Effect of Recognition Errors on Information Retrieval Performance

Alessandro Vinciarelli [a]

IDIAP–RR 04-08

February 2004

submitted for publication

Dalle Molle Institute
for Perceptual Artificial
Intelligence • P.O.Box 592 •
Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP

IDIAP Research Report 04-08

Effect of Recognition Errors on Information Retrieval Performance

Alessandro Vinciarelli

February 2004

submitted for publication

Abstract. This work shows experiments on the retrieval of handwritten documents. The performance of the same state-of-the-art Information Retrieval system is compared when dealing with manual (no errors) and automatic (Word Error Rate around 50%) transcriptions of the same handwritten texts. The results show that, in terms of the user effort required to find the desired items, the performance degradation due to the recognition errors can be considered acceptable.

1 Introduction

Information Retrieval (IR) techniques developed for digital texts can be applied to transcriptions of handwritten documents obtained with handwriting recognition systems. This paper evaluates the effect of the recognition errors by comparing the performance of the same IR system over both manual and automatic transcriptions of the documents belonging to a dataset. The manual transcriptions contain no errors and can be thought of as the *clean* version of the data. The automatic transcriptions are affected by a Word Error Rate (WER) of around 50% and can be thought of as the *noisy* versions of the same documents.

The retrieval task performed in our experiments is the *Known Item Search* (KIS): each query is supposed to retrieve a single specific document and no other documents are considered relevant to it. Two sets of queries have been used: the first contains the optimal queries obtained, for each document, with the Rocchio formula (see section 4). The second contains 40 queries produced by a user. The handwriting recognition system used in our experiments is based on continuous density Hidden Markov Models and Statistical Language Models (bigrams) [7]. The retrieval tasks have been performed with a state-of-the-art IR system based on the Vector Space Model (VSM) [2].

The output of the system is a ranking of the documents according to their matching with the query. In KIS, the retrieval process is considered successful when the Known Item is at the first position of the ranking (i.e. the users must check only one document to find what they are searching for). The results show that, when passing from clean to noisy data, around 15-20% of the documents ranking first fall to lower positions. On the other hand, the difference is reduced to less than 5% at the fifth position (see section 4). On average, when using noisy data, 85% of the Known Items can be found by checking less than 5 documents (90% of the cases with clean data). This means that the additional effort required to the user because of the noise can be considered acceptable.

Relatively few works have been dedicated to the retrieval of handwritten documents. In most of the cases [3][4][6], the approaches are aimed to the detection of specific keywords in the handwritten texts, while our work applies state-of-the-art IR techniques. Moreover, no comparison has been made, to our knowledge, between the performances over clean and noisy text versions of the same retrieval system.

The rest of this paper is organized as follows: section 2 describes our handwriting recognition system, section 3 presents our IR approach, section 4 shows experiments and results and section 5 draws some conclusions.

2 Handwriting Recognition

A full description of the offline handwriting recognition system used in this work can be found in [7]. The documents are first segmented into lines that are recognized one by one. Each line image is normalized with the technique described in [8], then it is converted into a sequence of feature vectors $O = (o_1, \ldots, o_L)$ through a sliding window approach: a fixed width window shifts column by column from left to right and, at each position, a feature vector is extracted (see [7] for details about the feature extraction process).

The recognition approach is based on continuous density Hidden Markov Models (HMM) and Statistical Language Models (SLM) and corresponds to finding the word sequence $\hat{W} = \{w_1, w_2, \ldots, w_M\}$ maximizing the a posteriori probability: $\hat{W} = \arg\max_W p(W|O)$. By applying the Bayes Theorem and by taking into account the fact that O is constant during the recognition, the last equation can be rewritten as follows:

$$\hat{W} = \arg\max_W p(O|W) \cdot p(W) \tag{1}$$

where it is possible to see the role played by the two available sources of information (handwritten data and language). The term $p(O|W)$ is estimated with continuous density HMMs and it models the handwritten data. The term $p(W)$ can be interpreted as the a priori probability of sentence W being written and it is estimated with Statistical Language Models (SLM). The role of the SLM is to constrain the search space by giving probability significantly different from zero to as few sentences as possible. The SLM used in this work is a bigram model and $p(W)$ is expressed as follows:

$$p(W) = \prod_{i=1}^{M} p(w_i|w_{i-2}, w_{i-1}) \tag{2}$$

where M is the number of words in W. The bigram is an example of a more general class of models called N-grams. In practice, N is never higher than three and we used the bigrams because (as shown in [7]) the low number of words per line (the average is around 10) makes it difficult for the trigrams to significantly outperform bigrams.

