IDIAP RESEARCH REPORT

Indexation de Documents Manuscrits

Alessandro Vinciarelli [a] [b]

IDIAP–RR 06-31

Juin 2006

paru dans
Proceedings du Colloque International Francophone sur l'Ecrit et le Document (CIFED06), article invité

a IDIAP - alessandro.vinciarelli@idiap.ch
b Ecole Polytechnique Fédérale de Lausanne (EPFL) - 1015 Lausanne (Switzerland)

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

Rapport de recherche de l'IDIAP 06-31

Indexation de Documents Manuscrits

Alessandro Vinciarelli

Juin 2006

paru dans
Proceedings du Colloque International Francophone sur l'Ecrit et le Document (CIFED06), article invité

Résumé. Les systèmes de reconnaissance automatique de l'écriture permettent de transfomer des collections de documents manuscrits en archives de documents numériques. L'avantage n'est pas tellement la réduction de l'espace nécéssaire pour stoquer les données, mais plutôt la possibilité d'appliquer les technologies de gestion du contenu normalement utilisées pour des textes numériques tels que pages web et e-mails. Le problème principal dans une telle démarche est que les transcriptions sont généralement bruitées, c'est-à-dire qu'elles sont caracterisées par un taux d'erreur qui peut atteindre, dépendemment des cas, les 50 pour cent. Cet article montre que cela ne constitue pas un problème majeur dans deux cas importants: Information Retrieval et Text Categorization. Une comparaison des résultats obtenus avec les mêmes technologies d'indexation sur les transcriptions manuelles (sans erreur) et automatiques (environ 40% de Term Error Rate) des mêmes documents montre en effet que l'impact sur la performance de l'application finale est négligeable.

1 Introduction

La reconnaissance offline de l'écriture a atteint des résultats remarquables dans des domaines tels que la transcription des addresses postales ou la lecture des chèques bancaires. La raison principale est que les informations complémentaires (code postal et montant du chèque écrit en chiffres) et les contraintes expérimentales limitent la taille du lexique et la variabilité des données à reconnaître [12]. La situation est différente en ce qui concerne la transcription automatique de textes génériques. Dans ce cas, le lexique doit nécessairement inclure plusieurs dizaines de milliers de mots, et la seule connaissance *a-priori* disponible est le language dans lequel le document est écrit. Les résultats des travaux qui s'occupent de reconnaissance de textes montrent un taux d'erreur qui parfois avoisine les 50% [11, 28].

Les transcriptions qui en resultent sont donc illisibles pour un être humain, mais elles contiennent quand même suffisamment d'information pour l'application des technolgies basées sur l'indexation des textes. En fait, cet article montre que, au moins en ce qui concerne l'*Information Retrieval* (IR) [2] et la *Text Categorization* (TC) [19], l'effet des erreurs sur la performance est négligeable. En effet, l'effort supplémentaire demandé aux utilisateurs, dû á la présence d'erreurs, est acceptable (cf. Partie 4 pour plus de détails). Les deux raisons principales sont les suivantes : premièrement, l'indexation se base sur des statistiques de présence/absence des mots dans les documents et souvent, les mots les plus représentatifs du contenu sont écrits plusieurs fois. La redondance permet donc de faire face à une partie des erreurs de reconnaissance. Deuxièmement, les mécanismes de *query matching* en IR et les modèles de catégorie en TC se basent sur plusieures mots, et même en présence d'un taux d'erreur élevé, la probabilité de les manquer tous est faible.

Jusqu'ici, les approches de modélisation du contenu des documents manuscrits se sont limitées, à notre connaissance, à l'application du *keyword spotting*, c'est-à-dire à l'identification des mots clés soumis par les utilisateurs dans les transcriptions [8, 10, 17] ou dans les images mêmes des textes manuscrits [4, 7, 14, 15, 20, 21, 22]. Cela correpond en IR à une approche booléenne depassée par d'autres méthodes [2], mais surtout cela empêche d'utiliser d'autres technologies comme la TC, la sommairization automatique, la *topic segmentation* et d'autres. Pour cette raison, l'application de l'indexing peut représenter une avancée importante.

Cet article est une synthèse de plusieurs travaux précédents [25, 24, 27, 26] contenant tous les détails qui, pour des raisons de place, ne seront pas reportés ici. Dans le reste de l'article, la Partie 2 présente les techniques d'indexation, la Partie 3 décrit le système de reconnaissance de textes manuscrits utilisé, la Partie 4 montre les experiences et résultats et la Partie 5 contient les considérations finales.

