

Un nouveau schéma de pondération pour la catégorisation de documents manuscrits

Sebastián Peña Saldarriaga¹ Emmanuel Morin¹ Christian Viard-Gaudin²

(1) LINA - UMR CNRS 6241, Université de Nantes

(2) IRCCyN - UMR CNRS 6597, Université de Nantes

{Prénom.Nom}@univ-nantes.fr

Résumé. Les schémas de pondération utilisés habituellement en catégorisation de textes, et plus généralement en recherche d'information (RI), ne sont pas adaptés à l'utilisation de données liées à des textes issus d'un processus de reconnaissance de l'écriture. En particulier, les candidats-mot à la reconnaissance ne pourraient être exploités sans introduire de fausses occurrences de termes dans le document. Dans cet article nous présentons un nouveau schéma de pondération permettant d'exploiter les listes de candidats-mot. Il permet d'estimer le pouvoir discriminant d'un terme en fonction de la probabilité *a posteriori* d'un candidat-mot dans une liste de candidats. Les résultats montrent que le taux de classification de documents fortement dégradés peut être amélioré en utilisant le schéma proposé.

Abstract. The traditional weighting schemes used in information retrieval, and especially in text categorization cannot exploit information intrinsic to texts obtained through an on-line handwriting recognition process. In particular, top n ($n > 1$) candidates could not be used without introducing false occurrences of spurious terms thus making the resulting text noisier. In this paper, we propose an improved weighting scheme for text categorization, that estimates a term importance from the posterior probabilities of the top n candidates. The experimental results show that the categorization rate of poorly recognized texts increases when our weighting model is applied.

Mots-clés : Catégorisation de textes, écriture en-ligne, n-best candidats, pondération.

Keywords: Text categorization, on-line handwriting, n-best candidates, weighting.

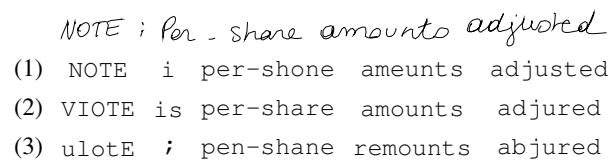
1 Introduction

Les avancées dans le domaine de la reconnaissance de l'écriture en-ligne permettent de produire, à partir d'un signal manuscrit, des textes en langue naturelle. Il devient alors possible d'appliquer des technologies de gestion du contenu normalement utilisées pour des textes électroniques tels que pages web et e-mails (Vinciarelli, 2006). Cependant, l'exploitation des données issues de la reconnaissance n'est pas aussi simple qu'il y paraît. En effet, les transcriptions sont souvent *bruitées*, c'est-à-dire qu'elles contiennent des suppressions, insertions et remplacements de mots par rapport au texte correspondant réellement au signal manuscrit.

La catégorisation automatique de textes est une problématique classique en intelligence artificielle liée au traitement automatique des langues. Toutefois, ce domaine n'a été exploré de

façon approfondie que pour les documents électroniques (Sebastiani, 2002), et peu de travaux existent sur la catégorisation de documents manuscrits. Des travaux récents se sont intéressés à cette problématique (Vinciarelli, 2005; Peña Saldarriaga *et al.*, 2009) et ont mis en évidence une différence, pouvant être significative, entre les résultats obtenus avec les données manuscrites et les textes électroniques originaux. L'ampleur de cette différence dépend de la quantité de bruit existant dans les documents issus du processus de reconnaissance de l'écriture.

Mais, si un système de reconnaissance induit du bruit dans les transcriptions qu'il produit, il peut également estimer la qualité du texte en sortie. En particulier, une probabilité peut être associée à chacun des mots du texte. De plus, une liste de candidats-mot à la reconnaissance peut également être obtenue comme le montre la figure 1.



