Participation de l'IRISA à DEFT 2011: expériences avec des approches d'apprentissage supervisé et non-supervisé

Christian Raymond[1,2,3] Vincent Claveau[1,4]
(1) IRISA, Campus de Beaulieu, 35042, Rennes, France
(2) Université Européenne de Bretagne
(3) INSA de Rennes,35708, Rennes, France
(4) CNRS, Rennes, France
christian.raymond@irisa.fr, vincent.claveau@irisa.fr

Résumé. Cet article présente la participation de l'équipe TexMex de l'IRISA à DEFT 2011. Nous avons participé aux deux tâches proposées et à toutes les pistes. Nous avons exploré différentes approches. Nous avons notamment employé des techniques d'apprentissage particulières à base de *boosting* et de *lazy-learning* et des pondérations issues du domaine de la recherche d'information. Ces différentes approches nous ont permis d'obtenir de bons résultats et de nous classer premiers sur la tâche de datation et d'obtenir une précision de 99 % et 99.5 % sur la tâche d'appariement.

Abstract. This article presents the participation of IRISA TexMex team at DEFT in 2011. We participated in the two proposed tasks and all tracks. We explored different approaches. We employed specific learning techniques based on *boosting* over decision trees and *lazy-learning* together with weights from the information retrieval field. These different approaches enabled us to obtain good results since we rank first on the task of dating and we obtained an accuracy of 99% and 99.5% on the pairing task.

Mots-clés : Classification, boosting, arbre de décision, bonzaiboost, Okapi, apprentissage paresseux, k-plus-proches voisins.

Keywords: Classification, boosting, decision tree, bonzaiboost, Okapi, lazy learning, k-nearest neighbours.

1 Introduction

Cet article décrit la participation de l'équipe TexMex de l'IRISA à ce défi. Il s'agit de notre première participation à cette compétition. Les deux tâches proposées dans le cadre du DÉfi Fouille de Texte (DEFT) en 2011 portaient sur la classification de documents, qui est un domaine d'intérêt pour nous.

Les délais très courts (phase d'entraînement de 5 semaines) imposés par les contraintes d'organisation nous ont orientés sur des mise-en-œuvre simples mais efficaces. Pour ce faire, nous nous sommes appuyés notamment sur des techniques classiques de recherche d'information ou des techniques d'apprentissage développées en interne. Malgré ce manque de temps, les bonnes performances obtenues et nos classements montrent le bien fondé de ces approches. Par ailleurs, nos différentes approches ont pour point commun de n'avoir pas recours à des connaissances externes.

L'article est structuré comme suit. Les deux sections suivantes décrivent les deux approches utilisées pour la tâche de datation (les runs 1 et 2 pour les deux pistes de la tâches 1 sont donc décrits en section 2 et le run 3 dans la section 3). La section 4 décrit l'approche utilisée pour la tâche 2 d'appariement résumé/article.

2 Contribution à la tâche 1 : classification

La première piste envisagée dans la résolution de la tâche 1 a été une approche à base de classification. Cela a permis dans un premier temps de cerner la difficulté de la tâche qui *a priori* devait être compliquée pour au moins trois raisons :

1. retrouver une période temporelle semble raisonnable, retrouver une année en particulier beaucoup plus difficile ;
2. il faut discriminer entre 145 années différentes ;
3. l'entrée est très bruitée en raison des erreurs induites par l'OCR, même si c'est le problème qui, de notre point de vue, ne paraissait pas le plus critique.

En raison des deux premiers points soulevés, une approche de classification classique ne semble pas adaptée, mais un test en condition réelle s'impose pour avoir une confirmation. L'idée principale était ensuite d'adapter le classifieur ou la présentation de la tâche elle-même pour la résoudre en utilisant une méthode d'apprentissage supervisé.

