre d'articles pour chaque décennie du corpus d'entraînement.

Décennie	1800	1810	1820	1830	1840	1850	1860	1870	1880	1890	1900	1910	1920	1930	1940
Nombre	252	252	252	252	252	252	251	252	252	224	218	220	221	221	223

TABLE 1 – Répartition des extraits du corpus d'apprentissage Décennie

De manière à éprouver la robustesse des systèmes, le corpus de test intègre des extraits provenant des quatre titres présents dans le corpus d'entraînement, plus un cinquième absent de ce dernier. En pratique une difficulté majeure réside dans la lisibilité du texte. En effet, ces articles ont été obtenus à la suite d'un traitement automatique de reconnaissance de caractères sur des images scannées des journeaux et souffrent d'un nombre très élevé d'erreurs de reconnaissance.

2.2 Origine géographique

L'identification de l'origine géographique de chaque document (pays d'origine) constitue la seconde piste de cette campagne. Elle repose sur des corpus de presse rassemblant deux titres provenant de France et deux autres du Québec. Le corpus d'apprentissage est composé de 3719 extraits : 60% des corpus d'origine, les 40% restants sont utilisés pour le test.

Sous-tâche 1 : Quel Pays ? Les deux pays à retrouver sont la France et le Québec. Le tableau 2 donne le nombre d'extraits de chaque pays présents dans le corpus d'apprentissage.

Pays	Québec (Q)	France (F)
Nombre	1991	1728

TABLE 2 – Répartition des étiquettes Pays sur le corpus d'apprentissage Origine

Sous-tâche 2 : Quel Journal ? Cette sous-tâche consiste à identifier de quel journal précisement provient l'extrait. Les quatre journaux sont « Le Monde » (étiquette M) et « l'Est républicain » (E) pour la France, « Le Devoir » (D) et « La Presse » (P) pour le Québec. Le tableau 3 donne le nombre d'extraits de chaque journal présents dans le corpus d'apprentissage.

Pays	Le Monde (M)	l'Est républicain (E)	Le Devoir (D)	La Presse (P)
Nombre	900	828	976	1015

TABLE 3 – Répartition des étiquettes Journal sur le corpus d'apprentissage Origine

3 Présentation des systèmes initiaux

Comme les années précédentes, chaque participant du LIA propose un ou plusieurs systèmes. Ces systèmes sont donc différents par la méthode de classification employée ou par les techniques de représentation des articles propres à l'appréhension personnelle de la tâche par chaque concepteur. Il s'agit donc conjointement d'optimiser chacun des systèmes initiaux mais aussi d'obtenir un ensemble aussi hétérogène que possible. Ce qui sera mis à profit par l'utilisation dans un second temps de techniques de fusion permettant d'obtenir le meilleur système global possible. Les techniques de fusion sont décrites dans la section 4.

3.1 Protocole expérimental

Nous présentons d'abord brièvement les choix communs. Puis, les sous-sections suivantes décrivent les différentes méthodes ainsi que chacun des systèmes de niveau 1.

Etiquettes morpho-syntaxique ou lemmatisation Lors de nos participations précédentes, l'utilisation des POS (Part-of-Speech) et lemmes était systématiquement proposé à chacun de nos systèmes. Cette année en revanche, la trop grande quantité de bruit dans les corpus due à l'OCRisation mais également à des problèmes d'encodage n'a pas permis d'effectuer une analyse syntaxique efficace des textes. Ainsi, l'utilisation des POS, lemmes ou stemmes a été abandonnée.

Validation croisée Afin de tester nos méthodes, de régler leurs paramètres et de palier au phénomène de sur-apprentissage, nous avons décidé de scinder l'ensemble d'apprentissage (A) de chaque corpus en 5 sous-ensembles approximativement de la même taille (en nombre d'articles à traiter). La procédure d'apprentissage a été la suivante : 4 des 5 sous-ensembles sont concaténés pour produire un corpus d'entraînement et le cinquième est utilisé pour le test. La procédure est effectuée cinq fois afin que chacun des sous-ensembles du corpus d'apprentissage soit utilisé une fois pour le test. Les ensembles ainsi concaténés seront appelés dorénavant ensembles de développement (D) et le restant ensemble de validation (V).

3.2 Systèmes extra-linguistique : *Mick_MLP* et *Mick_SVM*

3.2.1 Sac de mots

L'une des difficultés majeures de la tâche décennie réside dans la lisibilité du texte. En effet, les articles ont été obtenus à la suite d'un traitement OCR et souffrent énormément des erreurs de reconnaissances. Ces erreurs de reconnaissances introduisent un bruit dans les documents et peuvent dégrader les performances des classifieurs. Nous proposons une méthode de correction des erreurs de mots et une méthode de classification de texte basé sur les mots-outils.