NOTE i per-share amounts adjusted

(1) NOTE i per-shone ameunts adjusted

(2) VIOTE is per-share amounts adjured

(3) ulote i pen-shane remounts abjured

FIGURE 1 – Reconnaissance avec 3 candidats-mots

Nous pensons que l'utilisation de ces candidats-mot pour la catégorisation peut aider à réduire la différence de performances observée dans les travaux précités. Le travail proposé ici a pour but d'apporter une fonctionnalité de catégorisation en utilisant les listes successives de n-best candidats-mots à la reconnaissance, là où les approches explorées jusqu'ici se contentent d'utiliser la séquence de mots la plus probable donnée par le système de reconnaissance, contenant la plupart du temps le candidat-mot arrivé en tête de la liste.

Cependant, si l'utilisation des n-best candidats peut permettre de conserver l'information correspondant à un terme qui ne serait pas arrivé en première position, elle introduit également de fausses apparitions de mots avec un poids égal : le contenu du document s'en trouve altéré et la catégorisation aussi. Afin de réduire l'impact de ces fausses apparitions, il faut redéfinir les schémas de pondération classiques utilisés avec le formalisme vectoriel de représentation des données (Salton *et al.*, 1975), en nous basant sur la probabilité associée à chaque candidat-mot. De plus il serait convenable d'ajuster dynamiquement le nombre de candidats-mot, en seuillant sur la valeur des probabilités, limitant ainsi l'incidence des candidats très peu probables.

La section 2 s'attache à introduire la problématique de la reconnaissance de l'écriture en-ligne. Nous y décrivons également le moteur de reconnaissance utilisé et les ressources linguistiques qui lui sont associées. Le nouveau schéma de pondération basé sur les probabilités des candidats-mot est présenté dans la section 3. Afin de montrer l'intérêt du schéma de pondération proposé, nous décrivons en section 4 les expériences réalisées sur une base de documents reproduisant sous forme manuscrite les dépêches de l'agence Reuters bien connues dans le domaine de la catégorisation de textes (Debole & Sebastiani, 2005). Enfin, dans la section 5, nous concluons en évoquant les perspectives de ce travail.

2 Reconnaissance et écriture en-ligne

Souvent cantonnée à la saisie sur des terminaux de petite taille (PDA, smartphone), l'écriture en-ligne devient aujourd'hui une nouvelle source d'information en langue naturelle. Cela résulte

de l'émergence des dispositifs de saisie que sont les stylos électroniques couplés à des supports papier. Ils permettent de produire des véritables documents de diverses natures : notes de cours, copies d'examens, articles, formulaires, etc.

Dans le domaine de l'écriture en-ligne, un document se présente sous la forme d'une séquence de points ordonnés dans le temps $(x(t), y(t))$. Le tracé correspond à la trajectoire échantillonnée de l'instrument d'écriture, chaque point étant une position où le crayon s'est trouvé posé (cf. figure 2).

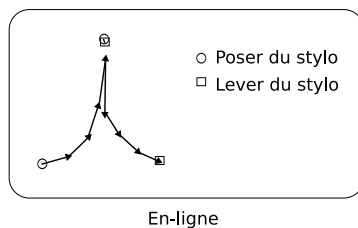


FIGURE 2 – Exemple de tracé en-ligne pour la lettre *i*

L'objectif de la reconnaissance en-ligne est de déterminer la suite de caractères la plus vraisemblable étant donné le signal correspondant au tracé manuscrit à l'aide d'informations fournies par un ensemble de connaissances *a priori* sur la langue (Perraud *et al.*, 2005).

Dans le cadre de cette étude, nous avons privilégié l'utilisation d'un moteur de reconnaissance stable et prêt à l'emploi : MyScript Builder¹. Ce moteur de reconnaissance permet d'associer différentes ressources linguistiques afin de guider et d'optimiser la reconnaissance (cf. figure 3).