2.1 Pré-traitement des données

Le pré-traitement des données semblait être une phase importante dans le traitement de cette tâche, par manque de temps nous n'avons malheureusement pas vraiment approfondi cette partie. Les traitements effectués (ou envisagés) sont (ont été) les suivants :

– la première des choses à laquelle on pense est la correction des erreurs de l'OCR, mais nous avons réalisé que la tâche en elle-même était compliquée : outre des erreurs de graphie, des erreurs de segmentation sont glissées ce qui rend non seulement la correction aussi bruitée que la non-correction, mais il ne semble pas évident qu'une correction parfaite puisse véritablement apporter un avantage indéniable dans la résolution de la tâche principale. Nous avons donc abandonné ce traitement ;
– comme le souligne (Oger *et al.*, 2010) dans l'édition de DEFT précédente, il est légitime de penser que les erreurs de l'OCR peuvent être dues 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. 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 conserverons les fréquences d'apparition des signes de ponctuation.

Nous avons par la suite adopté un pré-traitement classique : pour l' approche à base de classification nous avons mis à notre disposition trois attributs :

1. le texte lui-même (sauf la ponctuation) où à chaque mot est associé une étiquette

- premièrement, les étiquettes associées aux mots sont des étiquettes morpho-syntaxiques. En raison de l'entrée bruitée, nous n'avons pas utilisé d'analyseur, nous avons juste associé à un mot son étiquette la plus probable sans tenir compte du contexte.
- deuxièmement, nous avons enrichi ce jeu d'étiquettes par des étiquettes provenant de liste de connaissances *a priori* communes (*i.e.* villes, pays, mois, jours de la semaine, titres de noblesse, grade militaire, et quelques autres).
- troisièmement, nous n'avons conservé que les couples (mot/étiquette) dont l'étiquette appartient soit à la liste des connaissances *a priori* soit à l'ensemble suivant des étiquettes morpho-syntaxique (noms,adjectifs,verbes) supprimant ainsi, nous l'espérons, une partie du bruit et de mots insignifiants.

2. en guise de deuxième attribut, nous avons pour chaque mot ou étiquette du premier attribut associé sa fréquence d'apparition dans le texte. Nous utilisons par exemple le nombre de verbes utilisé, le nombre de titres de noblesses utilisé,*etc.*
3. le troisième attribut est identique au premier excepté que nous n'avons pas appliqué le troisième point de traitement, tout les mots ont été conservés en espérant retrouver d'éventuelles figures de style à l'intérieur.

Pour la suite, si aucune mention contraire n'est posée, les classifieurs utilisés vont s'appuyer sur des descripteurs *N*-grammes dans les attributs 1 et 3 (de taille $<= 2$ pour l'attribut 1 et de taille $[2, 3]$ pour l'attribut 3), étant données les deux informations en entrée (*i.e.* le mot ainsi qu'une étiquette correspondante) tous les *N*-grammes issus de la combinaison de ces deux informations sont générés. L'attribut deux sera interprété comme du « scoredtext », c'est-à-dire que des seuils sur la valeur numérique seront évalués (*i.e.* on peut apprendre des seuils sur les fréquences d'apparition d'un mot ou d'une étiquette). Afin d'éliminer une partie du bruit et surtout faciliter la sélection de descripteurs pertinents des filtres sont appliqués : seuls les descripteurs ayant été observés au moins 2 fois pour l'attribut 1 sont conservés. La coupure est faite à 10 pour l'attribut 3. Le choix de la taille des *N*-gramme ou du paramètre de filtrage pour les attributs 1 et 3 a été guidé par les objectifs suivants :
– attribut 1 : taille *N*-gramme filtrage réduits : capturer les mots ou expressions caractéristiques d'une année
– attribut 3 : taille *N*-gramme > 1 et filtrage important : capturer des informations caractéristique d'une époque (*e.g.* style)
Le logiciel BonzaiBoost (Raymond, 2010) est utilisé pour l'extraction *N*-gramme et pour toutes les approches de classification .

