IDIAP RESEARCH REPORT

Effect of Recognition Errors on Text Clustering

David Grangier [1] Alessandro Vinciarelli [2]

IDIAP–RR 04-82

September 2004

1 IDIAP, CP 592, 1920 Martigny, Switzerland, grangier@idiap.ch
2 IDIAP, CP 592, 1920 Martigny, Switzerland, vincia@idiap.ch

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 04-82

Effect of Recognition Errors on Text Clustering

David Grangier Alessandro Vinciarelli

September 2004

Abstract. This paper presents clustering experiments performed over noisy texts (i.e. texts that have been extracted through an automatic process like character or speech recognition). The effect of recognition errors is investigated by comparing clustering results performed over both clean (manually typed data) and noisy (automatic speech transcriptions) versions of the same speech recording corpus.

1 Introduction

Recent advances in digital technologies have led to the availability of large multimedia databases: in several application domains, images (e.g. photo archives), video recordings (e.g. video conferencing) and audio data (e.g. radio broadcast news) are collected and techniques like retrieval [1, 12, 25], browsing [7] and categorization [31] are thus essential to take full advantage of such collections. These technologies rely on different types of information that can be automatically extracted from the data (e.g. speaker identity in speech recordings, object detection in video, etc). Among this information, texts (e.g. speech transcripts or subtitle texts) are especially appropriate for indexing as they are related to the semantic content of the original data (e.g. arguments debated in a video conference, topics discussed in a radio recording). Moreover, users generally find it more natural to interact with a system through text queries rather than providing images or audio examples [11]. Also, the use of texts allows one to benefit from previous works on digital text databases [2].
However, the textual data automatically extracted from a multimedia source have an important difference with respect to manually typed texts: the extraction processes (e.g. Automatic Speech Recognition, *ASR*, or Optical Character Recognition, *OCR*) introduce *noise*. This means that some words are inserted, deleted or substituted with respect to the *clean* text actually contained in the original source [6][15]. This noise can significantly affect the data (e.g. $\sim 30\%$ of misrecognized words is not uncommon in ASR transcriptions) and the techniques developed for clean digital texts could suffer from a performance loss when applied to noisy data. However, this degradation has shown to be acceptable in the case of Information Retrieval (IR) and text categorization [12, 25, 31].
This work focuses on the effect of noise on document clustering. Clustering identifies groups of similar documents in a corpus and it has shown to be helpful to retrieval and browsing of clean texts [3, 5, 13]. Therefore, it could likewise be useful to similar tasks on noisy data, if it is robust to noise (i.e. if it is possible to group noisy documents with the same topic together with performances comparable to those obtained on clean data). To our knowledge, the effect of noise on clustering techniques has not been investigated to date. The previous results obtained on noisy text retrieval [1, 12, 25] are encouraging, but retrieval and clustering are different problems: retrieval only relies on a few terms (query terms and other related terms if query expansion is used [2]) to determine whether a document is relevant or not, whereas clustering is based on document comparisons in which all terms are used. This means that all recognition errors can potentially degrade the clustering results. Moreover, other techniques have been shown to be more sensitive to noise than retrieval (e.g. summarization [16]) and it is thus an open issue whether clustering techniques are robust to noise.
In order to evaluate the degradation introduced by noise, the same clustering experiments are performed on both clean and noisy versions of the TDT2 corpus (which consists of $\sim 25,000$ spoken documents recorded from broadcast news [8]). Three clustering techniques that differ in the way they compute similarity between documents are evaluated: the first method compares documents according to a geometrical criterion (i.e. vector inner product), the second relies on the number of shared terms between documents and the third is based on comparisons of Statistical Language Models (SLM).
In order to evaluate clustering effectiveness (i.e. its ability to group documents with the same topic in the same cluster while scattering documents with different topics in different clusters), we introduce an evaluation methodology that relies on IR ad-hoc queries (TREC SDR queries [10]). For any query, the documents that are relevant to it are considered to have the same topic and should hence be grouped together in clusters that contains few non-relevant documents (i.e. documents with different topics). According to this evaluation method (section 4), we observe that the degradation due to noise is limited even with a high amount of noise ($\sim 30\%$ Word Error Rate in our data).
The rest of this paper is organized as follows: section 2 gives an overview of previous work on document clustering, section 3 presents the clustering techniques we evaluated, section 4 explains the evaluation methodology we used, section 5 presents the experiments we performed and the results we obtained, section 6 draws some conclusions.

