

# Handwritten mail classification experiments with the Rimes database

Christopher Kermorvant and Jérôme Louradour

A2iA SA

40 bis, rue Fabert

75007 Paris France

{ck,jl}@a2ia.com

## Abstract

*In this paper, we consider the task of automatic handwritten mail classification and we investigate the relation between the transcription rate and the classification rate. Several configurations of a multi-word handwriting recognizer using different language models are tested and their word recognition rates on the documents to be classified are reported. For the document classification task, we have investigated three different classifiers (KNN, SVM, AdaBoost). All the experiments were conducted on the public database Rimes.*

## 1. Introduction

Automatic handwriting recognition has found industrial applications since the beginning of the 90's: at this moment, the recognizers's performances were sufficient to be used in applications such as postal address recognition [5] or bank check recognition [7]. However, the use of automatic handwriting recognition engines is still limited in other document image processing applications. This could be due to the fact that the recognition error rate on free-format handwritten documents seems to be very high compared to postal address or bank check recognition, for which the information to be recognized is very constrained: postal addresses are validated with a postal database and bank check are validated with other financial information. Conversely, on free-format document the information is *per se* unconstrained.

Today, one of the main usage of unconstrained handwriting recognition is in mailroom applications. Large organizations with a large number of private customers such as telephone companies or administrations daily receive from their clients a lot of paper documents that they must process rapidly. Very often, they also have a quality level commitment and they must process a mail

within 2 or 3 days after its reception. To cope with the varying amount of received mails, they have no choice but to automate the processing of the incoming mail.

Very few papers report results on both handwriting recognition and document classification. In [16], the authors reports handwriting recognition results on short paragraphs and classification results of these paragraphs into a small number of predefined classes. The database is not publicly available and the classification model is not standard, which make the results difficult to analyze. In [12], the authors studied the impact of on-line handwriting recognition performance on text categorization. They compared two standard classifiers (SVM and k-NN) on an handwritten version of a subset of the Reuters-21578 corpus. In [3], a theoretical model of the relation between the transcription error level and the classification rate is given.

The Rimes database [2] was collected in order to evaluate systems dedicated to the recognition and classification of handwritten mails sent by individuals to companies or administrations. The database is composed of handwritten mails written by 1300 volunteers according to predefined scenario but using their own words. Several competitions have been organized to compare the performance on handwriting recognition systems on isolated words [8] but no result yet has been published on multi-word transcription and document classification.

In this paper, we have tested 12 configurations of handwriting recognizers and 3 classifiers for automatic handwritten document classification on the Rimes database. Our goal was to explore the impact of the transcription errors on a document classification task. We chose the configurations of the recognition system not to minimize the transcription error rate but to be representative of different levels of transcription error rate. Since we have the human transcription of the documents, we can directly measure the degradation of the classifier performance with respect to the recognizer

transcription rate.

This article is organized in four parts : description of the handwriting recognizers, description of the classifiers, experiments on the Rimes database and conclusion.

## 2. Handwritten word recognition

### 2.1 Isolated word recognition

The isolated word recognizer is based on an explicit segmentation of the word in parts called graphemes. The segmentation is an over-segmentation which means that a grapheme is either a character or subpart of a character. For each detected grapheme, a set of 74 relatively simple features is computed (see [11] for details).

*Hidden Markov models* (HMM) are used to model the decomposition of the words into letters and each letter into graphemes. The topology of the model is a simple left-to-right Bakis model topology with three states for each letter HMM, each state corresponding to a grapheme. The model is a hybrid neural network HMM, where a multi-layer perceptron is trained to evaluate the posterior probability of each grapheme with respect to the feature vector.

### 2.2 Multi-word recognition

The first step of the multi-word line recognizer is the segmentation of the line into hypothesis of isolated words. The segmentation is given by a neural network which predicts for each grapheme if it belongs to the same word or to two different words. The result of this segmentation is a segmentation graph which encodes the different segmentation options and their probability. The isolated word recognizer is applied on each object of the segmentation graph, and the list of all recognized word hypothesis with their probability is stored in a word recognition transducer. This list can be pruned to reduce the decoding time.

### 2.3 Language modeling (LM)

We have included a Language Model (LM) in the multi-word recognizer. This LM defines the vocabulary of the isolated word recognizer and allows a re-scoring of the word recognition transducer with word sequence statistics collected on a text corpus. This LM is a smoothed n-gram encoded as a weighted finite state transducer [1]. The smoothing method used for our LM is the one proposed by Katz [10]. The re-scoring the word recognition hypothesis with the LM is obtained by composing the word recognition transducer and the

LM transducer. The best word sequence is obtained by applying a best-path algorithm on the composed transducer.