2.2 Premier aperçu avec un arbre de décision

Pour des raisons de diagnostic sur la difficulté de la tâche j'ai décidé de commencer par l'apprentissage d'un arbre de décision car le modèle produit est facilement interprétable. J'ai développé un arbre classique basé sur un critère de segmentation entropique et un critère d'arrêt statistique qui stoppe l'induction lorsqu'il considère que le gain obtenu est trop faible pour que le descripteur choisi soit véritablement pertinent, résultat : l'arbre ne se développe pas. Bien entendu en présence de 145 classes aucun critère ne peut apporter de gain significatif. Lorsque l'on continu l'induction de l'arbre en utilisant des critères d'arrêts plus conventionnel (taille des feuilles) on obtient hélas un arbre dont même les premiers choix ne semblent guère pertinent, le premier descripteur choisi est par exemple un descripteur *N*-gramme de l'attribut 1 : « etoit ». Les performances sur la tâche 1 sont de l'ordre de 0.14 à 0.16 avec un critère d'arrêt sur la taille minimale d'une feuille fixé entre 5 et 30 documents.

Ces résultats étaient plus ou moins prévisibles : il n'existe pas ou peu de descripteurs caractéristiques d'une année. On peut par contre espérer qu'il existe des descripteurs caractéristiques d'une période temporelle. De plus, la nature même du problème implique que sa résolution ne doit pas être envisagée par une classification brutale en année : en effet les classes ne sont visiblement pas indépendantes et se tromper d'un an dans la prédiction ne doit pas avoir la même répercussion que de se tromper de 100 ans. Pour modéliser cela, nous avons appris un arbre de décision où le critère de segmentation est la minimisation de la variance autour de l'année médiane d'une feuille. Une fois de plus, même si le classifieur est beaucoup mieux adapté à la tâche, les descripteurs sélectionnés par l'arbre sont peu convaincants : l'arbre sélectionne des descripteurs réduisant la variance sur les sous-nœuds gauche et droit, c'est-à-dire des descripteurs qui permettent de séparer au mieux des documents antérieurs à une date dans le nœud gauche et des documents postérieurs à cette même date dans le nœud droit : vraisemblablement, il n'en existent pas de véritablement pertinents : sont choisis alors des descripteurs qui répondent à ce critère de manière circonstancielle qui ont une réalité sur le corpus d'apprentissage mais ne sont pas assez généraux. On peut remarquer dans l'arbre de la figure 1 que de nombreux critères sur la fréquence de caractères de ponctuation sont

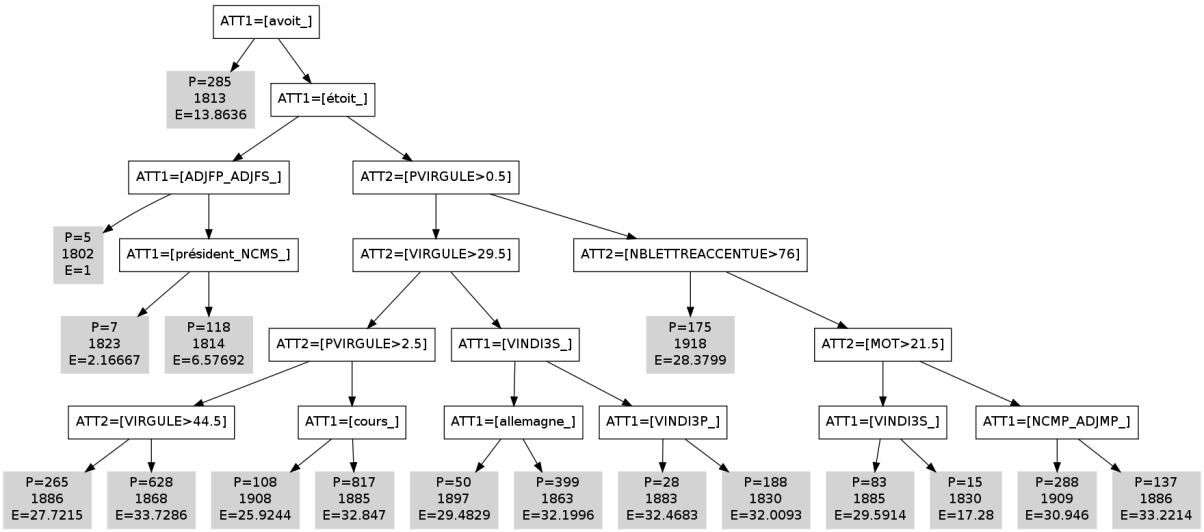

FIGURE 1 – Arbre de décision (élagué) construit en minimisant la variance autour d'une année médiane dans un nœud, chaque feuille indique : P=nombre de documents, l'année médiane, et E=l'écart type

sélectionnés confirmant l'hypothèse que la sur-reconnaissance de ceux-ci par l'OCR est potentiellement attaché à la mauvaise qualité des vieux documents. Les résultats sur la tâche 1 tourne autour de 0,17.