Correction des closures Les erreurs de reconnaissance issues d'un OCR sont multiple et peuvent être regroupées en 2 catégories : les erreurs de non-mots (*non-word errors*) et les erreurs de mots-réels (*real-word errors*). Une erreur de non-mot apparaît quand un mot est interprété comme une chaîne de caractère qui n'appartient pas à la langue française. Une erreur de mot-réel apparaît quand un mot est interprété comme une chaîne de caractère qui appartient à la langue française, mais n'est pas identique à la source du

texte. Par exemple, si la phrase source « le président de la république » est reconnu par un OCR comme « la président de la répablique » , *répablique* est une erreur de non-mot et *la* est une erreur de mot-réel.

Nous nous sommes intéressés ici aux erreurs de non-mot et plus particulièrement aux troncatures des mots. Par exemple le mot « président » peut être retranscrit après l'OCR comme « pré sident » . Ces erreurs introduisent du bruit dans les classifieurs et peuvent être corrigées très facilement à l'aide d'un dictionnaire. Pour chaque document, nous extrayons les bi-grammes de mots. Pour chacun de ces bigrammes, si la concaténation des mots le composant appartient au dictionnaire alors nous remplaçons le bigramme par le mot du dictionnaire. Le dictionnaire utilisé dans notre système a été constitué à partir des articles du journal Le Monde 1987-2003 ainsi que sur celui d'ESTER (Gravier *et al.*, 2004).

Classification sur les mots-outils La plupart des méthodes de classifications de textes sont basées sur un TF-IDF (*Term Frequency-Inverse Document Frequency*) et/ou un pré-traitement linguistiques (POS, lemmatisation). Ces techniques peuvent montrer des faiblesses lorsque les textes sont trop bruités (sortie d'un OCR, sortie de transcription, etc). (Stanislas Oger, 2010) a montré, dans le cadre de la classification de genre vidéo, que l'utilisation de la fréquence d'apparition des mots outils de la langue améliore nettement les résultats par rapport à l'utilisation de TF-IDF lorsque les données textuelles sont bruitées.

3.2.2 Extraction des entités nommées

A l'aide d'une encyclopédie numérique, les entités nommées d'un document (noms de personne, lieux, etc), peuvent nous donner des informations intéressantes sur sa date de rédaction. La détection d'entités nommées sur un corpus bruité n'est pas une chose facile (contexte bruité, noms propres mal orthographiés, etc). Nous proposons ici d'utiliser le systeme LIANE. Il permet de détecter les entités nommées sur des sorties bruités comme des sorties de transcription automatique de la parole ou issues d'un processus d'OCR.

Nous proposons donc un système à trois niveaux. Le 1er niveau va détecter les entités nommés. Le 2ième niveau va vérifier que l'entité nommée est bien écrite grâce à l'outil en ligne de suggestion d'orthographe de Google. Finalement, le 3ième niveau va rechercher sur une encyclopédie numérique (Wikipédia dans notre cas) les dates correspondant aux entités nommées détéctées. Un vecteur contenant toutes les dates est ainsi créé. Chaque indice du vecteur correspond au nombre de fois où la date a été vue sur Wikipédia. Ce vecteur est ainsi ajouté au vecteur d'observation du classifieur.

3.2.3 Extraction des caractères de ponctuations

Le taux d'erreur d'un document est dû en grande partie à la qualité du document numérisé. Les OCR sont assez sensibles à la qualité du papier, à la police de caractères, à l'encre utilisée, etc. On peut donc penser que le nombre d'erreurs rencontrées est lié à la date d'écriture du document. Au plus un document est ancien, au plus le taux d'erreur est élevé. Deux critères sont traditionnellement utilisés pour modéliser le taux d'erreur d'OCR : le nombre de mot hors vocabulaire et la perplexité du document selon un modèle de langage appris sur un corpus sans erreurs. Dans ce travail, nous proposons d'utiliser la ponctuation car les artefacts d'un document (tâches d'encre, etc.) sont souvent transformés par l'OCR en signes de ponctuation. Pour chaque document nous créons donc un vecteur contenant les fréquences d'apparition des signes de ponctuation. Ce vecteur est aussi ajouté au vecteur d'observation du classifieur.

3.2.4 Classifieur

Au total, le vecteur d'observation du classifieur est constitué des mots outils (soit environ 20 000 entrées), des fréquences des signes de ponctuation (13 entrées) et des entités nommées (15 entrées). Nous avons testé, sur le corpus de test, 2 classifieurs : les machines à vecteurs supports (*Support Vector Machine*, SVM) et les réseaux de neurones de type perceptron multi-couche (MLP). Nous avons choisi pour le SVM d'utiliser un noyau linéaire et pour le MLP une topologie à 3 couches (avec une couche cachée de 1000 neurones). Les performances globales des 2 classifieurs sont comparables. Par contre les réponses données par les classifieurs sont très différentes et donc nous pouvons espérer qu'en combinant ces 2 classifieurs dans un systèmes de fusion cela puisse améliorer les résultats.

