RESEARCH REPORT

On using alternative recognition candidates and scores for handwritten documents classification

a2ia-RR-2008-1

Christopher Kermorvant
Anne-Laure Bianne Bernard
Théodore Bluche
Jérôme Louradour

On using alternative recognition candidates and scores for handwritten documents classification

Christopher Kermorvant
Anne-Laure Bianne Bernard
Théodore Bluche
Jérôme Louradour

Abstract

This paper compares 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.

1 Introduction

Mailroom automation is today one of the main applications of handwritten documents recognition, together with historical documents indexing. On these kinds of free form documents, the error rate of automatic systems are still high, compared to the rates achieved on bank checks or envelops. This is due to the unconstrained nature of the handwritten texts: the content is very variable and there is no simple way to validate the recognition results, contrary to what can be done for example for postal address recognition with a postal databases. However, even with relatively high recognition error rates, automatic transcription allows to partially automate the classification of the huge amount of paper documents coming into large companies or organizations. In automated mailroom applications, handwritten documents are automatically classified on the basis of their content, reducing processing costs and delays.

2 Document image classification

The first step of a content-based document image classification is the automatic recognition of the sequence of words written on the image. Then a representation of this sequence is built: standard representation of the text such as *bag of words* vectors can be used to encode the output of the recognizer [22, 18]. Finally, state of the art machine learning techniques can be applied to the bag of words vector to classify the document according to a pre-defined classification scheme.

2.1 Bag of words representation for document image classification

A *bag of words* is a sparse vector whose size depends on the number of different words in the training set used for the classification task. Each dimension of the vector corresponds to a different word and the corresponding value is 0 if the word is not present in the document represented by the vector. If the word is present, the value can be 1 in case of a binary bag of words or a positive value related to the frequency of the word in the document. The name *bag of words* was given to this representation because it does not take into account the word order. Even if this aspect could appear as a limitation, in practice:

- classification rules (automated or manual) are based on the presence of absence of keywords without sequential order
- the high error rate of automated handwriting recognition limits the probability of having long sequence of correct words
- several studies have shown that using the word order information - for example with a bigram model- tends to degrade the classification rate [21].

Two approaches are possible to deal with keywords detection:

- either to detect them explicitly as proposed in [15] by using a recognizer with a small lexicon of keywords and a simple character model for all the other words;
- or to perform a recognition with a large generic lexicon and to rely on the classifier to implicitly select the right keywords.

Even if good results have been reported in [15] using the former method, we have chosen to use the latter since it does not require to define the keywords *a priori*.

2.2 Bag of words of recognizer's output

The standard approach for handwritten document classification consists in building a bag of words from the best recognition hypothesis, discarding all

the hypothesis considered less likely by the recognizer. From a classification point of view, error in the recognized words can be considered as noise on the bag of words representation. Several studies have shown the correlation between the level of noise, directly linked to the error rate of the recognizer, and the document classification rate [17, 7].

If we suppose that there are only a few important keywords needed to correctly classify the document, there are two kinds of recognition error with an impact on the classification rate: fail to recognize a key word and falsely recognize a keyword. If the number of keywords is very small compared to the number of possible words, the number of false detection is negligible compared to the number of missed keywords. In order to increase the number of keywords correctly detected, we show in this paper that we can take into account alternative recognition hypothesis when their associated confidence score is high enough. This approach was already proposed by [16] for the classification of document produced by a on-line handwriting recognizer.

If adding alternative recognized words should allows to detect more keywords, it will also introduce some noise in the representation with falsely detected keywords. This problem can be tackled by using the recognition confidence score given by the recognizer to each recognition hypothesis. Since the techniques used by the state of the art handwriting recognizers (hidden Markov models or recurrent neural networks) consist in optimizing the *a posteriori* probability of the words, the confidence score is proportional to the probability of presence of the word in the document. Moreover, if the confidence score are calibrated and normalized (between 0 and 1 and summing to 1), the confidence score can be directly used in the bag of words representation.

