De l'usage des scores et des alternatives de reconnaissance pour la classification d'images de documents manuscrits.

Jérôme Louradour — Théodore Bluche — Anne-Laure Bianne Bernard — Fares Menasri — Christopher Kermorvant

A2iA SA, 40 bis rue Fabert, 75007 Paris *{jl,tb,alb,fm,ck}@a2ia.com*

RÉSUMÉ. *Cet article explore différentes stratégies de représentation des transcriptions automatiques pour la classification de documents manuscrits. L'approche classique consiste à apprendre un classifieur statistique à partir du texte issu du reconnaisseur, mais elle ne prend pas en compte les spécificités du texte reconnu automatiquement : présence d'erreurs de reconnaissance, disponibilité d'un score de confiance voire d'alternatives de reconnaissance. Afin de prendre en compte ces aspects, nous proposons d'incorporer les scores de confiance en plus des mots reconnus comme pondération dans les vecteurs de caractéristiques utilisés par le classifieur ainsi que d'utiliser des alternatives de reconnaissance. Grâce à cela, nous apportons une amélioration systématique des résultats de classification pour différentes familles de classifieurs statistiques sur trois bases données de documents manuscrits.*

ABSTRACT. *This paper explores different strategies for automatic transcription representation in the scope of handwritten documents classification. The classical approach learns a statistical classifier directly from the recognizer's output, however it doesn't take into account the specificities of automatic text recognition: presence of errors and availability of confidence scores along with recognition alternatives. We propose here a method that considers these aspects. We suggest to use confidence scores as weights for the classifier's input features vectors and to take into account the n-best recognition alternatives. Using three handwritten documents databases and different families of statistical classifiers, we show that thanks to this approach, classification results are consistently improved.*

MOTS-CLÉS : *Classification automatique, reconnaissance de l'écriture manuscrite, HMM, réseaux récurrents, boosting, réseaux de neurones, SVM*

KEYWORDS: *Document classification, handwriting recognition, HMM, boosting, neural networks, SVM*

1. Introduction

Le traitement du courrier entrant (*mailroom automation*) est aujourd'hui, avec le traitement des archives, une des principales applications de la reconnaissance d'écriture manuscrite non contrainte. Contrairement à la reconnaissance d'écriture contrainte comme la reconnaissance d'adresses postales pour le traitement du courrier ou bien la reconnaissance de montants pour le traitement des chèques, les taux d'erreurs dans les transcriptions de documents non contraints sont encore relativement élevés. Cependant, même avec des erreurs de reconnaissance, la transcription automatique permet d'automatiser le traitement d'une partie du flux de documents qui parvient chaque jour aux entreprises et aux administrations. La reconnaissance d'écriture manuscrite est ainsi utilisée pour classer automatiquement des images de documents selon un plan de tri pré-défini permettant ainsi une réduction des coûts et des délais de traitement.

1.1. *Représentation "sac de mots" pour la classification d'images de document*

La classification d'images de documents en thèmes nécessite une première étape de reconnaissance de la séquence des mots présents dans chaque document. Vient ensuite l'analyse lexicale ou sémantique de ce contenu textuel pour catégoriser le document.

Les approches état de l'art consistent à appliquer des techniques d'apprentissage automatique de classification, généralement conçues pour des vecteurs d'attributs numériques, aux "*sacs de mots*" reconnus par le reconnaisseur (Vinciarelli, 2005, Saldarriaga *et al.,* 2008). Un *sac de mots* est un vecteur creux dont la taille dépend du nombre de mots différents présents dans la base de données d'entraînement de la tâche de classification. Chaque composante du vecteur correspond à un mot, et prend pour valeur 0 si le mot est absent, ou une valeur positive si le mot est présent. La valeur lorsqu'un mot est présent est 1 dans le cas d'un *sac de mots binaire*, ou un terme dépendant des fréquences d'apparition du mot. L'expression "*sac de mots*" fait référence au fait que l'ordre des mots dans le texte n'est pas pris en compte dans la modélisation utilisée pour la classification automatique. Bien que négliger l'ordre des mots puisse paraître limité, en pratique :

– les règles de classification de courrier (non automatique ou semi-automatique) se basent essentiellement sur la présence/absence de quelques mots clés, qui se suivent rarement.

– la reconnaissance automatique de texte n'est pas suffisamment robuste pour que soit fiable l'information selon laquelle plusieurs mots se suivent.

– des études ont déjà montré qu'utiliser une représentation 2-gram au lieu de 1-gram a tendance à dégrader les performances de classification automatique de documents manuscrits, même si une telle représentation permet d'améliorer la reconnaissance des mots (Toselli *et al.,* 2004)

1.2. *Motivations pour la classification de documents manuscrits*

L'approche état de l'art de la classification de documents manuscrits est de constituer les sacs de mots à partir des meilleures hypothèses de mots selon le reconnaisseur, sans tenir compte des autres mots que le reconnaisseur estime probables. Du point de vue de la classification, les erreurs du reconnaisseur constituent un bruit dans les *sacs de mots*. Des études ont confirmé et quantifié la corrélation qui existe entre ce niveau de bruit, estimé par les performances du reconnaisseur, et les performances de la classification sur le texte reconnu (Peña Saldarriaga *et al.,* 2010, Kermorvant *et al.,* 2010). On peut présumer que les "mots-clés", termes pertinents pour classifier les documents, sont en nombre limité. Ainsi, les erreurs de reconnaissance les plus coûteuses pour la classification sont

1) les échecs dans la reconnaissance des mots-clés et

2) les fausses détections de mots-clés.