3 Information Retrieval

The IR system used in this work is based on a state-of-the-art approach commonly applied for digital texts and no modifications have been made to deal with the recognition errors. The texts are first *preprocessed*: all non-alphabetic characters are removed (digits, punctuation marks, etc.) resulting in a stream of words that is given as input to the *normalization*. This last is supposed to remove the variability unuseful to the retrieval process and it is performed through two steps: *stopping* and *stemming*. The first step is the removal of all words supposed to be poorly related to the document content (articles, prepositions, words of common use like *to have* or *to be*, etc.). Stopping results, on average, in the elimination of around 50% of the words in a database. The stemming is the replacement of all morphological variants of the same word (e.g. *connection*, *connected*, *connecting*) with their stem (*connect*). On average, the stemming reduces the size of the dictionary (the list of unique terms appearing in the database) of around 30%.

After the normalization, the documents are available as streams of *terms*. This is not a suitable form for the retrieval process and the texts must be given a different representation through an *indexing* procedure. Our system is based on the Vector Space Model (VSM) [2]: the documents are represented as vectors where each component accounts for a term. This allows one to represent the database with the so-called *term by document* matrix where each column corresponds to a document and each row corresponds to a term. The generic element $A(i,j)$ can be written as a product:

$$A(i,j) = L(i,j) \cdot G(i) \tag{3}$$

where $L(i,j)$ is a local weight taking into accout only information contained in document j and $G(i)$ is a global weight using information extracted from the whole database (see [5] for a survey on weighting functions).

In this work, several weighting schemes are used: the first, called *binary*, has $G(i) = 1$ and $L(i,j) = 1$ when term i is present in document j (0 otherwise). The second is referred to as *tf* and has $G(i) = 1$ and $L(i,j) = tf(i,j)$, where $tf(i,j)$ is the *term frequency*, i.e. the number of times term i appears in document j. The third is the *tf·idf* weighting scheme, where $L(i,j)$ is the term frequency $tf(i,j)$, and $G(i)$ is the *inverse document frequency idf*(i):

$$idf(i) = \log\left(\frac{N}{N_i}\right) \tag{4}$$

where N is the total number of documents in the database and N_i is the number of documents containing term i. The fourth is associated with the Okapi formula and it is described below.

The previous steps are performed once for a given database and represent the *offline* part of the IR process. The *online* part consists of the matching between the queries (short texts expressing user information needs) and the database documents. The matching is performed through a measure giving as output the *Retrieval Status Value* (RSV) of a document d given a query q. Several matching measures are available in the literature. In this work we used the cosine between query and document vectors:

$$RSV(q,d) = \frac{\mathbf{q} \cdot \mathbf{d}}{||\mathbf{q}|| \cdot ||\mathbf{d}||} \tag{5}$$