A pruning procedure can also be used to discard recognition results which are too unlikely or by considering only the n-best recognition results [16]. However, this pruning is not necessary as soon as the confidence score are present in the bag of words representation.

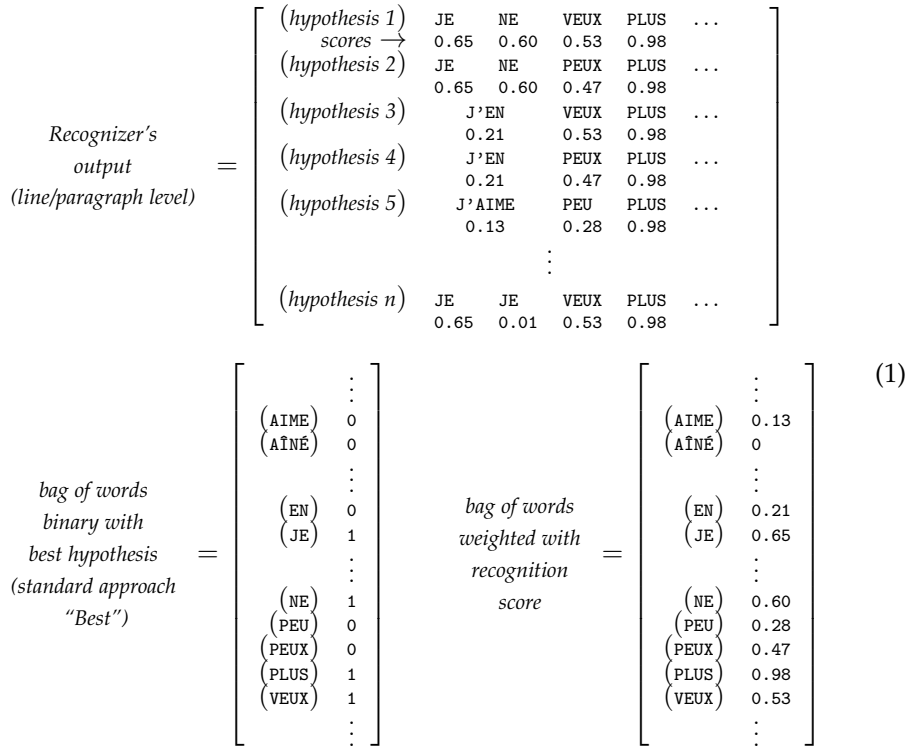
Figure 1 shows the usage of recognition results for a bag of words representation of document. In our case, the alternative recognition results are given at line level.

Let $\{w_k\}_{k=1\ldots K}$ be the set (of size K) of all the different words in the classification training set, $\hat{w}_i^t$ the word recognized at position t in the i^{th} recognition hypothesis of length T_i (the best line hypothesis can be denoted $\{\hat{w}_1^t\}_{t=1\ldots T_1}$). The corresponding binary bag of words is a sparse vector of size K such as:

$$x_k^{\text{bin}} = \begin{cases} 1 & \text{si} \quad w_k \in \{\hat{w}_1^t\}_{t=1\ldots T_1} \\ 0 & \text{sinon} \end{cases} = \max_{t=1\ldots T_1} \delta\left(\hat{w}_1^t, w_k\right), \tag{2}$$

where δ is the Kronecker symbol. If we denote by s_i^t the score associated to the word hypothesis $\hat{w}_i^t$, we propose to use the following bag of words, weighted by recognition scores:

$$x_k^{\text{scores}} = \max_i \max_{t=1\ldots T_i} s_i^t \times \delta\left(\hat{w}_i^t, w_k\right) \tag{3}$$



(1)

Figure 1: How to use recognition results in a bag of words representation for document classification.