L'approche précédente n'a pas fonctionné, mais il semble pourtant que cette approche n'est pas dénuée de sens, les mauvais résultats précédents viennent du fait qu'il n'existe pas de descripteur pertinent pour décider si un document est supérieur ou inférieur à une année (qui fonctionne avec la plupart des documents), mais on suppose toujours qu'il existe des descripteurs caractéristiques d'une période. Pour modéliser cela en s'affranchissant du problème précédent, nous avons tenté l'induction d'un arbre minimisant la variance dans le nœud gauche exclusivement provoquant l'induction d'un « peigne » où chaque branche gauche pourrait être caractéristique d'une époque particulière. Pour éviter l'établissement de branches du peigne avec un seul document, nous avons imposé une taille minimale. Cette fois-ci, les descripteurs sélectionnés semblent très pertinents, tout du moins dans la partie haute de l'arbre, car malheureusement les descripteurs caractéristiques semblent s'épuiser très rapidement, et cette arbre-peigne qui semblait prometteur est au final inefficace globalement car il ne peut se développer au delà d'un certaines profondeur de manière efficace. La figure 2 montre un arbre-peigne illustrant les descripteurs caractéristiques de certaines périodes historiques.

2.3 Un peu plus de souplesse

Dû au manque évident de caractéristiques fortement discriminantes, il semble nécessaire d'adopter une approche de classification moins rigide. L'idée est de faire voter des participants (dont l'opinion n'est pas parfaite) et de combiner les décisions des participants. Les méthodes de boosting sont des méthodes permettant de combiner des classifieurs faibles pour obtenir au final un classifieur puissant. Les classifieurs faibles (*i.e.* les participants) sont ici des arbres de décision limités à 1 niveau (deux nœud/feuilles). L'algorithme de boosting utilisé est Adaboost.MH (Schapire & Singer, 2000). Une utilisation classique de cet algorithme sur les données présentées comme décrit dans la section 2.1 obtient un score de 0.226 sur la piste 1 (avec 1300 rounds de boosting). Les résultats ne sont pas vraiment excellents mais nettement meilleurs par rapport à ceux obtenus avec des arbres de décision classiques. Là aussi, le principal problème est la rigidité liée aux erreurs entre classes, une prédiction erronée de 1 an est considérée comme fausse au même titre qu'une erreur d'un siècle. Avec plus de temps, nous aurions aimé étudier les arbres dont la construction est guidée par la minimisation de la variance combinés avec un algorithme de boosting. Nous avons opté pragmatiquement pour une transformation du problème : le problème de classification ne sera plus de trouver la bonne année, mais de retrouver un ensemble de période temporelles. Pour faire simple, le problème K-classes est décomposé en un ensemble de K problèmes binaires où est indiqué si une année de référence est supérieure ou non à une des 145 années de la liste (*e.g.* à un document de 1910 sera associé la liste de labels : SUP_{1801}, SUP_{1802},...,SUP_{1909}). Le même algoritme de boosting, AdaBoost.MH étant un algorithme