## 3. Text classification

For the task of mail categorization, the text recognition process is followed by the classification of recognized word sequences. We experimented three different types of classifiers: *k-Nearest-Neighbors* (KNN), *Support Vector Machines* (SVM) and *Adaptative Boosting* (AdaBoost). This section reviews these classifiers in their basic versions. The Machine Learning community has proposed numerous sophisticated variants to train these classifiers, but these improvements are out of the scope of this paper. Our goal here is to study the effect of text recognition accuracy on commonly available classification methods.

### 3.1 Bag-of-words representation (BOW)

For our classification purpose, all mentioned classifiers are applied on *Bag-of-Words* (BOW) representations of input mails. In other words, they do not take into account the sequential order of recognized words, and are invariant to word permutations within mails.

In this paper, we consider only *binary* BOW: each word encountered in the training set corresponds to a numerical feature that is 1 if the word is present in the mail, and 0 otherwise. Other representations are possible, for instance the *Term Frequency-Inverse Document Frequency* (TF-IDF) weighting [14]. But preliminary experiments showed that using TF does not improve, and TF-IDF even tends to degrade classification performance. Even if IDF is relevant to reduce the influence of common terms in text mining, it undervalues words that are important for a some classification tasks.

As a matter of fact, the input dimensionality is high when using BOW; It equals the vocabulary size of the training dataset. So facing the *curse of dimensionality* is a big challenge for the classifier, whose goal is to generalize well from BOW training dataset to unseen BOW test data.

### 3.2 k-nearest-neighbors (KNN)

KNN is a popular pattern classification algorithm as it leads to competitive results despite its simplicity. This instance-based lazy learner memorizes all training data. Most of the computational effort is done when classifying a test sample. This procedure consists in computing determining the *k* most similar training samples, and returning a confidence-rated prediction based on the

similarity of these  $k$  nearest neighbors. The parameter  $k$  controls the model complexity and can be chosen by random validation. The higher  $k$ , the smoother is the model.

The choice of the KNN similarity function is critical to obtain good classification results. The cosine between BOW is the state-of-the-art method for text categorization [13].

### 3.3 Support Vector Machine (SVM)

SVM [4] is also an appealing classifier since some methods have been proposed to tackle somehow large-scale problems [6]. Similarly to KNN, SVM is based upon a similarity measure, the kernel, whose choice is critical. The most common kernels are:

**lin-SVM** The dot product (linear),

**cos-SVM** The cosine, suitable for text categorization,

**rbf-SVM** The Gaussian kernel (radial basis function, exponentially decreasing w.r.t the squared distance times a parameter  $\gamma$ ).

Given a classification problem with two classes, SVM are trained to produce a confidence-rated prediction that can be expressed as a linear combination of kernel evaluations between the BOW test sample and a subset of the entire BOW training dataset (the *support vectors*). The objective function is a linear combination of the hinge loss and the soft margin loss, parameterized by a factor  $C$  to monitor model regularization. A high value of  $C$  will produce complex classifiers that can discriminate between very complex patterns, but that may also over-fit the training dataset and generalize poorly.  $C$  (as well as  $\gamma$  for rbf-SVM) can be chosen by random validation.

The generalization of SVM to classification problems with more than two classes can be done in several ways. Empirical studies have shown that one-vs-one scheme is suitable for practical use [9].

### 3.4 Adaptive Boosting (AdaBoost)

The AdaBoost algorithm used in our experiments is the Real AdaBoost MH algorithm described in [15]. The boosted weak learners are word selectors which classify samples based on the presence or not of a given word within an input BOW. For a given BOW training dataset, the AdaBoost algorithm yields weights to train the weak learners (one weight per sample and per class), so as to have the greatest improvement on the empirical objective function with each new weak learner. The optimal number of weak learners to keep after AdaBoost learning can be chosen by random validation.

## 3.5 Three different types of classifiers

The above-mentioned classifiers correspond to several designs of the classification decision function, and can be divided in three groups.

lin-SVM is among the smoothest possible models (linear discrimination between class pairs). cos-SVM is equivalent to lin-SVM after a spherical normalization all BOW inputs using *Term Frequencies*.

AdaBoost as described above acts like a feature (word) selector, such as decision trees. By this way, AdaBoost can easily discard words that are not relevant for the classification task. These useless words can constitute a cumbersome noise for KNN and SVM, which handle BOW through a rigid similarity function.

KNN and rbf-SVM are template-matching based methods. To work well, they need a good coverage of the input space by the training samples, and are the most severely affected by the *curse of dimensionality*.