2 Analyse des Textes numériques

Un système d'analyse de textes numériques se compose de deux parties : la première est l'*indexation* qui convertit les textes en vecteurs, objets plus faciles à manipuler pour un ordinateur. La deuxième est l'application spécifique qui constitue le but de l'indexation comme par exemple, dans le cas de ce travail, IR [2] et TC [6]. Une description exhaustive de IR et TC peut être trouvée en [2] et [6] respectivement. Les prochaines sous-sections, nous décriront les techniques principales d'indexation et les approches normalement utilisés en IR et TC.

2.1 Indexation

Le Vector Space Model (VSM) est l'approche la plus commune en indexation d'un corpus de documents. La raison de son nom est que les documents y sont representés sous forme de vecteurs dans les quels chaque composante correspond à un mot du *dictionnaire* (voir la suite pour plus de détails). D'autres techniques se basent sur des modèles probabilistiques ou booléens, mais ils n'ont pas l'efficacité du VSM [23]. Cela peut provenir de la difficulté à trouver les informations pertinentes pour l'entrainement, ou c'est parce que le modèle ne tient pas compte d'informations aussi importantes comme la fréquence des mots dans le documents et force les utilisateurs à exprimer leurs requêtes sous forme d'operations logiques, ce qui n'est pas toujours facile [2, 23]. Pour les raisons précedentes, cet article se concentre sur le VSM.

La première opération de l'indexation est le *preprocessing* pendant lequel tout symbol qui ne correpond pas à une lettre de l'alphabet (points, virgules, traits d'union, etc.) est eliminé. Cette opération est motivée par le fait que ces caractères ne sont pas liés au contenu des documents et peuvent donc être negligés. La deuxiéme opération est le *stopping* qui correspond à la suppression de tous les mots qui sont trop fréquents (ils n'aident donc pas à distinguer entre les documents) ou jouent un rôle purement fonctionnel dans la construction des phrases (articles, prépositions, etc.). Le résultat du stopping est que le nombre de mots dans la collection, ce qu'on appelle la *masse des mots*, est réduit en moyenne de 50%. Les mots à eliminer, connus comme *stopwords*, sont recoltés dans la *stoplist* qui contient en général entre 300 et 400 éléments.

Pour un ordinateur, les différentes *variations morphologiques* d'un même mot, par exemple *cuisine*, *cuisinier* et *cuisiner*, sont des mots différents. Pour cette raison, l'operation suivante dans le processus qui mène à l'indexation, est le *stemming*, c'est-à-dire le remplacement de tous les mots par leur racine (*stem* en anglais). Les termes *cuisine*, *cuisinier* et *cuisiner* sont ainsi remplacés par *cuisin*. Le stemming n'a pas d'impact sur la masse des mots, mais réduit de 30% en moyenne la taille du *dictionnaire*, c'est à dire la liste $T = \{t_1, \ldots, t_{|T|}\}$, $|T|$ étant le nombre d'éléments de T, des mots uniques qui apparaissent dans le corpus. Dans ce travail, le stemming a été effectué en utilisant l'algorithme de Porter [13].

L'étape finale de l'indexation est l'extraction de la matrice *terms et documents* A où l'élément a_{ij} est une mesure de l'impact du term t_i sur le document d_j. Il existe de nombreuses techniques qui permettent de calculer les valeurs a_{ij} (cf. [18] pour une étude exhaustive), mais en général on a toujours une expression du type :

$$a_{ij} = G(i)L(i,j) \tag{1}$$

où le facteur $G(i)$ est dit *global weight* parce qu'il ne depend pas du document et qu'il se base sur des informations extraites sur la totalité du corpus. Le facteur $L(i,j)$ est dit *local weight* parce qu'il utilise seulement de l'information contenue dans le document d_j. La conséquence d'une telle approche est que le même document est indexé différemment dans différentes collections. L'exemple le plus connu est le *tfidf* :

$$a_{ij} = tf(i,j) \cdot idf(i) = tf(i,j) \log\left(\frac{N}{N_i}\right) \tag{2}$$