2 Previous Work on Document Clustering

Document clustering identifies groups of similar documents in a corpus. Each group is called a *cluster* and can be assigned a representative called a *centroid*. If most of the documents about the same topic are grouped in a single cluster that contains few other documents, clustering can be useful to several application domains: in previous works, it has been used to improve efficiency of IR systems [5, 30, 32], to ameliorate retrieval effectiveness [18, 29, 30], to allow browsing [9, 13] and to distribute documents across a network [3, 34]. To our knowledge, document clustering has so far only been applied to clean digital texts. If the same clustering techniques can be applied to texts extracted from different media (e.g. noisy texts extracted from audio or video recordings), this would allow one to take advantage of the same applications for these media. This requires the verification that clustering techniques, which have been developed for clean texts, are robust to noise.
Different methods can be used to cluster a document corpus. They can be divided into two categories based on the cluster structure they lead to [14, 24]: partition (e.g. k-means) and hierarchical (e.g. single link clustering, Ward's method). Partition clustering splits the corpus into disjoint subsets. Hierarchical clustering produces a tree which can be examined in two directions: from top, one cluster is successively split into two parts until it is no longer possible (divisive view) and from bottom, *ndoc* clusters (with one document per cluster, *ndoc* being the corpus size) are successively joined until there is only one cluster (agglomerative view) [26]. The choice of a solution depends on the application (e.g. for distributed retrieval, the partition clustering is preferred while for efficiency improvement, the hierarchical method is generally used). Another key aspect of clustering techniques is the similarity measure used to determine which documents should be grouped in the same cluster. This measure should be high when comparing documents about the same topic and low otherwise. Two kinds of measures have been proposed in the literature: measures based on geometrical criteria [2, 21, 27] (e.g. inner product of document vectors in the $tf \cdot idf$ space) and measures based on Statistical Language Models [17, 19] (e.g. Kullback Leibler distance between unigram models).
In the following, different applications of document clustering mentioned above are presented in more detail. The improvement of the efficiency of an IR system is one of them [5, 30]: the search for relevant documents is restricted to a subset of the corpus which consists of the clusters whose centroids have the highest matching with the query. Clustering has also been used to improve retrieval effectiveness and there are three main approaches to this problem: a first solution is to enrich the document representation with information from its cluster [20], a second approach is to rely on the clusters to modify a first ranking obtained by an IR system (the documents belonging to the clusters of top-ranked documents are assigned higher scores [18]), a third approach is to restrict the search for relevant documents to the clusters with the highest query/centroid similarity scores [28, 30]. Furthermore, it has been noted that clustering can be helpful to browsing applications (i.e. tools which allow one to look through a corpus to find information of interest): users can identify more documents of interest in a limited amount of time when they are presented clusters rather than a list of single documents [13]. Distributed retrieval also can benefit from document clustering. In some cases, a large collection that cannot be stored on a single site for efficiency, reliability or storage constraints is split into different parts and each part is stored on a different site of a network. In such a case, it is possible to use a partition clustering to split the document collection [3, 34]. This approach has several advantages: when a query is submitted, the search for relevant documents can be restricted to sites containing the clusters with the highest query/centroid similarity scores, hence, the network traffic is limited and the computational cost due to the query submission affects only a few sites.
In the above applications, the benefit of using clustering depends on its ability to concentrate the documents with the same topic in few clusters that contain few documents with other topics. This property is difficult to evaluate directly without an application. That is why, in the most frequent approach [5, 13, 18, 30, 32], the performances of the methods using clustering are measured with respect to the final task and their results are compared to a solution without clustering. In this work, we do not rely on a specific task but we introduce a more general evaluation methodology (see section 4): we evaluate an intermediate step required by different applications [5, 18, 34]. In this step,

the system should identify (according to centroids) the clusters which are the most likely to contain the documents about a given topic. The identified clusters should contain most of the documents about the considered topic while containing few documents about different topics. This step is used in different contexts: it allows the system to restrict the search for relevant data to a subset of the corpus (for efficiency improvement [5]), to limit the number of sites involved in answering a request (in the case of distributed retrieval [34]) or to extract documents related to the user information needs (in order to improve retrieval effectiveness [18]).