## 4 Experiments

### 4.1 Databases

We describe in this section the data used to train and test each part of the complete document classification system.

#### 4.1.1 Word recognition

We have trained the same isolated word recognizer on two different databases. In both case, the recognizer is case insensitive and the accents are removed.

**IWR1** : this recognizer was trained on the same database used for the ICDAR 2009 handwriting competition [8]. This database is composed of 44,197 isolated words.

**IWR2** : this recognizer was trained on a database of isolated words extracted from envelopes and bank checks. This database is composed of 75,000 isolated words.

#### 4.1.2 Language modeling

We have tested three different corpora to train the language models (LM) used by the multi-word recognizer.

**Rimes** This corpus is the human transcription of the 1,151 handwritten letters of the Rimes database used for training the isolated word recognizer. It is independent from the test database used for evaluating the transcription and the classification. The total number of words in this corpus is 76,334.

**TelCo** This corpus is composed of the automatic transcription of printed documents collected from a client database. The client is a telephone company (TelCo) and the documents are scanned images of the correspondence between the company and its clients. We collected 35 000 documents mainly corresponding to printed letters. These documents were transcribed using an OCR. The resulting corpus is composed of 3,600,000 words. Even if this corpus contains many recognition errors, it can be used to train language models.

**LeMonde** This corpus is composed of 100,000 articles published in the french newspaper Le Monde between 1994 and 2002. This corpus is composed of 40,000,000 words.

For the three corpora, we have removed the punctuation and normalized all the words to upper case before training the LM. The lexicon used for the LM and for the multi-word recognizer was limited to the 10 000 most frequent words (in the case of Rimes, the total number of different words is below 10000). For each corpus, the size of the selected vocabulary and its out-of-vocabulary rate estimated on the classification database is presented on the following table:

Training corpus	Vocabulary size	OOV rate
LeMonde	10 000	3.45%
TelCo	10 000	4.60%
Rimes	5 879	4.81%

### 4.1.3 Classification

The documents in the Rimes database are classified into nine different classes, unevenly distributed, corresponding to the content of the mail. The following table gives the class description (Class name) with their absolute count (Count) and percent of total (%):

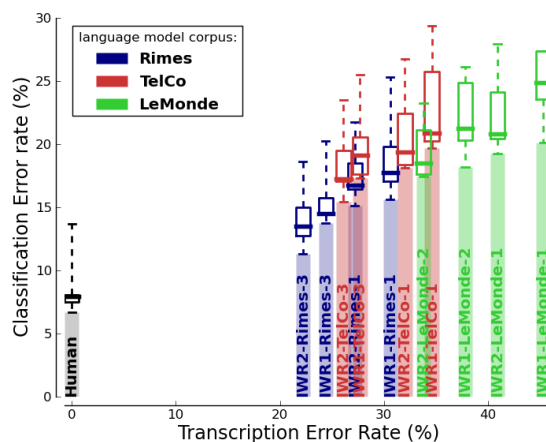
Class name	Count	%
account opening	108	5.42
financial difficulties	111	5.57
complaint	125	6.27
account closure	168	8.43
claim	189	9.48
reminder	196	9.83
change of personal information	219	11.0
request for information	374	18.8
change contract/order	503	25.2
Total	1993	100

All the classifiers were trained and tested on transcriptions of a subset of the Rimes database composed of 1993 handwritten mails in french, that were not used to train the word recognizer nor the language model. Only

the body text of the mails were used, in order to eliminate the errors due to the document layout analysis. This subset is splitted randomly to produce two subsets to train and validate the classifier (1196 and 397 mails) and another separate subset on which to evaluate the classifier (400 mails). To have a reliable evaluation, we repeated this random sub-sampling 10 times (in exactly the same way for all our experiments) and we present the average of these cross-validation results.

The parameters of each classification learning procedure were chosen so as to minimize the classification error rate on the validation dataset (379 mails at each validation iteration). We remind that these parameters are:  $k$  for the KNN,  $C$  for the SVM (besides  $\gamma$  for rbf-SVM) and the number of weak learner for AdaBoost. At each step of cross-validation, we restricted each KNN and SVM parameters search to a maximum of 20 trials, and the number of boosted weak learners to a maximum of 300.

## 4.2 Results



**Figure 2. Box plot of classification rates with respect to transcription rates**

Table 1 presents transcription and classification error rates for several systems that are combinations of a word recognizer, a language models and a text classifier. The different configurations were selected in order to cover a wide range of transcription error rate.