3.3 Modèle de langage à base de n-grammes de caractères : *Jmt* et *Jmt_basic*

Dans le contexte du défi DEFT'10, nous voulions savoir si les n-grammes de caractères permettaient de discriminer convenablement la classe des documents. Dans le cas affirmatif, cela présenterait plusieurs avantages par rapport aux n-grammes de mots. D'abord l'ensemble des n-grammes de caractères est considérablement plus petit que l'ensemble des n-grammes de mots. Dans l'utilisation des modèles n-grammes de caractères, on peut se passer des techniques de lissage ou de *Back-Off* (Manning & Shütze, 2000), car à la différence des mots, la plupart des caractères rencontrés dans les phases de test ont été observés. Enfin, bien que l'utilisation des caractères reste relativement différente entre les langues, la plupart des signes sont les mêmes entre les langues d'origine latine. Nous avons développé un classifieur classique incorporant des techniques élémentaires de n-grammes, mais en utilisant les caractères à la place des mots. Ces techniques, inspirées directement de l'approche probabiliste (Manning & Shütze, 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; El-Bèze *et al.*, 2007; Béchet *et al.*, 2008; Charton *et al.*, 2008). Pour une tâche de classification, on peut construire les modèles n-grammes associés aux classes recherchées, par exemple dans la tâche Origine, Pays et Journal $g \in \{P, J\}$. Le score du genre $\tilde{g}$ étant donné un document et une séquence de caracteres s, aurait pu être calculé selon le théorème de Bayes :

$$\tilde{g} = \arg\max_g P(g|s) = \arg\max_g \frac{P(s|g)P(g)}{P(s)} = \arg\max_g P(s|g)P(g) \quad (1)$$

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

combinée avec une interpolation simple. Cependant, nous avons voulu en particulier étudier les algorithmes originelement conçus pour l'identification de la langue (Cavnar & Trenkle, 1994). Pour préserver toute son efficacité, nous n'avons réalisé aucun filtrage de signes de ponctuation des corpus d'apprentissage pour ce système [2]. L'algorithme proposé opère en 2 phases : (a1 la création du modèle de langage (ML) pour chaque catégorie i de document, $i = 1, \cdots, c$ et (2) le calcul de distance d'un document inconnu par rapport aux ML_i :

Phase 1. Modèles de langage $\mathbf{M}_i$

1. Découper le texte en *tokens* (chaînes de caractères séparées seulement par des espaces).
2. Génération de tous les n-grammes possibles, pour $n = 1, \cdots, 5$.

2. Nous remercions Marc El-Bèze (LIA) pour ses scripts de conversion de caractères.

3. Créer une table triée inversée pour comptabiliser les occurrences des n-grammes.

Phase 2. Calcul de distance sur un document inconnu

1. Créer un ML_x du document inconnu x au même titre que dans la phase (1).
2. Au moyen des i modèles de langage ML_i, calculer une statistique simple du rang au moyen d'une mesure de distance.
3. Initialiser un score à 0. Pour chaque n-gramme $\in ML_x$, cette distance détermine dans quelle mesure la position d'un n-gramme dans ML_x est dans la même position dans chaque ML_i. Même position, score += 0, position différente : score += | distance entre la position du n-gramme(ML_x) et n-gramme(ML_i) |. Les n-grammes $\in ML_x$ qui ne sont pas présents dans un modèle ML_i seront pénalisés par une grande valeur fixée empiriquement.
4. Le document inconnu x est attribué à la classe i avec le score le plus bas (i.e. la distance la plus petite) par rapport au modèle ML_i.

Nous avons fixé la distance maximale pour les n-grammes inconnus égale au nombre de n-grammes générés lors de la phase de création du ML. Nous avons appliqué ce modèle appéle n-grammes « basique » au corpus d'apprentissage de la tâche Origine, sans faire d'autres traitements particuliers. Pour faire varier un peu la stratégie, nous avons modifié le modèle de n-grammes précédent pour savoir si la contrainte de considérer les n-grammes de caractères sur la longueur du texte et pas sur celle de mots apportaient des éléments discriminants. Ainsi les ML ont été générés avec des n-grammes construits sur le texte vu comme une chaîne complète.

3.4 Un BoosTexter tout simplement : *Boost_basic*

Algorithme de *boosting* 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 et les éléments choisis par les classifieurs simples sont alors des n-grammes sur ces composants.

Boost_basic Ce système utilise le classifieur à large marge *BoosTexter (Schapire & Singer, 2000)* basé sur l'algorithme de *boosting Adaboost* (Freund & Schapire, 1996). Il est paramétré pour faire 600 ré-exécution au maximum et à chaque exécution choisir un motif de 1 à 3 mots consécutifs.

Les paramètres d'entrée de ce système sont les formes lexicales de chaque document. Les champs <texte> et <titre> pour la tâche *Origine* et le champ <texte> pour la tâche *Décennies*. Ces entrées lexicales subissent simplement deux traitements simples.

