IDIAP RESEARCH REPORT

APPLICATION OF INFORMATION RETRIEVAL TECHNOLOGIES TO PRESENTATION SLIDES

Alessandro Vinciarelli * Jean-Marc Odobez *

IDIAP–RR 05-36

JUNE 2005

TO APPEAR IN
IEEE Transactions on Multimedia

* IDIAP Research Institute, Martigny, Switzerland

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 05-36

APPLICATION OF INFORMATION RETRIEVAL TECHNOLOGIES TO PRESENTATION SLIDES

Alessandro Vinciarelli Jean-Marc Odobez

JUNE 2005

TO APPEAR IN
IEEE Transactions on Multimedia

Abstract. Presentations are becoming an increasingly more common means of communication in working environments, and slides are often the necessary supporting material on which the presentations rely. In this paper, we describe a slide indexing and retrieval system in which the slides are captured as images (through a framegrabber) at the moment they are displayed during a presentation and then transcribed with an OCR system. In this context, we show that such an approach presents several advantages over the use of commercial software (API based) to obtain the slide transcriptions. We report a set of retrieval experiments conducted on a database of 26 real presentations (570 slides) collected at a workshop. The experiments show that the overall retrieval performance is close to that obtained using either a manual transcription of the slides or the API software. Moreover, the experiments show that the OCR based approach outperforms significantly the API in extracting the text embedded in images and figures.

1 Introduction

Presentations and talks are common events in many working environments (companies, schools, conferences, etc.). They often represent a valuable source of information, but their content is difficult to store. In most cases, after the presentation is given, no record is left and most of the information provided by the speaker is lost. The most simple solution for such a problem is to record the talks (with cameras and microphones) and then to make them available to potential users without further processing (the so-called *Record and Playback* approach [35]), but the resulting material quickly becomes difficult to use. After that few hours of recordings have been collected, to retrieve the few minutes concerning a specific topic or simply to know what a talk is about can require the manual examination of long recording segments [17], [34]. For the above reasons, there have been several research efforts in order to develop effective indexing and browsing techniques allowing one to go beyond the simple Record and Playback approach (see section 2).

Most of the literature focuses, to our knowledge, on so-called *instructional talks*, i.e. presentations based on slides containing, in a concise form, the core information conveyed by the speaker. The use of slides does not represent a restrictive constraint since it is common in a wide spectrum of situations and it represents the rule rather than the exception. In such a framework, it is possible to include (the list is not exhaustive) the processing of school courses in the e-learning domain [1], [34], the production of video proceedings for conferences [2] and the creation of smart environments aimed at capturing class or meeting participant experiences [1].

In all of the above examples, slides are widely recognized as a fundamental source of information, but so far they have been used, to our knowledge, only to partition the recordings into meaningful fragments: slide changes are detected (see section 2 for the techniques used) and the presentation videos are segmented in correspondence with them. The rationale behind such an approach is that within the presentation, only one topic is discussed during the time a slide is displayed, and thus that a topic change can only occur at a slide transition. The fundamental limit of the above approach is, in our opinion, that the topic itself is not taken into account at all. This means that when a user wants to find the segment corresponding to a certain topic, s/he must browse through the slides until s/he finds the one corresponding to it. This can be reasonable for a few presentations, but it becomes heavily time consuming when the number of talks increases. In our data set, 26 presentations collected at the MLMI workshop [4] result in 570 slides and this means that the user might be required to browse hundreds of slides in order to find what s/he is looking for. This can be especially problematic when the user accesses the system through a network (e.g. in distance learning) and the amount of data transmitted must be limited. A good solution to such problems is, in our opinion, to transcribe and index the slides in order to apply Information Retrieval (IR) techniques. In this way the user can first search the slides answering to her/his information needs and then watch the presentation video segments they correspond to.

The automatic transcription of slides can be performed with software that converts the most common formats used for presentations (pdf and ppt) into ASCII text, but this creates several problems. The first is that the transcriptions are not synchronized with the presentation

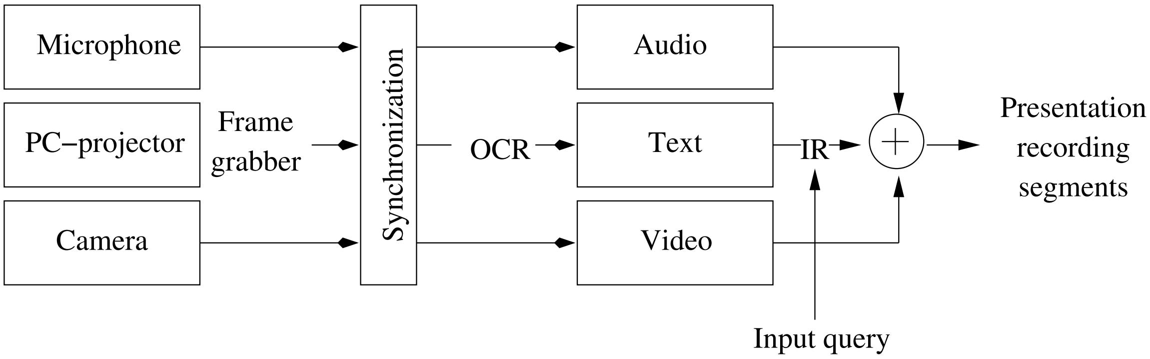

Figure 1: Retrieval approach. The presentations are recorded through three channels (audio, video and PC-projector). Slide images are obtained through the framegrabber and transcribed with an OCR system. By applying Information Retrieval (IR) techniques to the resulting text, it is possible to retrieve presentation recording segments relevant to an input query.

video. In other words, the information allowing for the linking of a slide with the video segment where it was displayed is not available. The second is that the conversion software is based on APIs that are potentially expensive and become obsolete after a relatively short lifespan because of the changes in commercial proprietary formats. The third problem is that the slides often contain text which is embedded in figures (workflows, system diagrams, plots, etc.) and the above converters cannot always access it. Moreover the speakers may leave the slideshow to use and demonstrate other software. With the converters, no text indexing and access points to these parts of the talk would be available.

In our opinion, the above problems can be solved by capturing the slides with a framegrabber (i.e. a device able to acquire and store as images that which is displayed through a projector) and then transcribe them with an Optical Character Recognition (OCR) system (see Figure 1). . The framegrabber output can in fact be synchronized with the presentation video (each slide can thus be linked with a video segment), the slide images are independent of the original format (ppt or pdf), and the text embedded in pictures can be transcribed as well as text displayed on the screen when the speaker projects something different from the slides. OCR technology has become one of the most successful applications in the field of pattern recognition. However, OCR systems have been designed to recognize characters on printed documents, and the application of this technology to other information sources such as images or videos remains a challenging problem [13, 27, 33, 16, 18, 5]. Slides are difficult to transcribe as they often contain a large variety of text fonts and sizes (from 10 to 130 pixels), images, plots and figures that can be misinterpreted as texts, layout changes for each slide and sometimes structured and complex backgrounds. Moreover, the use of linguistic information can be helpful only to a limited extent because presentations contain many proper names and acronyms that cannot be found in common linguistic resources (e.g. text corpora) used to build lexicon and language models.