3 Clustering Procedure

This section presents the algorithm we used to perform clustering: it is a partition algorithm that allows one to specify the similarity measure used to assign documents to clusters. Three different measures relying on different criteria (vector, set of terms or SLM comparisons) are evaluated by performing the same clustering experiments with each of them. The rest of this section is organized as follows: section 3.1 describes the algorithm and section 3.2 presents the similarity measures.

3.1 Clustering Algorithm

The clustering algorithm takes as input a collection of documents and splits it into K clusters through the following iterative process:

1. **Random initialization**
   The database is partitioned into K clusters containing the same number of documents through a random process.
2. **Iterative refinement**
   Each document is assigned to the most similar cluster according to the similarity measure chosen.
3. **Stopping criterion check**
   Step 2 is repeated until no performance improvement is observed on a set of *training queries*.

This algorithm is related to K-means algorithm. However, in the case of K-means, the goal is to minimize the mean square error when substituting a document representative by its cluster centroid and step 2 relies on Euclidean distance to compute similarities. In our case, the similarity measure can be specified (see section 3.2) which is advantageous as other measures can be more appropriate than Euclidean distance in the case of textual data [27]. The performance measure used for stopping criterion is another key aspect of this algorithm: after each refinement step, the quality of the clustering is evaluated according to the methodology described in section 4. At each step, the recall improvement with respect to the previous step is measured and the iterative process ends when no more improvement is observed. As for K-means, the hyper-parameter K (i.e. the number of clusters) should be a trade-off between the two following effects: if K is too small, the clustering will result in a few large clusters, possibly grouping documents about different topics in the same cluster. On the contrary, if K is too large, the clustering will result in many small clusters, possibly scattering documents about the same topic in different clusters. In our experiments, different K values have been evaluated to determine whether the choice of K has a great influence on the performance, or whether it can be set loosely without much impact on the results (see section 5).

3.2 Similarity Measures

The iterative refinement requires the computation of the similarity between documents and clusters in order to assign each document to the most similar cluster. This similarity should ideally be high when comparing a document and a cluster about the same topic and low otherwise. State-of-the-art measures access to such similarity through the comparison of document physical properties (e.g. term frequency, document length, etc.) [21, 27]. In this work, we evaluated three types of measures based on such properties. In the first approach, documents and clusters are represented as vectors which are compared according to their inner product. In the second approach, a term distribution is estimated for each document and each cluster and these distributions are compared using the Kullback-Leibler

distance. In the third approach, the set of shared terms between the document and the cluster is used to compute their similarity. Each of these approaches is described in the following.
In the first approach, the similarity between a document d and a cluster C is the inner product between the document vector $\mathbf{d}$ and the cluster vector $\mathbf{c}$:

$$sim(d, C) = \mathbf{d} \cdot \mathbf{c} = \sum_t d_t \cdot c_t$$

The document vector $\mathbf{d}$ is calculated as follows:

$$\forall t, d_t = ntf_{d,t} \cdot idf_t$$

where, $ntf_{d,t}$ is the normalized term frequency of term t in document d (the number of occurrences of t in d divided by the total number of term occurrences in d) and idf_t is the inverse document frequency of term t ($idf_t = log(N/N_t)$, where N_t is the number of documents that contain t and N is the total number of documents). The ntf factor gives more weight to terms occurring frequently in the document, based on the hypothesis that a term occurring several times is more representative of the document content. However, the tf factor alone is not sufficient. For example, a high frequency term occurring in most documents of the collection is not helpful to characterize the content of a document. This is why idf gives more weight to terms occurring in few documents, rare terms being considered more discriminative.
The vector $\mathbf{c}$ representing cluster C is the barycenter of the vectors representing the documents of C, weighted by their length:

$$\mathbf{c} = \frac{1}{\sum_{d \in C} l_d} \sum_{d \in C} l_d \cdot \mathbf{d}$$

where l_d is the length of d (i.e. the total number of terms in d). Longer documents are considered more representative of the cluster content and more reliable from a statistical point of view.
The second approach computes the Kullback-Leibler distance between the document and the cluster term distributions as introduced in [34].