Another representation is now mainly used for text classification: the TF-IDF [19] representation that can be expressed as:

$$x_k^{\text{TF-IDF}} = \sum_{t=1}^{T_1} \delta\left(\hat{w}_1^t, w_k\right) \times \log_{10} \frac{N}{\text{DF}_k} \tag{4}$$

where N is the number of documents in the classification training database, and DF_k is the number of documents of this database containing at least one occurrence of the word w_k in the best recognition hypothesis $\left\{\hat{w}_1^t\right\}_{t=1\ldots T_1}$. TF is the frequency of the word in the document and IDF is related to the inverse of the frequency of document containing this word. [16] proposed a generalization of TF-IDF taking into account recognition scores as:

$$x_k^{\text{scores/TF-IDF}} = \frac{\text{TF}_k^{\text{scores}} \times \log_{10} \frac{N}{\text{DF}_k}}{\sqrt{\sum_{k'=1}^{K} \text{TF}_{k'}^{\text{scores}} \times \log_{10} \frac{N}{\text{DF}_{k'}}}} \tag{5}$$

where $\text{TF}_k^{\text{scores}} = \max_i \sum_{t=1}^{T_i} s_i^t \times \delta\left(\hat{w}_i^t, w_k\right)$ is a generalization of TF.

3 Experiments

We describe in the section the experimental protocol used to evaluate the different representations of the handwriting recognizer's output using bag of words for document image classification.

3.1 Classification databases

We have made experiments on three different databases composed exclusively of handwritten documents.

The RIMES database

The RIMES database [1] was developed for the evaluation of automated systems for handwritten document processing. It is composed of 5599 handwritten pages, with human made ground-truth transcription and classification. The classification scheme (with the corresponding number of samples) is:

> *Change in contract* (1350), *Information request* (1038), *Claims handling* (611), *Address change* (599), *Follow-up* (596), *Account closing* (463), *Claims* (327), *Financial embarrassment* (312), *Account opening* (301), *Unknown* (2).

This database was collected with the help of more than 1300 voluntary writers how wrote each letters according to a predefined scenario but with their own words. This database can be considered as realistic but not real as the documents were too carefully produced. When the recognizer has been train on a subset of this database, not all the documents can be used for the classification experiments. In this case, only the 4251 documents not used for the training of the recognizer are considered.

The IAM database

The IAM database V3.0 [13] contains 1539 handwritten pages corresponding to handwritten copies of texts from the LOB corpus [5], made by 657 different writers. No classification task was defined on the database but as the texts were extracted from the LOB corpus, they correspond to different genres: press, novels, religious texts, etc. From the 90 original different kinds of text, [6] proposed to define 15 or 4 classes. We propose here a classification scheme in 7 different classes, presented hereafter with the number of samples in each class:

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".`

These seven classes are used for our experiments on the IAM database. This database is not composed of realistic documents but the classification task is difficult since the classes are defined both in terms of content and writing style.

The A2iA-ArSf database

This database is composed of real handwritten documents from paper mails send by clients to a large company using an automated mailroom system. The documents classification was done manually and a ground-truth transcription is also available. This database is composed of 1649 documents representing the following 14 classes (with sample numbers):

> *Termination* (689), *Complex changes* (322), *Promotional offers* (187), *Simple changes* (109), *Simple actions* (96), *Claims* (93), *Complex actions* (82), *Legal* (52), *Client account action* (8), *Account management* (5), *Mobiles phones* (3), *CTI* (1), *Transactions* (1), *Form* (1).

Since this database if composed of real documents, they show a large variety of style and writing quality that make the classification task particularly difficult.

3.2 Recognizers

We describe in this section the two kinds of isolated word recognizer used for the automatic transcription and our strategy for the recognition of a complete line using isolated word recognizers. More details on the recognition process can be found in [14].