In this paper, we study the application of Information Retrieval to automatic transcriptions of

slides. First we analyze the recognition performance of a video OCR system described in [5] on such data, then we show how the recognition errors affect the IR performance on several retrieval tasks. The experiments have been conducted on a database of 26 presentations (or 570 slides) gathered at a workshop [4]. They show that despite the use of transcriptions affected by recognition errors and a large amount of noise, the retrieval performance degradation with respect to the use of manual or API based transcriptions (no errors) is acceptable.
The rest of this paper is organized as follows: Section 2 presents a survey of related works, Section 3 describes the OCR system used in our experiments, Section 4 shows the Information Retrieval approach applied, Section 5 presents experiments and results and Section 6 draws some conclusions.

2 Previous work

This section presents a survey of the papers dedicated to presentations in the literature. First we will describe works dealing with single aspects of the processing (e.g. segmentation or browsing), then we will show articles where talks are used in a wider context and the processing is oriented to specific goals (e.g. e-learning or conference video-proceedings).
One of the main problems of presentations is that they are composed of long streams of information (e.g. audio or video recordings) that are difficult to handle as a whole by users. This makes it necessary to partition them into segments that are meaningful to users and that enable them to effectively use the information contained in the presentation. All of the works dedicated to the above problem perform a *thematic segmentation*, i.e. they try to identify segments characterized by a single and specific topic. The main reason is that talks are typically organized as a sequence of topics and most, if not all, of the presentation content is concentrated in the topics presented by the speaker.
The use of shot boundaries as a criterion to perform thematic segmentation has been quickly discarded because in many practical applications talk recordings are made of one single shot obtained with a fixed camera pointing at the speaker. More relevant information can be extracted from the audio. In [15], the analysis of prosody and silences as well as the detection of *Discourse Markers*, i.e. expressions that typically introduce a new argument, are used to segment and index university lectures. The main limit of such an approach is that it is strongly related to the style of the speaker and it is language dependent. A content-based segmentation can be obtained by applying approaches based on text analysis. In [12], the variations in frequency and cooccurrence of words appearing in neighbouring segments of a text are used to detect topic changes and to structure a text into sections and subsections. The problem of such an approach is that it works well for data like news where stories about completely different topics appear after each other, but it has more difficulty on texts where there is a single topic and different subtopics (as is the case in presentations). Moreover, in the case of talks, the system should work on transcriptions obtained through Automatic Speech Recognition and this can further reduce the effectiveness of the method.
The approach that has been preferred so far is to segment the presentations in correspondence with slide transitions [31], [20], [19], [14]. As mentioned in the previous section, this is rea-

sonable because it reflects the logical organization given by the speaker to his/her talk, but it neglects the actual topics being presented. In all of the cited works, the slide changes are detected using the video footage. The slide is first located in the images (the problem of the speaker often occluding the slide is solved by analyzing multiple frames) and then matched with the electronic versions of the slides assumed to be available to the system. Each time the electronic slide best matching the slide extracted in the video changes, a transition is assumed to take place. An alternative solution to the same problem is to capture the slides through a frame-grabber synchronized with the video cameras, and to detect the transitions as the points where the difference between two following displayed images exceeds some threshold. Such an approach is used in this work and it is simpler, but at the same time it requires more devices (projector, frame-grabber, synchronization devices). On the other hand, no electronic version of the slides is necessary and no slide format dependent APIs need be used (see previous section). The segmentation based on slides is used to browse the presentations. Some of the works presented above as well as several works in the e-learning domain (see below) make use of browsers allowing to display the slides of a presentation and to access their corresponding segment by acting over them. An interesting approach is presented in [29], where the authors segment the presentation in correspondence with slide changes and then use transcriptions of both slides and audio in order to find what they call *topical events*, i.e. points of the audio where the words in the slide occur together.

One of the most common applications of the systems analyzing presentations is *e-learning*, i.e. the use of computer based tools to improve or facilitate didactic activity. The works in this domain can be roughly divided into systems that try to capture the *experience* of students in a classroom (in other words they try to make attending a lecture directly or through a computer, equivalent) [35], [17], [23], [1], [19] and works that are aimed at the efficient transmission of lectures to students that cannot attend directly (this domain is oftent defined *distance learning*) [34], [6]. The latter aspect is out of the scope of this work and we thus analyze in thus more detail the first kind of application. In [35], [17], the attention is focused essentially on the devices used to capture the lectures. The main concern when designing an apparatus capturing presentations is that it should be very easy to activate (the speaker is supposed to just push a button), but at the same time to be complex enough to capture all the information produced during the lecture. Complex devices can allow very good processing but they are potentially expensive and difficult to manage, while simple systems can be very easy to manage (to activate a simple videocamera a single button is sufficient), but they can produce unstructured data that require the elaboration of complex processing software. In [1], [23], extensive experiments performed during a course showed that a system based on videos and slides allowed the students to take more advantage of the lectures. The system was evaluated through the number of accesses to the website where the information was stored and through questionaries. The main advantage provided by the system is that the students can avoid taking detailed notes during the courses and focus on main ideas and concepts presented by the teacher.

An important application that has been explored is *summarization* [10], [11], [14]. In [10], the summary is obtained by simply eliminating silences from the audio channel. This reduces the length of a speech recording by 15-20 percent. The same approach is used in [11],

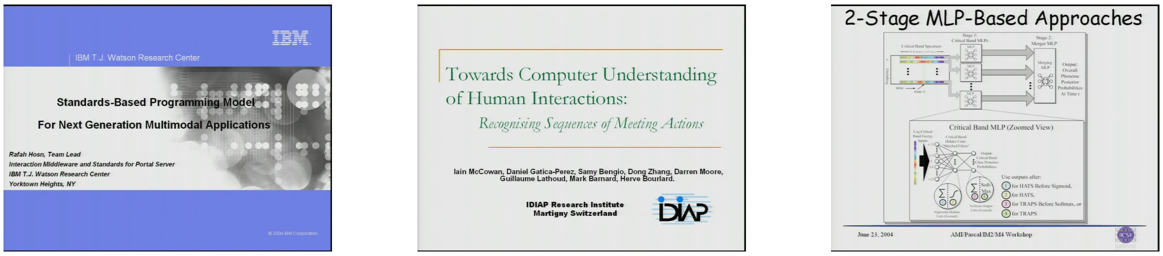

Figure 2: Some example of slides in the database

but further compression is achieved by selecting only the first seconds after slide changes. Such temporal segments are in fact assumed to summarize the content of the segment where a certain slide is displayed. In [14], the summarization is performed by first identifying segments between two slide transitions, and then by detecting gestures (e.g. pointing to specific elements in the slide) that are assumed to be related to important information. For each segment corresponding to a single slide, a few subsequences related to such gestures are thus extracted to build the summary.

The possibility of creating video-proceedings for conferences has been explored in [2]. This work is essentially aimed at the retrieval of speech segments (the audio of the talks is transcribed with an Automatic Speech Recognition System) and the video is used to browse the retrieved segments in order to fine tune the results of a query.

The only work dedicated to the retrieval of slides in the literature is presented, to our knowledge, in [21]. The main limitation of the system is, in our opinion, that it can access only proprietary formats of slides. This means that the system must be constantly updated in order to follow the version changes of commercial software for slide editing. Moreover, not all of the available formats are covered and some slide formats cannot be accessed.