Aussi, si l'on fait l'hypothèse que le nombre de mots-clés est très inférieur au nombre de mots possibles, les fausses détections sont négligeables en proportion par rapport aux carences de mots-clés. Afin de limiter la perte d'information induite par les erreurs de reconnaissance, nous montrons dans ce travail qu'il y a tout intérêt à tenir compte de toutes les alternatives que le reconnaisseur considère comme suffisamment probables pour un mot, une ligne ou une phrase localisé dans l'image d'un document. Cette approche a déjà été proposée par (Peña Saldarriaga *et al.,* 2009) pour la classification automatique de documents manuscrits en ligne. D'après (Peña Saldarriaga *et al.,* 2009), le gain relatif en précision pour la classification est d'autant plus marqué que la fréquence des erreurs de reconnaissance sur les mots est élevée. D'où la motivation pour enrichir la représentation des données disponibles pour la classification avec les alternatives de reconnaissance, notamment pour la classification d'images de documents manuscrits où le niveau de performance des reconnaisseurs est le plus critique.

2. Généralisation de la représentation sac de mots dans le cas d'une reconnaissance automatique de texte

Même si rajouter des alternatives de reconnaissance dans les *sacs de mots* permet de récupérer de l'information manquante dans l'approche de base, cela ajoute en contre-partie un certain niveau de bruit aux données. Si l'on suppose que le reconnaisseur est capable de renvoyer un score de confiance suffisamment fiable, alors une manière de remédier à ce problème est de tenir compte de ces scores. En fait, les techniques état de l'art d'apprentissage automatique en reconnaissance de texte consistent à optimiser des probabilités a posteriori sur les mots, par exemple dans les Modèles de Markov Cachés ou les Réseaux de Neurones Récurrents. Ainsi une généralisation directe d'un *sac de mots binaire* est un *sac de mots* où la valeur pour chaque mot est le maximum des probabilités estimées pour ce terme sur toutes les occurrences de mots détectées : la valeur est proche de 0 si le reconnaisseur est confiant que le mot n'est

pas présent dans un document, et proche de 1 lorsqu'il est quasiment sûr d'avoir vu le mot au moins une fois. Utiliser des scores de confiance calibrés (positifs et sommant à 1) est a priori crucial pour former des *sacs de mots* avec un *rapport signal sur bruit* maximal pour un reconnaisseur donné. Un élagage des scores est possible, par exemple en ignorant les scores inférieurs à un seuil étalonné, ou encore en se limitant aux n meilleures hypothèses (Peña Saldarriaga *et al.,* 2009) pour chaque unité de reconnaissance, qui peut être un mot, une ligne ou un paragraphe. Mais ce genre de technique n'est pas forcément utile pour améliorer la représentation du contenu textuel des documents, du moment que cette représentation prend bien en compte les scores de confiance. De plus, l'élagage est souvent appliqué au sein même du reconnaisseur en amont, pour des soucis d'efficacité computationnelle, par exemple lors de calculs de meilleur chemin dans un graphe de reconnaissance (*cf.* section 3.2.1).

La figure 1 illustre comment les résultats intermédiaires fournis par le reconnaisseur pour un document sont convertis en un vecteur *sac de mots*. L'exemple concerne un reconnaisseur qui fournit des alternatives au niveau ligne (ou paragraphe).

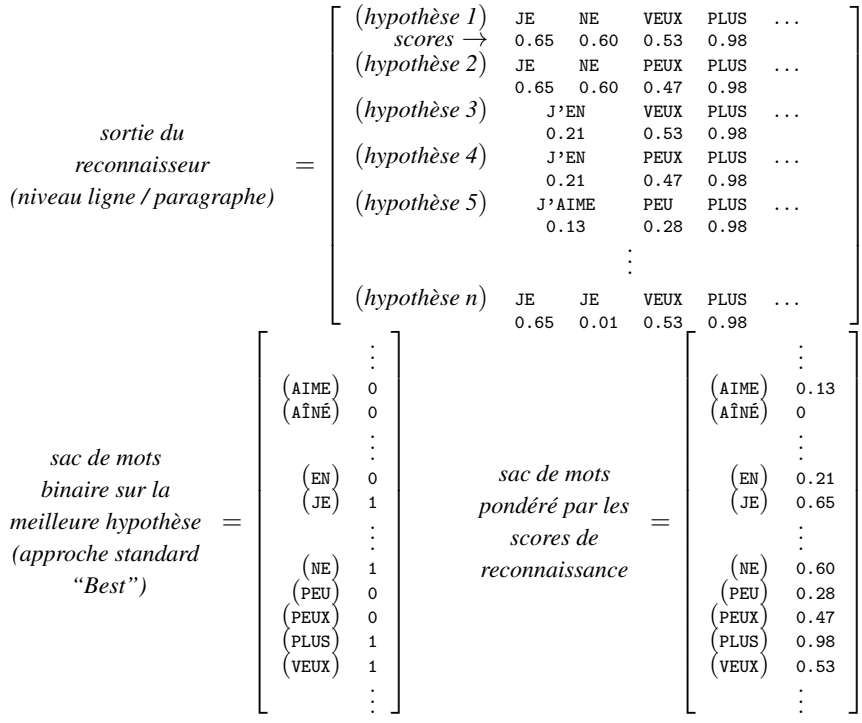

Figure 1. *Illustration de la méthode de conversion des résultats intermédiaires de la reconnaissance de texte en sacs de mots. Sur cet exemple, le reconnaisseur formule des hypothèses par ligne, et un calcul de score de confiance est fait au niveau mot.*