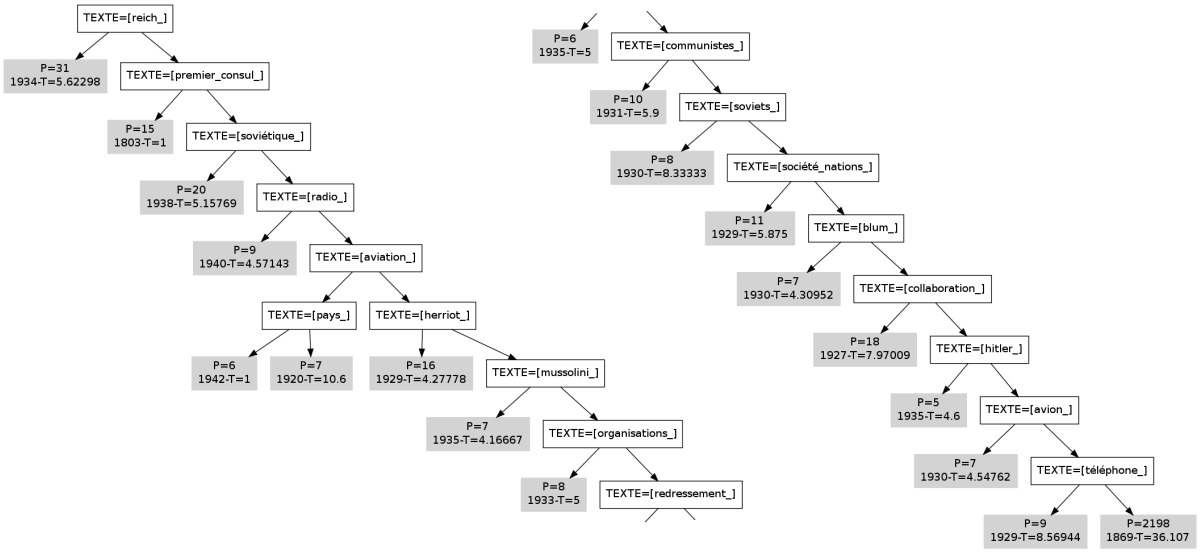

FIGURE 2 – arbre-peigne de profondeur 17 dont l'induction est basée sur la minimisation de la variance dans le nœud gauche. Chaque feuille indique : P=nombre de documents, l'année médiane, et T=l'écart type

multi-classes/multi-labels, est utilisé pour l'apprentissage. Ensuite une règle simple de vote permet de prendre une décision :

– pour chacun des labels possibles, s'il a été prédit, un vote est fait pour chacune des années supérieurs ou égales à l'année indiqué par le label, sinon le vote se porte sur chacune des années inférieures,
– on cumule le vote pour chacun des labels prédits pour un document,
– on sélectionne l'année qui obtient le plus de voix ;

Avec cette stratégie on espère obtenir les avantages suivants :

– chaque problème est plus simple à résoudre que le global
– chaque problème de séparation en deux profite de la sélection de multiples descripteurs au fur et à mesure des tours de boosting,
– on espère obtenir à la fin un ensemble d'informations qui permettent de localiser au moins grossièrement l'année même si le nombre d'erreurs est relativement élevé.

Cette méthode n'est au final pas forcément très convaincante non plus, car la rigidité de la barrière binaire (inférieur ou supérieur à une année) ne favorise pas la précision de l'algorithme (comme on pouvait s'y attendre mais dans le temps imparti nous n'avons pas eu le temps de creuser). Toutefois les résultats sont considérablement meilleurs que ceux obtenus par une approche directe sur l'année à prédire. Le résultat sur la tâche 1 étant aux alentours de 0,33. Cette méthode a été utilisé pour les runs 1 et 2 de la piste 1 et 2 de la tâche 1. La différence entre les deux runs concernent les filtres appliqués sur les descripteurs, donc sur le bruit des données d'entrées. Le tableau 1 illustre ce qui est appris durant les premiers tours de boosting.

2.4 Conclusion

L'approche par classification supervisée est limitée sur cette tâche. Les méthodes font de la sélection de descripteurs alors que peu de descripteurs sont véritablement discriminants. Le modèle obtenu a alors une vision très restreinte (exclusivement centrée sur les descripteurs qu'il a lui même retenus) et échoue dans la plupart des cas où il n'y a pas d'évidence. Une approche plus appropriée semble être de conserver un maximum d'information et de les exploiter de manière non-supervisée, voir section 3.