3 Text recognition system

Research efforts on the extension of OCR technologies to documents such as images and video started approximately 10 years ago. To the exception of some early works, most of the research in this field have adopted a top-down approach to the problem: text regions are first localized in the image, and a text recognition system is then applied on the extracted regions [13, 27, 33, 16, 18]. The method we employ in this article follows the same scheme (see [5] for a detailed description). In the next sections, we present an overview of the method, describing in more detail the aspects that have more impact on the retrieval performance.

3.1 Text line detection

The text line localization algorithm has two components. The first one consists in classifying each pixel of the image into either text or non-text. To achieve this task, vertical and horizontal edges are extracted using a Canny edge detector. Then, morphological operators are

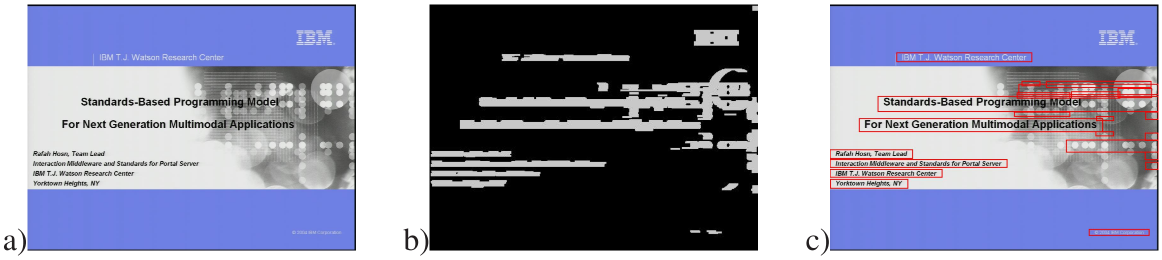

Figure 3: Text line detection process. (a) original image (b) potential text pixels (c) candidate text lines

applied to enforce the presence of both edge types in regions labeled as text. The image in figure 3b displays the result of this step applied to the image on the left of Figure 3.

The second part of the algorithm aims at identifying individual text lines from the generated text-labeled binary map. We perform this task by searching in a systematic way for the top and bottom baselines of horizontally aligned text string regions with enough density. The result of this algorithm applied on the binary image of figure 3b is shown in figure 3c. As can be seen, in the presence of structured background, the detection process generates a certain number of false alarms. These false alarms will be eliminated by identifiying noisy transcriptions generated by the recognition system, as described in the next section.

3.2 Text recognition

In this subsection, we first present an overall description of the recognition algorithm, and then we focus on the different strategies used to generate the text transcript.

3.2.1 Overall scheme

A simple approach to perform text recognition from localized image text lines consists of the application of a binarization algorithm on the text image followed by the use of standard OCR software. Although this approach can be sufficient ro recognize the majority of slide text, it still leads to many errors due to the following two issues:

- the binarization process can be affected by the fact that the distribution of gray-scale levels in the text region may not be bimodal. This can happen due to the presence of a structured background or layout, or because the localized text is part of an image, a drawing, or a plot.

- the exact text size and text font are unknown and difficult to estimate due to the limited amount of available text (which ranges from two characters to several words). As a consequence, the OCR is confused by similar-looking characters (*e.g.* l, I, 1, i,. . .), and its output is sensitive to the parameters of the binarization/text segmentation algorithm.

To address these issues, we proposed in [5] a scheme whose principle is illustrated in Figures 4 and 5, and works as follows: first, a segmentation algorithm that classifies the pixels

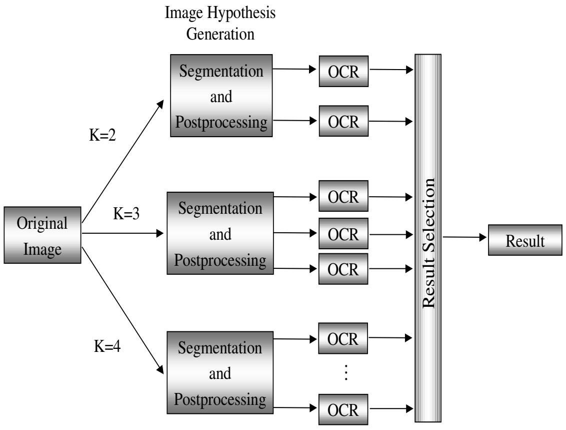

Figure 4: Text recognition process: overall text recognition scheme.

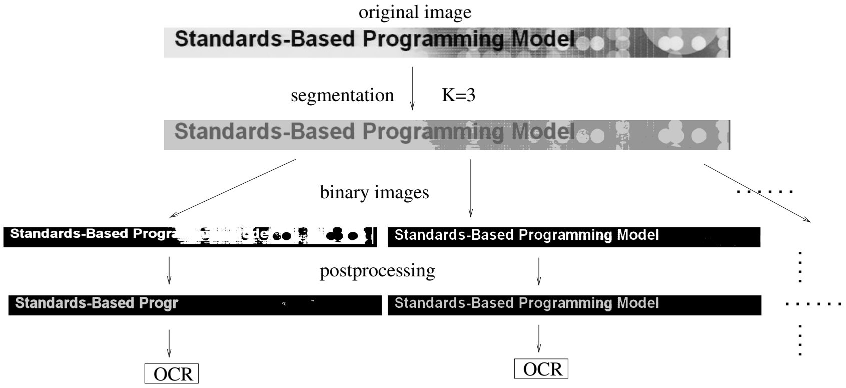

Figure 5: Text recognition process: individual segmentation and post-processing steps, for each value of K.

into K classes is applied to the text image. Then, the segmentation is exploited to produce binary text image hypotheses (*e.g.*, by assuming that a label, or a conjunction of labels, corresponds to the text layer). The resulting binary images are then passed through a post-processing step and forwarded to the OCR system, in this way producing different string hypotheses, from which the text result is selected.
In the current work, we used the K-Means algorithm to perform the image segmentation, as it was shown to have similar performance to more complex methods based on Markov Random Field [5, 32], and a connected component analysis step as post-processing, to remove regions corresponding to noise. More precisely, we only keep as character components the connected components that satisfy constraints on different parameters, such as size, aspect ratio, fill-factor and localization with respect to the text region boundaries. We then apply the OCR software on the resulting binary images to produce the text strings. The algorithm that selects the result from all the produced text strings is described in the next section.

3.2.2 Result selection and transcript production

The selection of the image text transcript from the set of strings generated by the segmentation step relies on a confidence value computed for each recognized string. This confidence value evaluation process exploits some prior information on text strings and on the OCR performance based on language modeling (applied to character sequences) and OCR recognition statistics. From a qualitative point of view, the system works by identifying characters which are more reliably produced when the segmentation is ideal (*i.e.* the original text is recognized with no error) than when the segmentation is noisy. For instance, when given text-like backgrounds or inaccurate segmentations, the OCR system produces mainly garbage characters like ., ,!, & etc and simple characters like i,l, and r, whereas characters like A or G are rarely produced in these situations.