Soit $\{w_k\}_{k=1...K}$ l'ensemble des mots du vocabulaire (de taille K) disponible pour l'apprentissage de la classification, $\hat{w}_i^t$ le mot reconnu à la position t dans la $i^{\text{ème}}$ hypothèse de longueur T_i. Alors le *sac de mots* binaire est un vecteur creux de dimension K dont les valeurs sont :

$$x_k^{\text{bin}} = \begin{cases} 1 & \text{si} \quad w_k \in \{\hat{w}_1^t\}_{t=1...T_1} \\ 0 & \text{sinon} \end{cases} = \max_{t=1...T_1} \delta\left(\hat{w}_1^t, w_k\right), \qquad [1]$$

où δ est le symbole de Kronecker. Si l'on note maintenant s_t^i le score associé au mot $\hat{w}_t^i$, alors nous recommandons d'utiliser un *sac de mots* pondéré par les scores qui est une généralisation du cas binaire :

$$x_k^{\text{scores}} = \max_i \max_{t=1...T_i} s_k^i \times \delta\left(\hat{w}_i^t, w_k\right) \qquad [2]$$

Une alternative à la représentation binaire [1], populaire pour la classification de textes en thème, est la représentation dite TF-IDF (Salton *et al.,* 1988) que l'on peut écrire :

$$x_k^{\text{TF-IDF}} = \sum_{t=1}^{T_1} \delta\left(\hat{w}_1^t, w_k\right) \times \log_{10} \frac{N}{\text{DF}_k} \qquad [3]$$

où N est le nombre de documents dans la base d'apprentissage, et DF_k est le nombre de documents de cette base comportant au moins une occurrence du mot w_k dans la meilleure hypothèse de reconnaissance $\{\hat{w}_1^t\}_{t=1...T_1}$. Le terme de gauche est un terme de fréquence du mot dans le document (TF), et le terme de droite fait intervenir l'inverse de la fréquence de document contenant le mot (IDF). (Peña Saldarriaga *et al.,* 2009) propose une généralisation de la pondération TF-IDF pour prendre en compte les scores de reconnaissance, qui peut s'écrire :

$$x_k^{\text{scores/TF-IDF}} = \frac{\text{TF}_k^{\text{scores}} \times \log_{10} \frac{N}{\text{DF}_k}}{\sqrt{\sum_{k'=1}^{K} \text{TF}_{k'}^{\text{scores}} \times \log_{10} \frac{N}{\text{DF}_{k'}}}} \qquad [4]$$

où $\text{TF}_k^{\text{scores}} = \max_i \sum_{t=1}^{T_i} s_k^i \times \delta\left(\hat{w}_i^t, w_k\right)$ est une généralisation du terme TF.

3. Expériences

Dans cette étude, nous menons une série d'expériences afin d'évaluer les différents aspects du processus de classification par le contenu d'images de document. Les différents aspects que nous faisons varier sont :

la base d'images : nous réalisons des expériences sur 3 bases, dont deux publiques (RIMES, IAM).

le reconnaisseur : nous testons deux types de reconnaisseurs, l'un basé sur des Modèles de Markov Cachés (HMM) hybrides et une segmentation graphème, et l'autre basé sur des Réseaux de Neurones Récurrents (RNN).

le type de représentation pour la classification : nous comparons différents types de sacs de mots à partir des résultats de transcription fournis par le reconnaisseur, comme décrit dans la section 2.

la méthode de classification de texte : nous évaluons trois types de méthodes de classification automatique, dont les Machines à Vecteurs de Support (SVM) très populaires pour la classification de texte, mais aussi les Réseaux de Neurones Artificiels (ANN) et le Boosting Adaptatif (AdaBoost).

Ces différents aspects sont décrits en détail dans les sections suivantes.

3.1. *Bases de données*

Nous menons des expériences sur trois bases d'images de documents exclusivement manuscrits.

3.1.1. *Base RIMES*

La base RIMES (Augustin *et al.,* 2006) a été développée afin d'évaluer les performances des systèmes de traitement de documents manuscrits. Elle comporte en particulier un ensemble de 5599 pages manuscrites pour lesquelles une transcription humaine et une classification thématique sont disponibles. Les différents classes de documents de la base RIMES (avec le nombre d'exemples correspondant) sont :

> *Modification de contrat* (1350), *Demande d'information* (1038), *Gestion de sinistre* (611), *Changement de données personnelles* (599), *Relance* (596), *Fermeture de compte* (463), *Réclamation* (327), *Difficulté de paiement* (312), *Ouverture de compte* (301), *Inconnu* (2).

Cette base a été constituée par des scripteurs volontaires qui ont rédigé les lettres selon des scénarios pré-définis mais avec leur propre formulation. Cette base peut donc être considérée comme réaliste car la formulation du contenu des documents était libre, mais l'écriture et la mise en page sont trop soignées pour en faire une base proche de problèmes industriels réels.

3.1.2. *Base IAM*

La base IAM (Marti *et al.,* 2002) version 3.0 est constituée de 1539 pages manuscrites correspondant à des textes extraits du corpus LOB (Johansson *et al.,* 1978). Ces textes ont été écrits par 657 scripteurs différents. Aucune tâche de classification n'a été définie sur la base par ses concepteurs, mais les textes étant issus du corpus LOB, ils correspondent à des types différents : presse, romans, textes religieux, etc. Les différents types de textes sont présentés par (Karlgren *et al.,* 1994) qui propose une réorganisation des 19 classes originales en 15 ou 4 classes plus générales. Nous proposons ici un niveau intermédiaire avec 7 classes, qui regroupe les classes originales de la façon suivantes (avec pour chaque classe, ne nombre d'exemples disponibles) :

Fiction (446) : `"General fiction/Novels", "Mystery and detective fiction/Novels", "Science fiction/Short stories", "Science fiction/Novels", "Adventure and western fiction/Novels", "Romance and love story/Novels", "Humour/Articles from periodicals", "Humour/Novels".`

Belles Lettres, etc. (342) : `"Popular lore/Popular politics psychology sociology", "Belles lettres biography essays/Biography memoirs".`

Press (313) : `"Press reportage/political", "Press editorial/Personal editorial", "Press editorial/Institutional editorial".`

Press Reviews (134) : `"Press reviews/Press reviews".`

Gov.Doc & Misc (130) : `"Miscellaneous/Government documents", "Learned and scientific writings/Natural sciences".`

Skills and Hobbies (92) : `"Skills trades and hobbies/Hobbies", "Skills trades and hobbies/Homecraft handiman".`

Religion (82) : `"Religion/books".`

Ces 7 classes sont celles utilisées pour la tâche de classification de documents sur la base IAM. Cette base n'est pas constituée de documents réalistes, mais le plan de classification est difficile car il se base à la fois sur le contenu et sur le style d'écriture.

