IDIAP RESEARCH REPORT

# Noisy Text Categorization

Alessandro Vinciarelli [a]

IDIAP–RR 04-03

January 2004

to appear in
IEEE Transactions on Pattern Analysis and Machine Intelligence

[a] IDIAP

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

IDIAP Research Report 04-03

# Noisy Text Categorization

Alessandro Vinciarelli

January 2004

to appear in
IEEE Transactions on Pattern Analysis and Machine Intelligence

**Abstract.** This work presents categorization experiments performed over noisy texts. By noisy we mean any text obtained through an extraction process (affected by errors) from media other than digital texts (e.g. transcriptions of speech recordings extracted with a recognition system). The performance of a categorization system over the clean and noisy (Word Error Rate between ∼10 and ∼50 percent) versions of the same documents is compared. The noisy texts are obtained through Handwriting Recognition and simulation of Optical Character Recognition. The results show that the performance loss is acceptable for Recall values up to 60-70 percent depending on the noise sources. New measures of the extraction process performance, allowing a better explanation of the categorization results, are proposed.

# 1 Introduction

Several media contain textual information that can be accessed only through an extraction process. Important examples are speech recordings and handwritten documents (that can be converted into text with a recognition system) as well as videos and images [6] (where text can be detected and recognized). In all of the above mentioned cases (the list is not exhaustive), the extraction process produces *noise*, i.e. word insertions, deletions and substitutions with respect to the actual *clean* text contained in the original source.

Applications dealing with clean text have been the subject of extensive research efforts in the last few decades: techniques developed in domains like Information Retrieval (IR) allow the management and the effective use of huge text corpora [3]. The possibility of extending the results of previous research from clean to noisy text would allow the indirect management of the sources from which the noisy texts are extracted. This represents, in our opinion, an interesting research direction that has been only partially explored. While the application of Information Retrieval to noisy texts (speech recording transcriptions in particular [14]) has been shown to be effective through extensive experiments, only moderate effort has been made, to our knowledge, towards noisy Text Categorization (TC). TC is the task of automatically assigning a document one or more categories belonging to a predefined set $C = \{c_1, c_2, \ldots, c_{|C|}\}$, where $|C|$ is the cardinality of $C$. This work focuses on such problem and shows how noisy texts extracted from different sources can be categorized.

The effectiveness of IR technologies over noisy data [14] seems to suggest that TC methods developed for clean texts could also be successfully applied to noisy texts. On the other hand, although based on the same document representation (the so called *bag of words* [3]) as IR, state-of-the art TC systems rely on Support Vector Machines (SVM) that are not necessarily as robust to noise as the algorithms used in IR (see end of section 2). In fact the recognition errors result into many irrelevant features in the vectors representing texts and SVM's have been shown to be sensitive to them [42].

The main problem in the application of TC to noisy text is that the most effective TC approaches developed so far involve algorithms that need to be trained (e.g. Multi Layer Perceptrons or Support Vector Machines [35],[23]). This is a problem in two main respects: the first is that training over noisy data leads to models that may fit the noise and it can be difficult to use them for data affected by noise with different characteristics. This might result in the need to have, for a certain category, a different model for each source from which the text can be extracted. The second is that, for certain media, it can be difficult to collect enough material for training. A possible solution to both problems is to train category models on clean digital texts (which are relatively easy to collect) and then apply them to noisy texts. The mismatch between training and test conditions due to the presence of noise is likely to degrade the performance, but, if the loss is acceptable, the solution can be a good trade-off between categorization performance and the experimental effort required to achieve it.

The experiments presented in this work are based on the Reuters-21578 clean text database [27], a well known and widely applied benchmark in TC research. A subset of 200 documents has first been extracted from its test set (see section 4 for details), then noise has been added to the texts it contains using two methods: the first is to manually write the documents of the subset and then to recognize them with an offline handwriting recognition system [40] (the use of 200 documents is due to the effort required to produce the handwritten versions of the texts). The second is to simulate an OCR based extraction process by randomly changing a certain percentage of characters [8]. The use of an automatic simulation allowed us to perform experiments not only over the above mentioned set of 200 texts (where it is possible to compare results over both OCR and handwritten data), but also over the whole test set of the Reuters-21578 corpus (where the comparison between Handwriting and OCR is not possible). In the OCR simulation, it has been possible to set different values of Character Error Rate (CER) leading to different amounts of noise. The categorization results show that the performance loss when passing from clean to noisy versions of the same texts is acceptable at low Recall values (less than 60-70 percent depending on the noise sources).

The most common measure for the degradation produced by an extraction process is the Word Error Rate (WER). The WER is used for both processes considered in this work (see section 4) as well as for

other cases (speech recognition, text detection in videos and images, manual typing, etc.). The WER values of the noisy texts considered in this work span from ∼10 percent to ∼50 percent depending on the source, while the categorization performance measures have a smaller range: e.g. the average Precision is between ∼85 (75) percent and ∼95 (80) percent for the 200 documents set (the whole Reuters test set). This happens because the WER is a measure oriented to the performance of the extraction processes and takes into account errors that have no influence on the categorization process (see section 4). For this reason, alternative measures are proposed that allow a better explanation of the final categorization results.
The rest of this paper is organized as follows: Section 2 provides a survey of the domain, Section 3 describes the TC system used in this work, Section 4 shows experiments and results and Section 5 draws some conclusions.

## 2 Previous Work

This section presents a survey of the literature dedicated to processing of noisy texts. Relatively few papers are available and of those, the majority concern the retrieval of speech recordings or scanned printed documents. In the first case, the text extraction process is Automatic Speech Recognition (ASR), in the second case it is Optical Character Recognition (OCR). The two processes produce different forms of noise: ASR systems give rise to word substitutions, deletions and insertions, while OCR systems produce essentially word substitutions (see section 4 for more details). Moreover, ASR systems are constrained by a lexicon and can give as output only words belonging to it, while OCR systems can work without a lexicon (this corresponds to the possibility of transcribing any character string) and can output sequences of symbols not necessarily corresponding to actual words. Such differences have a strong influence on the retrieval approach applied.
Most of the research on the retrieval of speech recordings has been made in the framework of the TREC conferences [14]: several groups worked on the same database (TDT-3 [16]) containing both manual (WER ∼10 percent) and automatic (WER ∼30 percent) transcriptions of broadcast news recordings. The TDT-3 dataset is composed of around 25,000 documents and in addition a set of queries with their respective relevance judgements. The participants equipped with an ASR system could use their own transcriptions which enabled the evaluation of the WER impact on the retrieval performance. The works presented in the TREC context do not try to model the noise: the techniques succesfully applied on clean texts have been shown to be effective also on noisy automatic transcriptions. All systems are based on the Vector Space Model (VSM) [3], where documents and queries are converted into vectors and then compared through matching functions. In most cases, the documents are indexed with *tf.idf* [3] (see section 3 for more details) and matched with the Okapi formula [24],[1],[13],[15], along with other approaches [36],[17],[26].
During the extensive experiments and comparisons performed in the TREC framework, at least two important conclusions emerge: (a) The retrieval is more effective over transcriptions at the word, rather than at the phoneme level. Some attempts were made to recognize documents as phoneme sequences and then to match them with the query words, but the performances were much lower than in the alternative approach [14]. (b) There is almost no retrieval performance degradation when increasing the WER from around 10 percent to around 40 percent [14].
Several works have been dedicated to the retrieval of OCR based transcriptions of printed documents [9]. The first approaches simply applied standard IR techniques to transcriptions [8],[45]. The results showed that Character Error Rates (CER) up to ∼ 5 percent can be tolerated without significant retrieval performance loss. A severe drop in performance is observed only when the CER is around 20 percent. In order to make the retrieval systems more robust with respect to the CER, several approaches have been proposed: in [30] the OCR confusion matrix and character bigram statistics are used to obtain multiple search terms from the query words. This is supposed to enable the matching between the correctly spelled terms of the query, and document terms potentially affected by OCR errors. An alternative way to solve the same problem is to use more flexible string matching

algorithms: in [28] approximate string matching and fuzzy logic are used to match clean terms and words containing OCR errors. Other works try to correct the output of the OCR system in order to reduce the noise [37],[10]. An alternative approach has been presented in [39], where the texts are compared only in terms of how similar are their images. This enables the comparison and the matching of texts without the need of OCR.

The retrieval of handwritten documents - for which the extraction process is a recognition process analogous to the case of speech recordings - has been investigated only recently [33],[32]. Both of these works focused on the detection of query words rather than on actual IR. In [33] the $N$ best transcriptions obtained through handwriting recognition are used in order to deal with recognition errors. In [32] word image matching techniques are used to perform word spotting.

Although both are based on the same *bag of words* document representation (see section 3 for more details), IR and TC use different algorithms. State-of-the-art IR systems rely essentially on similarity measures between vectors representing documents and queries, while state-of-the-art TC systems use Support Vector Machines [5]. The success of IR technologies over noisy texts is thus not a sufficient condition, in our opinion, to claim that also SVM based TC technologies may work effectively on noisy data. In IR, the systems measure the similarity between queries and vectors by looking for query terms in the documents and retrieve as relevant the documents where query terms are more frequent. The only errors actually affecting the retrieval results are thus those concerning the query terms. Since more important words are often repeated in the documents, it is relatively rare that a query term disappears; moreover, queries contain typically several words and, even with a high WER, the probability of misrecognizing all of them is low. On the contrary, the presence of errors can be more problematic for SVM's. In fact it has been shown that SVM performance is negatively affected by the presence of many irrelevant features (such as those introduced by recognition errors in vectors representing texts) in their input [42], and our experiments show that the effect of recognition errors in categorization is actually more significant than in retrieval, especially at high Recall values.