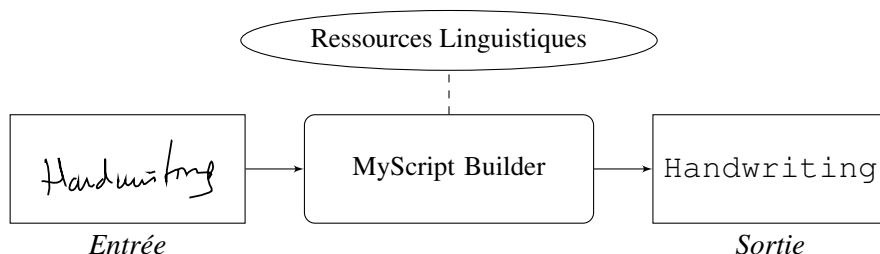


FIGURE 3 – Reconnaissance avec MyScript Builder SDK

Il est possible de définir des ressources spécifiques, soit sous forme de lexiques ou encore d'automates lexicaux, ou bien d'utiliser les deux ressources standard livrées avec MyScript Builder :

- **lk-text** est une ressource constituée d'un lexique standard et d'un modèle statistique du langage au niveau mot. Ce dernier permet de favoriser la reconnaissance des séquences de mots les plus probables. Ainsi, '*je tue*' sera prioritaire par rapport à '*je tu*'. Cette ressource permet également de reconnaître des éléments hors-lexique comme les dates, les codes postaux, etc.
- **lk-free** apporte peu de contraintes sur ce que l'on veut reconnaître. Il n'y a pas de lexique mais seulement un modèle de langage au niveau caractère. Cette ressource permet de favoriser les séquences de caractères les plus vraisemblables, par exemple '*MATIN*' sera prioritaire par rapport à '*MATIN*'.

1. <http://www.visionobjects.com/products/software-development-kits/myscript-builder/>

2.1 Évaluation de la reconnaissance

Le *bruit* induit par la reconnaissance est souvent mesuré au niveau des mots. Le taux d'erreur au niveau mot ou Word Error Rate (WER) correspond au pourcentage de mots mal reconnus sur la totalité de mots à reconnaître pour une séquence donnée :

$$WER = 1 - \frac{\sum_i^N \min(wf(i), wf'(i))}{\sum_i^N wf(i)} \quad (1)$$

Avec $wf(i)$ and $wf'(i)$ les fréquences du mot i dans le texte d'origine et le texte reconnu respectivement, et N le nombre de mots à reconnaître.

Une autre façon de mesurer le bruit, est de travailler au niveau terme. Le taux d'erreur au niveau terme ou Term Error Rate (TER) est plus adapté à la catégorisation car il tient compte de la normalisation de textes (*cf.* section 4.1). Puisque '*rêvas*' et '*rêves*' ont la même racine, reconnaître l'un à la place de l'autre ne modifie pas la liste de termes reconnus. Reconnaître '*pour*' à la place de '*par*' ne la modifie pas non plus, car quelque soit le mot reconnu, il sera filtré puisque c'est un mot outil.

Le TER est calculé grâce à la formule suivante (Vinciarelli, 2005) :

$$TER = 1 - \frac{\sum_i^N \min(tf(i), tf'(i))}{\sum_i^N tf(i)} \quad (2)$$

Avec $tf(i)$ and $tf'(i)$ les fréquences du terme i dans le texte d'origine et le texte reconnu respectivement, et N le nombre de termes de référence.

Dans nos expériences, nous utilisons ces deux mesures comme indicateurs de la qualité des documents produits en fonction de la ressource linguistique associée au moteur de reconnaissance.

3 Pondération et seuillage de candidats-termes

La mauvaise reconnaissance des documents engendre une catégorisation moins bonne (Vinciarelli, 2005; Peña Saldarriaga *et al.*, 2009). En effet, suite à la reconnaissance, un terme pertinent peut ne pas se trouver dans un document alors qu'il aurait dû y être. Or, les occurrences des termes sont au coeur de la réussite des algorithmes de catégorisation, et ce d'autant plus qu'ils utilisent le formalisme vectoriel et des schémas de pondération comme $tf \times idf$ (Spärck Jones, 1979). L'utilisation des n-best candidats-mot peut permettre de capturer l'information correspondant à un terme qui ne serait pas arrivé en première position. En effet, plus la liste de n-best est grande, plus le mot attendu a des chances de s'y trouver. Cependant, l'introduction artificielle de mots fausserait les résultats d'un algorithme de catégorisation. Dans ce contexte nous redéfinissons la pondération $tf \times idf$ pour tenir compte des probabilités des candidats des différentes listes de n-best. Dans un second temps, une stratégie de seuillage est proposée afin de filtrer des candidats très peu probables.