Grapheme-based Hybrid HMM recognizer

We have used a word recognizer based on a hybrid neural network and hidden Markov model (HMM) using a segmentation of the words into graphemes [8]. During the decoding phase, after the grapheme extraction, a feature vector describing each grapheme is computed (see Figure 2). The neural network

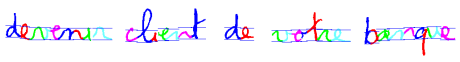


Figure 2: Decomposition of a text line into grapheme.

Figure 3: Grapheme cutting probability graph for a line (the weights on the arc are inverse log-probabilities). Arcs in bold correspond to the correct segmentation.

(a multi-layer perceptron) is used to compute the posterior probability of the each grapheme classes that are used by the HMM to describe the decomposition of a letter. The lexical constraints are modelled by a set of HMM corresponding to each word in the lexicon.

Recurrent neural network recognizer (RNN)

This recognizer is based on a Multi-Dimensional Long-Short Term Memory recurrent neural network (MDLSTM) [4]. This neural network is able to model long-term dependencies in sequences. The model is trained directly on the pixels sequence extracted from the image along four different scans: from the top, from the bottom, from the left, from the right. The model does not need to extract features on the image since the features are learned. The MDLSTM-RNN was trained with the RNNLib library [1], with a custom decoding to take into account a lexical constraint.

Word sequence recognition

Our approach for the recognition of a sequence of words in a paragraph is based on a segmentation into lines, followed by an explicit segmentation of the line into words. This approach relies on the supposition that the correct word segmentation can be found by exploring only a limited number of word segmentation hypotheses. This supposition is realistic in the case of documents with a good graphic quality and without noise, but can be false in case of dense writing or for Arabic handwriting for example. This approach is faster than the standard approach that consist in using a sliding window on the complete line. Once the segmentation hypotheses are built, the word recognizer is applied at each word position to build the recognition lattice.

The complete line recognition process is the following:

- compute the grapheme segmentation on the complete line (see Figure 2).

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

Figure 4: Word segmentation hypothesis.

- compute for each pair of consecutive graphemes, the probability that they belong to two different words (see Figure 3). We compute this probability with a neural network.
- build the segmentation graph from the n best paths in the grapheme cutting probability graph (see Figure 4).
- recognize each word position in the segmentation graph with the isolated word recognizer.
- build the recognition lattice by keeping only the N_m best recognition hypothesis at each word position.
- compose the recognition lattice with the language model if any.
- extract the N_l best path in the composed graph.

This procedure was used for the recognition of all the pages used in the document classification experiment.

3.3 Document classifiers

In this section, we describe the three kinds of classification algorithms that we have evaluated for the classification of document images based on the content extracted by automatic handwriting recognition.

In the following, the classification rates are always evaluated using 5-fold cross validation, 60% of the data being used for training the classifier, 20% for parameter validation and the remaining 20% for testing. In all the experiments, the lexicon size was limited to the 10,000 most frequent words of the training set (for each cross-validation fold).

Adaptive Boosting (AdaBoost)

AdaBoost is a classification algorithm based on a linear weighted combination of many simple classifiers called *weak learners*. In our experiments, we have used small depth (depth 1 or 3) decision trees as weak learners. For textual features, each node in the tree corresponds to a word in the lexicon and the following question is asked:

- is the word present in the feature vector in case of binary bag of words ?
- is the score associated to this word greater than a given threshold in case of feature vector using recognition score or TF-IDF ?

For a given document, the decision tree (weak learner) gives a weighted prediction for all the classes according to the final node reached. The words, the associated thresholds and the weights of the weak learners are learned with the *Real AdaBoost MH* algorithm [20].

Regarding the learning of the threshold associated to each words, [11] gave an efficient procedure to learn them.

Support Vector Machines (SVM)

The SVM algorithm selects from the training set the examples that define the classification boundaries with the largest margin. Is uses a similarity measure between examples, called the kernel, that must chosen carefully [3].

