On-line Handwritten Text Categorization

Sebastián Peña Saldarriaga[a], Christian Viard-Gaudin[b], and Emmanuel Morin[a]
[a]LINA UMR CNRS 6241, Université de Nantes, France;
[b]IRCCyn UMR CNRS 6597, Université de Nantes, France

ABSTRACT

As new innovative devices, accepting or producing on-line documents, emerge, managing facilities for these kinds of documents such as topic spotting are required. This means that we should be able to perform text categorization of on-line documents. The textual data available in on-line documents can be extracted through on-line recognition, a process which produces noise, i.e. errors, in the resulting text. This work reports experiments on categorization of on-line handwritten documents based on their textual contents. We analyze the effect of the word recognition rate on the categorization performances, by comparing the performances of a categorization system over the texts obtained through on-line handwriting recognition and the same texts available as ground truth. Two categorization algorithms (kNN and SVM) are compared in this work. A subset of the Reuters-21578 corpus consisting of more than 2000 handwritten documents has been collected for this study. Results show that accuracy loss is not significant, and precision loss is only significant for recall values of 60%-80% depending on the noise levels.

Keywords: Text categorization, noisy text, on-line handwriting recognition

1. INTRODUCTION

The categorization of electronic, ascii, documents is a well known research area and has been thoroughly studied.[1] But as new innovative devices which accept or produce on-line handwritten data emerge, this research area has to be extended to on-line documents in order to provide managing facilities such as automatic document organization into predefined groups, document routing or filtering and retrieval of documents with respect to a given topic. This means that we should be able to solve the problem of document categorization from an input format being on-line handwriting. Since the output of an on-line handwritten recognition engine contains errors, which will induce errors in subsequent processing stages of categorization application, a careful work has to be done to extract as precisely as possible the relevant terms which will guide the categorization process.

Due to the inherent difficulty of collecting large handwritten data sets, little research has been done on categorization of handwritten texts. Most of the research on noisy text categorization has so far been conducted with texts produced by Optical Character Recognition (OCR) systems,[2–4] which are quite efficient as long as the quality of the original document is not degraded.

On-line documents may have a rich internal structure that can include text, graphics, equations, tables and other non-textual elements. Categorization can be performed using non-textual features[5] but textual features are still a critical point that needs to be explored. To our knowledge only two works address the problem of categorization of handwritten documents.[6,7] Both of them dispose of off-line documents and of an off-line recognition system. Vinciarelli[6] used a subset of 200 documents from the Reuters-21578 corpus manually written by a single writer; the categorization models were trained over clean digital texts. Koch's experiments[7] are performed on a corpus of French business letters by spotting words of a reliable set of relevant terms.

A corpus of more than 2000 on-line documents has been collected for this study. In this paper we study various strategies enabling to produce from an on-line handwritten document a complete or a partial transcribed version of it, so that the text categorization (TC) engine described in section 2 can be used. To achieve this goal, we will use as a core module, the on-line handwriting recognition engine presented in section 3. It allows us to obtain different levels of accuracy by varying the linguistic knowledge attached to it.

Further author information: (Send correspondence to S.P.S.)
S.P.S.: E-mail: sebastian.pena-saldarriaga@univ-nantes.fr
C.V-G.: E-mail: christian.viard-gaudin@univ-nantes.fr
E.M.: E-mail: emmanuel.morin@univ-nantes.fr

2. TEXT CATEGORIZATION

The TC engine used in this work is based on machine learning approaches and the VSM (Vector Space Model) representation.[8] The TC process is composed of several steps which are described in the following subsections.

2.1 Pre-processing

The aim of pre-processing first step is to split character streams into a sequence of words. At the end of this step, *stopwords* are removed: articles, prepositions and such words, which are assumed to carry no information. The next step is stemming, or suffix stripping.[9] By stemming, the words *indexed*, *indexing* and *indexation* will have a single representative form: *index*. Stemming also reduces the number of distinct terms needed to represent a set of documents.