3.1.3. *Base A2iA-ArSf*

La base A2iA-ArSf est une base de documents constituée de courriers client d'une grande société ayant mis en place un système automatique de gestion du courrier entrant (mailroom). Ces documents ont été classés selon un plan de tri par des opérateurs et une transcription humaine du contenu a été réalisée. Cette base comprend 1649 lettres manuscrites réparties dans les 14 classes suivantes (avec le nombre d'exemples correspondant) :

> *Résiliations* (689), *Modifications complexes* (322), *Multimédia et offres diverses* (187), *Modifications simples* (109), *Actes simples* (96), *Réclamations* (93), *Actes complexes* (82), *Juridique* (52), *Gestion comptes clients* (8), *Actes de gestion* (5), *Information sur mobiles* (3), *CTI* (1), *Encaissements* (1), *Formulaires* (1).

Cette base est constituée de documents réels caractérisés une disposition, une qualité et un style d'écriture très variables.

3.2. *Reconnaisseurs*

3.2.1. *Reconnaissance multi-mots*

Notre approche pour la reconnaissance de documents complets est basée sur une segmentation en ligne puis sur une segmentation explicite de la ligne en mots. Cette

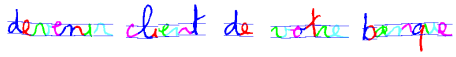

Figure 2. *Décomposition d'une ligne de texte en graphèmes.*

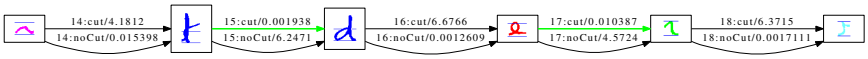

Figure 3. *Graphe des probabilités de segmentation en mot pour la ligne "devenir client de votre banque" (les poids sur les arcs sont les log-probabilités inverses). Seule la partie centrale de ligne autour du mot "de" est présentée. Les arcs en vert gras correspondent à la segmentation correcte.*

approche fait l'hypothèse que la segmentation correcte d'une ligne en mots peut être obtenue en testant un nombre limité d'options de segmentations. Cette hypothèse est souvent satisfaite pour la reconnaissance de textes de bonne qualité graphique mais n'est pas réaliste dans le cas d'une écriture tassées (par manque de place) ou pour l'écriture arabe. Cette approche a l'avantage d'être plus rapide qu'une approche par fenêtre glissante sur la ligne complète. Une fois les options de segmentation en mot construites, un reconnaisseur de mot isolé est appliqué sur chaque position de mot pour obtenir un treillis de reconnaissance. Le processus complet de reconnaissance d'une ligne de texte est donc :

– calculer la segmentation graphème sur la ligne complète (voir Fig. 2).

– en utilisant un réseau de neurones, calculer pour chaque couple de graphèmes consécutifs la probabilité qu'ils appartiennent à deux mots différents (voir Fig. 3).

– construire le graphe de segmentation en cherchant les S meilleurs chemins dans le graphe des probabilités de coupe entre graphèmes (voir Fig. 4).

– reconnaître chaque option de mot dans le graphe de segmentation avec le reconnaisseur de mot isolé.

– construire le treillis de reconnaissance en conservant les N_m meilleures options de reconnaissance pour chaque position de mot.

– composer le treillis de reconnaissance avec le modèle de langage.

– chercher les N_l meilleurs chemins dans le graphe composé.

Pour les expériences décrites dans cet article, nous avons choisi de ne pas appliquer de modèles de langage afin de tester les différentes représentations de documents et les différents classifieurs avec des transcriptions très diverses et bruitées.

Figure 4. *Liste des options possibles de segmentation en mot.*

3.2.2. *Reconnaisseur à base de Graphèmes et HMM Hybride (NN-HMM)*

Nous avons utilisé un reconnaisseur basé sur des Modèles de Markov Cachés (HMM) hybrides et une segmentation explicite en mot et en graphèmes (Knerr *et al.,* 1998). Lors du décodage, après la segmentation de la ligne en mots (en autorisant des options de segmentation) et l'extraction des graphèmes, un vecteur de caractéristiques décrivant chaque graphème est calculé. Un réseau de neurones de type MLP est utilisé pour prédire la probabilité de chaque classe de graphème et un HMM décrit les découpages possibles de chaque lettre en graphème. Les contraintes lexicales sont prises en compte en construisant les modèles HMM des mots du vocabulaire et, pour chaque mot dans les options de segmentation, en cherchant le meilleur chemin dans cet ensemble de HMM.

3.2.3. *Reconnaisseur à base de Réseaux de Neurones Récurrents (RNN)*

Ce reconnaisseur est basé sur des réseaux de neurones récurrents de type *Multi-Dimensional Long-Short Term Memory* (MDLSTM) (Graves *et al.,* 2009) qui permettent de modéliser des séquences en apprenant des dépendances à long-terme. Ces modèles combinent plusieurs parcours de l'image (haut/bas, gauche/droite) et l'extraction de caractéristiques est apprise directement à partir des valeurs de pixels. Les modèles ont été entraînés avec la librairie RNNLib [1] et le décodage a été adapté pour prendre en compte le vocabulaire et fournir des résultats de reconnaissance au niveau mot.