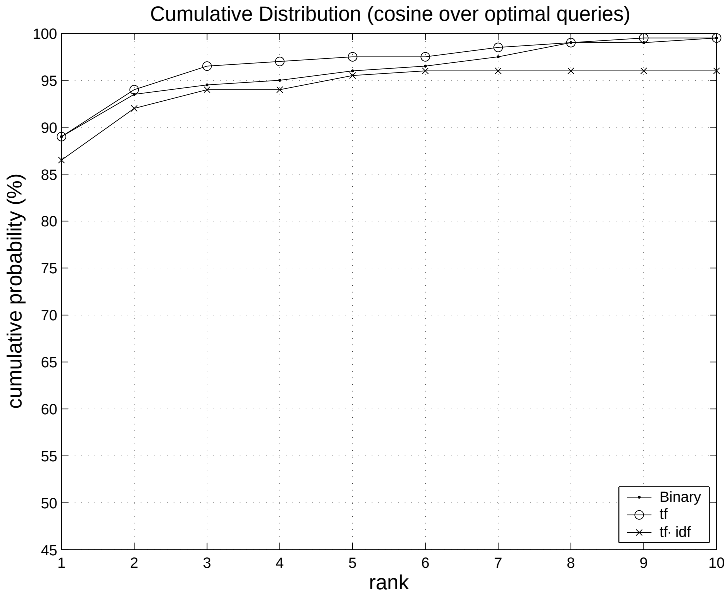

Figure 1: Retrieval performance over optimal queries using the cosine. The clean texts results are not reported because the clean texts are always at the top of the ranking. Three document representations are used.

as well as the Okapi formula:

$$RSV(q,d) = \sum_{i:t_i \in \{q_l\}} \frac{(k+1) \cdot tf(i,d) \cdot idf(i)}{k \cdot [1 - b + b \cdot ndl(d)] + tf(i,d)} \tag{6}$$

where the sum is performed over the terms belonging to the query, k and b are hyperparameters, and $ndl(d)$ is the normalized length of document d (the actual length divided by the average document length in the database). Each addend of the sum in the Okapi formula can be thought of as a component of the document vector $\mathbf{d}$. Such representation is peculiar of the use of the Okapi formula. The RSV is used as a criterion, given a query, to rank all the documents of the database. The relevant documents (i.e. the texts satisfying the information need expressed with the query) are expected to be at the top ranking positions.

4 Experiments and results

This section describes the KIS experiments performed in this work. Two different sets of queries have been used and the results show that, even if the WER is close to 50%, the recognition errors make necessary only a moderate additional user effort to find the items they are looking for. The next subsections are organized as follows: section 4.1 describes the data, and section 4.2 presents the retrieval results.

4.1 The data

The experiments performed in this work are based on the Reuters-21578 database [1], a well known and widely applied Text Categorization benchmark. The collection has been split into training (9603 documents) and test set (3299 documents) following the Modapté protocol [1]. A set of 250 documents

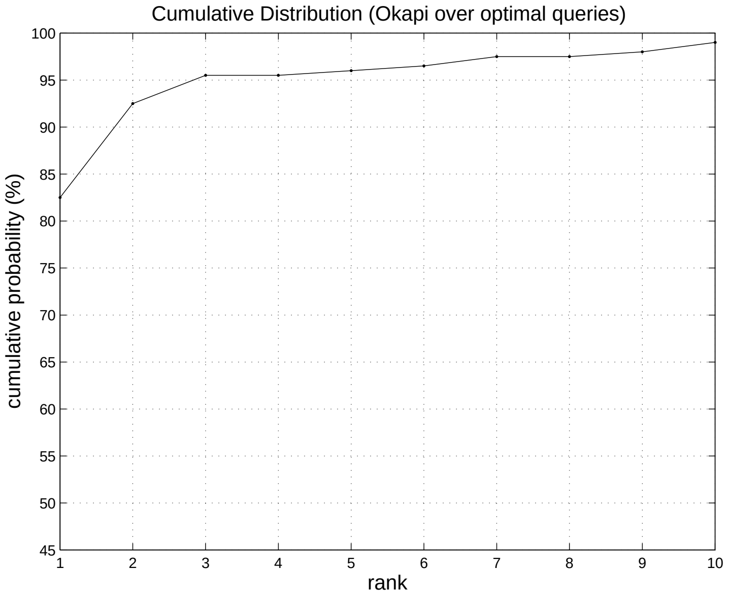

Figure 2: Retrieval performance over optimal queries using the Okapi formula. The clean texts results are not reported because the clean texts are always at the top of the ranking. Only one document representation can be used.

has been randomly selected from the test set and it has been manually written by a single person. This resulted in a collection of handwritten documents that has been partitioned into training (50 documents) and test set (200 documents). The total number of words is 31,484 (5,789 in the training set and 25,695 in the test set).