More formally, let $T = (T_i)_{i=1..l_T}$ denote a string where l_T denotes the length of the string and each character T_i is an element of the character set $\mathcal{T} = (0, \ldots, 9, a, \ldots, z, A, \ldots, Z, G_b)$, in which G_b corresponds to any other garbage character. Furthermore, let H_a (resp. H_n) denote the hypothesis that the string T or the characters T_i are generated from an accurate (resp. a noisy) segmentation. The confidence value is estimated using a variant of the log-likelihood ratio:

$$C_v(T) = \log\left(\frac{p(H_a|T)}{p(H_n|T)}\right) + l_T * b = \log\left(p(T|H_a)\right) - \log\left(p(T|H_n)\right) + l_T * b$$

when assuming an equal prior on the two hypotheses and b is a bias. We estimated the noise free language model $p(.|H_a)$ by applying the CMU-Cambridge Statistical Language Modeling (SLM) toolkit on Gutenberg collections[1]. The noisy language model $p(.|H_n)$ was obtained by applying the same software on a database of strings collected from the OCR system output while providing as input to the OCR either badly segmented texts or text-like false alarms coming from the text detection process. The use of the bias b is necessary to account for the string length and avoid the confidence evaluation process to over-weight short strings with only a few very reliable letters. More details can be found in [5].

The confidence value can be used for two purposes. The first one is the rejection of string results whose confidence value is not high enough. This usage is extremely useful to filter out false alarms in the detection step. For instance, in the example of Fig. 3, 13 out of the 14 erroneously detected regions did not produce any string with a confidence value above the threshold used in our experiments. They were thus considered as non-text regions and rejected.

The second purpose is the selection of the final text transcript from the set of all strings generated by our multi-hypothesis approach. In the experiments, we have considered the three following methods to produce the transcript:

1. **Trans2 :** in this case, we only considered a segmentation process with K=2 classes, resulting in the generation of two strings (one corresponding to the binary image which assumes bright characters on a dark background, and one based on the reverse assumption). The string with highest confidence is used as the transcript. This strategy

[1] www.gutenberg.net

corresponds to the usual binarization process used in most of the work on text recognition.

2. **TransBest :** as shown in Fig. 4, the recognition process is applied three times, by segmenting the image each time with a different K value. Specifically, we used a value of K equal to 2, 3 and 4. >From all the generated text string hypotheses, the string with highest confidence is used as the transcript. In [5], this method applied to videos was shown to significantly improve the recognition rate, at both the character and word level.

3. **TransAll :** in the current application, the transcripts are not intended to be read by people. They will be used for slide indexing in a retrieval task. For such an application, the most important point is to obtain a transcript with as many slide words as possible correctly recognized. To optimize this criterion, we propose to use the following strategy. >From the set of text strings obtained for a single value of K (see Fig.4), we keep the string with the highest confidence. In this way, we obtain three text strings T_{K2}, T_{K3} and T_{K4}. Then, we initialize the final text transcript T_{final} with the most confident of these strings. Finally, T_{final} is iteratively updated, by adding to it each word of the two other strings that is not yet in the transcript. With this strategy, we palliate the sensitivity of the OCR engine, which sometimes, due to the small amount of text material or to JPEG compression distortion noise, produces strings from different segmentations that only differ by one letter.

The transcripts obtained with any of these three methods will be used to index the slides as described in the next section.

4 Information Retrieval

Information Retrieval is the task of finding automatically in a large corpus the documents that are relevant to an information need expressed through a query. The literature proposes several approaches (see [3] for a survey) and in this work we use the so-called *Vector Space Model* (VSM) which is the most successful and widely applied. A system following such an approach is composed essentially of two parts. The first is defined *offline* and it is performed only once for a given database. The second is called *online*, and it is performed each time a query is submitted to the system. The offline part performs *normalization* and *indexing*, while the *online* part performs the actual retrieval. The next two subsections describe the steps of the process in detail.

4.1 Normalization and Indexing

Normalization and indexing compose the offline part of the system. The normalization takes the raw data as input and removes from it the variability which is not useful for the rest of the process. It is composed of three steps (*preprocessing*, *stopping* and *stemming*) and

it converts the original documents into streams of *terms*. The indexing takes as input the streams of terms and converts them into a form suitable for the retrieval process. In the case of the VSM, the documents are represented as vectors where each component accounts for a term of the dictionary (the list of unique terms appearing in the corpus). At the end of the indexing, the document vectors are arranged in the *term-by-document* matrix A where each column corresponds to a document and each line corresponds to a term.
The first normalization step is the *preprocessing*, which simply removes all non-alphabetic characters (punctuation marks, parentheses, digits, etc.) from the text. Such symbols are removed because they are supposed not to carry any useful information. After the preprocessing, the original documents have been transformed into streams of words and they are given as input to the *stopping*, i.e. the removal of all of the words belonging to a predefined set called *stoplist* [8]. The *stopwords* (i.e. the words of the stoplist) are typically articles, prepositions, pronouns and other words that play a functional rather than semantic role. In other words, the stopwords are needed to make a sentence grammatically and syntactically correct, but they are not representative of the sentence content. In some cases, the stoplist can contain words that are very frequent (e.g. *information* and *retrieval* in a collection of IR articles). The reason is that a word appearing in most of the documents of a collection does not help to discriminate between them. While a stoplist containing only functional words is general and it can be applied to any kind of data, a stoplist enriched with the most frequent words of a specific corpus becomes database dependent and cannot be used for other corpora [8]. In this work, we used a generic stoplist containing 384 words. After the stopping, the number of words in a corpus is reduced, on average, by 30-50 percent.
The normalization is completed by performing the *stemming*, i.e. by replacing all of the words with their stem (e.g. *connection*, *connected* and *connection* are replaced with *connect*). The rationale behind the stemming is that the meaning of the words is carried by their stem rather than by their morphological variations [9]. In this work we used the widely applied Porter stemming [24] resulting, on average, in a reduction by around 30 percent of the lexicon size.
After the normalization, the original documents have been converted into streams of *terms*. This is not yet a form suitable for the retrieval process and it is necessary to perform indexing in order to represent the documents as vectors. Indexing can be seen as the filling of a *term-by-document* matrix A where each column corresponds to a document and each row corresponds to a term in the dictionary. An element a_{ij} of A can be written as follows:

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

While $G(i)$ depends only on a term i, $L(i,j)$ depends on both a term i and a document j. For this reason, $G(i)$ can include information extracted from the whole corpus and is called *global* weight, while $L(i,j)$ can only include information coming from a single document and is called *local* weight. The weighting scheme plays an important role in the retrieval process [7] and a large number of alternatives have been proposed for both $G(i)$ and $L(i,j)$ (see [26] for a survey). In this work we applied the so-called Okapi formula [25] which is the most effective and widely applied in current state-of-the-art systems:

$$a_{ij} = \frac{tf(i,j) \cdot \log\left(\frac{N}{N_i}\right)}{k \cdot [1 - b + b \cdot NDL(j)] + tf(i,j)} \tag{2}$$

where $tf(i,j)$ is the number of times term i appears in document j (the *term frequency*), N is the total number of documents in the database, N_i is the number of documents containing term i, k and b are hyperparameters and $NDL(j)$ is the normalized document length (the length of j divided by the average document length in the database). The logarithm is referred to as *inverse document frequency* (idf) and gives more weight to the terms appearing in few documents because they are supposed to be more discriminative.
The processing steps described in this section are performed only once for a given database. The next section describes how the matrix A is used in the actual retrieval process, i.e. how the relevant documents are indentified when a query is submitted to the system.