3.1 Pondération

Dans la suite de ce document nous considérons qu'un candidat-terme est simplement l'entité représentative du sens d'un candidat-mot dans l'espace vectoriel. Autrement dit, il s'agit de la

racine (Porter, 1980) ou du lemme (Namer, 2000) d'un candidat-mot.

Définition 1 *Fréquence d'un candidat-terme*

Soit $p_n(i)$ la probabilité d'un candidat-terme i dans la n -ième liste de candidats, et N le nombre de listes de candidats-terme dans lesquels i apparaît au sein d'un document. La fréquence du candidat-terme i est donnée par la formule suivante :

$$ctf(i) = \sum_{n=1}^N p_n(i) \quad (3)$$

Nous pouvons multiplier la fréquence ainsi obtenue par le facteur idf classique pour obtenir une mesure $ctf \times idf$ adaptée à l'exploitation des listes de candidats-mots.

Définition 2 *Mesure candidate- $tf \times idf$*

Soit K le nombre de documents dans un corpus, et k_i le nombre de documents dans lesquels le candidat-terme i apparaît au moins une fois. La pondération $ctf \times idf$ peut être calculée grâce à la formule suivante :

$$ctf.idf(i) = ctf(i) \times \log \frac{K}{k_i} \quad (4)$$

Afin de réduire les effets engendrés par les différences de longueurs des documents, il convient de normaliser cette mesure, en particulier lorsqu'elle est utilisée avec des approches à base de distances ou mesures de similarité.

Définition 3 *Mesure candidate- $tf \times idf$ normalisée*

Soit M le nombre de termes de l'espace de représentation vectorielle et i un candidat-terme donné. La mesure $ctf \times idf$ normalisée est donnée par la formule suivante :

$$nctf.idf(i) = \frac{ctf(i) \times \log \frac{K}{k_i}}{\sqrt{\sum_{j=1}^M (ctf(j) \times \log \frac{K}{k_j})^2}} \quad (5)$$

La définition de ce nouveau schéma de pondération va permettre de calculer facilement le poids d'un candidat-terme i dans un vecteur. Des méthodes de catégorisation standard ou des systèmes existants (Peña Saldarriaga *et al.*, 2009) peuvent alors être utilisés sans modification majeure.

3.2 Seuillage

Le but de la stratégie de seuillage proposée ci-dessous est d'ajuster dynamiquement la taille des listes de candidats. Nous supposons que la liste de n -best candidats est triée par ordre décroissant probabilité et que $\sum_{i=0}^n p(i) = 1$ avec n le nombre de candidats dans la liste et $p(i)$ la probabilité d'un candidat-terme i . Nous cherchons à trouver les k ($k < n$) premiers candidats tels que $\sum_{j=0}^k p(j) \approx t$, $0 < t \leq 1$ où t est le seuil désiré. La figure 4 montre le comportement de notre stratégie de seuillage pour $t = 0,8$ et différentes listes de n -best candidats.

| | | | | | |
|--------------------|-------------|-------------------|-------------|-------------|-------------|
| exchequer | 0,61 | implication | 0,54 | fuel | 0,23 |
| <u>exoneration</u> | <u>0,22</u> | implications | 0,14 | gull | 0,21 |
| exonerator | 0,11 | <u>imputation</u> | <u>0,12</u> | gulf | 0,19 |
| excheqwr | 0,04 | imprecation | 0,11 | <u>full</u> | <u>0,19</u> |
| exchcqr | 0,02 | implicated | 0,09 | cruel | 0,18 |