3.3. *Méthodes de classification*

Dans cette section, nous présentons trois familles de techniques de classification automatique, sur lesquelles nous menons nos expériences de classification d'images de document à partir des sorties d'un reconnaisseur de mots.

1. `https://github.com/mathfun/RNNLIB`

Dans la suite, les performances de la classification sont évaluées par 5-*fold* validation croisée. En outre, pour chaque *fold*, 80% des données d'apprentissage sont utilisées pour l'optimisation de la classification, et les 20% restantes sont utilisées pour calibrer quelques hyper-paramètres d'apprentissage propres à chaque méthode.

3.3.1. *Boosting Adaptatif (AdaBoost)*

L'algorithme de classification AdaBoost que nous testons fait partie des méthodes dites de sélection de variables. Le principe consiste à combiner linéairement plusieurs classifieurs simples ("*weak learner*") qui sont dans notre implémentation des arbres de décisions binaires de faible profondeur. Lorsque les entrées sont textuelles, chaque nœud d'un arbre correspond à un des mots du vocabulaire, dont il teste :

– la présence dans le cas de *sacs de mot binaire*, ou

– la supériorité par rapport à un seuil dans les cas où l'on manipule des scores de reconnaissance ou une pondération TF-IDF.

Pour un document d'entrée, l'arbre de décision *weak learner* fait un vote pondéré sur toutes les classes selon le résultat du test de la feuille terminale. Pour sélectionner de manière incrémentale les mots (avec seuils numériques associés s'il y a lieu) et apprendre les poids des votes, nous avons utilisé l'algorithme *Real AdaBoost MH* comme décrit dans (Schapire *et al.,* 2000). Nous avons entraîné deux variantes : une avec des souches d'arbre ("*stumps*"), et une avec des arbres de profondeur 3.

Concernant la généralisation de *Real AdaBoost MH* à des *sacs de mots* non binaires, nous préconisons de considérer le seuil associé à chaque nœud d'arbre comme un paramètre libre du *weak learner*, au même titre que le mot sélectionné. Même si (**?**) décrit cette méthode comme trop onéreuse (et propose une alternative sous-optimale), l'optimisation des seuils peut en fait être calculée de manière très efficace en ordonnant, pour chaque mot, les valeurs numériques associés à ce mot dans la base de données. Cette représentation ordonnée des données, calculée une seule fois au début de l'apprentissage, rend possible une recherche incrémentale du seuil optimal pour chaque mot, avec un coût computationnel moindre. Cette astuce de calcul est décrite par (**?**) pour l'algorithme *Robust LogitBoost*.

3.3.2. *Machines à Vecteurs de Support (SVM)*

Les SVM sélectionnent les exemples d'apprentissage qui permettent de définir les frontières de décision offrant le plus de marge, en se basant sur une mesure de similarité, le noyau, dont le choix peut être crucial (Cortes *et al.,* 1995). Dans nos expériences nous avons testé trois fonctions noyaux :

1) le noyau linéaire, qui est un bon candidat pour bien généraliser sur des problèmes à haute dimension,

2) le noyau Gaussien, qui est le plus populaire en général, et

3) le noyau cosinus, qui est très populaire en classification de texte.

L'optimisation des SVM est faite en utilisant la librairie `libSVM` (Chang *et al.,* 2011). Les SVM font intervenir un hyper-paramètre C de régularisation, ainsi qu'un hyper-paramètre de localité γ dans le cas du noyau Gaussien. Dans nos expériences, nous utilisons les données de validation pour choisir la valeur de C parmi $\{1, 10, 100\}$[2], et γ parmi $\{1/d_{10\%}, 1/d_{50\%}, 1/d_{90\%}\}$, où $d_{50\%}$ (resp. $d_{10\%}$) est la valeur médiane (resp. le $10^{\text{ème}}$ centile) des distances calculées sur 1000 paires d'exemples d'apprentissage tirés aléatoirement[3].

3.3.3. *Réseaux de Neurones Artificiels (ANN)*

Au lieu de sélectionner les variables ou les exemples d'apprentissage, les Réseaux de Neurones Artificiels apprennent des caractéristiques linéaires sur les données qui sont ensuite combinées à travers une ou plusieurs couches de non-linéarité. Pour la classification, la technique état de l'art consiste à choisir des neurones de sortie du réseau analogues à des probabilités a posteriori de chaque classe (un neurone de sortie par classe) : la normalisation dite "*softmax*" permet cela. Dans nos expériences, l'optimisation des réseaux est faite par descente de gradient stochastique (LeCun *et al.,* 1998) sur l'opposé de la log-vraisemblance. Aussi le choix du pas de gradient et l'*early stopping* sont faits par validation.

Contrairement à AdaBoost, dont l'apprentissage est invariant par rapport aux transformations scalaires monotones sur les variables d'entrée, la normalisation des données est cruciale pour les réseaux neuronaux afin d'éviter des phénomènes de saturations qui compromettraient l'optimisation, dans le cas où les valeurs numériques en entrée peuvent varier dans un intervalle relativement large (ce qui est le cas avec TF-IDF). Dans nos expériences, nous normalisons les données de manière à avoir une moyenne nulle et une variance unitaire sur *toutes* les valeurs numériques présentes dans les données d'apprentissage. Il est important de ne pas faire la normalisation indépendamment pour chaque mot, afin de ne pas perdre l'information de la pondération IDF.

Nous testons deux types de réseaux de neurones : la régression logistique (qui est un réseau de neurones sans couche cachée) et une variante des Machines de Boltzmann Restreintes (RBM) pour la classification (Larochelle *et al.,* 2008). Les RBM pour la classification ont le même pouvoir de modélisation qu'un réseau de neurones à une couche cachée. La fonction de transfert est différente, et des expériences préliminaires ont montré que les RBM avaient de performances similaires et parfois meilleures qu'un réseau avec une couche cachée de neurones *tanh*.