où $tf(i,j)$ est la *term frequency*, c'est-à-dire le nombre de fois que le terme t_i apparaît dans le document d_j, et $idf(i)$ est l'*inverse document frequency*, c'est-à-dire le logarithme du rapport entre le nombre N de documents dans le corpus et le nombre N_i de documents qui contiennent le terme t_i. Ce schéma d'indexation donne plus de poids aux termes qui apparaissent avec une haute fréquence dans peu de documents. L'idée sous-jacente est que de tels mots aident à discriminer entre textes ayant différent sujet. Le *tfidf* a deux limites fondamentales : la première est que la dépendance de la term frequency est trop importante. Si un mot apparaît deux fois dans un document d_j, ça ne veut pas nécessairement dire qu'il a deux fois plus d'importance que dans un document d_k où il n'apparaît qu'une seule fois. La deuxième est que les documents plus longs ont typiquement des poids plut forts parce qu'ils contiennent plus de mots, donc les term frequencies tendent a être plus élevées. Pour aborder ces problèmes a été proposée une nouvelle technique d'indexation connue comme *Okapi Formula* [16] :

$$a_{ij} = \frac{tf(i,j)idf(i)}{[(1-b) + b \cdot NDL(d_j)] + tf(i,j)} \tag{3}$$

où $NDL(d_j)$ est la *longueur normalisée* de d_j, c'est à dire sa longueur (nombre de mots qu'il contient) divisée par la longueur moyenne des documents dans le corpus.

2.2 Information Retrieval and Text Categorization

Une fois qu'un corpus a été indexé, il est possible d'appliquer à ces documents différentes technologies. Cette partie illustre IR et TC.

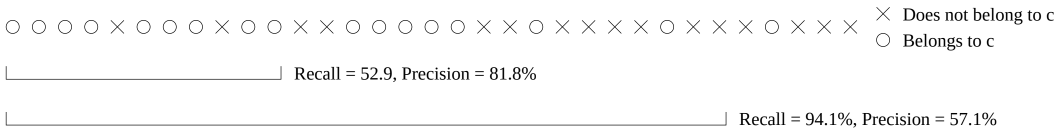

Fig. 1 – Precision et Recall. La figure montre comment on obtient les valeurs de Precision en fonction du Recall. Les cercles correspondent aux documents pertinents ou appertenant à une catégorie specifique.

Un système de IR est censé trouver, dans un corpus donné, les documents pertinents par rapport à une requête soumise en language naturel par un utilisateur. Cette tâche est accomplie en mesurant la similarité entre la requête, qui n'est rien d'autre qu'un court texte, et les documents du corpus. Cette opération permet d'associer une *Retrieval Status Value* $RSV(q, d)$ à chaque document d qui est calculée comme dans la formule suivante :

$$RSV(q, d_j) = \sum_{t_i \in q} a_{ij}. \tag{4}$$

Cela correpond à sommer les poids a_{ij} correspondants aux termes t_i qui apparaissent dans la requête q. Les documents qui n'ont aucun terme en commun avec la query ont donc une RSV nulle. Une fois que la RSV est disponible pour tous les documents du corpus, il est possible de faire un classement (en ordre descendant). On s'attend ainsi à trouver les documents pertinents en tête. Le système utilisé dans ce travail se base sur la formule d'Okapi qui permet d'obtenir des résultats comparables à l'état de l'art sur les tâches de retrieval plus fréquemment utlisées en litérature [2] comme benchmark.

La catégorisation consiste à attribuer chaque document à une catégorie c appartenant à un ensemble $C = \{c_1, c_2, \dots c_{|C|}\}$ prédéfini. L'approche qui a donné les meilleurs résultats jusqu'ici [19] utilise les *Support Vector Machines* (SVM) [3] pour décider sur l'attribution d'un document à une catégorie ou non. Les SVMs sont des classifieurs binaires qui peuvent être entrainés pour distinguer entre les documents qui appartiennent à une certaine catégorie et les autres. Typiquement, on entraîne une SVM pour chaque catégorie et chaque document est attribué aux catégories dont les SVM donnent des réponses positives. Cela permet de gérer le cas où les documents appartiennent à plusieures catégories en même temps. Dans le système utilisé dans ce travail, l'indexation utilise le *tfidf* décrit précédemment.