$$sim(d, C) = KL(d, C^*) = \sum_{t : tf_{d,t} \neq 0} p_{t,d} \cdot log \frac{p_{t,d}}{p_{t,C^*}}$$

where $C^* = C \bigcup \{d\}$ is the set containing the documents of C and document d. The term distribution in document d is estimated as follows: $p_{t,d} = \frac{tf_{d,t}}{\sum_{t'} tf_{d,t'}} = ntf_{d,t}$ and the term distribution in set C^* is estimated by considering a cluster as a single document resulting from the concatenation of the documents it contains: $p_{t,C^*} = \frac{tf_{C^*,t}}{\sum_{t'} tf_{C^*,t'}}$ where $tf_{C^*,t} = \sum_{d' \in C \bigcup \{d\}} tf_{d',t}$.
The third measure evaluated [4] compares a document and a cluster according to the set of shared terms in a similar way as a query and a document are compared in an IR system. Hence, identifying the most similar cluster with respect to a document is analogous to a retrieval problem where the most relevant document with respect to a query should be retrieved. A document is assigned to a cluster according to the following matching measure:

$$sim(d, C) = \sum_{t \in d} ndf_{C,t} \cdot icf_t$$

where $ndf_{C,t}$ is the normalized document frequency of t in C (i.e. the number of documents in C that contain term t divided by the number of documents of C) and $icf_t = log(K/K_t)$ is the inverse cluster frequency (K is the number of clusters and K_t is the number of clusters in which term t is present).
The three above similarity measures are referred to as $ntf \cdot idf$, KL and $df \cdot icf$ respectively in the following.

4 Evaluation Methodology

The goal of clustering is to identify groups of documents such that each group ideally contains all documents about a topic and no documents about different topics. This can be evaluated with the help of human assessors [22], but this approach can hardly be used for large databases, as it would require too much time. There is hence a need for an automatic evaluation methodology. In this work, we propose to verify the following property which is required by different applications relying on clustering (see section 2): given a topic, documents about this topic should be concentrated in few clusters that contain few documents about other topics. Moreover, the centroid should allow one to identify the prevailing topic of a cluster. For that purpose, we used IR queries: for each query, the relevant documents are considered to be about the same topic (i.e. the topic described by the query) while non-relevant documents are considered to be about different topics. Hence, given a query, we should verify whether it is possible to identify (using cluster centroids) a set of clusters that contain most of the relevant documents (i.e. in-topic documents) while containing few non-relevant documents (i.e. off-topic documents). This evaluation is performed in two steps. For each test query, we first rank the clusters according to the matching of their centroid with it using the following matching measure:

$$sim(q, C) = \sum_{t \in q} ntf_{c,t} \cdot icf_t$$

where the normalized term frequency of t in C is defined as the sum of the term frequencies of the documents of C divided by the sum of the lengths of the documents of C: $ntf_{C,t} = \sum_{d \in C} tf_{d,t} / \sum_{d \in C} l_d$. Second, at each position n in the ranking we measure *recall* (i.e. the percentage of relevant documents that appear in the clusters ranked above position n) and *selection rate* σ (i.e. the fraction of the corpus that the clusters ranked above position n account for). A good clustering should allow one to select, given a query, a fraction as small as possible of the database while preserving most of the in-topic documents (i.e. achieving a high recall at a low selection rate). We hence measure recall as a function of the selection rate $R(\sigma)$ and we average these results over several evaluation queries. As mentioned in section 3.1, this evaluation methodology is also used to define the stopping criterion of the clustering algorithm: the iterative process stops when the average recall ($\int_\sigma R(\sigma) d\sigma$) is not higher than at the previous step. In this case, a set of training queries distinct from the evaluation queries is used (section 5.1 describes the data used).

5 Experiments and Results

This section describes the experiments performed: Section 5.1 describes the data used and the experimental setup, section 5.2 presents the results.