---

2. Dans le cas de *sacs de mots* basés sur une pondération TF-IDF, nous avons utilisé la même normalisation des données que celle décrite dans la sous-section 3.3.3 pour les réseaux de neurones

3. voir `http://blog.smola.org/post/940859888/easy-kernel-width-choice`

3.4. *Résultats*

Le tableau 1 présente les performances de classification en fonction des types de représentations *sac de mots* sur les trois bases de données pour différentes méthodes de classification. Nous y donnons aussi à titre indicatif les taux d'erreurs de la transcription fournie par le reconnaisseur pour chaque base, ainsi que les taux d'erreurs de classification en utilisant les transcription humaine (qui peut être vue comme un reconnaisseur idéal). Pour chaque famille de méthode de classification présentée en section 3.3, nous avons moyenné les performances sur les différents choix techniques évoqués (par exemple, "SVM" présente les performances moyennés sur les trois types de noyaux essayés).

Les figures 5 et 6 permettent de comparer différents facteurs de la représentation *sac de mots* au moyen de boîtes à moustaches[4] de (Tukey, 1977), qui représentent schématiquement les distributions des performances de classification. Chaque boîte à moustaches correspond à une base de donnée, un classifieur et une famille de représentations *sac de mots*. Contrairement au tableau 1, ces figures permettent de comparer les différentes variantes des méthodes de classification entre elles.

La première conclusion est que malgré des taux d'erreur de transcription élevés, la classification des documents transcrits est réalisable avec des taux d'erreur du même ordre de grandeur qu'avec la transcription humaine. Aussi, l'utilisation des alternatives et des scores de reconnaissance permet dans tous les cas une amélioration des performances de classification par rapport à l'approche de base (figure 5). Les seuls cas où l'amélioration n'est pas significative correspondent à une classification par AdaBoost, notamment sur la base IAM.

En ce qui concerne la comparaison entre une représentation *sac de mots* binaire et une représentation TF-IDF, on note que l'utilisation de la représentation TF-IDF est bénéfique pour les SVM et les réseaux de neurones sur les bases RIMES et IAM. Une analyse plus approfondie nous a montré que le gain est essentiellement dû à la pondération TF pour la base IAM, et exclusivement dû à la pondération IDF pour la base RIMES (ce qui n'est pas surprenant étant donné qu'il y a peu de répétitions de mots au sein d'un même courrier). Cependant, la représentation TF-IDF n'apporte rien sur A2iA-ArSf, et nous avons observé la même chose sur plusieurs autres bases réelles de classification de courrier entrant. Notre interprétation est que la fréquence des mots dans les courriers (IDF) n'est pas liée à leur pouvoir discriminant pour déterminer la classe des courriers : en particulier, beaucoup de mots peu fréquents ne sont pas pertinents pour la classification. L'exception de la base RIMES pourrait venir du manque de spontanéité dans la rédaction des courriers : étant donné que les scripteurs ont écrit un courrier avec en tête un thème donné, ils ont peut-être fait moins de disgressions que dans les courriers spontanés.

Un résultat surprenant de cette étude est que AdaBoost ne semble généralement pas profiter des scores associés aux mots, qu'il s'agisse des scores de reconnaissance

4. traduction de *Box & Whiskers Plot*

Tableau 1. *Taux d'erreur de classification (%) en fonction de différentes représentations sacs de mots, pour plusieurs bases de données, reconnaisseurs et méthodes de classification. Les colonnes "10-Best" correspondent à l'approche proposée, sachant que les reconnaisseurs ne retiennent que les 10 meilleurs candidats pour chaque localisation de mot. Nous donnons à titre indicatif, en italique, les taux d'erreurs de la transcription fournie par le reconnaisseur pour chaque base (colonne de gauche), et les taux d'erreurs de classification en utilisant les transcription humaine (colonne "Transcription humaine"). Les taux d'erreurs en gras indiquent les meilleures performances de classification obtenues à partir des mots reconnus automatiquement, ainsi que les performances qui ne sont pas significativement moins bonnes selon un test de Student apparié (avec un intervalle de confiance bilatéral à 95%).*

Base & Reconnaisseur	**Méthode de classification**	**Sacs de mots**					
		Binaire [1]		Score [2]	TF-IDF [3]		TF-IDF Score[4]
		Transcription humaine	Reconnaisseur Best	10-Best	*Transcription humaine*	Reconnaisseur Best	10-Best
RIMES NN-HMM	SVM	*4.34*	8.20	6.36	*3.05*	7.02	**5.16**
	ANN	*4.59*	7.79	6.66	*3.59*	6.70	**5.48**
43% d'erreur en transcription	AdaBoost	*4.07*	8.18	**6.55**	*4.18*	9.27	**6.50**
RIMES RNN (*)	SVM	*5.03*	8.41	7.13	*3.82*	6.36	**4.94**
	ANN	*5.18*	7.73	6.89	*3.73*	6.69	**4.89**
25% d'erreur en transcription	AdaBoost	*4.85*	**7.75**	**7.55**	*5.21*	8.86	**7.76**
RIMES NN-HMM (*)	SVM	*5.03*	10.23	7.67	*3.82*	8.49	**6.92**
	ANN	*5.18*	9.88	7.90	*3.73*	8.42	**6.75**
49% d'erreur en transcription	AdaBoost	*4.85*	**9.93**	**9.27**	*5.21*	12.21	**9.67**
IAM NN-HMM	SVM	*25.41*	36.65	30.21	*18.00*	37.23	**26.71**
	ANN	*24.95*	36.00	31.38	*20.40*	39.44	**28.53**
50% d'erreur en transcription	AdaBoost	*29.30*	**39.18**	**41.98**	*29.50*	**41.00**	**39.05**
A2iA-ArSf NN-HMM	SVM	*32.74*	40.07	**38.07**	*31.04*	42.07	**37.11**
	ANN	*33.19*	41.33	**38.44**	*32.30*	43.33	**38.22**
59% d'erreur en transcription	AdaBoost	*33.63*	**40.52**	**40.07**	*34.07*	44.67	**39.33**