Dans un premier temps, les références aux entités XML sont remplacées par le caractère correspondant (*e.g.* &apos devient '). Dans un deuxième temps, le texte est nettoyé de toute ponctuation (! ?., ;) et autres caractères spéciaux (()#'&), chacun de ces caractères étant remplacé par une balise lexicale tenant compte de la présence ou non d'un espace autour du caractère. Par exemple, "suivant :" deviendra "suivant BAL-2POINTS-G" et "suivant :" deviendra "suivant BAL-2POINTS" tandis que " :encore" deviendra "BAL-2POINTS-D encore". Ainsi, la nouvelle phrase obtenue ne contient que des caractères aA-zZ et ne perd pas l'information de l'utilisation du caractère espace ' '.

3.5 Une autre approche du *boosting* : *Rk_icsiboost*

Pour ce système nous avons aussi utilisé un classifieur à large marge de type Adaboost spécialisé dans le traitement de données textuelles. L'outil ICSIBOOST [3] a été utilisé. Les paramètres d'entrée du système sont les entrées lexicales de chaque document, les champs <texte> et <titre> pour la tâche Origine et le champ <texte> pour la tâche Décennie. Nous avons choisi une représentation sous forme de n-grammes de mots, celle-ci ayant obtenu les scores les plus élevés lors des nos premiers tests. Afin d'améliorer les résultats sur la tâche Décennie, nous avons remplacé l'ensemble de la ponctuation par une balise lexicale, ainsi « , » devient « _VIR_ », « . » devient « _POINT_ », etc. Nous avons tenté par ailleurs certains prétraitements linguistiques classiques (Manning & Shütze, 2000) tels que le filtrage ou la racinisation sans amélioration notable des résultats. Concernant le nombre de tours de l'algorithme (paramètre T), les expériences ont montré qu'un optimum était atteint aux alentours des 1000 tours, ceci permettant des temps d'apprentissage assez court.

3.6 Système hybride pour résoudre la tâche Décennie

Confusion entre les classes Une des difficultés évidentes de la résolution de la tâche Décennie est le caractère continu des étiquettes. En effet, l'étude des matrices de confusion des systèmes sur le corpus de développement a mis en évidence que la majorité des erreurs étaient faites à une ou deux décennies près de celle de référence. A titre d'illustration, le tableau 4 donne la matrice de confusion du système *Mick_MLP* sur la partie (A.1) du corpus d'apprentissage.

Boost_basic_3classes Afin de palier ce problème, plusieurs systèmes initiaux ont été implémentés avec pour chacun le classement d'uniquement trois décennies consécutives. Ainsi, par exemple, le classifieur dont l'étiquette médiane est 1810 n'est appris que sur les exemples du corpus d'apprentissage étiquetés 1800 ou 1810 ou 1820. Ces classifieurs ont été implémentés selon le même protocole que le système initial Boost_basic. Les résultats de ces systèmes évalués en milieu fermé (c'est-à-dire sur des exemples du corpus de validation étiquetés selon le même sous-ensemble d'étiquettes que celui de développement) sont rapportés dans le tableau 5.

Les résultats obtenus sont meilleurs que ceux du même système appris sur les 14 classes (F-mesure moyenne d'environ 26%). Il est également intéressant de noter que les scores les plus élevés sont obtenus aux deux extrémités de l'intervalle de temps 1810 et 1930 avec des F-mesures dépassant les 50%.

3. http://code.google.com/p/icsiboost/

	1800	1810	1820	1830	1840	1850	1860	1870	1880	1890	1900	1910	1920	1930	1940
1800	28	13	2		1										
1810	4	24	6	4											
1820	2	29	14	7		1	1		2					1	
1830		3	4	22	8	2	2	2	1	1	1		1		
1840		2	1	15	11	6	3	11	6	2	1				
1850	1	7	2	10	3	7	7	7	5		1		1		
1860		4	1	7	3	4	6	4	3		3	2	2		1
1870		2		4	4	1	2	20	11	3	2	3	1		1
1880		1		5	1	1	5	7	13	6	4		2		3
1890		3	1	1	2	2	1	9	10	4	7	1	4	2	
1900	1	2	1	3	1	1	1	2	9	3	9	5	6		1
1910	1	1		1		1	1	2	5	4	4	11	6	4	13
1920			1	3			3	3	9	3	6	4	1	6	7
1930		1		2		1	1	1	5		4	2	7	11	12
1940		1		1				2	1		2			7	28

TABLE 4 – Matrice de confusion du système *Mick_MLP* sur la partie (A.1) du corpus d'apprentissage. Les colonnes correspondent aux décennies de référence tandis que les lignes correspondent à la décennie choisie par le système (score de confiance le plus élevé).

Etiquette médiane	1810	1820	1830	1840	1850	1860	1870	1880	1890	1900	1910	1920	1930
F-mesure moyenne	0.52	0.55	0.52	0.47	0.40	0.35	0.45	0.44	0.36	0.37	0.47	0.48	0.51

TABLE 5 – Performance de chacun des sous-classifieurs appris et testés sur 3 décennies consécutives.

Le choix de ne pas créer de classifieurs bi-classes pour les décennies 1800 et 1940 est également basé sur l'observation de bons résultats obtenus avec un classifieur 14-classes sur ces classes extrêmes.