5.1 Experimental Setup

Our goal is to compare document clustering results obtained over clean and noisy texts (i.e. texts that have been extracted from other media through an automatic process like ASR or OCR). We use a spoken document corpus (TDT2 [8]) which contains 600 hours of American broadcast news. This corresponds to $ndoc = 24,823$ documents that are available in both clean and noisy version. The clean version consists of manually typed texts (closed-caption for deaf people) while the noisy version is the output of the Dragon ASR system [33] and suffers from $\sim 30\%$ WER. The documents are brief news stories (clean and noisy document average lengths are 179.2 and 178.7 words). All data are preprocessed prior to clustering: content-poor words (e.g. "the", "what" or "through") are removed using a 389 stop-word list and all words have been replaced by their stem (e.g. "connection" and "connecting" are replaced by "connect") using Porter's stemmer [23].

The same experiments are performed on clean and noisy versions of the corpus: each version is clustered using the three clustering procedures presented in section 3.1. We also use a random split of the corpus, in which each cluster has the same size, as a baseline. The results are compared according to the evaluation methodology defined in section 4. All clustering techniques ($ntf \cdot idf$, KL, $df \cdot icf$

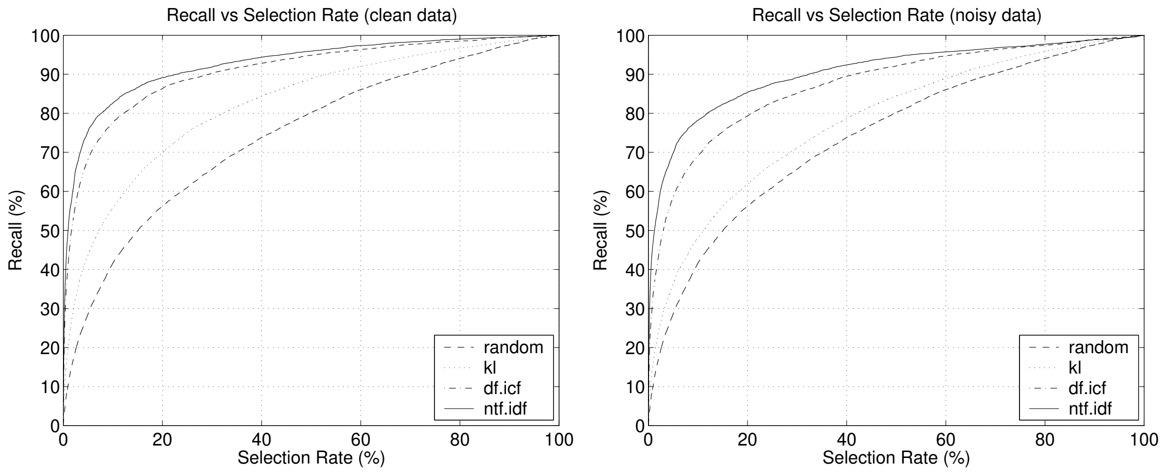

Figure 1: Recall vs Selection Rate for $K = 500$

and *random*) draw a random partition at the initialization step: the same initialization is used for all techniques in order to perform valid comparisons. Furthermore, all experiments were repeated with 10 different initializations and the results are averaged to avoid bias toward a specific initial random partition. K (i.e. the number of cluster) values ranging from 250 to $2,500$ (i.e. K varies from $\sim 1\%$ to $\sim 10\%$ of the corpus size $ndoc$) are tested to determine the influence of K on the clustering results. The IR queries required by the evaluation methodology (see section 4) are taken from TREC SDR: TREC8 subset is used for training (i.e. the queries used by the stopping criterion of the clustering algorithm) and TREC9 subset is used for evaluation. Each subset is composed of 50 queries, each query having on average 38.1 relevant documents.
The following section presents the results we obtained.