Tour	descripteur	présence	absence
1	étoit	[1813, 1944]	[1879, 1944]
2	VINDI3S	[1934, 1944]	[1802, 1937] 1942
3	PVIRGULE>0.5	[1802, 1944]	
4	au cours		[1802, 1944]
5	avoit	[1802, 1944]	
6	VIRGULE>29.5	1941 1943	[1802, 1944]
7	MOT>13.5	[1802, 1944]	
8	reich	1944	[1802, 1943]
9	M_Mme PRENOM MOTMAJ.	1804	[1802, 1803] [1805, 1944]
10	monsieur1 DETMS	[1826, 1944]	[1802, 1825]
11	long-temps	1824 [1810, 1820] [1827, 1944]	[1802, 1809] [1821, 1823] [1825, 1826]
12	NBLETTREACCENTUE>65.5	[1802, 1832]	[1833, 1944]
13	enfants		[1802, 1944]
14	la situation	[1937, 1941] [1943, 1944]	[1802, 1936] 1942
15	NBTITRENOBLESSE>0.5	1802 [1809, 1811] [1816, 1826] [1828, 1944]	[1803, 1807] [1812, 1815] 1827
16	écrit de	[1803, 1807] [1825, 1826] [1842, 1944]	1802 [1808, 1824] [1827, 1841]
17	télégraphie	1930 [1932, 1944]	[1802, 1931]
18	allemagne	[1802, 1811] [1932, 1944]	[1813, 1931]
19	cit MOT	[1802, 1944]	1944
20	enfans	1803, 04, 12, 13 [1815, 1819] [1828, 1944]	1802 [1804, 1814] [1816, 1827]
21	lit PREP DETMS	[1835, 1944]	[1802, 1834]
22	région		[1802, 1944]
23	DETMS « NCMS		[1802, 1944]
24	milieux	[1942, 1943]	1944 [1802, 1941]
25	président	[1935, 1944]	[1802, 1934]
26	DETMS NOMBRE PREP	[1802, 1944]	
27	VIRGULE>22.5	1927 [1922, 1925] [1934, 1944]	[1802, 1921] 1924, 26, 36, 43 [1928, 1934]
28	société des nations	[1935, 1944]	[1802, 1934]

TABLE 1 – Détails des 28 descripteurs choisis par les 28 arbres de décision construits au long de 28 tours de boosting. Pour chaque ligne la colonne « présence » montre les années pour lesquelles le classifieur donne son vote si le descripteur est présent dans le document, la colonne « absence » montre les années pour lesquelles le classifieur donne son vote si le descripteur est absent

3 Contribution à la tâche 1 : k-plus proches voisins

Cette section décrit une approche différente que nous avons expérimentée pour cette même tâche de classification diachronique. Elle repose sur un apprentissage paresseux, qui se veut plus souple et plus adapté à la tâche.

3.1 Vision de la tâche : *lazy-learning*

Cette approche repose sur un constat. Ces dernières années l'emploi de techniques d'apprentissage (principalement numériques et supervisées) s'est très largement répandu dans le domaine du TAL. Cependant, ces techniques sont souvent utilisées sans tenir compte des spécificités de la tâche à accomplir, des données et du classifieur. Il est ainsi courant de voir utilisées des approches à base de modèle alors que les instances d'une même classe sont connues pour apparaître sous des formes très diverses. Cela oblige à utiliser des classifieurs très complexes à mêmes de construire un unique modèle permettant de prendre en compte ces instances peu comparables entre elles. Dans beaucoup de cas, une approche par plus proches voisins est alors bien plus adaptée.

C'est ce même cas qui se présente dans cette tâche de datation. En effet, deux articles de la même année n'aborde pas forcément les mêmes sujets. Leurs descriptions, si elles sont basées sur les mots qu'ils contiennent, ont peu de chance d'être comparables. Chercher à apprendre un unique modèle sur ces instances est donc inutilement difficile. Une approche par k-plus proches voisins nous a donc paru plus adaptée.

Dans une approche par k-plus proches voisins, une instance inconnue est classée en trouvant les k instances connues les plus similaires et en lui assignant la classe majoritaire de ces instances. Il n'y a donc pas à proprement parler d'apprentissage, d'où le nom de *lazy-learning*, mais l'induction repose sur la calcul de similarité et la mise en œuvre du vote.