The first result of the transcription experiments is that the corpus on which the language model is trained has a huge impact on the recognizer performance: the error rate increases by 50% (from 22.2% to 33.8%) when using (LeMonde) a corpus large but far from the application domain, compared to (Rimes) a corpus very small but exactly on the domain. Using (TelCo) a small

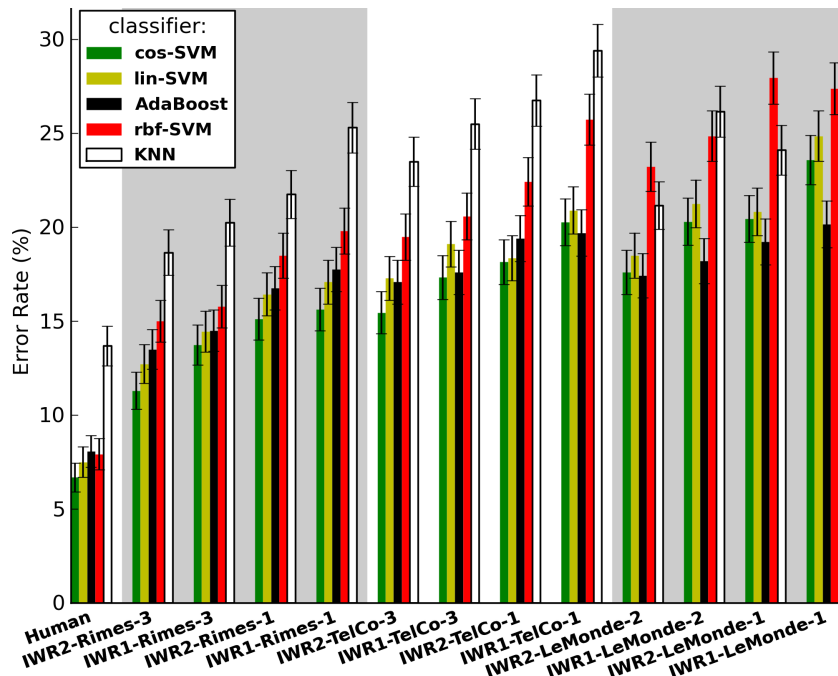


Figure 1. Classification results with respect to text recognizers and classifiers

corpus collected on similar documents, even with transcription errors, yields good performance, with only a 17.5% increase of error rate.

The second result of the transcription experiments is that, contrarily to language models, isolated word recognizer tend to perform better when it is trained on a large dataset even if it is not typical of the task of interest (IWR2), rather than on a smaller dataset that matches evaluation conditions (IWR1).

Fig.1 shows all classification results for the different text transcriptions, and Fig.2 shows classification rates averages with respect to the transcription error rate. 95% confidence intervals for all classification error rates are around +/-1% (they are shown in Fig.1).

For a given a dataset to train the LM, as expected, classification error rates increase as the transcription error rate increases. But in general, the classification error rate is not monotonous with respect to the transcription error rate: when using a LM trained on TelCo, classification results are worse than the ones obtained with a LM trained on Rimes with the same transcription accuracy (Fig.2).

Regarding the different classifiers, lin-SVM and cos-SVM outperform AdaBoost at a low transcription error level but AdaBoost is more robust to transcription errors. In our experiments, it outperforms other classifiers for transcription error rates higher than 33%. KNN and rbf-SVM always lead to the poorest performance:

they need more training data to generalize well given the high input dimensionality.

## 5. Conclusion and perspective

In this paper, we have presented a set of experiments on the Rimes database for an application to automatic handwritten mail classification. We have tested several combinations of isolated word recognizers, language models and classifiers.

We have found that even if the transcription accuracy affects classification results, the classification rate cannot be predicted using only the transcription rate. We are interested in finding alternative measures on the transcription to predict the classification performance.

Regarding the classifiers, the cosine SVM outperforms all the other classifiers at low transcription error level, whereas Adaboost is more robust to transcription errors. We also observed that classification on the human transcription still produce a certain amount of errors (around 7%). Bag-of-words are perhaps too limited to discriminate between mail classes.