2.2 Indexing

After pre-processing, we need to convert the resulting sequence of *terms* into vectors. Each vector component accounts for a term of the *feature space*. The feature space is composed of a given number of relevant terms extracted from the training set using Forman's round robin algorithm[10] over category specific scores obtained with the chi-squarestatistic.[11] Each vector component w_i is weighted by a statistical measure such as the normalized $tf \times idf$ score.[12]

$$w_i = \frac{f_i \times \log \frac{N}{n_i}}{\sqrt{\sum_{j=1}^{M} \left[f_j \times \log \frac{N}{n_j} \right]}} \tag{1}$$

Where f_i is the frequency of term i in a document, N the number of documents in the collection, M the number of distinct terms in the collection and n_i the frequency of i within the whole collection of documents.

Vectors can now be used within categorization algorithms. Two state-of-the-art categorization methods are used in our experiments : a k-nearest neighbour (kNN) algorithm and Support Vector Machines (SVM).

2.3 Categorization Methods

The two categorization methods used in this work have been chosen because their implementation is rather easy and they seem to be two of the dominant TC approaches developed in the recent years.[13–15] Both methods are shortly described below.

2.3.1 k-Nearest Neighbours (kNN)

The kNN algorithm is a lazy learning approach. In order to decide if a document d belongs to a category candidate c, the algorithm ranks the training document vectors according to their distance to d, and a score $p(c, d)$ is calculated using the similarity scores of the k-nearest neighbours.[16] If the score is high enough, a positive decision is taken, a negative decision is taken otherwise.

$$p(c, d) = \frac{\sum_{i=1}^{k} sim(d, x_i) \times I(x_i, c)}{\sum_{i=1}^{k} sim(d, x_i)} \tag{2}$$

Where

$$I(x_i, c) = \begin{cases} 1 & if\ category(x_i) = c \\ 0 & otherwise \end{cases} \tag{3}$$

And the similarity function $sim(d, x_i)$ is the cosine of the angle between the two vectors.

2.3.2 Support Vector Machines (SVM)

Support Vector Machines[17] have proven to be an efficient learning method for text categorization.[14] SVMs perform categorization by projecting the original data in a high dimension space where documents of a category ⬠ can be linearly separated from others by a hyperplane.

An optimal hyperplane (see figure 1) minimizes the empirical classification error and maximizes the geometric margin between positive training documents (⬠) and the negative ones (⬭) . The experiments presented in this paper were performed using the SVMlight V6.0 package by Joachims*.

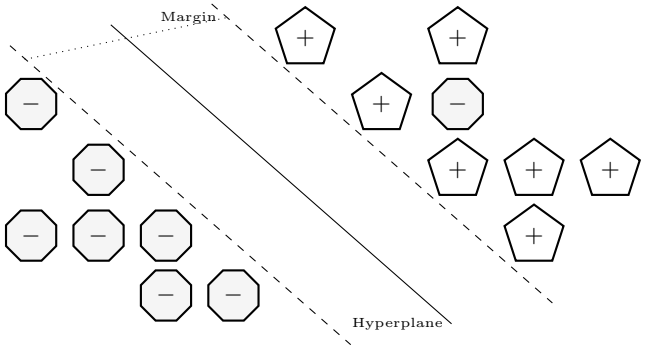

Figure 1. Optimal hyperplane

2.4 Evaluation

The most commonly used effectiveness measures for text categorization are recall (ρ) and precision (π). These measures may be estimated from the contingency table for a category on a given set of documents (see Table 1).

Table 1. Contingency table for category c

Category c		Reality	
		YES	**NO**
Predictions	**YES**	TP	FP
	NO	FN	TN

TP is the number of documents correctly classified under c (true positives), TN the number of correctly rejected from c (true negatives), FP the number of incorrectly classified (false positives) and FN the number of incorrectly rejected (false negatives).