Moreover, in some applications the noise has been shown to degrade significantly the performance of state-of-the-art systems. It is the case of Named Entity Extraction (NEE) and summarization [29],[25]. In [29], it is shown that for a state-of-the-art NEE system the F-measure (see section 4.5 for more details) decreases of 0.6 for each 1 percent of Word Error Rate. Moreover the absence of Named Entities in the vocabularies used by the recognizers (recognition processes can give as output only words belonging to their dictionaries) has a heavy effect on the final performance. In [25], the authors present an application summarizing voicemail messages through the extraction of salient information (message sender, call reason, etc.). The results show that, when passing from manual to automatic transcriptions of the messages, the correct identification of the sender by the receiver drops from 94 to 57 percent. On the other hand, the noise is not degrading the performance for other tasks like finding the subject of the message or defining the call priority.

While noisy text retrieval has been extensively investigated, noisy Text Categorization has been addressed, to our knowledge, in only a few studies [4],[18],[38]. In [4], in which the noisy texts are obtained from printed documents through OCR, the features describing a text are word substrings extracted with an iterative procedure. The selection of the features is task dependent and adapted to the data under consideration. Such an approach enables the modeling of the data noise and makes categorization more robust. However, if for example the OCR system changed, a new noise model could be required. In [18], German businness letters recognized with an OCR system (WER $\sim$20 percent) are attributed to one of five predefined classes. The experiments are performed over a test set of 42 letters. The rate of correct classification is 57 percent. In [38], the performance of a Naive Bayes classifier over 400 documents recognized with an OCR (WER $\sim$14 percent) is presented. Six categories (out of 52) are analyzed and the highest rate of correct classification achieved is 83.3 percent.

A comparison with results obtained using clean versions of the documents is made only in [38]. For this reason it is not always possible to say whether the categorization techniques are robust with respect to noise or not. Moreover, the works focus solely on OCR and other modalities such as speech or handwriting recognition have not been investigated.

# 3 Text Categorization

This section presents the TC system used in this work. Several approaches have been proposed in the literature (see [35] for a survey) and the best results have been obtained by representing the documents as vectors (like in the VSM [3]) and by categorizing them with algorithms that can be trained (e.g. Neural Networks and Decision Trees). The system used in this work is based on Support Vector Machines [7] and achieves state-of-the-art performances on the main benchmark tasks presented in the literature.
The next sections will describe in detail the individual steps of the categorization process: preprocessing, normalization, indexing, categorization and performance evaluation.

## 3.1 Preprocessing and Normalization

Preprocessing and normalization perform the so-called *lexical analysis*, i.e. the selection of the information in the texts that is useful for categorization [11].
The preprocessing takes as input the raw documents and removes all elements supposed to be category neutral. In our system, all non-alphabetic characters (digits, punctuation marks, dashes between connected words like *self-contained*, etc.) are eliminated. This solution has some disadvantages: acronyms cannot be distinguished from words composed of the same letters (e.g. *U.S.* is processed like *us*), connected words that should be used as a single index term are processed separately (e.g. *state-of-the-art* is treated like *state of the art*), etc.. On the other hand, this approach is very simple and can be applied with good results to many text corpora.
The preprocessing converts the original raw texts into streams of *words* that are then inout to the normalization stage. The normalization stage uses two techniques , *stopping* and *stemming*, to remove information that is not useful for categorization. Stopping is the removal of all words expected to be poor index terms (the so-called *stopwords*): articles, propositions, pronouns and other functional words that are not related to the content, or words that appear in too many documents to allow a discrimination between different texts. Stopwords occur with high frequency (e.g. the word *the* accounts for around 7 percent of the total amount of words in an average text corpus) and the application of the stopping results in an average reduction of the number of words by around 50 percent.
Stemming is the conflation of the morphological variants of the same word (e.g. *connection*, *connecting*, *connected*) into a common stem (*connect*) [12]. The application of the stemming is based on the hypothesis that different inflected forms of a certain word do not carry category dependent information. The experiments show that, in most cases, the stemming actually leads to an improvement of the categorization performance, although some counter-examples have been presented in the literature [35][23]. After the stemming, the number of unique terms in the text corpus is reduced, on average, by around 30 percent. This reduces the dimension of the vectors representing the documents (see section 3.2) leading to computational advantages and to a limitation of *curse of dimensionality* related problems [19]. Among the various stemming techniques proposed in the literature (see [12] for a survey), the most commonly applied is the one proposed by Porter [31]. Such an algorithm represents a good trade-off between complexity and effectiveness and it is used in our system.

## 3.2 Indexing

After preprocessing and normalization, the original documents are available as streams of terms containing only the information supposed to be useful for the categorization. A stream of terms is not a suitable representation and an indexing procedure must be applied in order to represent the document content in a way allowing the actual categorization step.
The indexing techniques applied in TC are the same as those used in IR. The VSM [3] is used in most cases [35][23]: the documents are represented with vectors where each component accounts for

one of the terms belonging to the dictionary (the list of all unique terms contained in the text corpus after preprocessing and normalization). The VSM is based on the *bag-of-words* approximation: the terms are assumed to be independent of each other and the order of the terms in the original texts is not taken into account. Although simple, the VSM has been shown to be effective in both IR and TC [3][23]. More complex representations give rise to improvements not worth the effort they require [35].
The indexing is typically based on a *term by document* matrix $A$ where each column $j$ corresponds to a document of the text corpus under consideration and each row $i$ corresponds to a term of the dictionary. The element $A_{ij}$ is the component of document vector $j$ related to term $i$. The matrix element can be thought of as a product of two functions:

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

where $L(i,j)$ is a local weight using only information contained in document $j$ and $G(i)$ is a global weight using information extracted from the whole corpus (an extensive survey about $L(i,j)$ and $G(i)$ functions can be found in [34]). The most commonly applied weighting scheme is the so-called $tf \cdot idf$:

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

where $tf(i,j)$ is the term frequency, i.e. the number of times term $i$ appears in document $j$, $N$ is the total number of documents in the database and $N_i$ is the number of documents containing term $i$ (the logarithm is called *inverse document frequency*). In this scheme, a term is given more weight when it occurs more frequently (it is assumed to be more representative of the document content) and when it is contained in few documents (it is assumed to be more discriminative). The dimension of the data is very high (several tens of thousands) and the vectors are typically sparse. Only few terms (less than 1 percent on average) of the dictionary are represented in each text and this results in few non-zero components in the vectors.

### 3.3 Categorization

Several categorization approaches have achieved state-of-the-art performance [35]. In this work we apply Support Vector Machines (see [7],[5] for a good introduction) because there is theoretical evidence that they are especially suitable for data with characteristics typical of document vectors [23]: very high dimensionality (several thousands of dimensions), sparseness (few features have values different from zero), high number of relevant features (every term is important even if it appears only a few times), distribution of the term frequences following Zipf's Law [46], and a high level of redundancy (several features account for the same category). Based on such properties it is possible to bound the expected prediction error [23] of the SVM's. This supports their use not only on the basis of empirical performance evaluation, but also from a theoretical point of view.
Given a document vector $\mathbf{d}$, the SVM related to $c$ calculates the following score:

$$f(\mathbf{d}) = \sum_i \alpha_i y_i \Phi(\mathbf{s}_i)\Phi(\mathbf{d}) + b = \sum_i \alpha_i y_i K(\mathbf{s}_i, \mathbf{d}) + b \tag{3}$$

where the $\alpha_i$'s and $b$ are coefficients obtained during training, the $\mathbf{s}_i$ are the Support Vectors and the $y_i$'s are labels ($y_i = 1$ when $\mathbf{s}_i$ belongs to category $c$, and $y_i = -1$ otherwise). The function $\Phi(\mathbf{x})$ is supposed to map the data into a space (the *feature space*) where the documents belonging to $c$ can be more easily separated from others with a hyperplane. Since a positive score is associated to Support Vectors belonging to $c$, the higher $f(\mathbf{d})$, the higher the probability that $\mathbf{d}$ belongs to category $c$.
The product $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{y})$ can be replaced with a function $K(\mathbf{x},\mathbf{y})$ (called *kernel*) satisfying appropriate conditions [7],[5]. This allows the use of infinite dimensional feature spaces or mappings that cannot be known explicitly.
Because of the characteristics mentioned at the beginning of this section, the document space has the

properties expected in a feature space. For this reason, it is possible to achieve statisfactory results by simply using a linear kernel [21]: $K(\mathbf{x},\mathbf{y}) = \mathbf{x}\cdot\mathbf{y}$ (we used the SVM Light package [22]). The main advantage is that a linear kernel has no hyperparameters to be set manually and this simplifies the training of the SVM's using it. More sophisticated kernels [7] lead to better performances, but the goal of this work is to show the effect of the recognition errors on a categorization system rather than to achieve the highest possible performance. For this reason our experiments involve only linear kernels.

A different SVM is trained for each category to distinguish between the documents belonging to it and the others. This makes it possible to cope with documents that belong to more than one category and texts that do not belong to any of the categories in the predefined set $C$: since the decisions about different categories are made separately and do not influence each other, a certain document can be accepted by more than one SVM or it can be rejected by all of them: in the first case, the document is assumed to belong to more than one category, in the second case it is assumed to be a document that does not belong to any $c \in C$.

### 3.4 Performance Evaluation

The performance evaluation of a TC system is an open problem. Several measures are available, but none of them describes exhaustively the system performance. Moreover, depending on the application the appropriatenes of a measure can vary. In this work we use several measures in order to give a description comprehensive of the results.