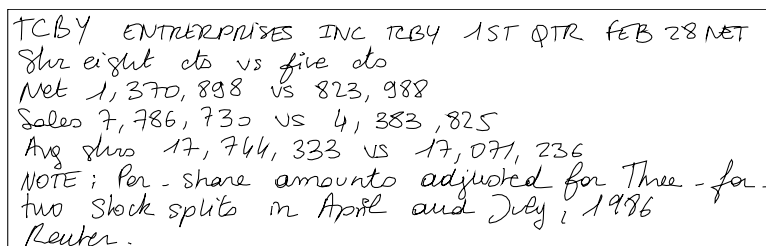
FIGURE 4 – Exemple de seuillage sur différentes listes de candidats

4 Expériences

Afin de valider le nouveau schéma de pondération, nous avons mené plusieurs expériences sur le corpus présenté ci-dessous. En premier lieu, nous avons effectué la reconnaissance des documents manuscrits et observé l'évolution des taux d'erreur en fonction du nombre de candidats-mot acceptés. Ensuite, nous avons catégorisé les documents en utilisant la sortie standard du système de reconnaissance ainsi que la sortie comportant des listes de candidats-mot à la reconnaissance. Les résultats de ces expériences sont présentés et commentés dans les sous-sections 4.2 et suivantes.

4.1 Données et paramètres expérimentaux

Pour la réalisation des expériences, nous avons utilisé un jeu de données composé de 2 029 dépêches du corpus Reuters-21578 reproduites sous forme manuscrite et réparties sur 10 classes. L'ensemble d'entraînement est constitué de 1 625 documents et celui de test de 404 dépêches. La partition en ensembles d'entraînement et de test suit le protocole ModApté (Apté *et al.*, 1994). Les données sont mono-catégorie, c'est-à-dire que les documents n'appartiennent qu'à une seule classe. La figure 5 montre un exemple de document manuscrit de notre base.



TC BY ENTERPRISES INC TC BY 1ST QTR FEB 28 NET
 Shr eight cts vs five cts
 Net 1,370,888 vs 823,988
 Sales 7,786,730 vs 4,383,825
 Avg shr 17,744,333 vs 17,071,236
 NOTE: Per-share amounts adjusted for Three-for-two stock splits in April and July, 1986
 Reuter.

FIGURE 5 – Document du corpus manuscrit

Nous avons choisi d'utiliser deux méthodes de catégorisation simples mais performantes. Il s'agit de la méthode des k-Plus Proches Voisins (kPPV) et des Séparateurs à Vaste Marge (SVM) (Vapnik, 1995), ces deux approches étant reconnues parmi les approches les plus performantes développées durant la décennie (Yang & Liu, 1999; Joachims, 2002; Debole & Sebastiani, 2005).

Avant l'application de ces algorithmes, une étape de normalisation a lieu. Elle consiste à segmenter les textes en occurrences de forme, filtrer les mots outils et appliquer l'algorithme de racinisation de Porter (1980). Durant la phase d'entraînement, l'ensemble des termes de l'espace de représentation des documents est choisi en utilisant la statistique du χ^2 (Yang & Pedersen, 1997) couplée à l'algorithme de Forman (2004).

L'évaluation du système se fait sur la base du document, c'est-à-dire en utilisant la micro-moyenne de la précision et du rappel. Comme les données sont mono-catégorie, sans rejet, la précision et le rappel inter-classes sont égaux (Beney, 2008). De ce fait, nous donnerons une seule mesure de la qualité d'un classifieur que nous appellerons *taux de classification*, correspondant à la micro-moyenne de la précision ou le rappel.

4.2 Reconnaissance

Les documents du corpus manuscrit sont reconnus en utilisant les deux ressources décrites précédemment : *lk-text* et *lk-free*. La figure 6 montre l'évolution du WER et du TER en fonction du nombre de candidats-mot.