Recall is the proportion of correctly classified documents within all the documents of a given category (c).

$$\rho(c) = \frac{TP}{TP + FN} \tag{4}$$

Precision is the proportion of correctly classified documents within all the documents classified under a given category (c).

$$\pi(c) = \frac{TP}{TP + FP} \tag{5}$$

*This package can be found at http://svmlight.joachims.org

In order to evaluate categorization over several categories, the above category-specific measures must be averaged. There are two averaging methods: macro- and micro-averaging. Macro-averaging gives equal weight to each category; scores for each category are computed separately before being averaged. The macro-averaged measures are computed as follows.

$$\rho^M = \frac{\sum_{c=1}^{|C|} \rho(c)}{|C|} \qquad \pi^M = \frac{\sum_{c=1}^{|C|} \pi(c)}{|C|} \tag{6}$$

On the other hand, micro-averaging gives equal weight to every document, thus micro-averaged scores are heavily affected by the performances on frequent categories. Contingency tables for individual categories are added as shown in equation 7.

$$\rho^\mu = \frac{\sum_{c=1}^{|C|} TP_c}{\sum_{c=1}^{|C|} TP_c + FN_c} \qquad \pi^\mu = \frac{\sum_{c=1}^{|C|} TP_c}{\sum_{c=1}^{|C|} TP_c + FP_c} \tag{7}$$

In our application context, documents are available at different moments in time; hence system evaluation should be done on a document basis. Moreover, as we work with single label documents; micro-averaged recall and precision are equal. Hence, a single accuracy measure will be given as system evaluation.

In order to give a comprehensive description of the system performance when handwritten documents are used, interpolated precision versus recall curves[18] are also presented for the 11 standard recall values $[0, 0.1, 0.2, \ldots, 1.0]$.

3. HANDWRITING RECOGNITION

Handwriting recognition is the process of automatically converting a handwriting signal into a character stream usable within text-processing applications. Handwriting recognition can be performed on documents of two different types : off-line and on-line documents. The former are raw bitmap images that contain no other information than the image itself. The latter are produced with digital pens or PDA stylus and consist in a sequence of points, corresponding to the device trajectory, which is sampled during the writing process.

In this work, as we dispose of a set of on-line documents, an on-line recognition engine is used. The on-line recognition process is shown in figure 3. The recognition engine of MyScript Builder[†] is used in our experiments.

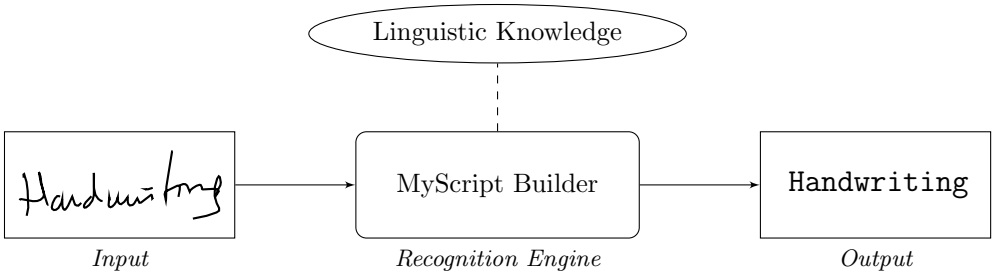

Figure 2. On-line handwriting recognition

Recognition systems achieve poor recognition performances without linguistic resources. Linguistic resources give *prior* knowledge to the recognizer about what it is likely to recognize. Hence, we can have a good (or bad) influence on the recognition process by varying the linguistic knowledge attached to the recognizer.

MyScript Builder offers the possibility to create our own resources and also contains two convenient standard built-in resources:

[†]MyScript Builder SDK can be found at http://www.visionobjects.com/products/software-development-kits/myscript-builder

- **lk-text** is composed of a standard lexicon of English words and a statistical language model. This model allows the recognition of a word according to words around it, for example, *'this is'* will have priority over *'this in'*. **lk-text** has also the capability to recognize a variety of specific out-of-lexicon language elements such as dates, numbers, postcodes, etc.

- **lk-free** has an extended out-of-lexicon recognition capability that helps the recognition of an uppercase letter when combined with other uppercase letters, a lowercase letter with other lowercase characters and a digit with other digits. When the recognizer is unable to differentiate between an *'I'* and a *'1'*, if the other characters in the word are recognized as uppercase letters then *'I'* will have priority over *'1'*.

3.1 Noise Measures

Recognizers are not perfect, and recognized documents often contain *noise*, i.e. errors induced by the recognition process. The Word Error Rate (WER) is a common performance measure of a handwriting recognition system. One drawback with this measure, however, is that no account is taken of the effect that different types of errors may have on the document representation and end applications.