4.2 Retrieval

Once the document database has been indexed, the system can perform the actual retrieval task. Each time a query is submitted, the system calculates a score called *Retrieval Status Value* (RSV) for all of the documents. The documents can then be ranked according to their RSV (the better the RSV, the higher the position) and the documents relevant to the query are expected to occupy the top positions.
The RSV expression mostly depends on the indexing technique applied and, in the case of Okapi, it is the sum of the index values corresponding, for a given document d, to the query terms:

$$RSV(q,d) = \sum_{t \in Q} a_{td} \tag{3}$$

where Q is the set of the terms contained in the query q, and a_{td} is an element of the term-by-document matrix A. Since the value of a_{td} is zero when term t does not appear in document d, the above RSV expression tends to be higher when d and q share more terms. However, not all of the common terms contribute in the same way. The presence of $tf(t,d)$ at the numerator of a_{td} (see Equation 2) gives more weight to the terms appearing more times in d (they are supposed to be more representative of its content). The inverse document frequency makes the contribution of terms appearing in few documents higher (they are supposed to be more discriminative). The main limit of such an approach is that long documents tend to have higher scores because the probability of sharing terms with a query is higher [28]. The presence of the NDL in Equation 2 is aimed at smoothing such an effect by reducing the contribution of terms belonging to longer texts.

4.3 Evaluation

This section presents the metrics used to assess the retrieval performance in this work. Several measures are available in the literature, but none of them provides an exhaustive description of the retrieval results [3]. Moreover, depending on the application, some measures can

be more appropriate than others. For the above reasons, in order to give a complete description of the system performance, we apply several different measures.
Given a query q, the set of the documents relevant to it is $R(q)$ and the set of the documents identified as relevant by the system is $R^*(q)$. The two fundamental measures in IR are *Precision*:

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

and *Recall*:

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

Precision can be considered as the probability that a document identified as relevant by the system is actually relevant, while Recall can be thought of as the probability of a relevant document being identified as such by the system. The value of $\pi(q)$ is often calculated in correspondence of a predefined set of ρ values (typically 10, 20,. . ., 100 percent) resulting in the so called *Precision vs Recall* curves. In order to obtain such a curve for a query set rather than for a single query, it is possible to perform a *macroaverage*, i.e. for each predefined value of ρ the plotted Precision is the average of the π values obtained for different queries:

$$\pi^M = \frac{1}{|T|} \sum_{q \in T} \pi(q) \tag{6}$$

where T is the query set.
The Precision vs Recall curves give an overall view of the retrieval performance, but they are difficult to use in comparisons between different systems. For this reason two different techniques have been proposed in order to obtain a single number assumed to be representative of the whole curve [3]. The first leads to the *Average Precision* (avgP) and consists in calculating the average value of the Precision along the curve. The second leads to the *Break Even Point* (BEP) and consists in calculating the Precision at the curve point where $\pi(q) = \rho(q)$. The BEP can be easily obtained by measuring the Precision at the ranking position corresponding to the number of relevant documents $|R(q)|$. In fact, if $\pi(q) = \rho(q)$, then $|R^*(q)| = |R(q)|$ (see Equations 4 and 5). An ideal system (i.e. a system able to put all of the relevant documents at the top of the ranking) has BEP 100 percent. The lower the BEP, the more a system is far from such an ideal situation. Both avgP and BEP can be averaged over all of the queries in a set T in order to evaluate a retrieval task composed of different queries.
Since most of the IR systems provide the user with the ranking of the documents (ordered following their RSV), the evaluation can be performed in terms of Precision at position N, i.e. the percentage of relevant documents in the first N positions of the ranking. Such a measure is closely related to the perception of the users that typically, after submitting a query, check the documents following the ranking provided by the system. The more relevant documents appear at the top of the ranking (i.e. the higher the precision at position N), the better the user perception. A different Precision at position N plot can be obtained for each query in a set T and then, through a macroaverage, a single plot can be obtained for T as a whole.

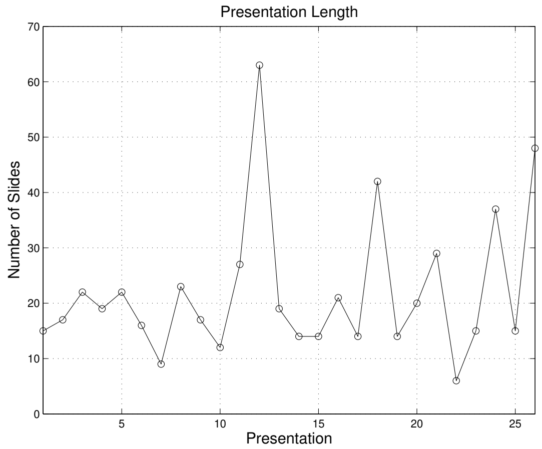

Figure 6: Number of slides. The plot reports the number of slides contained in each presentation.

A different evaluation metric is used when each query has only a single relevant document (such a task is often referred to as *Known Item Search*). In this case, what is important is the ranking position occupied by the relevant documents. For this reason, the evaluation is made by calculating (for a set T of queries), the percentage of times a relevant document appears in the top, top two, . . ., top N positions of the ranking. The result is the cumulative distribution of the ranking positions of the relevant documents.

5 Experiments and Results

This section presents the experiments performed in this work. The slides displayed during a conference have been acquired with a framegrabber and transcribed with the OCR system described in Section 3. The IR system presented in Section 4 was then used to perform several retrieval tasks. In the next subsections the slide database, the OCR performance and the retrieval experiments are presented in detail.

5.1 The data

The slide database used in this work has been collected during a conference (Machine Learning in Multimodal Interfaces, MLMI) held in June 2004 [4]. The slide authors were not aware of our experiments and they prepared their slides without respecting any constraint. In other words, the data was not created in a laboratory, but collected in a real working environment. They thus represent a realistic benchmark with respect to similar situations. In total, we collected 26 presentations containing 570 slides (the number of slides per presentation is shown in Figure 6). The average number of slides is 21.9 (minimum and maximum are 6 and 63 respectively). All of the slides have been acquired with a framegrabber (i.e. a device capturing

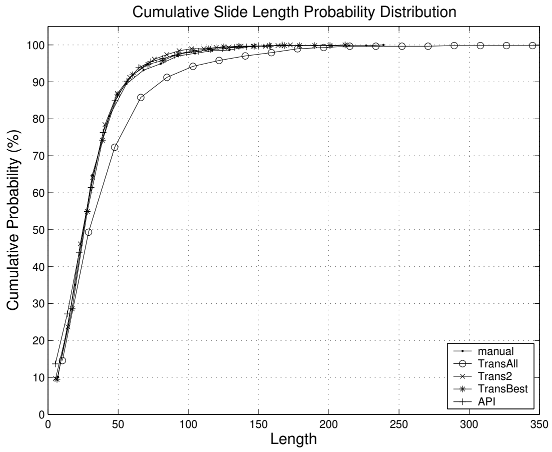

Figure 7: Cumulative slide length probability distribution. The plot shows the cumulative probability distribution of the number of terms per slide. The distribution curve for the TransAll transcription is lower than the other since the use of multiple OCR outputs tends to make the documents longer.

the images displayed through a projector) and compressed in jpeg, resulting into 570 images of dimension 1036×776 pixels (91.2 dpi resolution).

The text contained in the presentations has been transcribed in three different ways. The first is by manually typing the content of the slides (this version is used as reference and will be referred to as *manual*). The second is by applying the different versions of the OCR system described in section 3 to the slide images captured through the framegrabber (the transcriptions will be referred to as *Trans2*, *TransAll* and *TransBest*). The third is by using software converting the electronic versions of the slides (i.e. the Powerpoint or PDF files) into text (this version will be referred to as *API*).