Les textes reconnus avec la ressource *lk-free* sont fortement dégradés. En effet plus d'un mot sur deux est perdu en moyenne, alors qu'avec *lk-text* 77% des mots et autant de termes présents dans les textes sont correctement reconnus. Introduire des modèles de langage permet d'améliorer considérablement le taux d'erreur (Perraud *et al.*, 2005). Quand il n'y a pas d'*a priori* apporté par un tel modèle, comme c'est le cas de la ressource *lk-free* les performances d'un système de reconnaissance sont très mauvaises.

Nous observons également que plus la liste de candidats-mots est grande, plus le terme attendu a des chances de s'y trouver, le taux d'erreur s'en trouve alors diminué comme le montre la figure 6.

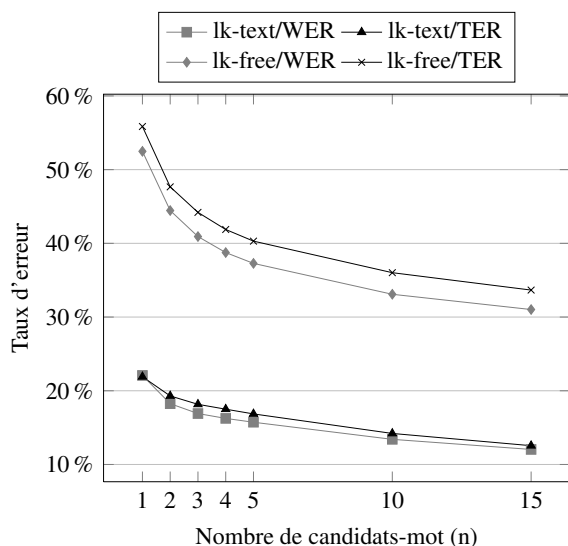


FIGURE 6 – Taux d'erreur en fonction du nombre de candidats-mot

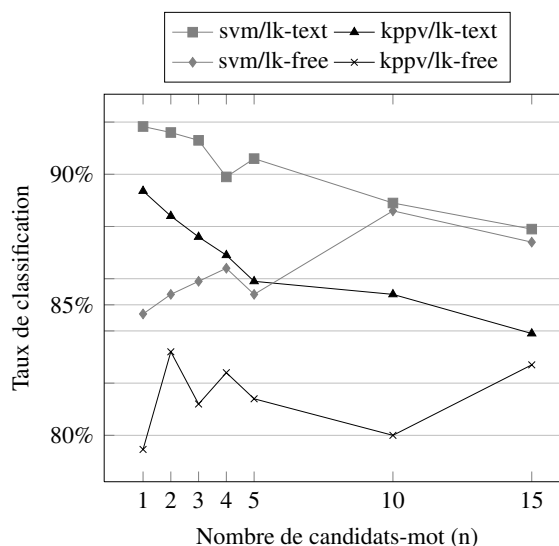


FIGURE 7 – Taux de classification en fonction du nombre de candidats

4.3 Catégorisation

La catégorisation des 404 documents d'évaluation a été effectuée aussi bien sur les documents issus de la reconnaissance avec un seul ou avec plusieurs candidats-mot.