5.2 Results

When comparing the different clustering methods (see fig. 1), the same conclusions can be drawn from noisy and clean data experiments: $ntf \cdot idf$ and $df \cdot icf$ methods outperform KL and *random*. The $ntf \cdot idf$ technique especially leads to good results at low selection rates (see fig. 2 for recall at $\sigma = 5\%$). Two key characteristics of $ntf \cdot idf$ can explain these results. First, the use of statistics extracted from the whole corpus (idf) gives more weight to discriminative terms (i.e. terms occurring in few documents): those terms are more characteristic of the document content and have a lower probability to occur in two documents with different topics. Second, $ntf \cdot idf$ considers longer documents as better representatives of a cluster content: the content of these documents can be more reliably estimated from a statistical point of view and they can be assigned to a cluster with more confidence. The $df \cdot icf$ technique also uses statistics extracted from the whole corpus (icf) which possibly explain why its results are not far from those obtained with $ntf \cdot idf$. Contrary to the other methods, KL leads to poor results, close to or worse than random clustering (depending on the K value, see fig. 2). This might be due to the briefness of the documents (~ 179 words): no reliable term distribution can be estimated from such a small sample. Moreover, contrary to $ntf \cdot idf$ and $df \cdot icf$, KL does not make any distinction between terms based on statistics extracted from the whole corpus.
Concerning the choice of K (i.e the number of clusters), the results show that it has a moderate influence on $ntf \cdot idf$ and $df \cdot icf$ results (see fig. 2). This means that the hyper-parameter does not need to be carefully tuned within a range between $\sim 1\%$ and $\sim 10\%$ of the corpus size. On the contrary, KL is shown to be more sensitive to the choice of K. KL leads to better performance with small K values: with fewer clusters, each cluster is larger and, hence, a better language model can be estimated with this larger amount of data which possibly explains KL's better results. Random

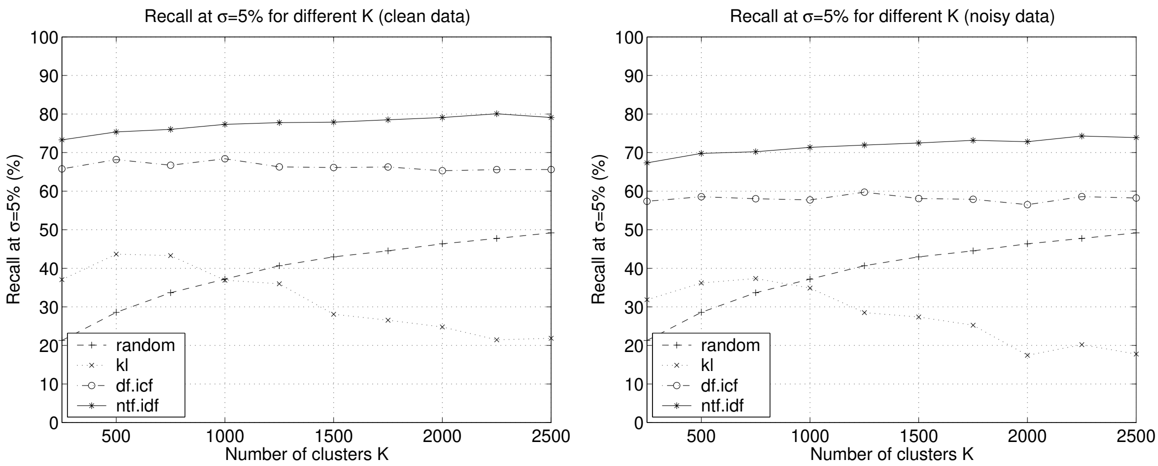

Figure 2: Recall at $\sigma = 5\%$ for different K

clustering results are better when K is larger as, in this case, the clusters are smaller, which means that less off-topic documents are present in the clusters containing at least one in-topic document.