We have also shown that the training corpus for the language model is a key point in this kind of application: a very small corpus very close to the application domain is better than a large corpus far from the application domain. We have shown that when there is no electronic corpus available to train the language model,

**Table 1. Text recognition and classification error rates**

Word Reco.	Language model		Transcription Error Rate (%)	Classification Error Rate (%)				
	Corpus	Order		cos-SVM	lin-SVM	AdaBoost	rbf-SVM	KNN
<i>Human transcription</i>			—	<b>6.7</b>	7.5	8.1	7.9	13.7
<b>IWR2</b>	<b>Rimes</b>	<b>3-gram</b>	22.2	<b>11.3</b>	12.7	13.5	15.0	18.6
<b>IWR1</b>	<b>Rimes</b>	<b>3-gram</b>	24.4	<b>13.7</b>	14.4	14.5	15.7	20.2
<b>IWR2</b>	<b>Rimes</b>	<b>1-gram</b>	27.2	<b>15.1</b>	16.4	16.7	18.5	21.7
<b>IWR1</b>	<b>Rimes</b>	<b>1-gram</b>	30.6	<b>15.6</b>	17.1	17.7	19.8	25.3
<b>IWR2</b>	<b>TelCo</b>	<b>3-gram</b>	26.1	<b>15.4</b>	17.3	17.1	19.5	23.5
<b>IWR1</b>	<b>TelCo</b>	<b>3-gram</b>	27.7	<b>17.3</b>	19.1	17.6	20.6	25.5
<b>IWR2</b>	<b>TelCo</b>	<b>1-gram</b>	32.0	<b>18.1</b>	18.3	19.4	22.4	26.7
<b>IWR1</b>	<b>TelCo</b>	<b>1-gram</b>	34.6	20.3	20.9	<b>19.7</b>	25.7	29.4
<b>IWR2</b>	<b>LeMonde</b>	<b>2-gram</b>	33.8	17.6	18.5	<b>17.4</b>	23.2	21.1
<b>IWR1</b>	<b>LeMonde</b>	<b>2-gram</b>	37.8	20.3	21.2	<b>18.2</b>	24.8	26.1
<b>IWR2</b>	<b>LeMonde</b>	<b>1-gram</b>	40.9	20.4	20.8	<b>19.2</b>	27.9	24.1
<b>IWR1</b>	<b>LeMonde</b>	<b>1-gram</b>	45.3	23.7	24.8	<b>20.1</b>	27.4	27.4

one can achieve reasonable performance by training it on a corpus of printed transcriptions. We are interested in further exploring the use of automatic transcriptions, from printed or handwritten documents, to train language models.

Finally, there is also room for improvement regarding the classification rate on the human transcription, for example by going beyond the bag-of-words representation.

## References

- [1] C. Allauzen and M. Mohri. Generalized optimization algorithm for speech recognition transducers. In *Proc. of the Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 352–355, 2003.
- [2] E. Augustin, M. Carré, E. Grosicki, J.-M. Brodin, E. Geoffrois, and F. Preteux. Rimes evaluation campaign for handwritten mail processing. In *Proc. of the Workshop on Frontiers in Handwriting Recognition*, pages 231–235, 2006.
- [3] J. Brodin. Ocr-classification interaction mathematical model. In *Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*, 2008.
- [4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
- [5] M. El-Yacoubi, J.-M. Bertille, and M. Gilloux. Conjoined location and recognition of street names within a postal address delivery line. In *Proc. of the Int. Conf. on Document Analysis and Recognition*, page 1024, 1995.
- [6] R. Fan, P. Chen, and C.-J. Lin. Working set selection using second order information for training svm. *Journal of Machine Learning Research*, 6:1889–1918, 2005.
- [7] N. Gorski, V. Anisimov, E. Augustin, O. Baret, and S. Maximov. Industrial bank check processing: the a2ia checkreader. *International Journal on Document Analysis and Recognition*, pages 196–206, 2001.
- [8] E. Grosicki and H. ElAbed. Icdar 2009 handwriting recognition competition. In *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2009.
- [9] C. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Trans. on Neural Networks*, 13(2):415–425, 2002.
- [10] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoustics, Speech, and Signal Processing*, Assp-35(3):400–401, 1987.
- [11] C. Kermorvant, F. Menasri, A.-L. Bianne, and L. Likforman-Sulem. N-best combinaison of isolated word recognizers with neural networks. In *Proc. of the Int. Conf. on Handwriting Recognition*, volume submitted, 2010.
- [12] 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*, 2009.
- [13] S. Saldarriaga, E. Morin, and C. Viard-Gaudin. Categorization of on-line handwritten documents. In *Proc of the Int. Workshop on Document Analysis Systems*, pages 95–102, Los Alamitos, CA, USA, 2008. IEEE Computer Society.
- [14] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & management*, 24(5):513–523, 1988.
- [15] R. Schapire and Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168, 2000.
- [16] A. Toselli, A. Juan, and E. Vidal. Spontaneous handwriting recognition and classification. In *Proc. of the Int. Conf. on Pattern Recognition*, pages 433–436, 2004.