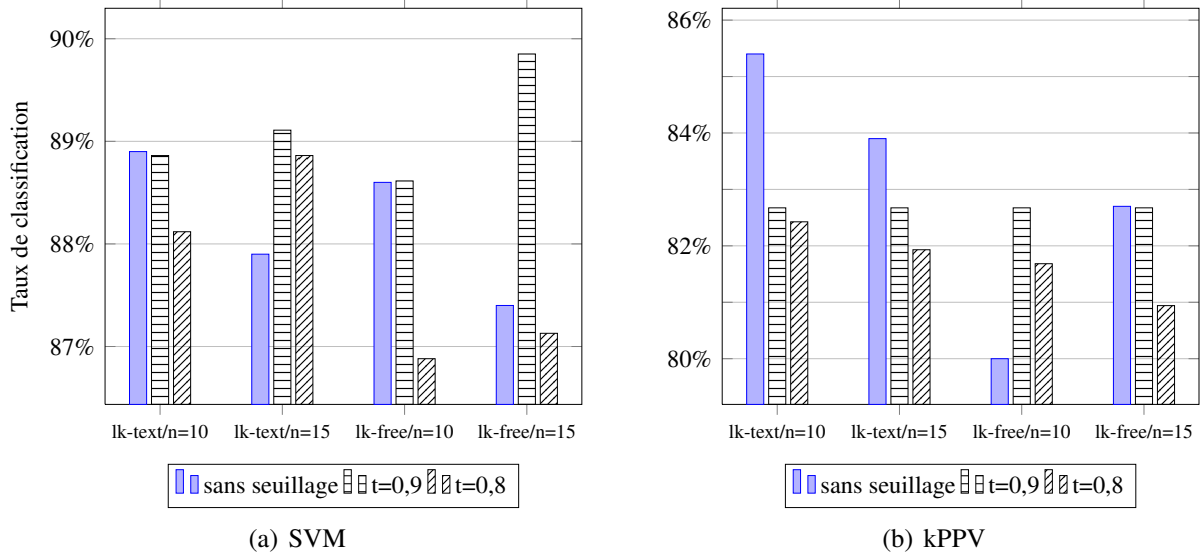


FIGURE 8 – Taux de classification avec seuillage

Les paramètres des classifieurs ont été optimisés afin d’obtenir le meilleur taux de classification. Nous avons utilisé pour cela, un sous-ensemble de l’ensemble d’entraînement reconnu avec un seul candidat-mot. Les SVM sont utilisés avec un espace vectoriel de 1000 termes, et les kPPV avec 300 termes et $k = 15$.

La figure 7 montre le taux de classification en fonction de l’algorithme, de la ressource et du nombre de candidats-mot utilisés.

Lorsque nous utilisons une liste de candidats-terme avec *lk-text* le taux de classification baisse avec l’augmentation du nombre de candidats. En revanche, lorsque les documents de *lk-free* sont utilisés, pour tout $n > 1$ le taux de classification est supérieur à celui obtenu en ne prenant que le premier candidat. Une augmentation moyenne de 2,1 % avec un écart-type de 1,2 peut être signalée et ce, quel que soit le classifieur utilisé.

Nous avons utilisé la stratégie de seuillage présentée en 3.2 pour $n = 10$ et $n = 15$, les résultats obtenus en fonction de la valeur du seuil t utilisés sont donnés par la figure 8. L’application du seuillage n’a pas permis d’améliorer, de façon générale, le taux de classification avec la ressource *lk-text*. L’amélioration observée pour les SVM et $n = 15$ paraît logique : le seuillage a eu pour effet de réduire n , le point le plus bas de la courbe *svm/lk-text* de la figure 7 va donc remonter car la courbe décroît de façon quasi-monotone en fonction du nombre de candidats.

Notre stratégie apparaît plus efficace lorsqu’elle est appliquée avec la ressource *lk-free*. Avec SVM et $n = 15$, une amélioration importante est observée, le taux de classification est même supérieur à celui obtenu avec des documents moins dégradés (*lk-text*) dans une configuration identique. Une légère augmentation avec kPPV et $n = 10$ se produit également, mais ne permet pas de dépasser les résultats obtenus avec les documents de *lk-text* dans la même configuration.

Nous pouvons observer qu’un seuil fort ($t = 0,9$) permet d’obtenir des meilleurs résultats par rapport à des seuils plus faibles. Cependant, les seuils utilisés ont été définis manuellement et peuvent ne pas être optimaux, ce qui pourrait expliquer en partie les résultats de notre stratégie de seuillage.

5 Conclusion

Le travail présenté dans cet article décrit un nouveau schéma de pondération pour l'utilisation des listes de n -best candidats-terme dans un processus de catégorisation textuelle de documents manuscrits en-ligne. Nous pensons que l'utilisation de ces listes permettrait d'atteindre un taux de catégorisation supérieur à celui obtenu avec juste le premier candidat.