3 Système de Reconnaissance

Le système de reconnaissance utilisé dans cet article est décrit en détail dans [28]. Le système se base sur l'approche *sliding window* : une fenêtre de largeur fixe parcourt l'image et, à chaque position, un feature vector est extrait. Le résultat est la conversion de l'image des lignes de texte en séquences $O = \{o_1, \dots, o_{|O|}\}$ de vecteurs. La transcription est effectuée en choisissant la séquence de mots $\hat{W} = \{\hat{w}_1, \dots, \hat{w}_{|W|}\}$ appartenant au lexique telle que la probabilité *a-posteriori* est maximisée :

$$\hat{W} = \arg\max_{W} p(W|O)p(W). \tag{5}$$

La likelihood $p(W|O)$ est estimée avec des HMMs et des mixtures de Gaussiennes comme probabilités d'émission [5]. La probabilité *a-priori* que la séquence de mots soit W est estimée avec des bi-grams [5]. Le dictionnaire contient les 20000 mots les plus fréquents dans l'ensemble d'apprentissage (training) du corpus Reuters-21578.

La performance en reconnaissance est mesurée avec le *Term Error Rate* (TER) :

$$TER = 1 - \frac{\sum_i min(tf(i), tf^*(i))}{\sum_k tf(k)} \tag{6}$$

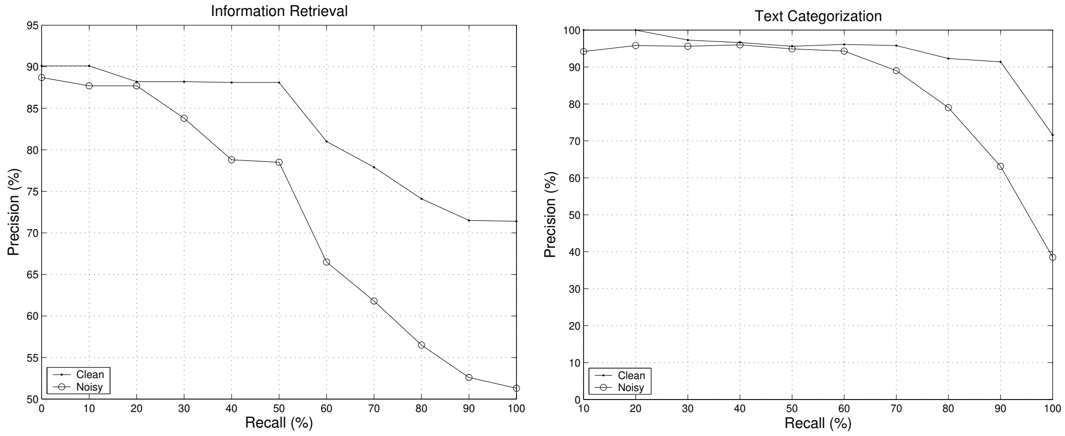

FIG. 2 – Resultats. Le plot à gauche montre les resultats des experience de IR, le plot à droite montre les resultats des experiences de TC.

où $tf(i)$ et $tf^*(i)$ sont les term frequencies du terme i respectivement dans le groundtruth et dans la transcription automatique. Le TER est mesuré après avoir effectué stopping et stemming (cf. Partie 2) pour obtenir une mesure qui correspond mieux à l'utilisation de la transcription pour l'indexation. Le TER sur l'ensemble de test (test set) de 200 documents utilisé dans ce travail est de 40.7% (cf. Partie 4).

4 Expériences et Résultats

Les expériences effectuées dans ce travail se basent sur le corpus Reuters-21578, un benchmark largement utilisé dans la littérature [9]. La collection a été partitionnée en training et test set en utlilisant le *ModApté* split [1]. Le training set a été utilisé pour entraîner les modèles de language mentionnés dans la Partie 3. Dans le test set, 250 textes ont été choisis au hasard et écrits à la main par une seule personne. Ce dernier ensemble de documents a été partagé un deux sous-ensembles, le premier contient 50 documents et le deuxième en contient 200. Le premier ensemble a été utilisé pour entraîner le système de reconnaissance de l'écriture, le deuxième a été utilisé pour effectuer les expérience de IR et TC décrites dans le reste de cette partie.

Les prochaines sous-sections présentent les mesures de performance appliquées et les résultats en termes de retrieval et catégorisation.