Given a category $c$, $R(c)$ is the set of the documents actually belonging to it and $R^*(c)$ is the set of the documents identified as such by the system. The two fundamental measures of TC performance rely on such sets and are called *Precision* [3]:

$$\pi(c) = \frac{|R^*(c) \cap R(c)|}{|R^*(c)|} \tag{4}$$

and *Recall* [3]:

$$\rho(c) = \frac{|R^*(c) \cap R(c)|}{|R(c)|}. \tag{5}$$

The value of $\pi$ is typically calculated at standard values of $\rho$ $(10\%, 20\%, \dots, 100\%)$ resulting in the so-called *Precision vs Recall* curve [3]. The comparison between different systems can be difficult if only these curves are available. For this reason, there are two ways of representing the curve with a single value: the first is to average over the $\pi$ values along the curve to obtain the *average Precision* (avgP) [3]. The second is to find the *Break Even Point* (BEP), i.e. the point of the curve where $\pi = \rho$ [3]. In both cases, the single measure is less informative than the whole curve, but it is easier to use in comparisons.

The above measures must be averaged over different categories in order to obtain a global performance measure. The average can be obtained in two ways: *Macroaveraging* and *Microaveraging* [35]. In the first case, the average $\pi$ value at a certain $\rho$ level is obtained as follows:

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

where the index $M$ stands for macroaverage. In the second case, the average is calculated as:

$$\pi^\mu = \frac{\sum_c |R^*(c)| \cdot \pi(c)}{\sum_{c'} |R^*(c')|} \tag{7}$$

where the index $\mu$ stands for microaverage (the expression of the micoraveraged recall can be obtained in the same way). The macroaverage is a category pivoted measure: all categories are given the same weight independently of how much they are represented. The performance of categories occurring few times is emphasized. The microaverage is a document pivoted measure: all documents are given the same weight and the most represented categories have more influence on the final performance measure (the $R^*$ set is typically bigger for the categories occurring more frequently). Macroaverage and microaverage can be very different when the number of occurrences changes significantly depending on the category. Whether one or the other should be used depends on the application. When a TC system is used to organize a document collection (e.g. the creation of directories in an archive) the macroaverage is more appropriate. In applications like mail filtering or routing where the categorization is performed document by document, the microaverage is a better measure. In this work, both averages will be used (wherever is possible) in order to give a comprehensive description of the performance.

The measures described so far are typically used in a laboratory setup and are oriented to a general description of the system performance. In application environments, different measures can be used that better reflect the requirements of the specific task the system must perform. An application oriented measure is the *Precision at position* $n$: all the documents of the test set are ranked according to the score obtained with the SVM related to category $c$ and the $\pi$ value at position $n$ is measured. Such a measure is interesting for applications where the system deals with document databases where, rather than taking a decision, it ranks the documents depending on their score. A good system is supposed to rank all documents belonging to $c$ at the top positions (this allows the user to find all the needed documents without browsing the whole dataset), i.e. to have high Precision at position $n$. A further example is the *application oriented Precision* used in contexts where the decisions are taken document by document: the system attributes category $c$ to a certain document when the score obtained using the related SVM (see section 4.4) is above a thresold $\theta(c)$. The threshold set $\Theta = \{\theta(c_1), \theta(c_2), \ldots, \theta(c_{|C|})\}$ must be obtained using a validation set independent from the test set. The thresholds are selected in order to achieve a specific task (e.g. a certain $\pi$ value) and the results obtained over the test set show whether the system can achieve it.

# 4 Experiments and Results

This section describes the experiments performed in this work. The first step is the creation of noisy documents starting from a set of clean texts. Two methods are used: the first is based on handwriting recognition, the second on the simulation of an Optical Character Recognition (OCR) process. The categorization performance over both clean and noisy versions of the same documents is compared. The results show that the performance loss is acceptable for Recall values up to 60 or 70 percent depending on the sources, but it is difficult to relate noise measures with categorization results. For this reason, new noise measures are proposed in order to better relate extraction process and categorization performance.

## 4.1 Experimental setup

The experiments performed in this work are based on the Reuters-21578 database [27], a well known and widely applied TC benchmark. The database contains 12,902 documents partitioned into training (9,603 documents) and test set (3,299 documents) following the so-called ModApté split [2]. The database is composed of articles extracted from the Reuters newswire bulletin. The number of categories is 115, but not all of them are sufficiently represented to allow training of a model. The ten most represented categories account for more than 90 percent of the database, while the less represented categories occur only in the training set (or only in the test set) and cannot thus be considered. For this reason, this work focuses on the ten most frequently occurring categories.

The noisy versions of the documents were obtained through OCR simulation and handwriting recognition (see next subsection for more details). In OCR case it is possible to use the whole Reuters-21578 test set (referred to as *full*). In handwriting case, because of the heavy effort required to produce manuscript data, it was necessary to select a subset of the full test set. For each of the ten most represented categories, 20 examples were randomly selected, resulting in a collection of 200 documents referred to as *small* test set. All the experiments presented in this section are performed over both *full* and *small* test sets.

### 4.2 The noisy data

In order to obtain noisy versions of the documents, two methods are applied: the first is to change a certain percentage of characters into other symbols through a random process [8]. This simulates Optical Character Recognition (OCR), or manual typing and the Character Error Rate (CER) can be set arbitrarily leading to different levels of noise. The second is to manually write the documents and then recognize them using an offline cursive handwriting recognition system. In this case, the noise level cannot be set in advance, but is only measured after the recognition is performed.

The OCR simulation is simple, and several phenomena occurring in real systems are not simulated (e.g. systematic errors for certain characters or frequent confusion between couples of letters like $i$ and $l$). On the other hand, what is most important for our work is the way OCR errors affect categorization results: when a character is misrecognized, it rarely transforms the word it belongs to is transformed into another meaningful term. When a document is indexed, only terms in the dictionary are taken into account, thus most of the words where a character is misrecognized simply disappear, and no longer play a role in the categorization process. From this point of view, the effect of our simulation on the categorization process is similar to the effect of a real OCR system. The use of simulation, makes it possible to add noise to large amounts of data and to perform experiments over the *full* test set.

Handwriting Recognition is the second technique we use to add noise to the data. A complete description of the handwriting recognizer used can be found in [40]. The transcription is performed line by line (reducing the the search space dimension) through several steps: *normalization*, *feature extraction* and *recognition*. During the first step, the system removes *slant* and *slope*. The slant is the angle between the vertical direction and the direction of the strokes supposed to be vertical in an ideal handwriting model (e.g. the vertical bar of a $t$ which is often inclined due to personal handwriting style). The slope is the angle between the horizontal direction and the direction of the line the words are aligned on (the acquisition process often rotates the data resulting in lines that are not horizontal). A full description of the normalization technique can be found in [41].

Feature extraction is performed with a sliding window technique: a fixed width window shifts column by column from left to right and, at each position, a feature vector is extracted from its content (see [40] for more details). After feature extraction, the line image is converted into a sequence of vectors $O = \{o_1, o_2, \ldots, o_M\}$ and it is possible to perform the recognition. In the approach we use [20], we look for the word sequence $\hat{W} = \{w_1, w_2, \ldots, w_N\}$ maximizing the probability $p(W|O)$:

$$\hat{W} = \arg\max_{W} p(W|O). \tag{8}$$

By applying the Bayes theorem, the last equation can be rewritten as

$$\hat{W} = \arg\max_{W} \frac{p(O|W)p(W)}{p(O)} \tag{9}$$

and since $p(O)$ is constant during the recognition, this corresponds to:

$$\hat{W} = \arg\max_{W} p(O|W)p(W). \tag{10}$$

In our system, $p(O|W)$ is estimated with Continuous Density Hidden Markov Models (HMM) and $p(W)$ is estimated with bigrams [20].

The HMM's are trained using a set of 50 documents (40 for training and 10 for validation) independent of the 200 documents used in the small test set. A HMM is built for each letter and word models are

| Source | WER (small) | TER (small) | WER (full) | TER (full) |
|---|---|---|---|---|
| Handwriting | 49.4% | 40.7% | none | none |
| OCR (2%) | 11.3% | 24.5% | 11.4% | 16.1% |
| OCR (4%) | 20.7% | 35.5% | 21.4% | 27.8% |
| OCR (6%) | 29.9% | 44.8% | 30.4% | 38.5% |
| OCR (8%) | 37.2% | 54.2% | 28.1% | 47.3% |
| OCR (10%) | 44.2% | 54.6% | 44.8% | 54.9% |

Table 1: Word and Term Error Rates. This table reports the noise level of the documents in terms of Word and Term Error Rate for both small (second and third columns from left) and full (fourth and fifth columns from left) test set.

obtained by concatenating single letter models. All letter models have the same number of states $S$ and Gaussians $G$ per state. The values of $G$ and $S$ are selected through validation: systems corresponding to several $(S, G)$ couples are trained over the training set and tested over the validation set and the couple leading to the highest recognition results is retained as optimal.
The bigram Statistical Language Model is trained over the training set of the Reuters-21578 corpus. The lexicon is composed of the 20,000 most frequent words in the same set. There are two main reasons behind this choice: the first is that we can expect that the most frequent words in the training set are frequent also in the test set (the hypothesis is that the training set is representative of the test set data), the second is that the estimations performed over more frequent words are statistically more reliable. The same lexicon is used as the dictionary for the handwriting recognition and TC systems. Since the training and test set are independent of each other, around 4 percent of the words in the latter set are not represented in the lexicon and thus cannot be correctly recognized. This happens especially for proper names that typically appear several times in a few documents concentrated in a certain time interval (the documents in the corpus are ordered by time). On the other hand, it is not possible to extract the lexicon from the test set since no information obtained from it can be used when building the system (for more details about this experimental setup, see [40]).