Notre hypothèse de départ n'est pas entièrement confirmée par les résultats expérimentaux. En effet, l'utilisation des candidats-termes n'a pas permis d'améliorer le taux de catégorisation des documents où environ 77% des termes d'indexation sont correctement reconnus. En revanche, l'utilisation de la liste des n -best mots s'est révélée bénéfique pour des documents où plus de la moitié de l'information est perdue. Aussi bien avec les kPPV qu'avec les SVM, une augmentation moyenne de 2,1 % du taux de catégorisation a été observée.

La stratégie de seuillage proposée ne semble pas suffisante pour limiter l'influence des candidats très peu probables dans la catégorisation des documents de *lk-text*. Elle a permis cependant d'améliorer les résultats obtenus avec *lk-free* et un nombre de candidats important ($n \geq 10$). Ces résultats apparaissent prometteurs mais une question se pose, comment définir le seuil optimal ? De nouvelles expériences doivent être effectuées afin d'explorer différentes techniques d'estimation de ce seuil (validation croisée, leave-one-out, méta-heuristiques).

De manière générale, les résultats présentés dans cette contribution montrent que l'utilisation des listes de n -best candidats-terme permettent d'obtenir des niveaux convenables de catégorisation sur des documents fortement dégradés.

Remerciements

Ces travaux ont été soutenus par la Région Pays de la Loire à travers le Projet MILES et par l'Agence Nationale de la Recherche à travers le programme Technologies Logicielles (ANR-06-TLOG-009).

Références

- APTÉ C., DAMERAU F. & WEISS S. M. (1994). Towards language independent automated learning of text categorization models. In *Proceedings of the 17th Annual International ACM SIGIR Conference (SIGIR '94)*, p. 23–30.
- BENEY J. (2008). *Classification supervisée de documents*. Hermès Science / Lavoisier.
- DEBOLE F. & SEBASTIANI F. (2005). An analysis of the relative hardness of reuters-21578 subsets. *Journal of the American Society for Information Science and Technology*, **56**(6), 584–596.
- FORMAN G. (2004). A pitfall and solution in multi-class feature selection for text classification. In *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*.
- JOACHIMS T. (2002). *Learning to Classify Text using Support Vector Machines*. Kluwer Academic Publishers.

- NAMER F. (2000). Flemm : Un analyseur flexionnel du français à base de règles. *Traitement Automatique des Langues*, **41**(2), 523–547.
- PEÑA SALDARRIAGA S., VIARD-GAUDIN C. & MORIN E. (2009). On-line handwritten text categorization. In *Proceedings of Document Recognition and Retrieval XVI, IS&T/SPIE International Symposium on E.I. (DRR '09)*, volume 7247, p. 724709.
- PERRAUD F., VIARD-GAUDIN C., MORIN E. & LALLICAN P. M. (2005). Statistical language models for on-line handwriting recognition. *IEICE Transactions on Information and Systems*, **E88-D**(8), 1807–1814.
- PORTER M. F. (1980). An algorithm for suffix stripping. *Program*, **14**(3), 130–137.
- SALTON G., WONG A. & WANG C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, **18**(11), 613–620.
- SEBASTIANI F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, **34**(1), 1–47.
- SPÄRCK JONES K. (1979). Experiments in relevance weighting of search terms. *Information Processing & Management*, **15**, 133–144.
- VAPNIK V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.
- VINCIARELLI A. (2005). Noisy text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**(12), 1882–1895.
- VINCIARELLI A. (2006). Indexation de documents manuscrits. In *Actes du 9ème Colloque International Francophone sur l’Ecrit et le Document (CIFED '06)*, p. 49–54.
- YANG Y. & LIU X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference (SIGIR '99)*, p. 42–49.
- YANG Y. & PEDERSEN J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML '97)*, p. 412–420.