Hybride_Boost_MLP Un système hybride a été élaboré à partir des systèmes *Mick_MLP* et *Boost_basic_3classes*. Le principe est le suivant :

1. la décision de *Mick_MLP* détermine la classe médiane ;
2. le système *Boost_basic_3classes* correspondant à cette classe est appliqué sur l'exemple ;
3. l'étiquette finale est celle obtenant le plus haut score de confiance pour *Boost_basic_3classes*.

Au final, les scores de confiance de l'ensemble des étiquettes est mis à 0 sauf ceux ayant obtenus un score dans *Boost_basic_3classes* où leur score est conservé.

4 Systèmes de fusion

Lors de l'évaluation chaque participant propose 1 à n systèmes et trois soumissions sont faites : meilleur classifieur pour la tâche considérée, vote majoritaire et fusion probabiliste, décrits dans les sous-sections suivantes.

4.1 Fusion majoritaire

Une des manières les plus simples de prendre en compte l'ensemble des décisions est d'effectuer un vote majoritaire. Le système de fusion majoritaire prend en considération les systèmes suivants :
– Tâche Décennie : *Boost_basic*, *Hybride_Boost_MLP*, *Mick_MLP*, *Mick_SVM* et *Rk_icsiboost* ;

– Tâche Origine : *Jmt*, *Jmt_basic*, *Boost_basic*, *Mick_SVM* et *Rk_icsiboost*.

Le vote de chaque système correspond à l'étiquette ayant obtenu le score de confiance le plus élevé. Une voix est accordée à chacun des systèmes de base définis pour la tâche donnée. Par ailleurs, un classement des systèmes selon leur performance sur le corpus d'apprentissage est établi pour chaque tâche. Ainsi, en cas d'égalité de voix sur deux ou plusieurs étiquettes, l'étiquette choisie parmi ces dernières sera celle élue par le système le plus performant.

Chaque étiquette en lice (choisie par au moins un des systèmes de base) se voit attribué un score de confiance. Ce score correspond au quotient du nombre de voix obtenues sur le nombre de votants. En cas d'égalité, l'étiquette choisie grâce au classement des performances des classifieurs est augmentée d'une voix, ainsi que le nombre de votants. Toutes les autres étiquettes obtiennent un score de confiance nul.

4.2 Fusion par classification

Afin d'exploiter au mieux l'information fournie par les différents systèmes de classification développés pour ces tâches, appelés ici systèmes initiaux, nous avons mis en place un mécanisme de prise de décision basé sur des classifieurs.

4.2.1 Architecture

Nous avons opté pour une architecture originale à deux niveaux au-dessus des systèmes initiaux, représentée dans la figure 1. Le premier niveau consiste en un ensemble de classifieurs qui sont entrainés séparément à classer les documents en prenant en entrée les supervecteurs rassemblant les prédictions des systèmes initiaux. Pour chaque classe, ces classifieurs fournissent un score de prédiction tenant compte des résultats fournis par les systèmes initiaux. Le second niveau est constitué d'un classifieur qui va prendre la décision finale à partir des prédictions fournies par le premier niveau. L'entrée de ce dernier classifieur est un supervecteur constitué des scores de prédiction pour chaque classe produit par le premier niveau de classification.

Les classifieurs du premier niveau de classification peuvent être vus comme des angles d'observation différents, chacun étant adapté à seulement une partie des documents. Nous espérons ainsi que pour chaque document au moins un classifieur propose la bonne décision. Le classifieur du second niveau sert alors à prendre la décision finale en fonction des prédictions de ces points de vue complémentaires.

4.2.2 Paramètres

Notre système de fusion par classification est conçu pour recevoir en entrée un score par classe et par document pour chaque système à fusionner. Techniquement, les scores de prédiction obtenus pour chacune des classes par les différents systèmes sont concaténés dans un supervecteur de paramètres qui sera fourni à notre système de fusion par classification. Les systèmes que nous avons inclus sont tous les systèmes élémentaires décrits dans cet article, auxquels s'ajoute le système de fusion par vote majoritaire. En effet, un système de fusion reste un système et fournit des prédictions au même titre que les systèmes initiaux, mais ses scores ont la particularité de contenir une information globale sur l'ensemble des systèmes.

4.2.3 Classifieurs pour les 2 niveaux

Nous avons sélectionné 21 classifieurs de fusion de premier niveau. Parmi ceux proposés par l'outil WEKA [4], ont été retenus ceux qui donnaient les meilleurs résultats sur le corpus d'entraînement. Il y a donc 8 arbres de décision, un réseau bayesien, 3 SVM, 3 régressions, 2 votes et 4 classifieurs basés sur des règles. Les multiples instances de classifieurs au sein d'une famille correspondent à différentes implémentations ou différents réglages fournissant des résultats différents.

Le classifieur de niveau 2 retenu est celui qui fournissait les meilleures performances sur le corpus d'apprentissage : une régression logistique.