The comparison of clean and noisy results allows one to evaluate the robustness of each clustering technique with respect to noise: for that purpose, we plot relative recall degradation as a function of selection rate (fig 3). Even with $\sim 30\%$ WER, the degradation is limited for $ntf \cdot idf$ and $df \cdot icf$ (less

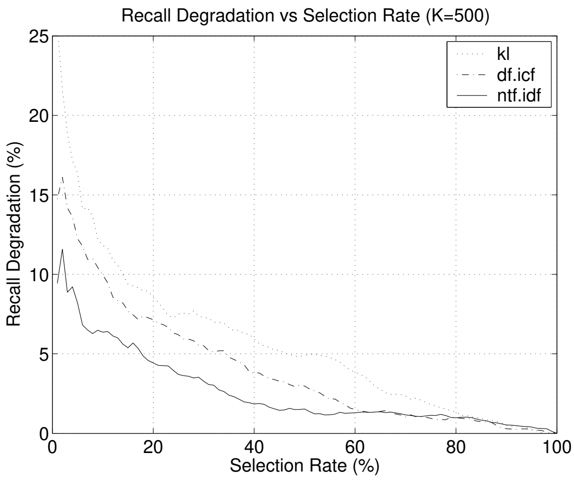

Figure 3: Relative recall degradation at $K = 500$

than 15%). The $ntf \cdot idf$ technique is the most robust to noise (less than 10% at $\sigma = 5\%$). On the contrary, KL is more affected by noise than the other methods. The robustness of $ntf \cdot idf$ is possibly due to the fact that longer documents are given more weight to compute the cluster representatives: longer documents have a higher level of redundancy (i.e. some terms are repeated and different related terms are present in those documents) which means that a recognition error has less impact on such documents (i.e. it is unlikely that all repetitions of the same term are corrupted). In our data, terms occurring more than once represent 28% of the 10% longest documents (as opposed to 15% in the other documents) and only 10% of these repeted terms are not preserved (i.e. all their occurrences have been mis-recognized by the ASR system), as opposed to 22% for terms occurring once. The results also suggest that techniques leading to good results on clean data ($ntf \cdot idf$ and $df \cdot icf$)

are also effective on noisy data. This means that techniques developed for clean texts can be used, without modifications, to cluster other types of data (like broadcast news recordings) through the use of automatically extracted texts, even in the presence of an important amount of recognition errors.

6 Conclusions

In this work, we focused on the clustering of noisy texts, i.e. texts that have been extracted from other media through an automatic process (e.g. ASR, OCR). Such techniques could be helpful to retrieval and browsing of multimedia databases (e.g. video conference recordings or ancient manuscript archive), if shown to be robust to noise. In order to measure this robustness, we performed the same experiments on clean (manually typed) and noisy (ASR output with $\sim 30\%$ WER) versions of a same corpus (TDT2 which consist of $\sim 25,000$ documents from broadcast news).
Three different clustering techniques have been evaluated. Those techniques differ in the way they compute the similarity between a document and a cluster which is a key aspect in a clustering procedure. The first technique ($ntf \cdot idf$) assigns to documents and clusters a vector representation and uses the inner product for comparison. The second method (KL) compares documents and clusters according to term distributions estimated from them. The last method ($df \cdot icf$) relies on the set of shared terms to compute similarities.
In order to determine the clustering performance, a quantitative evaluation methodology has been introduced: given a topic, we verified whether it is possible to identify few clusters that contain most of the in-topic documents, while containing few off-topic documents (see section 4).
The results suggest that $ntf \cdot idf$ and $df \cdot icf$ techniques lead to good results in both the clean and the noisy case. The $ntf \cdot idf$ method is besides shown to be the most robust to noise when comparing results on clean and noisy data. The $df \cdot icf$ technique is also leading to good results with a slightly higher level of degradation due to noise. However, the effect of noise on both $ntf \cdot idf$ and $df \cdot icf$ clustering can be considered as limited in contrast with the level of noise in our data ($\sim 30\%$ WER). On the contrary, KL achieved poor results on TDT2 data, certainly because of the briefness of the documents (~ 180 words) which prevents one from estimating reliable term distributions.
These results are promising and suggest that document clustering developed for clean texts can be applied to noisy texts. This would allow one to apply such techniques to various sources from which texts can be extracted (speech recordings, handwritten documents, video databases, etc) and benefit from the retrieval and browsing techniques that rely on clustering, which is a potential future work. It would also be interesting to verify whether clustering techniques are also robust in the presence of higher levels of noise (i.e. with data from worse recording conditions).

Acknowledgments

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] A. Amir, M. Berg, S.-F. Chang, and W. Hsu. IBM research TRECVID-2003 video retrieval system. In *NIST TREC Video*, 2003.