Figure 7 shows the cumulative distribution of slide lengths (after stopping and stemming). The number of terms is an important parameter because the probability of a relevant document being identified as such by a retrieval system tends to increase with its length [28]. The reason is that if a document contains many terms, the probability that a query contains one of them is higher. This can be an important source of problems for this work because slides contain often figures, plots, pictures and other kinds of visual information that limit the amount of space left to text. The average number of terms per slide ranges in fact from 33.6 (Trans2) to 48.3 (TransAll) while it is 217.1 ($\sim$4 to $\sim$7 times higher) for the Wall Steet Journal Corpus [22] (one of the main IR benchmarks).

OCR method	slide		presentation		database	
	TR	TP	TR	TP	TR	TP
Trans2	72.4	77.3	67.5	77.0	71.4	77.4
TransBest	77.0	78.4	72.6	77.3	76.7	79.0
TransAll	80.9	65.5	76.2	62.3	80.8	62.0
API	81.1	89.5	75.3	87.1	80.8	88.9

Table 1: Term recall TR and term precision TP averaged other slide documents, presentation documents, or computed on the whole database.

5.2 OCR performance evaluation

In this section we evaluate the quality of the OCR transcripts. To do so, we consider as performance measures the *term recall* TR and *term precision* TP, which are defined by:

$$TR(d) = \frac{\sum_t \min\left(tf^{\star}(t,d), tf(t,d)\right)}{\sum_t tf(t,d)} \tag{7}$$

and

$$TP(d) = \frac{\sum_t \min\left(tf^{\star}(t,d), tf(t,d)\right)}{\sum_t tf^{\star}(t,d)} \tag{8}$$

where $tf(t,d)$ denotes the number of times the term t really appears in the document d (d will be either a slide, a presentation, or the whole database), and $tf^{\star}(t,d)$ denotes the number of times the term t appears in the transcript of the document d. The term recall can be interpreted as the percentage of terms in the document that have been correctly recognized by the OCR, while the term precision indicates the proportion of recognized terms that are actually true. Although the use of word recall or word precision measures would have reflected more directly the intrinsic OCR performance, the use of the term recall and term precision measures is more adequate in our context. In fact, from a retrieval point of view, we do not care for instance whether stop words were well recognized or not, as this has no influence on our task. Hence, while still being characteristic of the OCR performance, we can expect the proposed measures to reveal the discrepancy existing between the document representations used in the retrieval process and built from either the true transcript (we use the manual annotation as reference) or from the API or OCR transcripts.

Table 1 provides the average term recall and precision computed over either slides, presentations or on the whole database. The overall values are good, showing that around 3 out of 4 terms are correctly recognized by the OCR systems, which means an average of 25 correct terms per slide document. These numbers, however, hide a large recognition variance depending on the slide type. While slides containing plain text only usually have term recall above 85%, slides containing images, plots or screen shots have lower and more diverse TR values. This diversity of recognition according to the slide type can be appreciated by looking at the distribution of the slide TR, Figure 8: while more than 50% of the slides have a term recall higher than 85%, 10% of them have a term recall lower than 50%. It is interesting

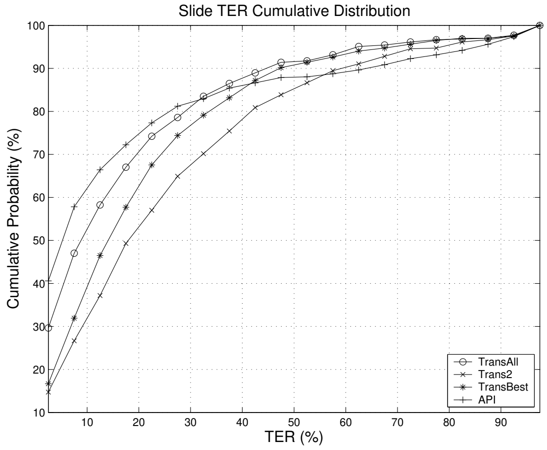

Figure 8: Cumulative distribution of the Term Error Rate (TER), defined as $1 - TR$, over the slide documents.

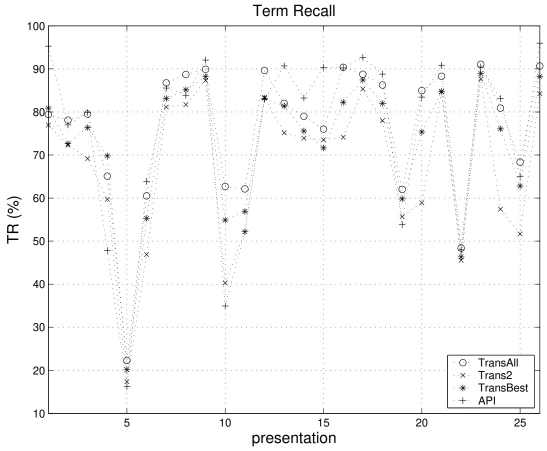

Figure 9: Term recall value computed for each presentation.

to note that the different types of slides are not evenly spread across presentations. This can be illustrated by looking at the variabilities in the performance computed per presentation, shown in Figure 9. In the extreme case of the 5th presentation (about Mountains, Learning Media and Interaction), 11 out of the 22 slides contain only geographical maps, with many names embedded in clutter, and few words in the remaining slides. Therefore, the number of hard to recognize terms largely dominates, leading to the poor term recall we reported, between 18 and 23%. As other difficult cases, presentation 10 and 11 contain screen captures of dialogue interfaces (presentation 10) and meeting/presentation browsers (presentation 11) comprising large amount of small size text corrupted by jpeg noise, conducting to medium recognition rates. At the other end of the spectrum presentations like the 16th or the 23rd contain mainly slides with text and the OCR achieves up to 90% term recall on these. Alltogether, these examples illustrate that presenters may have completely different styles with some of them being more 'visual', or that there are different presentations types (e.g. presentation reporting only facts). It is thus important to build a system that is able to handle these different kinds of presentations.

The comparison of the OCR performance with the API results shows that the difference between the two methods is not so large overall. While the OCR transcriptions are noisier, as indicated by the lower term precision, the term recall of the best performing OCR is equivalent to the API one (cf Table 1). Still, as expected, the API and OCR systems have different behaviours. While the API is almost error-less on text slides, it misses most of the text on slides with images, diagrams or plots, and performs some errors on these. This can be observed by noting the differences in the performance per presentation, Figure 9, which depends on the content type, as commented in the previous paragraph. Alternatively, we can notice this from the curves in Figure 8, which show that the API is performing better on slides with high term recall -mainly text slides-, but perform worse on the slides with medium to low TR values typical of slides with embedded figures.