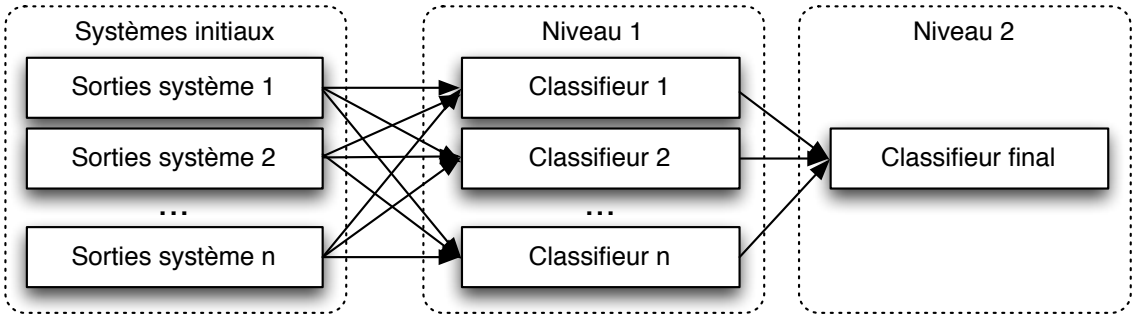

FIGURE 1 – Architecture du système de fusion par classification

5 Évaluation

5.1 Évaluation stricte

Comme décrit dans la section 2, le but du défi est de classer des documents dans un nombre fini de classes. Pour la tâche 1, il s'agit de les classer suivant leur décénnie de rédaction et pour la tâche 2 suivant le pays ou le journal de publication. Pour chaque tâche, un corpus d'apprentissage est mis à notre disposition afin de développer les algorithmes. Ceux-ci sont évalués sur des corpus de test (T) avec des caractéristiques semblables à celui d'apprentissage, en calculant la *F-mesure* des documents bien classés, moyenné sur tous les corpus :

$$F - mesure(\beta) = \frac{(\beta^2 + 1) \times \langle Précision \rangle \times \langle Rappel \rangle}{\beta^2 \times \langle Précision \rangle + \langle Rappel \rangle} \tag{3}$$

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

$$\langle Précision \rangle = \frac{\sum_{i=1}^{n} Précision_i}{n} \; ; \; \langle Rappel \rangle = \frac{\sum_{i=1}^{n} Rappel_i}{n} \tag{4}$$

Étant donné pour chaque classe i :

$$Précision_i = \frac{\{\text{Nb de documents correctement attribués à la classe} i\}}{\{\text{Nb de documents total attribués à la classe} i\}} \tag{5}$$

4. http://www.cs.waikato.ac.nz/~ml/weka/

$$Rappel_i = \frac{\{\text{Nb de documents correctement attribués à la classe}i\}}{\{\text{Nb de documents appartenant à la classe}i\}} \quad (6)$$

5.2 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 la probabilité pour un document d'appartenir à une classe d'opinion donnée. La F-mesure pondérée par l'indice de confiance a été utilisée pour l'évaluation des systèmes soumis à DEFT'10 si un indice de confiance était fourni par les participants. Dans la F-mesure pondérée, la précision et le rappel pour chaque classe ont été pondérés par l'indice de confiance. Ce qui donne :

$$Précision_i = \frac{\sum_{\text{AttribuéCorrect}_{i=1}}^{\text{NbAttribuéCorrect}_i} \text{Indice\_confiance}_{\text{AttribuéCorrect}_i}}{\sum_{\text{Attribué}_{i=1}}^{\text{NbAttribué}_i} \text{Indice\_confiance}_{\text{Attribué}_i}} \quad (7)$$

$$Rappel_i = \frac{\sum_{\text{AttribuéCorrect}_{i=1}}^{\text{NbAttribuéCorrect}_i} \text{Indice\_confiance}_{\text{AttribuéCorrect}_i}}{\{\text{Nb de documents correctement attribués à la classe } i\}} \quad (8)$$

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'10, le calcul de la F-mesure retenue par les organisateurs est celui de la formule 3 avec les précision et rappel des formules 7 et 8. Cette réécriture suppose évidemment que β soit égal à 1 de façon à ne pas privilégier la précision ou le rappel.

5.3 Discussion

Pour la tâche 1, étant donné qu'il s'agit de positionner des documents sur une échelle continue de temps, le simple contrôle de l'appartenance à une décennie semble un peu brutal : un article écrit en décembre 1899 est considéré comme faux s'il est étiqueté 1900. Le calcul de l'erreur pourrait tenir compte de l'éloignement "temporel" entre la décennie donnée et celle de référence, ou bien utiliser des classes avec recouvrement et considérer comme juste les deux classes possibles en cas de chevauchement.

De plus, la méthode d'évaluation par F-mesure pondérée est utilisée systématiquement pour l'évaluation de DEFT'10 par les organisateurs. De notre point de vue, il serait plus judicieux de présenter les deux. En effet dans la mesure où nous n'avons aucun contexte précis pour utiliser les scores de confiance fournis par les systèmes seul le maximum de leur distribution peut être utilisé d'un point de vue opérationnel. Par ailleurs, n'étant pas informés de ce mode d'évaluation nous n'avons pas tenté d'optimiser nos scores de confiance pour maximiser la F-mesure pondérée comme on aurait pu le faire par exemple en appliquant une fonction de transformation des scores de confiance (*mapping*).