4.1 Precision et Recall

La performance des systèmes de IR et TC est mesurée généralement à l'aide des courbes de *Precision* π en fonction du *Recall* ρ. Dans les deux applications, les systèmes donnent en sortie un classement des documents du corpus. Dans le cas du IR, le classement se fait sur la base de la pertinence par rapport à une requête q, dans le cas de TC, le classement se fait par rapport à une catégorie c. Les documents pertinents relativement à q ou appartenant à c sont censés apparaître en tête du classement, mais la situation la plus commune est celle décrite dans la Figure 1. En correspondance des documents desirés (representés avec des ronds dans la Figure), on peut mesurer ρ, c'est-à-dire le pourcentage de documents pertinents ou appartenants à c correctement identifiés. En correspondance des mêmes documents on peut mesurer aussi π, c'est-à-dire la pourcentage auquel correpondent les documents desirés par rapport au nombre total de documents dejà rencontrés lors du classement.

Cette opération permet d'obtenir une courbe de π en fonction de ρ. Si le système marche parfaitement, la courbe est toujours à 100%, mais dans les cas plus réalistes, elle tend plutôt à descendre pour les valeurs plus élevées de ρ. Pour evaluer un système de IR, on crée un ensemble de requêtes et on obtient la courbe pour chacune. Ensuite, on effectue un *macroaverage*, c'est à dire qu'on prend la moyenne des valeurs pour $\rho = 10, 20, \ldots, 100\%$. La même opération peut être effectuée pour la catégorisation. On obtient une courbe différente pour chaque catégorie $c \in C$ et puis on en fait la macroaverage.

4.2 IR experiments

L'ensemble des requêtes (query set) Q utilisé dans ce travail contient 20 requêtes (on peut les trouver dans [26]) qui ont, en moyenne, moins de quatre documents pertinents. Puisque cela correpond à moins de 2% du corpus, la probabilité de repérer les documents pertinents au hasard est faible et la tâche peut être qualifiée de difficile [2]. Le même système de IR est testé sur les transcription manuelles sans erreurs (clean) et sur les transcriptions automatiques avec un TER moyen de 40.7% (noisy), les résultats sont illustrés par le graphique gauche de la Figure 2.

Puisqu'on a en moyenne quatre document pertinents, une Precision de 50% à 100% de Recall (comme dans le cas des transcriptions automatiques) signifie qu'on trouve tous les documents pertinents dans les huit premières positions. Dans le cas de la transcription clean, la Precision à 100% de Recall est à peu près 75%, ce qui correpond à dire qu'on trouve, en moyenne, tous les documents pertinents dans les cinq premières positions. Cela signifie donc que les erreurs de transcription n'ont pas un impact majeur sur la perception de l'utilisateur. En fait, les moteurs de recherche les plus communement utilisés présentent les résultats en groupes de dix et, dans le cas des experiences presentées ici, les documents pertinents seraient tous sur la première page dans les deux cas. Pour cette raison, l'effort de l'utilisateur serait à peu près le même.

4.3 TC experiments

Une SVM différente a été entraînée pour chacune des dix catégories les plus representés dans le corpus Reuters-21578. Cela correpond à une configuration expérimentale normalement utilisée dans la littérature [19] et permet d'être certain que chaque catégorie est representée dans le corpus de 200 documents utlisés comme test set. La courbe à droite de la Figure 2 montre le macroaverage des courbes de Precision en fonction du Recall obtenus pour les différentes catégories.

La différence de π est modeste jusqu'á un niveau de Recall de 70%. La raison de l'écart après un tel seuil est que pour chaque catégorie il y a quelques documents pour les quels le TER est plus haut que la moyenne et, pour cette raison, se retrouvent á la fin du classement. Cela suffit à baisser de plusieurs points π. En général, ce phénomène concerne des documents courts dans lesquels il n'y a pas beaucoup de termes liés à la catégorie. Quelques erreurs de reconnaissance sont donc suffisants à rendre le texte difficile a classer correctement pour les SVM. Dans le cas des documents de longueur moyenne (ou élevée), même en présence d'un haut TER, il y a toujours plusieurs termes liés à la catégorie qui sont correctement transcrits et cela suffit à les garder à la tête du classement.