The handwriting recognition system and the language model have been trained using the techniques shown in [7]. The resulting recognizer has a WER of 44.2% on the test set. The substitution rate is 35.6%, the insertion rate is 0.3% and the deletion rate is 8.6% (the lexicon size is 20,000). The original 200 texts extracted from the Reuters test set are the clean data, while their transcriptions affected by recognition errors are the noisy version of the same data.

4.2 Information Retrieval Results

The aim of this work is to evaluate the effect of the recognition errors on the performance of an IR system. For this reason we measure the performance of the same IR process in a KIS retrieval task using both clean and noisy versions of the same data. The KIS task is based on two different sets of queries. The first query set is obtained as follows: for each clean document j we compute the optimal query $\hat{\mathbf{q}}(j)$ that should be submitted to the system in order to retrieve it. By optimal, it is meant the query that gives the document j an RSV higher than any other document in the data set. In other words, when the documents are ranked according to such query, $\mathbf{d}_j$ will occupy the first position. The optimal query can be obtained with the Rocchio formula [2]:

$$\hat{\mathbf{q}} = \frac{1}{|R|} \sum_{\forall d_j \in R} \mathbf{d}_j - \frac{1}{N - |R|} \sum_{\forall d_j \notin R} \mathbf{d}_j \qquad (7)$$

where R is the set of the documents relevant to $\hat{\mathbf{q}}$ and N is the total number of documents in the database. The optimal query $\hat{\mathbf{q}}(j)$ for document $\mathbf{d}_j$ is found when $R = \{\mathbf{d}_j\}$. The optimal queries can be extracted from the clean database and then applied to both clean and noisy versions of the data.

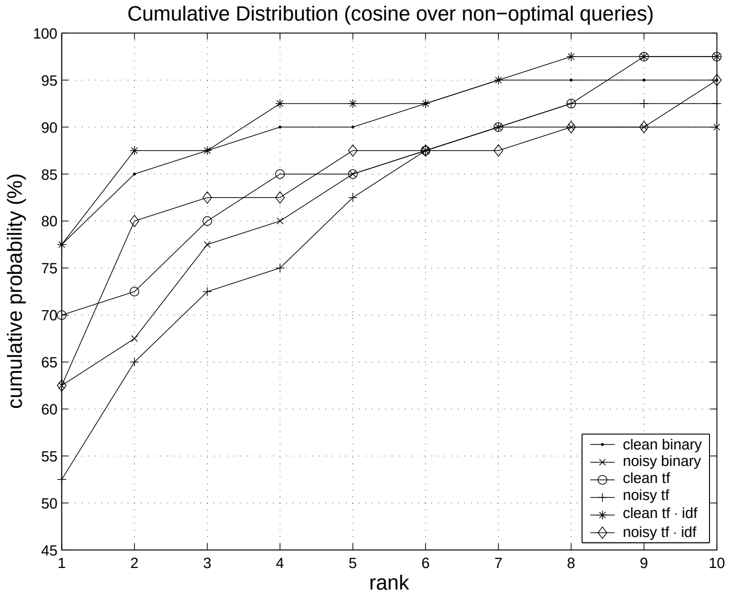

Figure 3: Retrieval performance over non-optimal queries using the cosine. Three document representations are used.

When the $\hat{\mathbf{q}}(j)$ queries are used on the clean data, the document j is always at the first position of the RSV ranking (see section 3). When they are used on the noisy data, some documents, because of the recognition errors, occupy lower positions. The distribution of the relevant documents percentage as a function of the position can thus be used as a measure of the noise effect.

The results are shown in Figure 1 (for cosine) and Figure 2 (for Okapi formula) for different document representations. The plots show that at least 80% of the documents (89% in the best case) rank first also when they are affected by noise, thus, in most of the cases, the retrieval process is robust with respect to the recognition errors. This is furtherly confirmed by the fact that for both matching measures (independently of the representation) less than 5% of the documents fall after the fifth position.

Further results have been obtained by using a set of 40 queries written by a user. Each query is supposed to retrieve a Known Item and the evaluation is performed as in the previous experiments.