6 Résultats

Afin de ne pas surcharger de chiffres cet article, nous avons choisi de ne noter que la F-mesure finale de chaque système, obtenue pour chaque tâche. Plus de détails sont donnés uniquement pour le meilleur système. Les résultats sont donnés dans les tableaux 6 et 7.

	Apprentissage			Test			Test (F-m pondérée)		
Systèmes	Décennies	Journal	Pays	Décennies	Journal	Pays	Décennies	Journal	Pays
Mick_MLP	29,0	-	-	33,8	-	-	19,0	-	-
Mick_SVM	30,3	71,7	91,9	31,3	74,8	92,6	25,9	69,8	90,3
Jmt	-	67,3	87,9	-	68,3	89,6	-	68,3	89,6
Jmt_basic	-	68,4	88,9	-	68,3	90,0	-	68,2	90,0
Boost_basic	25,6	79,3	95,0	26,2	83,0	96,6	8,3	37,9	82,0
Rk_icsiboost	20,8	74,9	94,2	23,9	74,3	94,6	21,5	72,7	94,7
Hybride_boost_MLP	28,8	-	-	29,0	-	-	24,6	-	-
Fusion majoritaire	33,3	79,1	95,4	34,3	80,4	96,3	29,4	74,1	93,2
Fusion par classification	36,1	84,0	96,9	36,3	83,0	97,8	26,5	70,5	96,4

TABLE 6 – Ensemble des résultats obtenus par les systèmes initiaux et après fusion.

On remarque que tous les systèmes ont leur performance affectée par le passage au scoring avec pondération. Le système le plus touché est *Boost_basic* qui chute sur le corpus de test de 26,2% à 8,3% pour la tâche Décennie, de 83% à 38% pour les Journaux et de 96,6% à 82% pour les Pays. Ceci s'explique par le fait que les scores fournis ne représentent pas la probabilité de décision pour chaque étiquette mais un score de confiance compris entre 0 et 1 pour chaque décision indépendamment des autres étiquettes. La plupart des scores fournis sont compris entre 0,3 et 0,7. Il va de soi que ces scores auraient été paramétrés différemment si l'on avait eu conscience de leur importance dans le calcul des résultats. En revanche on remarque que ces systèmes sont robustes et obtiennent parfois de meilleurs résultats sur le corpus de test quelle que soit la tâche.

Pour le système Rk_icsiboost, les résultats obtenus sur la tâche Origine géographique sont d'excellentes qualités pour la **Sous-tâche 1**. Pour la **Sous-tâche 2**, ceux-ci restent corrects malgré un nombre conséquent d'erreurs, principalement des documents mal classés entre les deux journaux canadiens (D et P) selon la matrice de confusion. Nous attribuons la faiblesse des résultats obtenus sur la tâche Décennie au bruit produit par les erreurs d'OCR récurrentes dans le corpus. On observe cependant une bonne robustesse du système sur l'ensemble des deux tâches puisque les résultats obtenus sur les corpus de développement et celui de test sont proches.

Concernant le système *Hybride_boost_MLP*, il obtient des résultats honorables pour la tâche Decennies mais n'est pas le plus performant des systèmes initiaux et donc mériterait de meilleurs résultats vis à vis des efforts fournis (14 systèmes différents au total !).

Les modèles de n-grammes de lettres ont été testés uniquement sur la tâche Origine (sous-tâches Journal et Pays). Les performances en terme de F-mesure sur les ensembles de developpement sont autour de 89% pour les Pays et 68% pour les Journaux, le modèle "basic" étant légèrement meilleur que l'autre. Dans les deux cas, les deux types de modèles se sont avères relativement stables en deveppement et en test (90% pour les Pays et 68,3%).

La fusion majoritaire obtient toujours de meilleurs résultats que l'ensemble des systèmes initiaux mais est

surpassée par la fusion par classification. Là encore, on remarque qu'au niveau du calcul des scores pondérés la relation s'inverse et la fusion majoritaire obtient un meilleur résultat que la fusion par classification.

Concernant la fusion par classification, il est intéressant de mesurer le gain apporté par l'architecture à deux niveaux proposée. On peut le faire en comparant les performances du meilleur classifieur de niveau 1 et les performances du niveau 2. Ces résultats se trouvent respectivement dans les lignes "Meilleur niveau 1" et "Score niveau 2" du tableau 7. On constate une diminution de 0,1% absolu du taux de classification pour la tâche Pays, aucun changement pour la tâche Journal mais un gain de 1% absolu pour la tâche Décennie. On peut donc dire que cette architecture apporte un gain par rapport à une fusion utilisant un seul classifieur pour la tâche Décennie et qu'elle ne dégrade globalement pas les performances obtenues pour les autres tâches.