In our experiments, we have tested the three types of kernel:

1. a linear kernel, which shows good generalization properties in high dimension (our case)
2. a Gaussian kernel, which is a standard kernel
3. a cosine kernel, which is often used in text classification

The training was used using the `libSVM` [2] library. The parameters of the algorithms, C the regularization parameter and γ the width parameter for the Gaussian kernel need to be optimized. We have searched the optimal values on a validation set:

- in $\{1, 10, 100\}$ for C
- in $\{1/d_{10\%}, 1/d_{50\%}, 1/d_{90\%}\}$ for γ , where $d_{50\%}$ (resp. $d_{10\%}$) is the median value (resp. the 10^{th} centile) of distances between 1000 pairs of samples randomly chosen[2].

In the case of feature vectors using TF-IDF, we have used the same data normalization as in section 3.3.

Artificial Neural Networks (ANN)

Instead of choosing variables like boosting or training samples like SVM, neural networks learn how to weight the features and combine them using non linear functions. For classification, state of the art techniques use a *softmax* normalization that maps each output of the networks to the posterior probabilities of one class. In our experiment, the training was done using a stochastic gradient descent [10] on negative log-likelihood. The learning rate and early stopping are optimized on a validation set.

For neural networks, the data normalization is a key point to avoid the saturation of the network's activation. This saturation occurs when the input

[2] see `http://blog.smola.org/post/940859888/easy-kernel-width-choice`

data can vary in a large range of values, which is the case with TF-IDF coefficients. In our experiments, we have normalized our data such as they have a zero mean and unity variance. This normalization was done globally (not independently for each word) so that the IDF weighting is kept.

We have tested two kinds of neural networks: logistic regression (which is a neural network without hidden layer) and Restricted Boltzmann Machine for classification [9]. In preliminary experiments, RBM have proven to be as good and sometime better than the standard multi-layer percepton.

4 Results

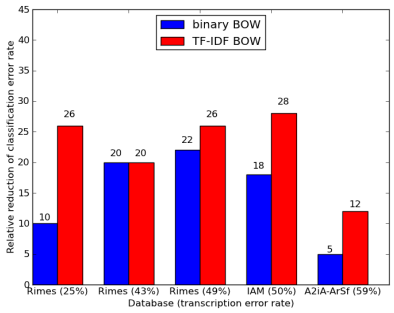


Figure 5: Relative reduction of classification error rate (in %) when using alternative recognition results compared to using only the best recognition result both with binary and TF-IDF bag of words (BOW).

Table 1 shows the results of all the experiments we have carried out on the three databases, with the two recognizers and with the three kinds of classifiers. The transcription error rates on each database for each recognizer are given. The classification experiments were also conducted with the human transcription: theses results can be seen as a reference upper bound for the results obtained with an automatic transcription. For each family of text classifier (Adaboost, SVM and Neural networks), we show the mean classification error rate for all the variants in this family, for example, all the types of kernel in the SVM family. These results are analysed in details in the following sections.

Impact of using recognition alternatives

We first study the use of recognition alternatives for document classification. Figure 5 shows the relative reduction of classification error rate when recognition alternatives are used compared to using only the best recognition results.

Database & Recognizer	**Classification method**	**Bag of words**					
		Binary			TF-IDF		
		Human Trancript.	Recognizer Best	Recognizer 10-Best	*Human Transcript.*	Recognizer Best	Recognizer 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% transcription error rate	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% transcription error rate	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% transcription error rate	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% transcription error rate	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% transcription error rate	AdaBoost	*33.63*	**40.52**	**40.07**	*34.07*	44.67	**39.33**

Table 1: Classification errror rate (%) for different kinds of bag of words, databases, recognizers and classification method. The "10-Best" column is our method with recognition scores and 10 best alternatives. The transcription error rate is given for each database and recognizer. Classification error rate on the human transcription is also given as a reference upper bound. The classification error rates showed in bold are the best or the rates that are not significantly different from the best one according to a two-sided paired Student test at 95%.