In our categorization context, the recognition of inflected forms of a word into another form of the same word (e.g. *dollar* as *dollars*) will not affect the categorization because stemming will correct this. The same goes for confusion of stopwords with other stopwords (e.g. *is* as *in*) since either way they will be removed.

The Term Error Rate (TER) [5] provides a better estimation of the noise in the categorization context. TER is the percentage of incorrectly extracted terms with respect to the original text and is defined as follows:

$$TER = 1 - \frac{\sum_i^N \min\left(tf(i), tf'(i)\right)}{\sum_k^N tf(k)} \tag{8}$$

Where $tf(i)$ and $tf'(i)$ are the frequencies of the term i in the clean and recognized text respectively, and N the total number of terms. But all these terms are not relevant regarding our categorization task, the loss of relevant terms is more important than the loss of other terms. A measure that estimates the proportion of relevant terms incorrectly extracted is thus necessary: the Relevant Term Error Rate (RTER). The RTER is defined as follows:

$$RTER = 1 - \frac{\sum_i^M \min\left(rtf(i), rtf'(i)\right)}{\sum_k^M rtf(k)} \tag{9}$$

Where $rtf(i)$ and $rtf'(i)$ are the frequencies of the relevant term i in the clean and recognized text respectively, and M the total number of relevant terms. In this paper we report the noise level of recognized documents in terms of WER, TER and RTER for our handwritten corpus.

4. EXPERIMENTS AND RESULTS

This section presents the experiments conducted in this work. First, the data used in our experiments is described in subsection 4.1. In subsection 4.2 results of recognition performed on this data are reported. The remainder of this section reports categorization results. Categorization has been performed on both recognized and clean versions of the documents and the performance difference measured from a statistical point of view.

4.1 Datasets

The handwritten documents used in this study are a subset of the Reuters-21578 newswire collection.[19] The Reuters-21578 is a well-known benchmark collection widely used in TC research.[15] The collection is composed of economic and business newswires appeared in 1987. The newswires are labeled into 135 categories. Out of these 135 categories only 90 are represented in both, training and test sets and the 10 most represented categories account for about 90% of the collection.

The newswires cover different matters such as corporate acquisitions, agricultural (coffee, sugar, etc.) or energy (crude, gasoline, etc.) commodities and interest rates for instance.

The database has been split into train and test sets following the ModApt split[20] and the documents not belonging to the top ten categories were discarded. 2000 documents were randomly chosen from the train set and 500 from the test set. These documents were used to collect handwritten data. Currently we dispose of 2029 documents out of these 2500; they have been collected over a period of four months from more than 1500 writers. The handwritten newswires were collected using a digital pen and Anoto paper. Figure 4.1 shows a sample from our on-line data set.

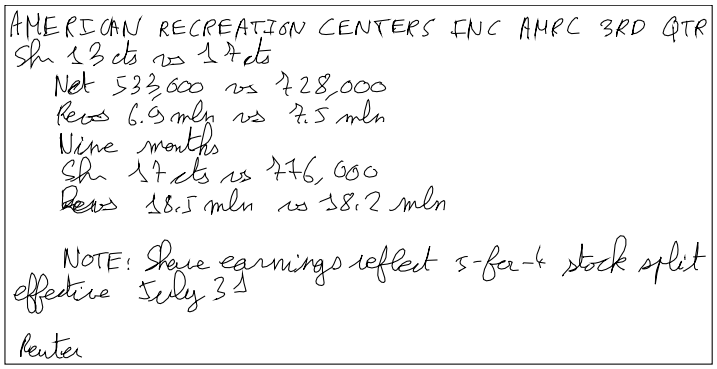

Figure 3. On-line handwritten text sample

Table 2 summarizes the number of documents per category for the training and the test sets. The categorization experiments reported in this paper are based on these documents and their electronic counterparts.

Table 2. Documents per category

Category	Train	Test
Earnings	642	108
Acquisitions	349	72
Grain	125	51
Foreign Exchange	177	49
Crude	80	40
Interest	76	30
Trade	59	21
Shipping	54	13
Sugar	32	10
Coffee	30	10
Total	1625	404

4.2 Recognition

A recognition engine is used to perform transcription of the handwritten documents. For each handwritten document the ground truth is also available, but only at the global text level. Hence, we compute WER and TER, by aligning the recognized word sequence with the ground truth using a string-to-string edit distance. Table 3 and table 4 show the performance of the recognition engine according to the linguistic knowledge resource used.