## 4.3 Word and Term Error Rates

The effect of both extraction processes can be measured with the Word Error Rate (WER), i.e. the percentage of words incorrectly extracted from the original text. The WER is obtained through an alignment (performed with Dynamic Programming) between the original text and the text obtained after the extraction process and it can be used as a general measure of the noise. On the other hand, the WER is not a good noise measure in the categorization context because it takes into account errors that have no influence on the document representation: the recognition of stopwords as other stopwords (e.g. *there* as *these*) and the transcription of inflected forms of a word into other inflected forms of the same word (e.g. *yields* into *yield*). Both these errors are corrected during the normalization: stopwords are removed and morphological variants of the same words are replaced with their stem (see Section 3). Moreover, the WER takes into account the order of the words while in the bag-of-words approximation the position of the terms is not important.
The WER is thus an overestimation of the noise and it is better to use the Term Error Rate (TER), i.e. the percentage of terms lost during the extraction process:

$$TER = 1 - \frac{\sum_i \min(tf(i), tf^*(i))}{\sum_k tf(k)} \tag{11}$$

where $tf(i)$ and $tf^*(i)$ are the frequencies of term $i$ in the clean and noisy text respectively. The values of WER and TER for the different noisy versions of the documents are shown in Table 1. While in the case of handwriting recognition, the TER is lower than the WER, in the OCR simulation, the TER is higher than the WER. The reason for this is that, in handwriting recognition, the transcription is performed on a word basis. The recognizer finds the sequence of words belonging to its dictionary that maximizes the likelihood of the handwritten data [40]. Since the recognition rate is lower for shorter words, the stopwords (that are shorter than other words on average) tend to be misrecognized more frequently leading to a TER lower than the WER. In the case of the OCR simulation, the transcription is performed on a letter basis. The shorter the word, the higher the probability of transcribing correctly all of its characters. This favors the stopwords and penalizes the content words leading to a TER higher than the WER.

### 4.4 Categorization

This section presents the results of the categorization experiments performed over the different versions of the above described data sets. The results have been collected for clean texts (small and full test set), handwriting based transcriptions (small test set) and OCR simulations with CER between 2 percent and 10 percent (small and full test set). The performance is measured in terms of Precision vs Recall curves, BEP, avgP, Precision at position $n$ and application oriented Precision (see section 3.4 for details about the measures). The results are presented both as macroaverages and microaverages (see section 3.4) where possible.

The category models used in the experiments are trained over the training set of the Reuters-21578 database. The mismatch between training (the texts are clean) and test (the documents are affected by noise) conditions is likely to degrade the categorization performance. On the other hand, if the loss is acceptable, the possibility of using models trained over clean texts to categorize noisy documents extracted from different sources has two important advantages: the first is that clean texts are relatively easy to collect and it is thus possible to have enough training material. The second is that it is not necessary to train a different model for each source of noisy text.

An SVM has been trained for each of the ten most represented categories. Good categorization results can be obtained with a linear kernel [21]: this corresponds to the use of the document space as feature space (see section 3.3) and it has the advantage that no hyperparameters must be set. An SVM related to category $c$ is trained to give positive answer when the document actually belongs to $c$ and negative answer otherwise. The decisions about different categories are made separately and independenlty of each other. This allows the system to assign a document more than one category (when more than one SVM gives a positive answer) or none of the predefined categories (when all SVMs give a negative answer).

In order to evaluate the results, the documents of the test set are ranked according to the score assigned to them by the SVM related to a certain category $c$. The documents at the top of the ranking are used as set $R^*(c)$ of the texts identified as belonging to $c$ by the system (see section 3.4). This allows one to measure $\rho$ and $\pi$ for each category. When the system works correctly, all the documents actually belonging to $c$ occupy the first positions of the ranking, then high Precision is achieved for all values of Recall. When some documents belonging to $c$ fall towards the last positions of the ranking, the $\pi$ values become low because $R^*(c)$ contains many documents not belonging to $c$.

Figures 1 and 2 show the Precision vs Recall curves for the small and full test set respectively. For the small test set, the Wilcoxon test [43] shows that, with a confidence level of 95 percent, the differences between the curve obtained from the clean texts and the curves obtained from the other sources are not statistically significant for Recall values up to 70 percent. The long documents tend to be more robust to noise. The reason is that they contain multiple occurrences of the terms relevant to their category, thus the probability that these are preserved during the extraction process is higher. The Precision drops at Recall values higher than 70 percent for the small test set are due to few documents that, because of the noise, fall at the bottom positions of the document ranking. The results are similar in both microaverage and macroaverage. The reason is that, on average, the performance of the system is similar for all categories and the category distribution is relatively uniform.

In the case of the full test set, the impact of the noise on the categorization performance is more evident: for Recall values higher than 50-60 percent, the performance loss is high and the system is

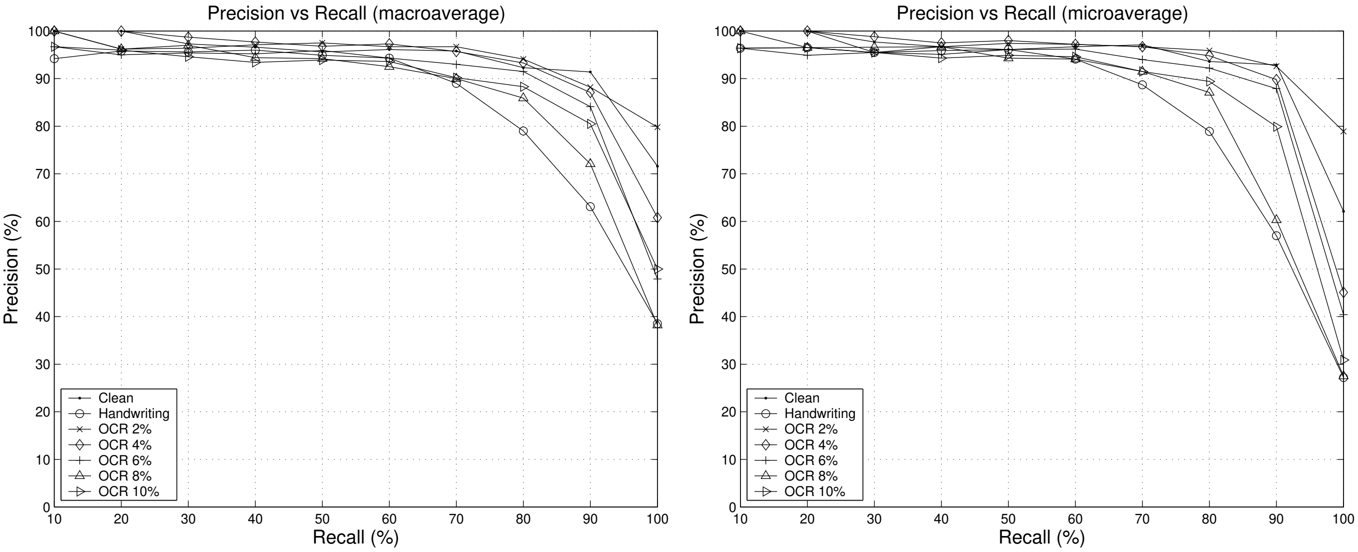



Figure 1: Precision vs Recall (small test set). The Precision is plotted as a function of the Recall. The curves are obtained by both macroaveraging (left plot) and microaveraging (right plot) over the different categories.

less robust with respect to the noise. The BEP and avgP values extracted from the curves of Figure 1 and 2 are shown in Table 2 and 3 respectively. BEP and avgP values are heavily influenced by the last points of the plot. For this reason, sources like handwriting or OCR at CER$\geq$8 percent have a low value compared to the clean case.

The curves of the Precision at position $n$ allow one to evaluate a different aspect of the categorization performance. When using the previous measures, the user is assumed to be interested in finding all the documents belonging to a certain category in the database. When using the Precision at Position $n$, the user is assumed to be interested to find only $n$ documents belonging to the category. The performance is influenced in this case by the highest positions in the ranking (i.e. low Recall points of the curve) where the systems are closer to each other. This can be observed in Figure 3, where the curves do not show the same differences as in the case of the Precison vs Recall curves. At $n = 20$, the precision is between $\sim$90 percent and $\sim$95 percent ($\sim$80 percent and $\sim$95 percent) for the small (full) test set. This means that, on average, for the small test set at least 18 documents (16 for the full test set) out of the 20 top ranking ones actually belong to the category under consideration. Since Precision at Position $n$ measures take into account only the top of the ranking, the increase of the overlap in the score distributions (see above) has no effect. For this reason, the results over small and full test set are for this measure more similar. No distinction can be made between macroaverage and microaverage: since the performance is considered at a specific position rather than at a certain Recall level, the value of $|R^*(c)|$ (see equation 7) is the same for all categories and it simply corresponds to $n$. For this reason, the weight given to each category becomes $\frac{1}{|C|}$ and this results in the equivalence between microaverage and macroaverage.