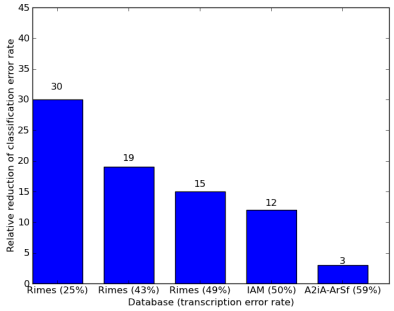


Figure 6: Relative reduction of classification error rate (in %) when using a TF-IDF weighting instead of a binary bag of words (with alternatives).

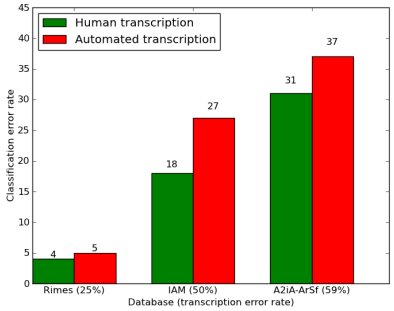


Figure 7: Classification error rates on human transcription and automatic transcription for the three databases, with the best recognizer and the best classifier (SVM).

The classification error rate reduction was computed for the best classification techniques show in Table 1. The gain with the alternative is clear: both with binary and TF-IDF bag of words, on all database using the alternative recognition results reduces the classification error rate, from 5% to 28% depending in the database.

Impact of using TF-IDF bag of words

The use of TF-IDF weighting is now standard in electronic text classification. However, we did not observed in the past a significant gain when using TF-IDF for real handwritten documents classification: the Figure 6 presents a possible explanation. This figure shows the relative reduction of classification error rate when using the TF-IDF weighting instead of a binary bag of words.

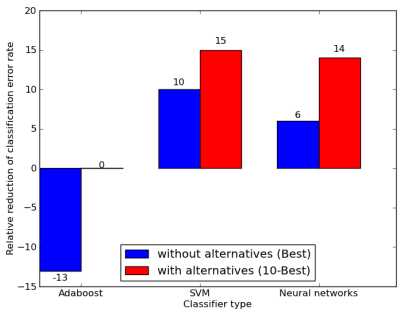


Figure 8: Classification error rate relative reduction (in %) when using the recognition score with (10-Best) or without (best) recognition alternatives for the three families of classifier.

The classification error rate reduction was computed for the best classification techniques showed in Table 1. We see that even if there is always a reduction of error rate when using TF-IDF, this reduction is larger in case of a low transcription error rate. For the real database A2iA-ArSf, the reduction is only of 3% whereas it reaches 30% on the Rimes database. Since we usually work on real databases with high transcription error rate, we did not observed the possible gain with TF-IDF in the past.

Impact of recognition error on classification.

In this section, we study the impact of the recognition errors on a task of handwritten documents classification. Since the error rates of the automatic transcriptions are still high on handwritten documents, compared to printed documents for example, one may think that these documents can not be automatically processed. We know form our experience that this is not the case. Figure 7 shows a comparison of the classification error rates on the three databases using either a human or an automatic transcription. We can see that the presence of recognition errors does not prevent automatic classification, even at high error rate. We know from experience that the main reason for the a failure in automatic classification of document images is not the quality of the recognition engine but the complexity of the classification task.

Choice of the text classifier

We have compared three kinds of state of the art statistical text classifiers: Boosting, SVM and neural networks. Boosting is a good candidate for industrial applications because of the relative simplicity of its training procedure: contrary to the other methods, it has no parameter whose value needs to be optimized on a validation set (excepted the number of weak learners

that we usually optimize to avoid over-training). Moreover, the Boosting's training procedure can be programmed efficiently simply with algorithmic optimization [11], without the burden of parallel or multi-thread programming. However, as shown on Figure 8, Boosting does not take advantage of the recognition scores: on average, using the recognition scores with only the best recognition hypothesis degrades the classification results (13% increasing of classification error rates), whereas the classification error rate is decreased with both SVM and neural networks. Using both the recognition scores and the alternatives has no effect on the classification rates, whereas the gain is significant for both SVM and neural networks (around 15% of relative error rate reduction). This could be due to the fact that recognition scores are not reliable enough for a greedy algorithm such as boosting.