(*) Ces reconnaisseurs ont été optimisés sur une partie de la base RIMES. Nous avons dû réduire le nombre de données consacrées à l'évaluation de la classification sur RIMES (4251 au lieu de 5599), afin d'évaluer sur une base de données *disjointe* de celle utilisée pour entraîner les reconnaisseurs.

ou de la pondération TF[5]. Il semble que les poids associés aux mots soient trop bruitées pour un algorithme glouton de sélection de variables, en particulier la pondération TF qui a tendance à dégrader les performances d'AdaBoost.

4. Conclusion

Dans cet article, nous avons présenté et étudié différentes représentations d'images de documents pour la classification. Nous avons mené des expériences avec deux types de reconnaisseurs, trois familles de classifieurs, et trois bases de données. Cette étude confirme l'intérêt de prendre en compte les différentes alternatives de reconnaissance avec leurs scores associés, notamment pour la classification en thèmes de courriers

5. Notons que le facteur IDF, indépendant pour chaque mot, n'a aucune influence pour AdaBoost.

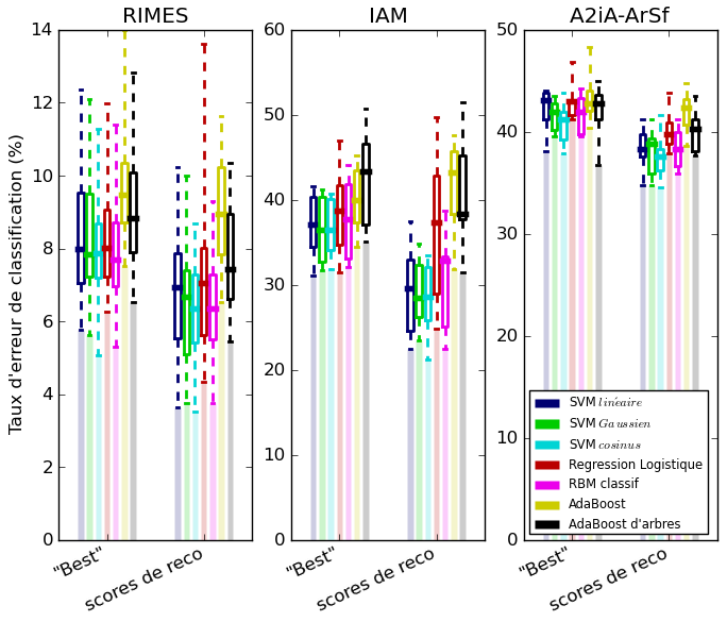

Figure 5. *Comparaison des performances entre les sacs de mots standard ("Best" : équations [1] et [3]) et les sacs de mots prenant en compte plusieurs alternatives de reconnaissance avec leurs scores ("scores de reco" : équations [2] et [4]). Les boîtes à moustaches représentent la médiane, les $25^{ème}$ et $75^{ème}$ centiles ainsi que les minima et maxima des performances mesurées par fold. Les résultats sur la base RIMES correspondent aux résultats cumulés sur trois reconnaisseurs (cf. tableau 1).*

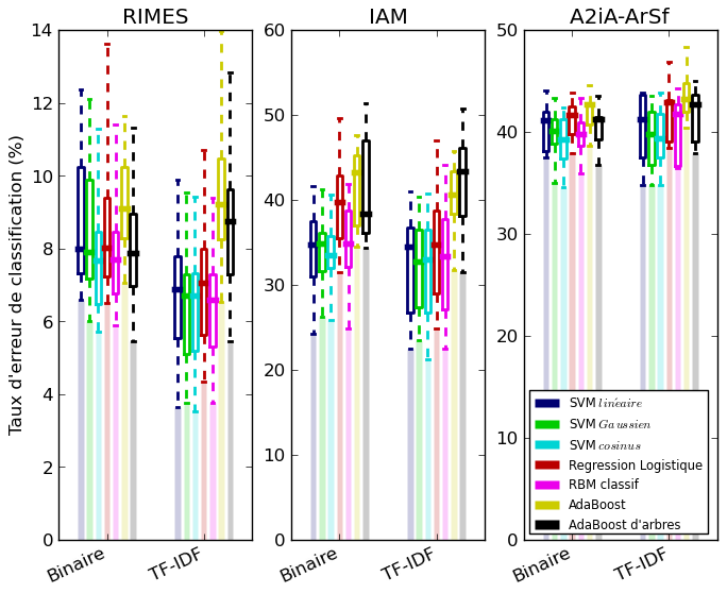

Figure 6. *Comparaison des performances entre les sacs de mots ne tenant pas compte des fréquences des mots ("Binaire" : équations [1] et [2]) et les sacs de mots basés sur une pondération TF-IDF ("scores de reco" : équations [3] et [4]).*

manuscrits. En particulier, les techniques de classification automatique par SVM et réseaux de neurones bénéficient d'une représentation enrichie par rapport à la représentation *sac de mots* standard qui prend en compte uniquement les meilleures hypothèses de mots reconnus. Le gain est moins marqué pour un algorithme de sélection de variables comme AdaBoost. Nous avons aussi discuté de l'intérêt mitigé de la représentation TF-IDF.