The evaluation can be performed also in a different application perspective. A system can be used to separately assign a category to input documents. This is necessary when input texts must be routed to different users or destinations. In this case, the category $c$ is assigned to a document if the score obtained with the $c$ related SVM is higher than a threshold $\theta(c)$. The threshold set $\Theta = \{\theta(c_1), \theta(c_2), \ldots, \theta(c_{|C|}\}$ must be obtained from an evaluation set different from the test set. In our experiments we used the Reuters-21578 training set as the evaluation set. The thresholds $\theta(c)$ are

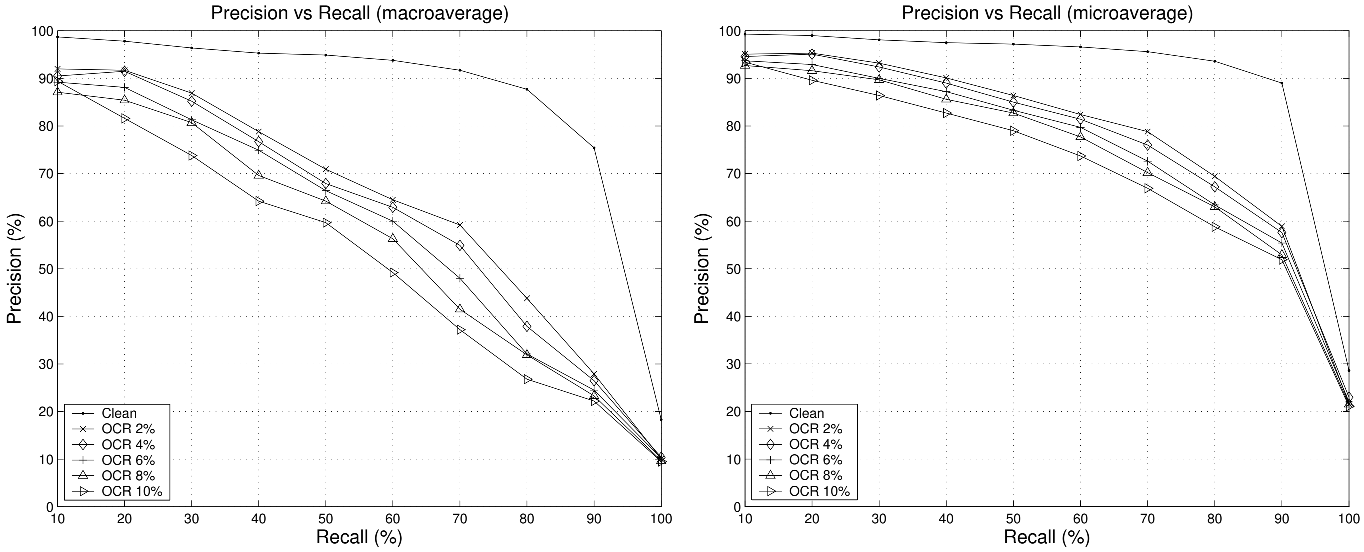



Figure 2: Precision vs Recall (full test set). The Precision is plotted as a function of the Recall. The curves are obtained by both macroaveraging (left plot) and microaveraging (right plot) over the different categories.

selected in correspondence of the BEP as measured over the evaluation set. Figure 4 shows the values of $\pi$ and $\rho$ obtained for the different noisy sources. In the small test set, most of them remain close to the $\pi = \rho$ line, i.e. tend to reproduce the BEP also over the test set. In the case of handwriting, the Precision of the BEP is preserved, but at the price of a high Recall loss. This is still helpful in applications where the user is not necessarily interested in high Recall, but in the selection of a subset with high Precision. A similar phenomenon is observed for all the sources over the full test set. The system tends to select fewer documents (all the points have lower Recall with respect to the BEP on the validation set), but with high Precision. This further confirms the fact that the highest scoring documents are less affected by the noise.

All experiments performed so far make use of the whole Reuters training set, but often less material is available. For this reason, we trained the category models using only the first 10, 20, . . . 100 percent of the available training data (the Reuters documents are ordered by time and this corresponds to use the documents available at a certain moment to categorize those which are provided in the future). The corresponding systems are tested over the test set and the performance (in terms of microaveraged BEP) is shown in Figure 5 for both full and small test set. The results show that the effect of the training set size is essentially the same for both clean and noisy documents. After a certain amount of material is used, to add training data does not result in significant improvements.

Another problem that can occur is that only noisy material is available, thus it is not possible to train over clean texts. Figure 6 shows the Precision vs Recall curves obtained over the full test set when the training set contains noisy texts (training and test set have the same CER, from 2 to 10 percent). The Precision vs Recall curves are essentially the same as those shown in Figure 2 and this seems to suggest that the training is robust with respect to noise.

## 4.5 The Coverage Plan

The results of the previous section show that there is no clear correlation between noise measures (WER and TER) and categorization results. Data affected by very different levels of WER and TER

| Source | M BEP (%) | M avgP (%) | $\mu$ BEP (%) | $\mu$ avgP (%) |
|---|---|---|---|---|
| Clean | 90.8 | 93.7 | 93.3 | 91.7 |
| Handwriting | 80.3 | 84.0 | 81.0 | 82.6 |
| OCR (2%) | 89.4 | 94.3 | 90.7 | 95.1 |
| OCR (4%) | 88.4 | 92.8 | 89.0 | 91.4 |
| OCR (6%) | 86.3 | 88.9 | 87.7 | 88.9 |
| OCR (8%) | 82.9 | 86.0 | 84.0 | 84.5 |
| OCR (10%) | 81.2 | 85.0 | 82.3 | 83.4 |

Table 2: Categorization Performance (small test set). This table shows BEP and avgP for each source of text. Both macroaverages and microaverages are reported.

| Source | M BEP (%) | M avgP (%) | $\mu$ BEP (%) | $\mu$ avgP (%) |
|---|---|---|---|---|
| Clean | 85.0 | 90.5 | 91.4 | 95.2 |
| OCR (2%) | 65.4 | 66.5 | 77.3 | 81.4 |
| OCR (4%) | 63.5 | 64.5 | 76.3 | 80.8 |
| OCR (6%) | 60.3 | 61.3 | 74.0 | 77.9 |
| OCR (8%) | 59.6 | 58.8 | 73.5 | 76.6 |
| OCR (10%) | 56.0 | 55.5 | 73.0 | 74.5 |

Table 3: Categorization Performance (full test set). This table shows BEP and avgP for each source of text. Both macroaverages and microaverages are reported.

have similar categorization performances and, in some cases, data with higher WER or TER have lower avgP or BEP. This means that the noise measures used so far are not more appropriate in this context.

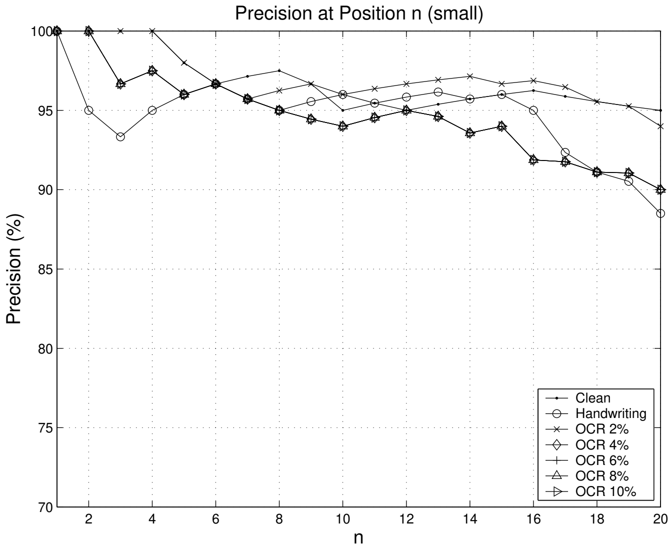
The TER solves some of the problems of the WER, but still it neglects an important aspect of the extraction processes: certain errors result in spurious information that can mislead the categorization process. This can be taken into account by measuring not only how many terms are preserved, but also what fraction of the document transcription they account for. This is done by introducing *Term Recall* and *Term Precision*. The measures used so far (WER and TER) are focused on the substitution errors, i.e. on the words incorrectly extracted. This neglects the noise introduced by other errors: insertions (a word is incorrectly split into two or more words) and deletions (two or more words are merged together and extracted as a single word). These errors are especially frequent in the handwriting and speech recognition cases and might result in a significant difference in the number of terms of the clean and noisy version of the texts. In order to observe the effect of this difference, it is necessary to plot the points corresponding to the different noisy documents on a plan (that we call *Coverage Plan*) where the coordinates are the *Term Recall* (percentage of terms preserved through the extraction process) and the *Term Precision* (percentage of terms in the transcription actually corresponding to terms of the original clean text). The Term Recall can be calculated as:

$$TR = \frac{\sum_i \min(tf(i), tf^*(i))}{\sum_k tf(k)} \tag{12}$$

while the Term Precision is:

$$TP = \frac{\sum_i \min(tf(i), tf^*(i))}{\sum_k tf^*(k)}. \tag{13}$$

Figure 7 shows the positions of the noisy documents extracted through handwriting recognition (only small test set) and OCR with CER between 2 percent and 10 percent (small and full test set). The



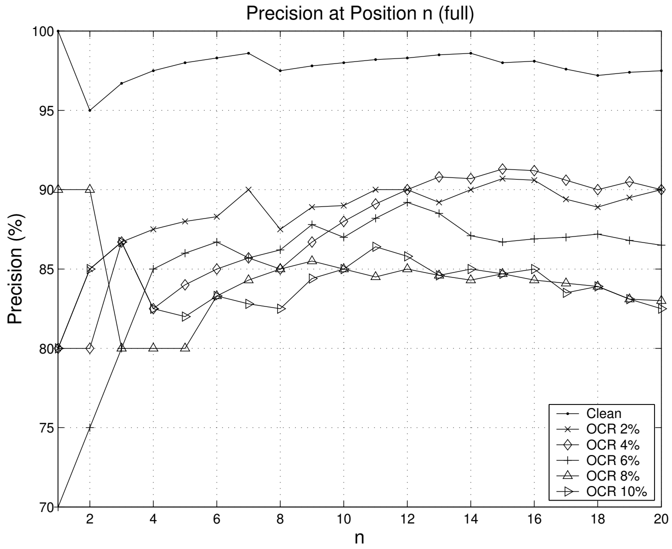




Figure 3: Precision at position $n$. The plots show the Precision at position $n$. Since the minimum number of documents per category is 20, the Precision is computed from position 1 to position 20.