Table 3. Noise levels for the training set

Resource	WER	TER
lk-free	52.47%	55.75%
lk-text	22.30%	23.01%

Table 4. Noise levels for the test set

Resource	WER	TER	RTER
lk-free	52.48%	55.85%	51.85%
lk-text	22.08%	21.90%	19.35%
lk-text$^+$	20.62%	18.61%	14.52%

Unsurprisingly lk-text clearly outperforms lk-free. The latter recognizes better short words. Hence, a substantial loss of information after stopword deletion and stemming is observed. Since lk-text works on a word basis, recognition of terms and words is substantially better. Moreover, for lk-text WER and TER are low considering that this resource contains no prior linguistic knowledge specific to this kind of documents, which yet feature a lot of acronyms, and out of lexicon terms.

When comparing table 3 and table 4, it can be seen that the training and the test sets behave very similarly with respect to WER and TER, which is consistent with the random selection of these two subsets. Of course, RTER is only applicable for the test set, since the relevant terms are defined using the chi-square statistic, among the recognized terms of the training set.

Concerning the test set, in addition to the two generic resources already mentioned, namely lk-free and lk-text, we have also introduced an additional resource, lk-text$^+$, which includes a dedicated lexicon competing with the general lexicon of lk-text. This lexicon is composed of all the words of the lk-text recognized training set corresponding to a relevant term. Our goal is to increase the likeness of test documents and lk-text training documents, thus increasing the chances of correct matching at categorization time. By using this new resource, the RTER is significantly decreased: nearly 5%, which corresponds to a relative improvement of 25%.

4.3 Categorization

The categorization results obtained on the test set using either the ground truth texts (termed as clean) or the recognized texts for training are reported below. It is worth to note that the *textual* training sets which are used for extracting the relevant terms are actually the output of the recognized system using various resources, except first line of table 5 where ground truth is used as well for training and test data sets.

The parameters of the classifiers have been tuned to achieve maximum accuracy. This has been done on the *clean* training set. 90% of the set is used for training of the categorization models, then categorization is performed on the remaining 10% set with several parameters until maximum accuracy is obtained. The optimal parameters for the kNN algorithm are 15 neighbours and 300 relevant terms, while 1000 relevant terms are optimal for the SVM classifier.

Table 5. Categorization accuracy for the test data sets

Dataset	kNN	SVM
clean	90.10%	93.56%
lk-text$^+$	89.36%	91.47%
lk-text	89.36%	91.83%
lk-free	79.46%	84.65%

The results presented in table 5 are obtained by sorting the category confidence scores per test document and assigning the top-ranking category to documents. When compared with the clean version of the texts, an accuracy loss of about 10% is observed using the lk-free resource regardless of the categorization method. This is explained by the poor capability of this resource to extract the relevant terms; it has to be related with the 51.85% of RTER. The recognition results obtained with lk-free are not suitable for efficient TC. On the other hand, lk-text performs well with both methods. Performance loss is about 2% for SVM and 1% for kNN.

Unfortunately, there's no clear contribution of lk-text$^+$ to categorization performance despite the boosting in the relevant term extraction performance. This means that RTER is not sufficient to assess the performances of the categorization system. We have to keep in mind that the relevant terms are selected based on the results of recognition on the training set: we could be very efficient to recognize these terms on the test set, but due to recognition errors on the training set the relevant term selection would be not so good.

Precision vs recall curves are obtained by ranking the document confidence scores per category and following the algorithm described by Baeza-Yates.[18] Figure 4 shows that micro-averaged performances of SVM and KNN with clean texts are close for recall levels up to 60%, above SVM is noticeably better. lk-text$^+$ performances confirm that the RTER is not a good indicator of noise. lk-text obtains acceptable results with both methods for recall values up to 50%.