Finally, comparing the different OCR systems between each other, we can draw the following conclusions. First, the standard approach consisting of binarizing the text image (Trans2, see 3.2.2) is not performing as well as the two other methods. For instance, the method that considers alternative numbers of grayscale classes in the input text image and selects the best OCR output (TransBest) from the set of generated candidate strings improves significantly (by approx. 5%) the term recall with respect to the Trans2 OCR, without any degradation in the term precision measure. This demonstrates the validity of both the use of the multi-class strategy and the string selection scheme. Second, compared with the TransBest approach, the TransAll strategy further improves the term recall (by approx. 4%), but this is done at the expense of the term precision, which drops by around 16%, from 78% to 62%. This effect is understandable, as this method consists of adding complementary transcripts from different multi-class segmentations. A net effect is to produce longer transcripts (cf previous section) in which the additional terms (w.r.t. TransBest) are less reliable. A gross analysis of the numbers indicates that only 20% to 25% of the added terms are indeed correct. However, as most of the erroneous added terms do not correspond to true terms, and are not susceptible of being part of a query, their impact on the Retrieval Status Value of documents for a given query should be negligeable in principle. Hence, from a retrieval point-of-view, such a strategy should lead to better results.

5.3 The Retrieval Tasks

The effectiveness of a retrieval system is measured through a *retrieval task*, i.e. a set of queries (designed to evaluate a certain aspect of the system performance) and related *relevance judgements* (the list of the documents relevant to each query). In this work, we created three retrieval tasks that will be referred to as *general*, *author* and *image* respectively. The first is composed of queries written in natural language (e.g. *multimodal interaction in meetings*) and aimed at finding all documents concerning a certain topic (the documents relevant to each query have been identified by human assessors). This task measures essentially the quality of the matching measure used to calculate the similarity of queries and documents.
The second task uses as query the last names of the authors and a slide is considered rel-

Task	queries	<length>	<relevant>
general	46	3.5	6.4
author	72	1	2.4
image	85	3.6	1

Table 2: Retrieval tasks statistics. This table reports, for each retrieval task, the number of queries (second column), the average query length (third column) and the average number of relevant documents (fourth column).

evant if it contains the name used as query. This task is essentially a keyword search and it is designed to measure the effectiveness of the OCR system. In fact, if the author name is correctly transcribed, the relevant slide is certainly retrieved, while if the author name is misrecognized, the relevant slide is certainly missed. The use of the whole set of author names avoids the potential bias due to arbitrary selection of keywords.
The third task is built by using as queries a list of terms appearing, for a given slide, in figures, but not in the text of that slide (e.g. the axis labels of a plot when they appear only in the plot). The task has been created by randomly selecting 85 figures containing text and by extracting from each of them a few keywords. All of the keywords appear only in the figure and not in the text appearing in the same slide. The goal of this task is to measure the effectiveness of the OCR in capturing not only the main body of text introduced in the slide (easily accessible through programs converting electronic versions of the slides into text), but also the text appearing as a bitmap in the figures. For each query, we considered as relevant only the slides containing the image the keywords were extracted from.
Table 2 reports, for each task, the number of queries, the average query length (in words) and the average number of relevant documents per query. The query length is an important parameter because long queries tend to have better results. The use of too many keywords makes it in fact more probable to match the terms in the slides leading to unrealistic high performance. The average number of relevant documents per query gives an idea of how hard is the retrieval task: the lower the percentage of the corpus accounted by the documents relevant to a query, the lower the probability of retrieving them by chance. A task is considered difficult when no more than 2 percent (or less) of the documents are relevant, on average, to a query [3]. Such a condition applies to all of the retrieval tasks proposed in this work.

5.4 General Task Results

This section presents the results obtained over the general task (see Section 5.3). The goal of this task is to find all of the documents in the corpus that are relevant to the information need expressed through a natural language query. Figure 10 reports the Precision vs Recall curves. The Precision achieved is higher on manual and API transcriptions (especially at high Recall) than on OCR based transcriptions. On the other hand, from a user's point of view, such a difference does not require too much additional effort in order to find all of

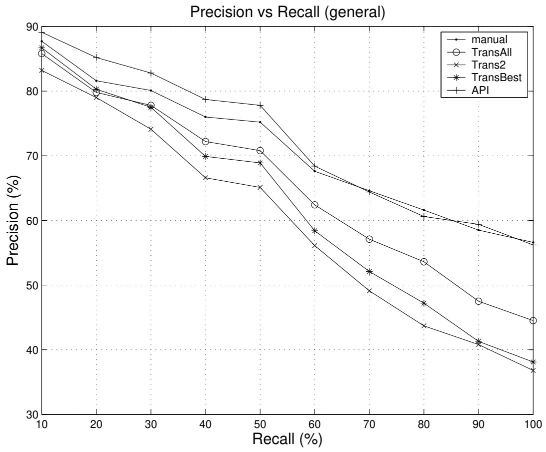

Figure 10: General. The plot shows the Precision vs Recall curves for the general task. The curves are reported for both manual and automatic transcriptions.

the relevant documents. At ρ=50 percent, the π values range from 65.1 percent (Trans2) to 77.8 percent (API). Since the average number of relevant documents per query is 6.4 (see Table 2), this means that the first 3 relevant documents can be found in the top 4 (API) to 5 (Trans2) positions. In other words, in order to find half of the relevant documents, a user must browse, on average, four documents when using the API and manual transcriptions and five documents when using the OCR based transcriptions. This means that in the case of the OCR transcriptions, the user must browse only one additional document (on average) in order to have the same performance as in the case of the manual or API transcriptions.
By applying the same considerations for the ρ=100 percent point, it is possible to say that the number of documents to be browsed along the ranking in order to find all of the relevant documents is 11 for manual and and API transcriptions, 14 for the TransAll transcription and 17 for Trans2 and TransBest transcriptions. Since most of the IR interfaces present the retrieval results in pages containing 10 documents (this is the case for the most popular web search engines), this means that all of the transcriptions require the user to go to the second page in order to find all the documents s/he need. The additional effort required to the user because of the recognition errors can thus be considered, in our opinion, acceptable.
This can be more easily observed in the Precision at top N curves (see section 4) shown in Figure 11. The plots report the average percentage of relevant documents appearing in the first N positions of the ranking after the retrieval process. The differences are never higher than 10 percent. At N=5, the π values range from 44.7 (Trans2) percent to 52.3 (API) percent and this means that the average number of relevant documents in the first five positions is between 2.2 (Trans2) and 2.6 (API). The same difference can be observed at N=10, where the average number of relevant documents goes from 3.3 (Trans2) to 3.8 (API). The performance of the system in the top ranking positions is thus only moderately affected by the presence of the recognition errors.

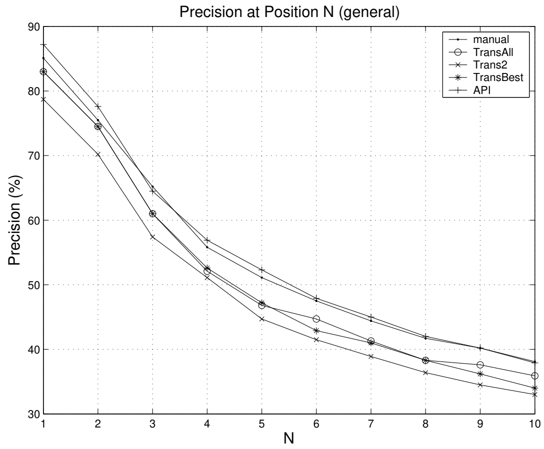

Figure 11: Precision at Position N. The figure shows the Precision at Position N plots for the different transcriptions.

Transcription	AvgP (%)	BEP (%)
manual	71.3	63.4
TransAll	66.0	58.7
Trans2	60.6	55.3
TransBest	62.3	58.1
API	72.8	65.8

Table 3: Average Precision and Break Even Point. This table reports the AvgP and BEP values achieved for the general task when using different transcriptions.