3.2 Mesures de similarité

Dans le cas présent, la similarité entre deux articles de presse est simplement calculée en utilisant les mesures classiques utilisées en recherche d'information. Nous avons exploré l'utilisation de deux de ces mesures pour cette tâche. Nous avons tout d'abord implémenté une similarité inspirée de la mesure OKapi BM-25 (Robertson *et al.*, 1998). Celle-ci repose sur une pondération donnée dans l'équation 1 qui indique le poids du terme t dans le document d.

$$w_{BM25}(t,d) = TF_{BM25}(t,d) * IDF_{BM25}(t) = \frac{tf * (k_1+1)}{tf + k_1 * (1 - b + b * dl/dl_{avg})} * \log \frac{N - df + 0.5}{df + 0.5} \quad , \quad (1)$$

où $k_1 = 2$ and $b = 0.75$ sont des constantes, tf le nombre d'occurrences du terme t dans le document d, dl la longueur du document, dl_{avg} la longueur moyenne des documents, N le nombre total de documents et df le nombre de documents contenant le terme t. Les deux parties de l'équation peuvent être interprétés comme un TF et un IDF.

En RI, une pondération spécifique pour le terme dans la requête est utilisée. Dans notre cas, la requête étant aussi un article, nous avons adopté la même pondération TF. Nous avons également donné un poids plus important à l'IDF pour améliorer la précision de l'appariement. Finalement, la similarité entre un article à dater d et un article connu D_{ex} est :

$$sim(d, d_{ex}) = \sum_{t \in d \cap d_{ex}} TF_{BM25}(t,d) * TF_{BM25}(t,d_{ex}) * (IDF_{BM25}(t))^3$$

Une autre mesure de similarités a été testée mais n'a pas fait l'objet d'une soumission de run. Il s'agit d'une mesure basée sur le modèle de langue proposé pour la RI par Hiemstra et Kraaij (Hiemstra & Kraaij, 1999). Cette mesure offre plus de possibilités grâce aux différentes stratégies de lissage qui peuvent être utilisées. Mais elle implique des temps de calculs plus importants puisque même les mots absents des documents doivent être pris en compte. Ces temps de calcul et le nombre limité de runs par équipe ne nous a pas permis d'évaluer ces variantes des modèles de langue à la Hiemstra.

Pour appliquer ces mesures, nous pré-traitons les textes simplement en les étiquetant avec TreeTagger et en ne retenant que les noms communs et propres, verbes et adjectifs. Ce sont donc sur ces termes que sont calculées les similarités. Bien entendu, du fait des erreurs d'OCR, de l'ancien français ou d'anciennes orthographes dans certains articles, l'étiquetage produit lui-même des résultats très bruités. Comme nous l'avons dit précédemment, des pré-traitements sur les textes permettraient d'améliorer considérablement cette représentation et certainement les résultats, mais le manque de temps ne nous a pas permis de les mettre en œuvre.

3.3 Procédure de vote

Dans les runs fournis, les cinquante plus proches voisins ont été retenus. À partir des dix années ainsi collectées à partir de ces voisins, différentes stratégies peuvent être mise en œuvre pour décider de l'année à attribuer à l'article inconnu.

Il est par exemple possible de faire un vote et de garder l'année majoritaire, ou de calculer une moyenne ou une médiane sur les années. Dans notre cas, nous avons implémenté un vote pondéré par le score de similarité. Pour tirer au mieux parti du caractère continu des classes, il est important de faire en sorte que les années proches des années des articles voisins soient également considérées. Pour savoir quels poids donner à ces années voisines, nous utilisons la même fonction gaussienne que celle utilisée pour l'évaluation : l'année n du voisin reçoit un poids de $1 * sim(d, d_{ex})$, les années $n-1$ et $n+1$ reçoivent $0.969 * sim(d, d_{ex})$... Finalement, chacun des cinquante voisins vote donc pour son année de parution et les années connexes, pondéré par le score de similarité entre ce voisin et l'article, et l'année obtenant le poids le plus important est proposé.