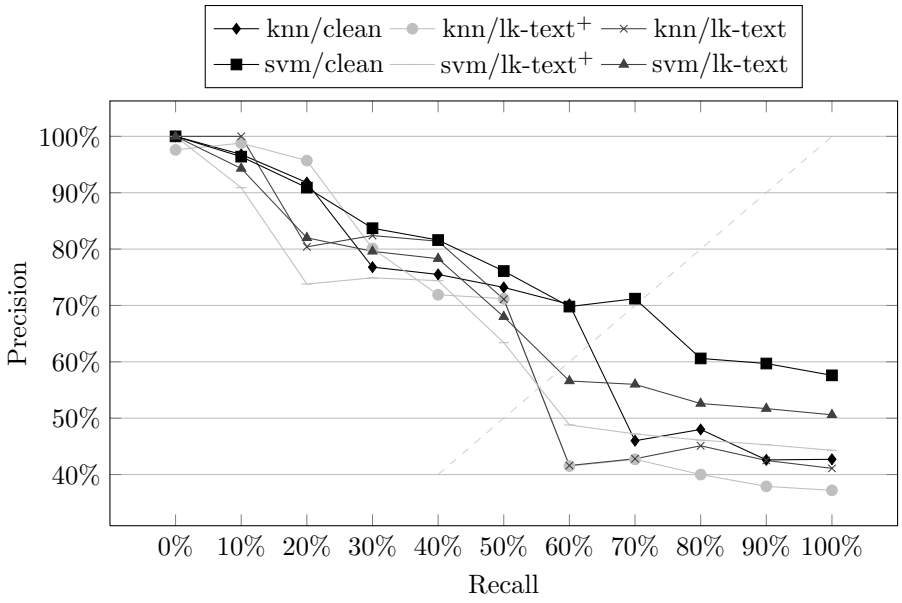

Figure 4. Micro-averaged precision vs recall curve

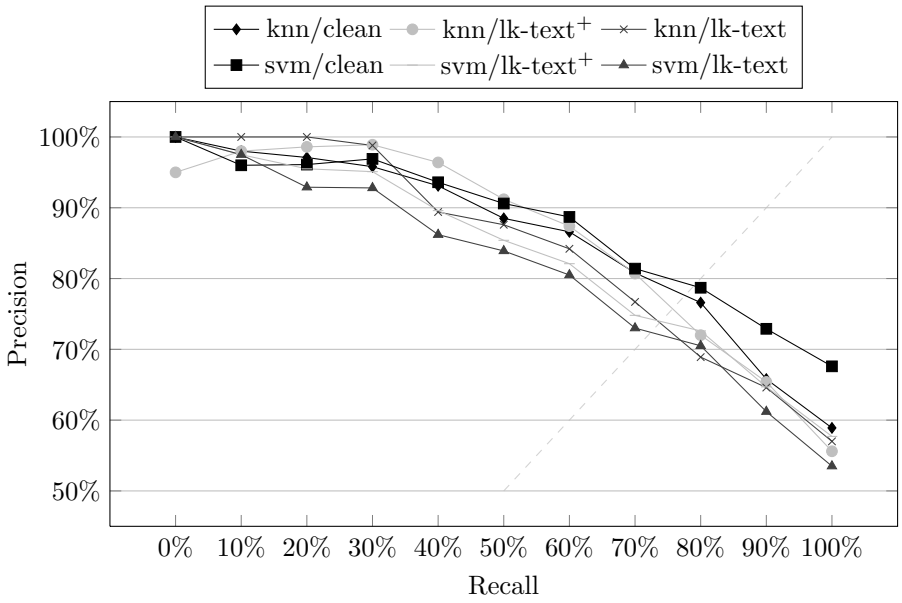

Figure 5. Macro-averaged precision vs recall curve

For recall values up to 80%, figure 5 shows no clear distinction between kNN and SVM macro-averaged precision for clean texts. Yet, in most situations, SVM outperforms kNN.

kNN is less sensitive to distorted texts than SVM, this can be a result of kNN optimal feature set parameterization. Tables 6 and 7 show the feature set overlapping rate between feature sets extracted from the different datasets.