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

[3] M. Bawa, G. Manku, and P. Raghavan. SETS: Search enhanced by topic-segmentation. In *ACM Special Interest Group on Information Retrieval*, 2003.

[4] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collection with inference networks. In *ACM Special Interest Group on Information Retrieval*, pages 21–28, 1995.

[5] F. Can, I. S. Altingvde, and E. Demir. Efficiency and effectiveness of query processing in cluster-based retrieval. *Information Systems*, 2004 (accepted for publication).

[6] D. Chen, J-M. Odobez, and H. Bourlard. Text detection and recognition in images and video frames. *Pattern Recognition*, 3(37):595–608, 2004.

[7] M. Christel and C. Huang. Enhanced access to digital video through visually rich interfaces. In *IEEE International Conference on Multimedia and Expo*, 2003.

[8] C. Cieri, D. Graff, M. Liberman, N. Martey, and S. Strassel. The TDT-2 text and speech corpus. In *DARPA Broadcast News Workshop*, 1999.

[9] D. Cutting, D. Karger, J. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *ACM Special Interest Group on Information Retrieval*, 1992.

[10] J. Garofolo, J. Lard, and E. Voorhees. Overview of the trec-9 spoken document retrieval track. In *NIST Text Retrieval Conference*, 2000.

[11] G. Gaughan, A. F. Smeaton, C. Gurrin, H. Lee, and K. McDonald. Design, implementation and testing of an interactive video retrieval system. In *ACM SIGMM international workshop on Multimedia information retrieval*, pages 23–30. ACM Press, 2003.

[12] A. Hauptman, R.V. Baron, M.-Y. Chen, and M. Christel. Analyzing and searching broadcast news video. In *NIST TREC Video*, 2003.

[13] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *ACM Special Interest Group on Information Retrieval*, pages 76–84. ACM, 1996.

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

[15] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, Massachusetts, 1998.

[16] K. Koumpis and S. Renals. Automatic summarization of voicemail messages using lexical and prosodic features. *ACM Transactions on Speech and Language Processing*, 2004. Submitted for publication.

[17] V. Lavrenko and W. B. Croft. Relevance based language models. In *ACM Special Interest Group on Information Retrieval*, pages 120–127. ACM, 2001.

[18] K. S. Lee, Y. C. Park, and K. S. Choi. Re-ranking model based on document clusters. *Information Processing and Management*, 37(1):1–14, 2001.

[19] X. Liu and W. B. Croft. Statistical language modeling for information retrieval. *Annual Review of Information Science and Technology*, 39, 2003.

[20] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *ACM Special Interest Group on Information Retrieval*, pages 186–193. ACM, 2004.

[21] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 3(2):159–165, 1958.

[22] J.M. Odobez, D. Gatica-Perez, and M. Guillemot. Spectral structuring of home videos. In *International Conference on Image and Video Retrieval*, 2003.

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

[24] E. Rasmussen. Clustering algorithm. In W. F. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, pages 419–442. Prentice Hall, 1992.

[25] S. Renals, D. Abberley, D. Kirby, and T. Robinson. Indexing and retrieval of broadcast news. *Speech Communication*, 32:5–20, 2000.

[26] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, UK, 1996.

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

[28] W. M. Shaw, R. Burgin, and P. Howell. Performance standards and evaluations in ir test collections: cluster-based retrieval models. *Information Processing and Management*, 33(1):1 – 14, 1997.

[29] A. Tombros, R. Villa, and C. J. Van Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing and Management*, 38(4):559–582, 2002.

[30] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[31] A. Vinciarelli. Noisy text categorization. In *International Conference on Pattern Recognition*, 2004.

[32] E. M. Voorhees. The efficiency of inverted index and cluster searches. In *ACM Special Interest Group on Information Retrieval*, pages 164 – 174, 1986.

[33] S. Wegmann, F. Scattone, I. Carp, L. Gillick, R. Roth, and J. Yamron. Dragon systems' 1997 broadcast news transcription system. In *NIST Broadcast News Transcription and Understanding Workshop*, 1998.

[34] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *ACM Special Interest Group on Information Retrieval*, pages 254–261, 1999.