OCR based transcriptions show high precision. This happens because, when one or more characters of a word are uncorrectly recognized, the resulting word very likely does not belong to the dictionary and then it is not counted as a term. The only terms remaining in the transcription are thus terms that were present also in the clean version of the document (for this reason the $TP$ is close to 1). In the handwriting recognition case, the misrecognitions are made at the word level and certain terms are transcribed into other terms. Such terms do not appear in the clean version of the documents and this makes the $TP$ lower. This is an important problem because the spurious terms can potentially mislead the categorization system.

If the transcription contains all the terms of the clean version and no other terms, its position on the coverage plan is the point $(1, 1)$. A transcription plotted in such a position gives the same categorization results as the clean text from where it is extracted. The position of a transcription on the coverage plan can be described using the so called $F_\beta$ measure [3]:

$$F_\beta = \frac{(\beta^2 + 1) \cdot TP \cdot TR}{\beta \cdot TP + TR} \tag{14}$$

where $\beta$ is a parameter allowing one to give more weight to $TP$ or $TR$ (when $\beta = 1$ both measures have the same importance). The average value of $F_\beta$ can be used as a measure of the noise produced by the extraction process. When $\beta < 1$ (more weight is given to $TP$) the values of $F_\beta$ have a good correlation with avgP and BEP over the small test set: an increase of $F_\beta$ corresponds to an improvement in categorization performance. This is no longer true when $\beta \geq 1$ ($TR$ is given the same or more importance than $TP$).

The categorization performance appears thus to be more affected by errors leading to spurious terms. This can explain why the handwritten data (that have, on average, low $TP$) lead to results lower than the OCR simulations. This suggests that the extraction processes must be oriented to achieve good Term Precision rather than high Term Recall.

In the case of the full test set, the results appear to be better correlated with the TER. This is not surprising since the $TP$ is very similar for all OCR simulations ($TP \sim 1$ for all documents) and the only parameter showing significant variability is $TR$. This means that it is not possible (in

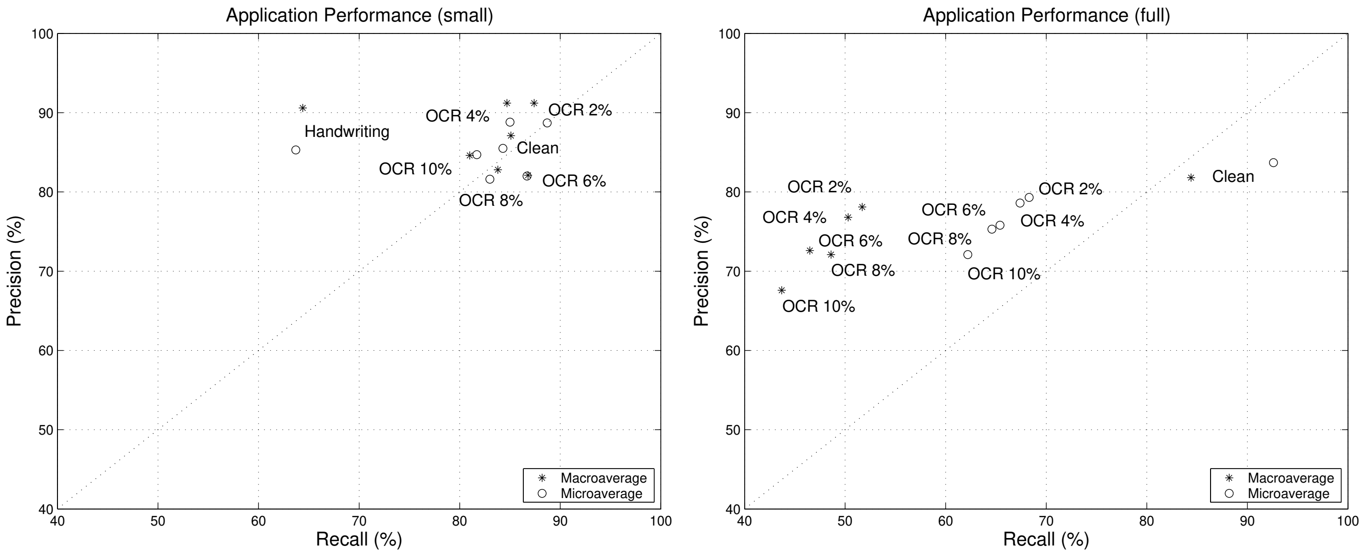



Figure 4: Application Performance. The plot shows the performance over different noisy sources using the thresholds extracted from the test set of the Reuters-21578 database. The diagonal is the line $\pi = \rho$.

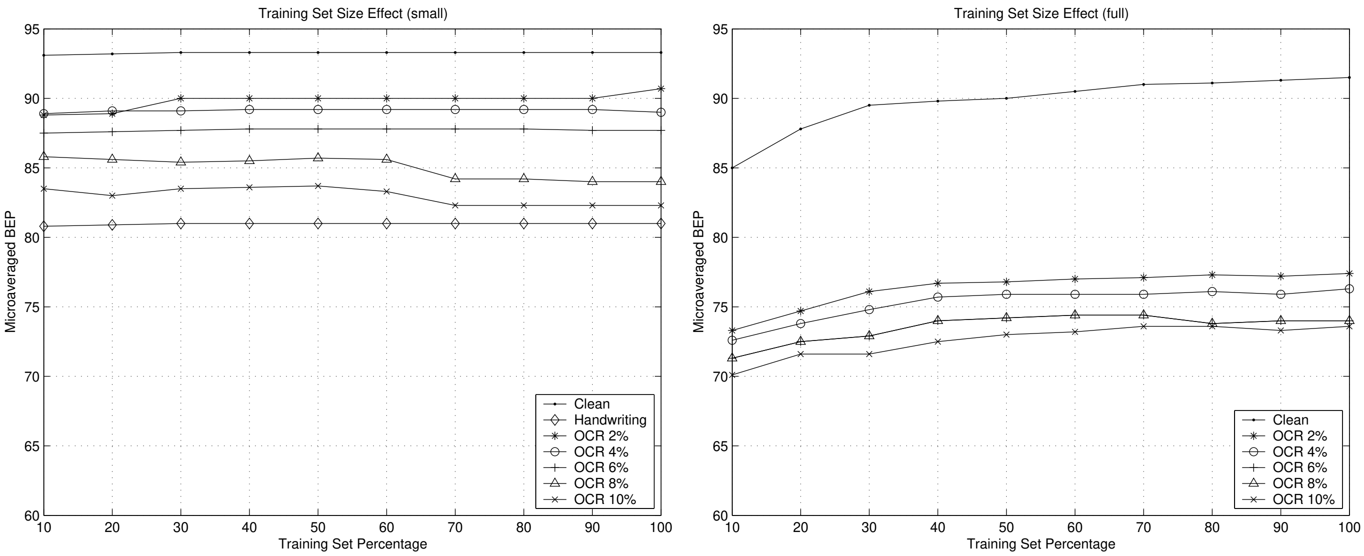



Figure 5: Training Set Size effect. The microaveraged BEP is plotted as a function of the training set fraction used for both the whole training set (left plot) and the subset of 200 documents used in previous experiments (right plot).

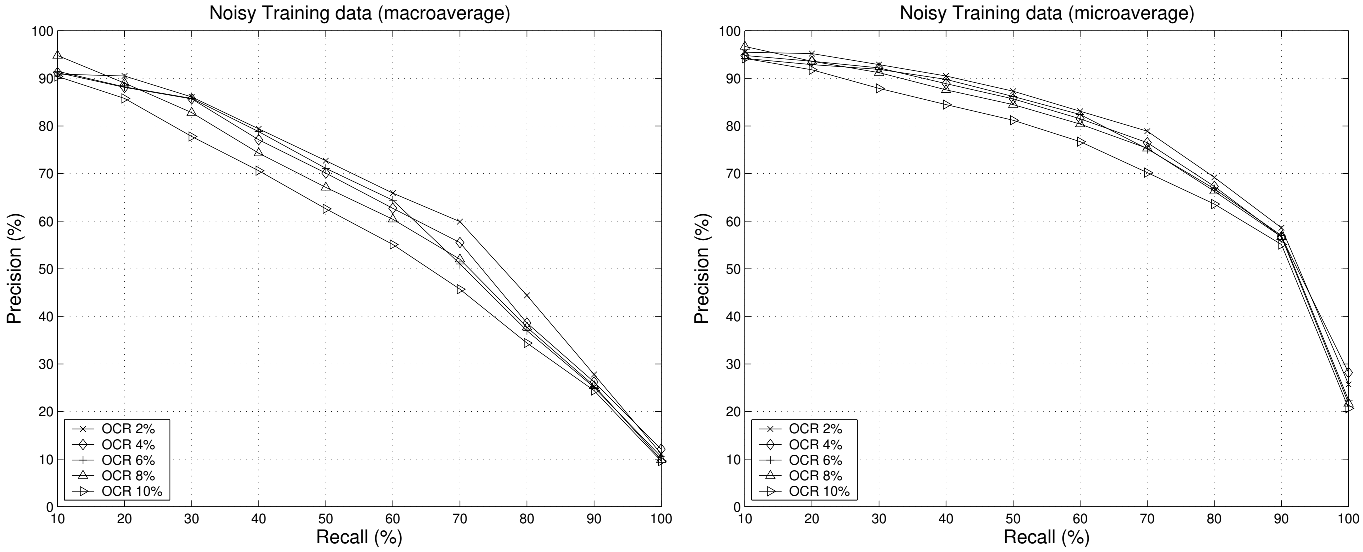



Figure 6: Training on noisy data (full test set). The plots show the Precision vs Recall curves obtained over noisy data when also the training set contains noisy texts. Both macroaverage (left plot) and microaverage (right plot) are shown.

the case of the OCR simulations) to measure the impact of the $TP$ loss and the variablity in the categorization performance is determined essentially by the $TR$. Since $TR = 1-$TER, this results in a better correlation between TER and categorization performance. This can be considered as a further positive effect of high $TP$ transcriptions.