Table 6. SVM feature set overlapping rate (1000 terms)

	clean	lk-text	lk-free
clean	100%	77.40%	42.30%
lk-text	77.40%	100%	42.10%
lk-free	42.30%	42.10%	100%

Table 7. kNN feature set overlapping rate (300 terms)

	clean	lk-text	lk-free
clean	100%	78.00%	51.33%
lk-text	78.00%	100%	50.33%
lk-free	51.33%	50.33%	100%

We can see that overlapping rate between lk-text and clean is almost the same, no influence on the categorization is expected. Furthermore, the overlapping rate between lk-free and clean is higher for kNN, yet accuracy loss with lk-free is slightly higher for kNN (10.64%) than for SVM (8.91%). Nevertheless, as the 300 terms used are the more relevant terms out of the 1000 extracted from the noisy training set, these 300 terms are more likely to be recognized as they were in the training set.

4.4 Significance tests

In order to assess which performance loss is *acceptable* or *little* in a statistical sense, we conducted several Wilcoxon rank tests[21] on categorization results. By comparing performances with this test, we can tell if a significant loss is observed when we use the noisy datasets instead of the clean documents.

Table 8 shows the results of the significance tests based on the paired accuracy values of individual categories. Accuracies of each noisy dataset are compared to those obtained with the clean dataset. A significant loss is observed for lk-free with both methods while no significant loss is observed with the other datasets.

Table 8. Accuracy based significant tests

Noisy dataset	kNN	SVM
lk-free	yes	yes
lk-text	no	no
lk-text$^+$	no	no

Considering that half the relevant terms are lost, performances obtained with lk-free are rather good. Still, a significant loss is observed when we compare lk-free results with performances obtained on clean documents. This confirms our previous observations about the suitability of the lk-free resource for efficient TC.

Tables 9 and 10 show the results of the Wilcoxon test for precision vs recall curve comparison. Paired precision values for each recall level are compared between the noisy datasets and the ground truth set.

Table 9. Micro-precision based significance tests

Noisy dataset	kNN	SVM
lk-free	yes	yes
lk-text	no	yes
lk-text$^+$	no	yes

Table 10. Macro-precision based significance tests

Noisy dataset	kNN	SVM
lk-free	yes	yes
lk-text	no	yes
lk-text$^+$	no	yes

The results presented above show that a significant loss is observed for every noisy set with SVM. Recognition errors may result into irrelevant features, and SVM can be very sensitive to them.[22] However, if we look at figures 4 and 5, we can see that at some points curves are closer to each other. If we want to know at which recall levels a significant loss exists, we have to compare curves at a given point. The tests presented in tables 11 and 12 are based on the paired precision values of individual categories : for a given recall level, precision is computed for each category and pairwise comparison performed.

Table 11. Curve comparison for kNN

Noisy dataset	Significant loss at
lk-free	30%-100%
lk-text	none
lk-text$^+$	none

Table 12. Curve comparison for SVM

Noisy dataset	Significant loss at
lk-free	50%-100%
lk-text	60%, 70%
lk-text$^+$	60%-80%

The results show that for recall values greater than 20%-40% depending on the TC method, the system is unable to handle documents where half the relevant information is lost, once again this confirms that the lk-free resource is not suited for effective TC. As for lk-text and lk-text$^+$, a significant loss is observed for recall levels between 60% and 80% with SVM. This the reason why micro- and macro-precision based significance tests conclude to a significant loss for SVM with all the noisy sets.

5. CONCLUSION

On-line handwritten documents can be transcribed through a recognition process that produces *noise* in the resulting text. By noise we mean word insertions, deletions and substitutions produced during the recognition process. In this paper, we considered an on-line text categorization engine consisting of four stages: on-line handwriting recognition, text pre-processing, text indexing, and categorization. Previous works[6, 23] reported similar categorization systems and showed that categorization of noisy texts can be performed using models trained over electronic, ascii, texts. Whether effective categorization can be done using models trained over noisy texts is one of the questions we address in this paper.

A subset of the Reuters-21578 dataset has been used as ground truth and a great effort has been made to collect an on-line handwritten version of it. Several linguistic resources are used within the on-line recognition engine in order to obtain different levels of noise, hence, obtaining different noisy versions of the clean set. First, categorization is performed on the ground truth dataset and its effectiveness is measured, then the process is repeated over all the noisy versions of the data set.