Afin de mesurer le gain global apporté par cette architecture, nous avons comparé les performances du meilleur système initial avec les résultats fournis par le second niveau de classification. La ligne intitulée "Meilleur système" du tableau 7 contient le taux de classification obtenu par le meilleur système initial et la ligne "Score niveau 2" contient le taux de classification final obtenu par la combinaison. On constate que la combinaison apporte une amélioration systématique par rapport au meilleur système initial sur les trois tâches considérées.

	Apprentissage			Test		
Mesure	Décennie	Journal	Pays	Décennie	Journal	Pays
Meilleur système	30,0	81,2	94,8	34,3	80,8	96,5
Oracle systèmes	64,8	94,6	99,9	64,5	94,6	99,9
Meilleur niveau 1	35,1	84,0	97,0	36,4	83,3	97,8
Oracle niveau 1	81,5	96,1	99,6	81,6	95,7	99,5
Score niveau 2	36,1	84,0	96,9	36,3	83,0	97,9

TABLE 7 – Performances en terme de F-mesure pour les trois tâches de classification considérées sur les corpus d'apprentissage et de test, mesurées aux différents niveaux du système de fusion par classification.

Afin de mesurer l'évolution des performances de la fusion par classification nous avons refait sur le corpus de test les mesures que nous avions faîtes sur le corpus de développement et qui nous avaient permis de valider l'approche. La partie droite du tableau 7 contient les résultats de ces mesures. On constate que les résultats obtenus sur les trois tâches à la sortie de la fusion (ligne "Score niveau 2") sont meilleurs que ceux du meilleur des systèmes initiaux (ligne "Meilleur système"), ce qui signifie que la fusion par classification permet toujours d'améliorer les résultats. L'architecture de classification à deux niveaux que nous avons utilisé permet donc de tirer parti de la complémentarité des classifieurs constituant le niveau 1, et qu'elle permet un gain de classification important sur la tâche Décennie par rapport à une architecture plus simple. Globalement, la fusion par classification que nous avons proposé améliore significativement les résultats obtenus par le meilleur système initial.

7 Conclusion et perspectives

La classification de documents est une tâche qui peut être très difficile en fonction du type de textes. Comme cela avait constaté lors des défis précédents, *"La lecture directe ne suffit pas toujours pour se forger un avis et privilégier une classification par rapport à une autre."* (Torres-Moreno *et al.*, 2009). Comme dans le passé, nous avons utilisé des approches de représentation numériques et probabilistes, afin de rester aussi indépendant que possible des sujets traités. Concernant les systèmes de base, Rk_icsiboost obtient de bonnes performances générales sur la tâche Origine mais se heurte aux erreurs d'OCR sur la tâche Décennie qui introduit du bruit dans le modèle en n-grammes de mot. Les modèles de n-grammes s'avérent intéressants dans la mesure où ils sont combinés avec d'autres méthodes, car ils permettent de capturer certaines caractéristiques très fines des documents. Nous pensons les améliorer en les combinant avec l'approche probabiliste de Bayes et en utilisant des mesures de distance non linéaires. Enfin nous avons présenté deux stratégies de fusion de méthodes qui se sont avérées robustes et performantes, dans tous les cas au-dessus des moyennes des meilleures soumissions initiales.

Références

BÉCHET F., EL-BÈZE M. & TORRES-MORENO J.-M. (2008). En finir avec la confusion des genres pour mieux séparer les thèmes. In *DEFT'08*, p. 161–170.

CAVNAR W. & TRENKLE J. (1994). N-gram-based text categorization. *Ann Arbor MI*, **48113**, 4001.

CHARTON, ERIC, CAMELIN, NATHALIE, ACUNA-AGOST, RODRIGO, GOTAB, PIERRE, LAVALLEY, REMI, KESSLER, REMY, FERNANDEZ & SILVIA (2008). Pré-traitements classiques ou par analyse distributionnelle :application aux méthodes de classification automatique déployées pour deft08. In *DEFT'08*, Grenoble, France.

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 *DEFT'05*, volume 2, p. 125–134.

EL-BÈZE M., TORRES-MORENO J.-M. & BÉCHET F. (2007). Un duel probabiliste pour départager deux présidents. *RNTI E-10*, p. 117–126.

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

GRAVIER G., BONASTRE J. F., GEOFFROIS E., GALLIANO S., MC TAIT K. & CHOUKRI K. (2004). The ESTER evaluation campaign of rich transcription of French broadcast news. In *LREC*, p. 885–888.

MANNING C. D. & SHÜTZE H. (2000). *Foundations of Statistical Natural Language Processing*. Cambridge, Massashusetts : MIT Press.

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

STANISLAS OGER, MICKAEL ROUVIER G. L. (2010). Transcription-based video genre classification. In *ICASSP*, p. 5114–5117.

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 *DEFT'07*, p. 119–133.

TORRES-MORENO J. M., EL-BÈZE M., BÉCHET F. & CAMELIN N. (2009). Fusion probabiliste appliquée à la détection et classification d'opinions. In *DEFT'09*.