The results are shown in Figure 3 (for cosine) and Figure 4 (for Okapi formula): also in this case, there is a loss of around 15% at the first position, but after the five top ranking texts, the difference is less than 5%. This seems to confirm the results obtained with optimal queries and suggests that the IR process is substantially robust with respect to noise: the additional user effort required to retrieve the Known Items in presence of noise can be considered acceptable.

5 Conclusions and Future Works

This paper presented experiments on the retrieval of handwritten documents. The effect of the recognition errors on the retrieval performance has been measured by comparing the results of the same IR system when dealing with both manual and automatic transcriptions of handwritten documents. To our knowledge, previous works concentrated on detecting keywords in the handwritten transcriptions [3][4][6] rather than on actually applying IR techniques. Moreover no comparison has been made, to our knowledge, between IR performances over clean and noisy versions of the same texts.

The results show that in 85% of the cases (when using noisy data), the user can find the desired

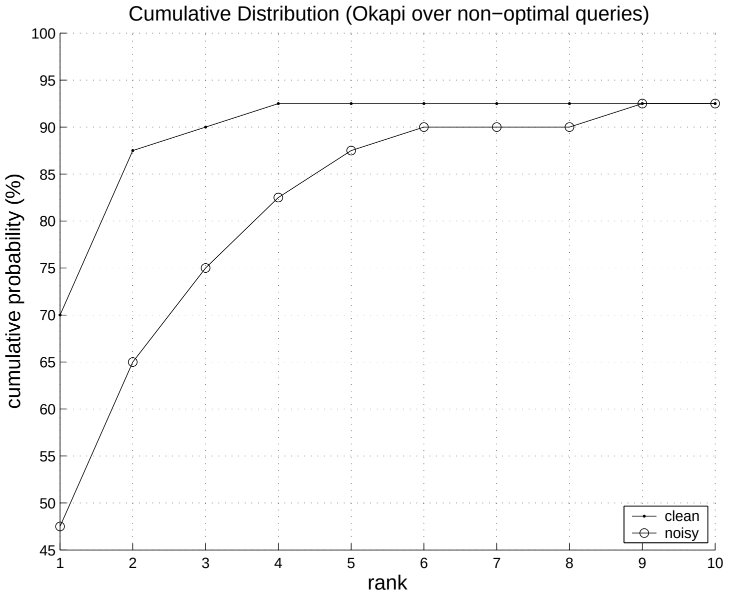

Figure 4: Retrieval performance over non-optimal queries using the Okapi formula. Only one representation can be used.

item by browsing less than five documents of the database. This represents an acceptable loss with respect to the clean text case, where the same result is achieved in 90% of the cases. This suggests that the user effort required to identify the searched item is only slightly increased by the presence of recognition errors. The experiments have been performed using both optimal and non-optimal queries (see section 4) and the results are similar in both cases.

The above results suggest that the retrieval process is robust with respect to the recognition errors even if the WER is close to 50%. As a future work we plan to perform more retrieval experiments (using queries with more than one relevant document) and to increase the size of our database in order to further support such conclusion.

Acknowledgement The author wishes to thank D.Grangier for technical contribution. 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] C. Apté, F. Damerau, and S.M. Weiss. Automated learning decision rules for text categorization. *ACM Trans. on Inf. Syst.*, 12(3):233–251, 1994.

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

[3] T.M Rath and R. Manmatha. Features for word spotting in historical manuscripts. In *Proceedings of IEEE ICDAR*, pages 218–222, 2003.

[4] G. Russell, M.P. Perrone, and Y.M. Chee. Handwritten document retrieval. In *Proceedings of IWFHR*, pages 233–238, 2002.

[5] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Proc. and Mgmt.*, 24:513–523, 1988.

[6] C.I. Tomai, B. Zhang, and V. Govindaraju. Transcript mapping for historic handwritten document images. In *Proceedings of IEEE ICDAR*, pages 413–418, 2002.

[7] A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of large vocabulary cursive handwritten text. *accepted for publication by IEEE Trans. on PAMI*, 2004.

[8] A. Vinciarelli and J. Luettin. A new normalization technique for cusive handwritten words. *Patt. Rec. Lett.*, 22(9):1043–1050, 2001.