Significance tests based on pairwise comparison between performances obtained with clean and noisy texts were performed. The results showed that no significant loss is observed with respect to performance achieved on the ground truth set. Effective TC, using state-of-the-art categorization methods, can be performed using categorization models trained over noisy texts. However, a deep understanding of the relationship between the level and type of noise which is present in document corresponding to the output of an online handwriting recognition system and its effects on the categorization is still lacking. As we dispose by now of a significant handwritten data set, complementary experiments will be conducted to further investigate on this issue.

ACKNOWLEDGMENTS

This work is funded by *La Région Pays de la Loire* under the MILES Project and by The French National Research Agency under the framework of the program *Technologies Logicielles* (ANR-06-TLOG-009).

The authors are indebted with the anonymous writers who willingly helped them with the tedious task of rewriting economic newswires.

REFERENCES

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

[2] Murata, M., Busagala, L. S. P., Ohyama, W., Wakabayashi, T., and Kimura, F., "The impact of OCR accuracy and feature transformation on automatic text classification," in [*Proceedings of the Seventh IAPR Workshop on Document Analysis Systems (DAS '06)*], 506–517 (February 2006).

[3] Ittner, D. J., Lewis, D. D., and Ahn, D. D., "Text categorization of low quality images," in [*Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR '95)*], 301–315 (April 1995).

[4] Junker, M. and Hoch, R., "An experimental evaluation of OCR text representations for learning document classifiers," *International Journal on Document Analysis and Recognition, IJDAR* **1**(2), 116–122 (1998).

[5] Chen, N. and Blostein, D., "A survey of document image classification: Problem statement, classifier architecture and performance evaluation," *International Journal on Document Analysis and Recognition, IJDAR* **10**(1), 1–16 (2007).

[6] Vinciarelli, A., "Noisy text categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(12), 1882–1895 (2005).

[7] Koch, G., *Catégorisation Automatique de Documents Manuscrits : Application aux Courriers Entrants*, PhD thesis, University of Rouen (2006).

[8] Salton, G., Wong, A., and Wang, C. S., "A vector space model for automatic indexing," *Communications of the ACM* **18**(11), 613–620 (1975).

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

[10] Forman, G., "A pitfall and solution in multi-class feature selection for text classification," in [*Proceedings of the Twenty-First International Conference on Machine Learning (ICML '04)*], (July 2004).

[11] Yang, Y. and Pedersen, J. O., "A comparative study on feature selection in text categorization," in [*Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97)*], 412–420 (July 1997).

[12] Spärck Jones, K., "Experiments in relevance weighting of search terms," *Information Processing and Management* **15**, 133–144 (1979).

[13] Yang, Y. and Liu, X., "A re-examination of text categorization methods," in [*Proceedings of the Twenty-Second Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*], *22nd Annual International SIGIR* , 42–49 (1999).

[14] Joachims, T., [*Learning to Classify Text using Support Vector Machines*], Kluwer Academic Publishers (2002).

[15] Debole, F. and Sebastiani, F., "An analysis of the relative hardness of reuters-21578 subsets," *Journal of the American Society for Information Science and Technology, JASIST* **56**(6), 584–596 (2005).

[16] Aas, K. and Eikvil, L., "Text categorisation: A survey," tech. rep., Norwegian Computing Center, http://www.nr.no/files/samba/bamg/tm_survey.ps (1999).

[17] Vapnik, V., [*The Nature of Statistical Learning Theory*], Springer-Verlag (1995).

[18] Baeza-Yates, R. and Ribeiro-Neto, B., [*Modern Information Retrieval*], ch. Retrieval Evaluation, 73–99, Addison-Wesley (1999).

[19] Lewis, D. D., "An evaluation of phrasal and clustered representations on a text categorization task," in [*Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*], 37–50 (1992).

[20] Apté, C., Damerau, F., and Weiss, S. M., "Towards language independent automated learning of text categorization models," in [*Research and Development in Information Retrieval*], (1994).

[21] Conover, W. J., [*Practical Nonparametric Statistics*], John Wiley & Sons (1998).

[22] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V., "Feature selection for svms," *Advances in Neural Information Processing Systems* **13**, 668–674 (2000).

[23] Peña Saldarriaga, S., Morin, E., and Viard-Gaudin, C., "Categorization of on-line handwritten documents," in [*Proceedings of the Eight IAPR International Workshop on Document Analysis Systems (DAS '08)*], (2008).