3.4 Résultats

Les résultats de cette approche correspondent au run 3 de l'équipe TexMex pour les deux pistes de la tâche. Ces runs se classent premiers pour les deux pistes et les scores obtenus sur cet échantillon de test correspondent aux

évaluations menées en *leave-one-out* durant la phase d'entraînement. Ces résultats témoignent du bien fondé de notre approche, même si une large part à l'amélioration existe.

Outre ces résultats bruts, il est intéressant de noter de cette approche par k-plus-proches voisins est calculatoirement légère. Il n'y a en effet aucune phase d'apprentissage, et l'ajout de nouveaux documents datés peut bénéficier immédiatement aux résultats.

4 Contribution à la tâche 2 : appariement résumé/article

Cette section décrit notre participation à la tâche d'appariement entre résumés et articles scientifiques. Deux pistes étaient proposées, se différenciant par la présence ou non des introductions et des conclusions dans les bases d'article. Nous avons soumis un seul run sur chacune de ces pistes.

4.1 Vision de la tâche

Cette tâche d'appariement apparaît clairement comme une tâche classique de recherche d'information où le résumé joue le rôle de requête. Nous avons donc là encore utilisé une simple approche de calcul de similarité par pondération Okapi-BM25 (cf. supra). Comme précédemment, les documents ont été étiquetés à l'aide de TreeTagger et seuls les noms, verbes et adjectifs ont été conservés pour représenter les documents.

Lors de la phase d'entraînement, l'évaluation a montré d'excellents niveaux de performances, avec une précision de l'ordre de 99 % à 100 %. Nous n'avons pas cherché à améliorer ces résultats, étant donné le peu d'intérêt pour une tâche si simple et le manque de temps. En particulier, aucune adjudication n'a été faite ; un même article peut donc avoir été assigné à différents résumés.

4.2 Résultats

Les runs soumis à la piste 1 (articles complets) et à la piste 2 (articles sans les introductions et conclusions) obtiennent respectivement des précisions de 99.5 % et 99 %. Ces résultats correspondent bien aux évaluations que nous avons effectuées sur le training set et soulignent encore une fois la facilité de cette tâche.

5 Conclusion

Cette première participation de l'IRISA à DEFT se traduit donc par de bons résultats, malgré un calendrier trop serré pour réellement permettre le développement de méthodes innovantes. Ces bons résultats sont donc le fruit de l'emploi de techniques classiques, mais choisies de manière à être bien adaptées aux tâches et aux données.

Les deux tâches proposées sont très différentes par leur niveaux de difficultés. La tâche de datation mêle en effet plusieurs niveaux de difficultés (données bruitées, classification en classes continues, diversité des exemples,...) qui peut rendre l'analyse des résultats difficile. À l'inverse, la trop grande simplicité de la tâche d'appariement fait que les équipes ont toute obtenu des niveaux de performances comparables proche de la perfection.

En revanche, ces deux tâches relevaient bien du même paradigme de comparaison de documents, paradigme plus proche de la recherche d'information que de la fouille de données. Nos runs les plus efficaces se sont donc inspirés des techniques de RI pour mener à bien ces deux tâches.

Références

HIEMSTRA D. & KRAAIJ W. (1999). Twenty-one at trec-7 : ad-hoc and cross-language track. In *Proceedings of the 7th Text Retrieval Conference TREC-7, NIST Special Publication 500-242*, p. 227–238.

OGER S., ROUVIER M., CAMELIN N., KESSLER R., LEFÈVRE F. & TORRES-MORENO J. M. (2010). Système du lia pour la campagne deft'10 : datation et localisation d'articles de presse francophones. In *DEFT'10*.

RAYMOND C. (2010). Bonzaiboost. http ://bonzaiboost.gforge.inria.fr/.

ROBERTSON S. E., WALKER S. & HANCOCK-BEAULIEU M. (1998). Okapi at TREC-7 : Automatic Ad Hoc, Filtering, VLC and Interactive. In *Proceedings of the 7th Text Retrieval Conference, TREC-7*, p. 199–210.

SCHAPIRE R. E. & SINGER Y. (2000). BoosTexter : A boosting-based system for text categorization. *Machine Learning*, **39**, 135–168. `http://www.cs.princeton.edu/~schapire/boostexter.html`.