5 Conclusion

In this article, we have tested four variants of bag of words representation for handwritten documents image classification: using or not the recognition scores and using or not the recognition alternatives. We used in our experiments a modified version of the standard TF-IDF coefficients adapted to automatic recognition results. We have evaluated the different kinds of bag of words on three different databases with three families of classification algorithms.

We have shown that a bag of words representation based on recognition scores and alternatives improve the document classification rates when used with SVM or neural networks. Boosting algorithm seems not to take advantage of this enriched representation. We also have shown that the expected gain using a TF-IDF bag of words is higher when the transcription error rate is low. Finally, our experiments show that handwritten document images classification can be done automatically despite a relatively high transcription error rate.

In the future, we will explore the use of more advanced natural language processing techniques. Theses techniques are very sensitive to the recognition errors as it has been shown by Lopresti [12] for printed text. Techniques such as lemmatization or semantic analysis would be useful for handwritten document classification but they need to be adapted to be robust to the errors produced by the recognizer.

References

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

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

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sept. 1995.

[4] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–68, May 2009.

[5] S. Johansson, G. Leech, and H. Goodluck. 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.

[6] J. Karlgren and D. Cutting. Recognizing text genres with simple metrics using discriminant analysis. In *International Conference on Computational Linguistics*, volume 2, 1994.

[7] C. Kermorvant and J. Louradour. Handwritten mail classification experiments with the Rimes database. In *International Conference on Frontiers in Handwriting Recognition*, 2010.

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

[9] H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In *International Conference on Machine Learning*, pages 536–543, 2008.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] P. Li. Robust LogitBoost and Adaptive Base Class (ABC) LogitBoost. In *Conference on Uncertainty in Artificial Intelligence*, number 2, 2010.

[12] D. Lopresti. Optical character recognition errors and their effects on natural language processing. *International Journal of Document Analysis and Recognition*, 12(3):141–151, Sept. 2008.

[13] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, Nov. 2002.

[14] F. Menasri, J. Louradour, A.-l. Bianne-bernard, and C. Kermorvant. The A2iA French handwriting recognition system at the Rimes-ICDAR2011 competition. In *Document Recognition and Retrieval Conference*, volume 8297, 2012.

[15] T. Paquet, L. Heutte, and G. Koch. A Categorization System for Handwritten Documents. *International Journal of Document Analysis and Recognition*, 2011.

[16] S. Peña Saldarriaga, E. Morin, and C. Viard-Gaudin. Using top n Recognition Candidates to Categorize On-line Handwritten Documents. In *International Conference on Document Analysis and Recognition*, number 3, pages 881–885, 2009.

[17] S. Peña Saldarriaga, C. Viard-Gaudin, and E. Morin. Impact of online handwriting recognition performance on text categorization. *International Journal on Document Analysis and Recognition*, 13(2):159–171, 2010.

[18] S. Saldarriaga, E. Morin, C. Viard-Gaudin, and L. U. M. R. Cnrs. Categorization of On-Line Handwritten Documents. In *International Workshop on Document Analysis Systems*, pages 95–102, 2008.

[19] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[20] R. E. Schapire and Y. Singer. BoosTexter : A Boosting-based System for Text Categorization. *Machine Learning*, pages 135–168, 2000.

[21] A. H. Toselli, A. Juan, and E. Vidal. Spontaneous Handwriting Recognition and Classification. In *International Conference on Pattern Recognition*, pages 433–436, 2004.

[22] A. Vinciarelli. Noisy text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1882–95, Dec. 2005.