### 4.6 Information Gain Plan

The measures described so far take into account only how many terms are uncorrectly extracted and not which terms are uncorrectly extracted. The loss of certain terms during the the extraction process is more important than the loss of others and this should be considered in the noise measure. Certain terms are more representative of the categories of their documents. A measure of such property is the so-called *Information Gain* (IG) [44]:

$$IG(k) = H(c) - p(k) \cdot H(c|k) - p(\bar{k}) \cdot H(c|\bar{k}) \tag{15}$$

where $IG(k)$ is the IG of term $k$, $p(k)$ is the fraction of documents where $k$ is present, $p(\bar{k})$ is the fraction of documents where term $k$ is absent and $H(.)$ is the entropy. The entropy $H(c)$ is estimated as $-\sum_c p(c) \log p(c)$, where $p(c)$ can be obtained as follows:

$$p(c) = \frac{n(c)}{\sum_{c'} n(c')} \tag{16}$$

where $n(c)$ is the number of times category $c$ is represented. The same defintion is used for $H(c|k)$ and $H(c|\bar{k})$, but on the set of the documents where term $k$ is present and absent respectively.
For most terms, $p(k)$ is close to zero, $p(\bar{k})$ is correspondently close to one and $H(c|\bar{k}) \simeq H(c)$. The resulting IG value is close to zero and the term is assumed to be not representative of the category of the documents it belongs to. In order to have a high IG value, a term must appear in many documents ($p(k) >> 0$) and such documents must belong to few categories ($H(c|k) << H(c)$).

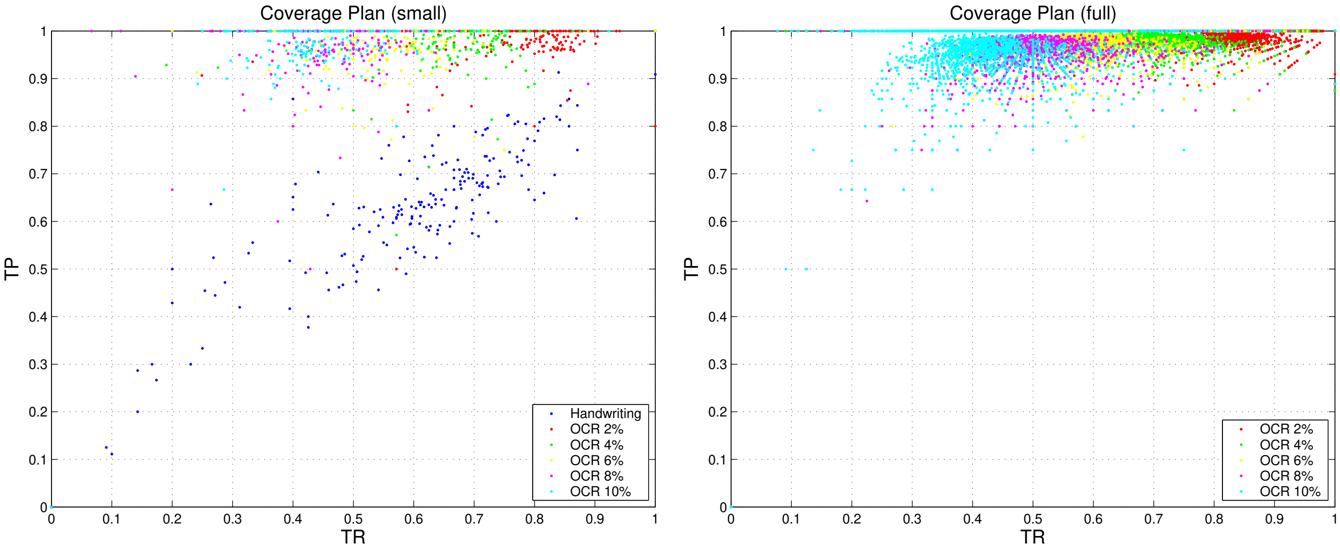



Figure 7: Coverage Plan. The simulated OCR based transcriptions have Term Precision close to 100 percent.

It is possible to think of the total IG contained in a document as the sum of the term frequencies multiplied by the corresponding IG values:

$$IG(d) = \sum_k tf(k) \cdot IG(k) \tag{17}$$

where $d$ is a document. After an extraction process, only a certain fraction of the original IG will be preserved, while word insertions, deletions and substitutions introduce a certain amount of *spurious* IG. This leads to two noise measures that can be called *Information Gain Recall* (IGR):

$$IGR = \frac{\sum_i \min(tf(i), tf^*(i)) \cdot IG(i)}{\sum_k tf(k) \cdot IG(k)} \tag{18}$$

and *Information Gain Precision* (IGP):

$$IGP = \frac{\sum_i \min(tf(i), tf^*(i)) \cdot IG(i)}{\sum_k tf^*(k) \cdot IG(k)} \tag{19}$$

(see section 4.1 for the meaning of symbols). The two measures can be calculated for each document in the data set and this leads to a point on a plan (that we call *Information Gain Plan*) where the coordinates are the values of $IGR$ and $IGP$ (see Figure 8). While in OCR simulations $IGP$ is always close to 1, in the handwriting case, important $IG$ fractions are spurious, i.e. they are due to terms that are the result of a misrecognition. When a document contains terms that have high IG with respect to a certain category $c$, the categorization system tends to identify it as belonging to $c$, thus the presence of spurious high IG terms can mislead the system.
The $F_\beta$ measure can be used as in the case of the coverage plan (see previous section) to measure the noise of the transcriptions. Also in this case, the better correlation with the categorization results over the small test set is obtained for $\beta < 1$. This confirms that extraction processes introducing few spurious terms lead to better categorization performances. The $IGP$ should thus be privileged with respect to the $IGR$.

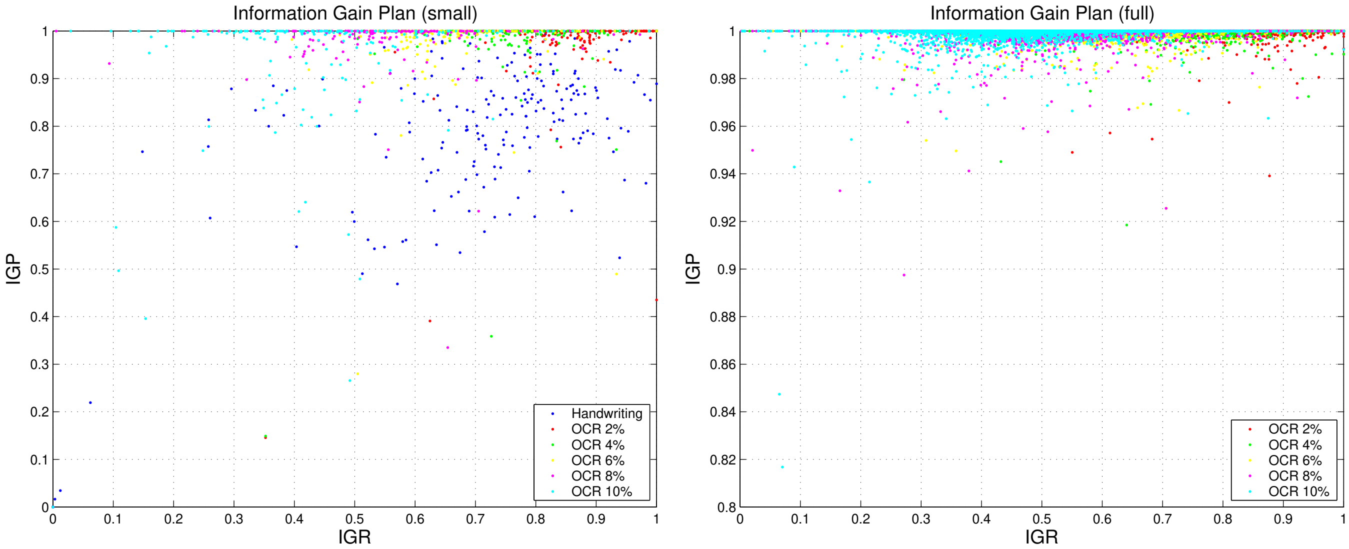



Figure 8: IG Plan. Each point corresponds to a document. The impact of the presence of spurious terms in the case of the handwriting is evident from the position of the documents.

In the case of the full test set, the same considerations made at the end of the previous subsection apply. Also in this case, the only parameter showing significant variability is $IGR$ and it is not possible to evaluate the impact of the $IGP$ loss. On the other hand, high $IGP$ seems to lead to better correlation between categorization performance and TER.