Cette étude doit être continuée en essayant d'améliorer la qualité de la transcription automatique, notamment en appliquant des modèles de langage. Aussi, nous prévoyons d'analyser en profondeur les performances du reconnaisseur afin de pouvoir regrouper automatiquement les mots fréquemment confondus dans la représentation *sac de mots*, toujours dans le but de rendre la classification plus robuste aux erreurs de transcription.

5. Bibliographie

Augustin E., Brodin J.-m., Carré M., Geoffrois E., Grosicki E., Prêteux F., « RIMES evaluation campaign for handwritten mail processing », *Proc. of the Workshop on Frontiers in Handwriting Recognition*, number 1, 2006.

Chang C.-C., Lin C.-J., « LIBSVM : A library for support vector machines », *ACM Transactions on Intelligent Systems and Technology*, vol. 2, p. 27 :1-27 :27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Cortes C., Vapnik V., « Support-vector networks », *Machine Learning*, vol. 20, n° 3, p. 273-297, September, 1995.

Graves A., Liwicki M., Fernández S., Bertolami R., Bunke H., Schmidhuber J., « A novel connectionist system for unconstrained handwriting recognition. », *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, n° 5, p. 855-68, May, 2009.

Johansson S., Leech G., Goodluck H., Manual of Information to accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital Computers, Technical report, Department of English, University of Oslo, Norway, 1978.

Karlgren J., Cutting D., « Recognizing text genres with simple metrics using discriminant analysis », *Proceedings of the 15th conference on Computational linguistics* -, 1994.

Kermorvant C., Louradour J., « Handwritten mail classification experiments with the Rimes database », *Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*, 2010.

Knerr S., Augustin E., « HMM Based Word Recognition and its Application to Legal Amount Reading on French Checks », *Computer Vision and Image Understanding*, 1998.

Larochelle H., Bengio Y., « Classification using discriminative restricted Boltzmann machines », *Proc of the 25th international conference on Machine learning - ICML '08*, ACM Press, New York, New York, USA, p. 536-543, 2008.

LeCun Y., Bottou L., Bengio Y., Haffner P., « Gradient-Based Learning Applied to Document Recognition », *Proceedings of the IEEE*, vol. 86, n° 11, p. 2278-2324, 1998.

Marti U.-V., Bunke H., « The IAM-database : an English sentence database for offline handwriting recognition », *International Journal on Document Analysis and Recognition*, vol. 5, n° 1, p. 39-46, November, 2002.

Peña Saldarriaga S., Morin E., Viard-Gaudin C., « Using top n Recognition Candidates to Categorize On-line Handwritten Documents », *International Conference on Document Analysis and Recognition*, number 3, Ieee, p. 881-885, 2009.

Peña Saldarriaga S., Viard-Gaudin C., Morin E., « Impact of online handwriting recognition performance on text categorization », *International Journal on Document Analysis and Recognition*, vol. 13, n° 2, p. 159-171, 2010.

Saldarriaga S., Morin E., Viard-Gaudin C., « Categorization of On-Line Handwritten Documents », *Proc of the Int. Workshop on Document Analysis Systems*, IEEE Computer Society, Los Alamitos, CA, USA, p. 95-102, 2008.

Salton G., Buckley C., « Term-weighting approaches in automatic text retrieval », *Information Processing & Management*, vol. 24, n° 5, p. 513-523, 1988.

Schapire R. E., Singer Y., « BoosTexter : A Boosting-based System for Text Categorization », *Finance*, vol. 39, p. 135-168, 2000.

Toselli A. H., Juan A., Vidal E., « Spontaneous Handwriting Recognition and Classification », *Proc. of the Int. Conf. on Pattern Recognition*, p. 433-436, 2004.

Tukey J. W., *Exploratory Data Analysis*, Addison-Wesley, 1977.

Vinciarelli A., « Noisy text categorization. », *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, n° 12, p. 1882-95, December, 2005.

ANNEXE POUR LE SERVICE FABRICATION
A FOURNIR PAR LES AUTEURS AVEC UN EXEMPLAIRE PAPIER
DE LEUR ARTICLE ET LE COPYRIGHT SIGNE PAR COURRIER
LE FICHIER PDF CORRESPONDANT SERA ENVOYE PAR E-MAIL

1. ARTICLE POUR LES ACTES :

   Colloque International Francophone sur l'Ecrit et le Document

2. AUTEURS :

   Jérôme Louradour — Théodore Bluche — Anne-Laure Bianne Bernard — Fares Menasri — Christopher Kermorvant

3. TITRE DE L'ARTICLE :

   De l'usage des scores et des alternatives de reconnaissance pour la classification d'images de documents manuscrits.

4. TITRE ABRÉGÉ POUR LE HAUT DE PAGE MOINS DE 40 SIGNES :

   De l'usage des scores et des alternatives

5. DATE DE CETTE VERSION :

   12 décembre 2011

6. COORDONNÉES DES AUTEURS :

   – adresse postale :
   A2iA SA, 40 bis rue Fabert, 75007 Paris
   {jl,tb,alb,fm,ck}@a2ia.com
   – téléphone : 00 00 00 00 00
   – télécopie : 00 00 00 00 00
   – e-mail : Roger.Rousseau@unice.fr

7. LOGICIEL UTILISÉ POUR LA PRÉPARATION DE CET ARTICLE :

   LaTeX, avec le fichier de style `article-hermes.cls`,
   version 1.2 du 03/03/2005.

8. FORMULAIRE DE COPYRIGHT :

   Retourner le formulaire de copyright signé par les auteurs, téléchargé sur :
   `http://www.revuesonline.com`

SERVICE ÉDITORIAL – HERMES-LAVOISIER
14 rue de Provigny, F-94236 Cachan cedex
Tél : 01-47-40-67-67
E-mail : revues@lavoisier.fr
Serveur web : http://www.revuesonline.com