5 Conclusion

Ce travail a montré les résultats de l'application de technologies habituellement utilisées pour des textes numériques aux transcriptions des documents manuscrits. Les expériences concernent notamment Information Retrieval et Text Categorization, deux applications qui essaient respectivement de modeliser le contenu des documents afin de réperer les documents pertinents à une requête et d'attribuer les documents à une catégorie prédéfinie. Les transcriptions automatiques ont, en moyenne, un TER de 40.7% (cf. Partie 3). Les résultats montrent que, pour IR ainsi que pour TC, la dégradation due aux erreurs de transcriptions a un impact modéré sur la perception des utilisateurs. Dans la tâche de IR utilisée, chaque requête a, en moyenne, quatre documents pertinents et il peuvent être repérés dans les huit premières positions (cinq premières dans le cas des transcriptions manuelles). Dans le cas de TC, la différence se remarque surtout pour un Recall supérieur à 70% et c'est l'effet de quelques documents qui, à cause des erreurs, se retrouvent à des positions plus basses dans le classement. Les limitations de ces experiences sont au nombre de deux : premièrement, les documents sont écrits par une seule personne, et deuxièmement, le corpus ne contient que 200 documents. Les résultats semblent donc montrer que les techniques d'indexation sont robustes pour des taux d'erreur élevés, mais des confirmations ultérieures sont nécéssaires en employant des bases de données plus larges et en incluant plus d'écrivains.

Remerciements L'auteur remercie les organisateurs de la conférence pour l'invitation et Agnès Just pour sa collaboration.

Références

[1] C. Apté, F. Damerau, and S.M. Weiss. Automated learning decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3) :233–251, 1994.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] C.J.C. Burges. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2) :121–167, 1998.

[4] A. Jain and A. Namboodiri. Indexing and retrieval of on-line handwritten documents. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 655–659. 2003.

[5] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

[6] T. Joachims. *Learning to Classify Text using Support Vector Machines*. Kluwer, 2002.

[7] A. Kolcz, J. Alspector, M. Augusteijn, R. Carlson, and G. Viorel Popescu. A line oriented approach to word spotting in handwritten documents. *Pattern Analysis and Applications*, 3 :153–168, 2000.

[8] T. Kwok, M.P. Perrone, and G. Russell. Ink retrieval from handwritten documents. In *Proceedings of Data Mining, Financial Engineering, and Intelligent Agents, Second International Conference*, pages 461–466. 2000.

[9] D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of 15th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–50. 1992.

[10] S. Marinai, M. Marino, and G. Soda. Indexing and retrieval of words in old documents. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 223–227. 2003.

[11] U.V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1) :65–90, 2001.

[12] R. Plamondon and S.N. Srihari. Online and offline handwriting recognition : a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1) :63–84, 2000.

[13] M.F. Porter. An algorithm for suffix stripping. *Program*, 14(3) :130–137, 1980.

[14] T.M Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 218–222. 2003.

[15] T.M Rath, R. Manmatha, and V. Lavrenko. A search engine for historical manuscript images. In *Proceedings of ACM SIGIR conference*, pages 369–376. 2004.

[16] S.E. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life : Okapi at TREC. *Information Processing and Management*, 36(1) :95–108, 2000.

[17] G. Russell, M.P. Perrone, and Y.M. Chee. Handwritten document retrieval. In *Proceedings of International Workshop on Frontiers in Handwriting Recognition*, pages 233–238. 2002.

[18] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24 :513–523, 1988.

[19] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1) :1–47, 2002.

[20] C.L. Tan, W. Huang, Z. Yu, and Y. Xu. Imaged document text retrieval without OCR. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6) :838–844, 2002.

[21] C.I. Tomai, B. Zhang, and V. Govindaraju. Transcript mapping for historic handwritten document images. In *Proceedings of International Workshop on Frontiers in Handwriting Recognition*, pages 413–418. 2002.

[22] S. Uchiashi and L. Wilcox. Automatic index creation for handwritten notes. In *Proceedings of International Conference on Acoustic, Speech and Signal Processing*, pages 3453–3456. 1999.

[23] C.J. Van Rijsbergen. *Information Retrieval*. Buttersworth, 1979.

[24] A. Vinciarelli. Effect of recognition errors on information retrieval performance. In *Proceedings of International Workshop on Frontiers in Handwriting Recognition*, pages 275–279. 2004.

[25] A. Vinciarelli. Noisy text categorization. In *Proceedings of International Conference on Pattern Recognition*, pages 554–557. 2004.

[26] A. Vinciarelli. Application of information retrieval techniques to single writer documents. *Pattern Recognition Letters*, 26(14-15) :2262–2271, 2005.

[27] A. Vinciarelli. Noisy text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12) :1882–1295, 2005.

[28] A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of large vocabulary cursive handwritten text. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6) :709–720, 2004.