# 5 Conclusions

Several media can be converted into texts through a process producing noise, i.e. word insertions, deletions and substitutions with respect to the actual clean text contained in the source. This work shows the effect of the noise over the performance of a Text Categorization system. The categorization effectiveness has first been measured over the clean version of a dataset, then over several noisy versions of the same data. The noisy versions have been obtained with two methods: the first is to manually write the documents and then to transcribe them with an offline handwriting recognition system, the second is to simulate an OCR recognition by randomly changing a certain percentage of characters. The OCR simulation allowed the use of different Character Error Rates (from 2 percent to 10 percent). The noise produced by the two different sources has different characteristics and it is representative of different extraction processes. The noise produced by the handwriting recognizer is similar to the noise produced by speech recognizers. The noise produced by OCR is similar to the one produced by systems detecting and recognizing texts in images and videos or manual typing. The two sources of noise have been selected in order to be representative of a wider spectrum of situations.
The results show that, for Recall values up to 60-70 percent depending on the sources, the categorization system is robust to noise even when the Term Error Rate is higher than 40 percent. The categorization performance has been measured by using a wide spectrum of metrics focusing on the needs of different potential users. Precision vs Recall curves give a general idea of the TC system behaviour from low to full Recall. It has been shown that, for low Recall values, the noise has almost no effect and the gap between different sources is not statistically significant. A larger difference can be observed for higher Recall values.
The Precision at position $n$ (measured for $1 \leq n \leq 20$) shows that all systems perform at the same

level in the top ranking positions. This means that higher scoring documents (on average longer than others and with a higher degree of redundancy) are moderately affected by the noise and they remain at the top ranking positions even for high levels of mismatch with respect to their clean version.
A specific experimental setup has been used to obtain an application oriented evaluation. A set of thresholds has been set in order to achieve BEP precision. The same thresholds have been used over the noisy texts showing that, while the system was able to keep the same Precision level of the evaluation set for all the noise sources, there was a significant Recall loss in the case of handwriting and OCR simulations over the full test set.
For any metric used, the results from the handwritten data appear to be lower than those obtained from OCR simulations. The main difference is that OCR transcriptions are more *precise*, i.e. few spurious terms are introduced during the extraction process. This seems to suggest that to loose terms produces less problems than transcribing them into other, potentially misleading, terms. This can be important in the development of extraction processes oriented towards the categorization: the use of mechanisms able to reject terms recognized with a confidence level too low could be applied.
The analysis of the results shows that Word Error Rate and Term Error Rate provide only a partial insight about the extraction process. Both measures account for the similarity between a noisy text and the clean text from where it is extracted, but do not consider the impact of different errors on the final categorization performance. Different measures have been proposed that better explain the final categorization results. It has been shown that the insertion of spurious terms (frequent in the case of handwritten data) has higher influence than the simple loss of terms. This explains why the categorization performance for handwritten data is lower than in the other cases and suggests that extraction processes should aim at high Term or IG Precision. This is important because an optimization of the extraction processes in terms of WER or TER leads to high Term or IG Recall that does not necessarily lead to good categorization results.
The possibility of categorizing noisy texts gives the possibility of organizing and managing databases of sources different from digital text. This is important when more and more databases of images, speech recordings, videos, etc. are collected. Our experiments showed that it is possible to train category models over clean material (relatively easy to collect and manage) and then apply them over different kinds of noisy texts. This is an important advantage because it gives the possibility of having a single system for different databases (each containing a different source) or databases containing different kinds of sources. On the other hand, the results appear to be close to those obtained over clean texts only for low Recall values.
Our experiments show that it is possible to extend SVM based categorization techniques from clean to noisy texts (at least in the cases we have considered). In a future work, we plan to use the information that can be extracted from sources together with text (in speech recordings it is possible to extract speaker identity, dialogue annotations, emotional states, etc., in videos it is possible to process images, motion, audio, etc.). This will not only help to fill the performance gap between clean and noisy texts, but also to explore categorization possibilities that cannot be exploited for purely textual documents.

**Acknowledgements** The author is indebted with the anonymous reviewers for their valuable comments. S.Bengio, F.Camastra and J.M.Odobez are thanked for useful discussions. D.Grangier is gratefully acknowledged for technical support. This work is supported by the Swiss National Science Foundation through the National Center of Competence in Research on Interactive Multimodal Information Management (IM2).

# References

[1] D. Abberley, S. Renals, D. Ellis, and T. Robinson. The THISL SDR system at TREC-8. In *Proceedings of $8^{th}$ Text Retrieval Conference*, pages 699–706, 1999.

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

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

[4] T. Bayer, U. Kressel, H. Mogg-Schneider, and I. Renz. Categorizing paper documents. *Computer Vision and Image Understanding*, 70(3):299–306, 1998.

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

[6] D. Chen, J.M. Odobez, and H. Bourlard. Text detection and recognition in images and videos. *Pattern Recognition*, 37(3):595–609, 2004.

[7] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[8] B. Croft, S.M. Harding, K. Taghva, and J. Borsack. An evaluation of information retrieval accuracy with simulated OCR output. In *Proceedings of Symposium on Document Analysis and Information Retrieval*, pages 115–126, 1994.

[9] D. Doermann. The indexing and retrieval of document images: a survey. *Computer Vision and Image Understanding*, 70(3):287–298, 1998.

[10] D. Doermann and S. Yao. Generating synthetic data for text analysis systems. In *Proceedings of Symposium on Document Analysis and Information Retrieval*, pages 449–467, 1995.

[11] C. Fox. Lexical analysis and stoplists. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval. Data Structures and Algorithms*, pages 102–130. Prentice Hall, 1992.

[12] W.B. Frakes. Stemming algorithms. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval. Data Structures and Algorithms*, pages 131–160. Prentice Hall, 1992.

[13] M. Franz, J.S. McCarley, and R.T. Ward. Ad hoc, cross-language and spoken document information retrieval at IBM. In *Proceedings of $8^{th}$ Text Retrieval Conference*, pages 391–398, 1999.

[14] J.S. Garofolo, C.G.P. Auzanne, and E.M. Voorhees. The TREC spoken document retrieval track: A success story. In *Proceedings of $8^{th}$ Text Retrieval Conference*, pages 107–129, 1999.

[15] J.L. Gauvain, Y. de Kercadio, L. Lamel, and G. Adda. The LIMSI SDR system for TREC-8. In *Proceedings of $8^{th}$ Text Retrieval Conference*, pages 475–482, 1999.

[16] D. Graff, C. Cieri, S. Strassel, and N. Martey. The TDT-3 text and speech corpus. In *Proceedings of Topic Detection and Tracking Workshop*, 2000.

[17] B. Han, R. Nagarajan, R. Srihari, and M. Srikanth. TREC-8 experiments at SUNY at Buffalo. In *Proceedings of $8^{th}$ Text Retrieval Conference*, pages 591–596, 1999.

[18] R. Hoch. Using IR techniques for text classification in document analysis. In *Proceedings of $17^{th}$ ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 31–40, 1994.

[19] A.K. Jain, P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

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

[21] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of European Conference on Machine Learning*, pages 137–142, 1998.

[22] T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods*. MIT Press, 1999.

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

[24] S.E. Johnson, P. Jourlin, K. Spärck-Jones, and P.C. Woodland. Spoken document retrieval for TREC-8 at Cambridge University. In *Proceedings of $8^{th}$ Text Retrieval Conference*, pages 197–206, 1999.

[25] K. Koumpis and S. Renals. Evaluation of extractive voicemail summarization. In *Proceedings of ISCA Workshop pn Multilingual Spoken Document Retrieval*, pages 19–24, 2003.

[26] W. Kraaij, R. Pohlmann, and D. Hiemstra. Twenty-one at TREC-8 using language technology for information retrieval. In *Proceedings of $8^{th}$ Text Retrieval Conference*, pages 285–300, 1999.

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

[28] D. Lopresti and J. Zhou. Retrieval strategies for noisy text. In *Proceedings of Symposium on Document Analysis and Information Retrieval*, pages 255–270, 1996.

[29] D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. Named entity extraction from noisy input: speech and OCR. In *Proceedings of $6^{th}$ Conference on Applied Natural Language Processing*, pages 316–324, 2000.

[30] M. Ohta, A. Takasu, and J. Adachi. Retrieval methods for english text with misrecognized OCR characters. In *Proceedings of IEEE International Conference on Document Analysis and Recognition*, pages 950–956, 1997.

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

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

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

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

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

[36] A. Singhal, S. Abney, M. Bacchiani, M. Collins, D. Hindle, and F. Pereira. AT&T at TREC-8. In *Proceedings of $8^{th}$ Text Retrieval Conference*, pages 317–330, 1999.

[37] K. Taghva, J. Borsack, and A. Condit. Expert system for automatically correcting OCR output. In *Proceedings of SPIE-Document Recognition*, pages 270–278, 1994.

[38] K. Taghva, T. Narkter, J. Borsack, Lumos. S., A. Condit, and R. Young. Evaluating text categorization in the presence of OCR errors. In *Proceedings of IS&T SPIE 2001 International Symposium on Electronic Imaging Science and Technology*, pages 68–74, 2001.

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

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

[41] A. Vinciarelli and J. Luettin. A new normalization technique for cursive handwritten words. *Pattern Recognition Letters*, 22(9):1043–1050, 2001.

[42] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svm's. In *Advances in Neural Information Processing Systems 13*, pages 668–674, 2000.

[43] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

[44] Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of $14^{th}$ International Conference on Machine Learning*, pages 412–420, 1997.

[45] C. Zhai, X. Tong, N. Milic-Frailing, and D.A. Evans. OCR correction and query expansion for retrieval on OCR data - CLARIT TREC-5 confusion track report. In *Proceedings of $5^{th}$ Text Retrieval Conference*, pages 341–344, 1996.

[46] G. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.