Similar conclusions can be drawn from the AvgP and BEP (see section 4) values reported in Table 3. The highest performance difference in terms of AvgP is 12.2 percent (between Trans2 and API), but if we consider the best OCR transcription (TransAll) the difference is only 6.8 percent. This means that, on average, at each Recall level, the number of documents to be browsed is increased by only 6.8 percent when passing from the API to the TransAll transcription. Similar considerations can be made for the BEP which accounts for the Precision at the ranking position equal to the number of relevant documents (see section 4). The highest difference is 12.5 percent (between Trans2 and API) and, since the average number of relevant documents is 6.4, it corresponds to 0.8 documents.

All of the performance metrics used show that the degradation introduced by the OCR errors leads to moderate effects on the ranking produced by the retrieval system. For this reason the impact on the user effort required to collect the whole set of relevant documents is not significant. The OCR based transcriptions can thus be a reliable alternative to the use of APIs to extract the text from slides when the goal is to perform retrieval.

Transcription	Recognition (%)	Retrieved (%)
manual	88.7	88.7
TransAll	87.6	90.3
Trans2	82.5	87.5
TransBest	84.8	87.5
API	86.9	86.9

Table 4: Author task results. This table reports the percentage of author names correctly recognized (second column) and the number of relevant documents given an RSV different from zero (third column).

5.5 Author Task Results

In the author task, the queries correspond to the last names of the authors appearing on the first slide of each presentation (each name is used separately). Each slide containing the name of an author is considered relevant to the corresponding query and the task is thus a keyword search rather than a retrieval experiment. The interest of such a task is that it allows a more explicit evaluation of the effect of transcription errors. In fact, a relevant slide can be retrieved if and only if the keyword in the slide is correctly recognized. The final performance is thus determined essentially by the transcription quality while in the general task an important role was played also by the retrieval algorithms.

Table 4 shows the percentage of author names that have been transcribed in the same way as they appear in their corresponding query after normalization and indexing. Even in the case of the manual and API transcriptions, some names are not correctly transcribed. The reason is the preprocessing (see section 4). Some names are written on the slides using the initial of the first name like in *J.Smith*. The preprocessing removes the points and transforms such an expression into *jsmith*. The reason is that the points often appear in acronyms (e.g. *U.S.A.*) that must be kept as a single term. Since the query is the last name (*Smith* in the case of the example), this leads to some errors also for for manual and API transcriptions. In the case of the API, some more errors are due to the presence of fonts and symbols that create problems. The OCR transcriptions are affected by the same preprocessing effects and by some misrecognitions.

In a keyword search task, the relevant documents (i.e. the documents that contain the keyword submitted as query) appear always at the top of the ranking and they are the only documents with an RSV different from zero. For this reason, the Precision vs Recall curves are not appropriate and it is better to use, as a measure of the performance, the percentage of relevant documents that are retrieved (i.e. that have a RSV different from zero), which is reported in Table 4. While in the case of API and manual transcriptions this simply corresponds to the percentage of correctly recognized keywords, in the case of the OCR systems there is a small improvement due to the fact that points are sometimes recognized as spaces (*J.Smith* is thus transcribed as *J Smith*) and the preprocessing problem described above does not take place.

Transcription	Retrieved (%)
manual	98.8
TransAll	88.2
Trans2	76.5
TransBest	87.0
API	45.9

Table 5: Percentage of retrieved relevant slides (image task). This table reports the percentage of slides where at least one of the query terms embedded in figures have been correctly recognized.

Also in this task, the use of OCR transcriptions leads to retrieval performances comparable with those obtained over manual and API transcriptions.

5.6 Image Task Results

In the most common slidewares, the text can be inserted only through apposite functions. This allows one to store the text as it is typed by the slide author and to avoid (if an API is available) a recognition process in order to extract it. For this reason, the API based converters lead to transcriptions that are almost exempt of errors. On the other hand, the authors insert many texts through figures (diagrams, plots, maps, logos, etc.) that are often represented as bitmaps and where the written information may only be accessed through an OCR process. The goal of the image task is to measure the effectiveness of the OCR system used in this work in accessing textual information.

For each query we consider to be relevant only the slide containing the figure from where it has been extracted. In some cases, the queries contain terms that are present also in other slides, so the relevant query is not always at the first position of the ranking. At the same time, if all of the query terms are incorrectly recognized, the relevant slide is given a null RSV and it is not retrieved. Table 5 reports the percentage of relevant slides with RSV higher than zero, i.e. where at least one of the query terms has been correctly recognized. The results show that the OCR is almost twice as effective as the API in extracting the text in pictures. The reason is that the API cannot recognize the texts available as bitmaps in embedded pictures, while the OCR can.

The same difference can be observed in the retrieval results. The plots in Figure 12 show the percentage of relevant documents ranking in the first N positions. The curves can be also interpreted as the cumulative probability distributions of relevant documents' ranking positions. The relevant document is at the top of the ranking around 80 percent of the time for the manual transcriptions, around 55 percent of the time for the OCR based transcriptions and around 35 percent of the time for the API. At the tenth position (i.e. at the end of the first results page in many IR system imterfaces), the percentage of relevant documents rises to 94.1 percent, 80 percent and 43.5 percent for manual, TransAll and API transcriptions respectively. The OCR is thus almost two times more effective than the API based system in

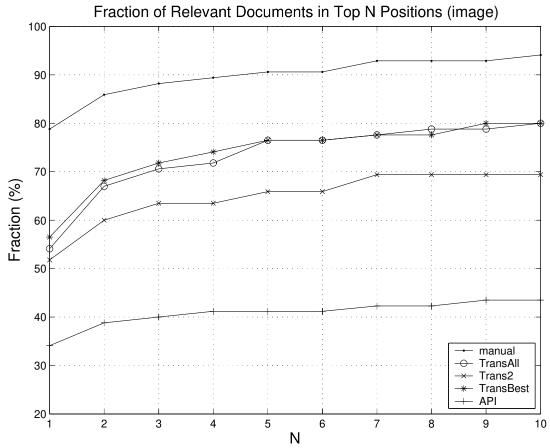

Figure 12: Fraction of relevant documents at position N. The plots reports the percentage of relevant documents appearing at the first N positions of the ranking.

indexing the text contained in figures.

6 Conclusions

Presentation slides represent a valuable source of information. They are often the only record left after a presentation is given and they are used more and more to replace reports and memos as a mean of communication in large organizations [30]. Limited efforts have been made, to our knowledge, to index and retrieve them in order to effectively use the information they contain. This paper presented retrieval experiments performed over slide transcriptions obtained by first capturing the slide images (with a framegrabber) and then by applying an OCR process. The results show that the transcription errors affect only to a limited extent the retrieval results. In other words, the performance achieved on such transcriptions is close to the one achieved over transcriptions obtained with an API based system able to capture without errors the text inserted in slides. Moreover, the OCR based system outperforms significantly the API based one in extracting and capturing the text embedded in figures and images (often not accessible to APIs).
The use of an OCR rather than API based transcription system has at least two main advantages. The first is that the APIs are expensive and need to be changed each time a different slideware is used to create the presentation. Moreover, proprietary formats are subject to change and this makes the APIs obsolete after a relatively short lifespan. The OCR process is robust to the above problems because it works on slide images stored in a format (jpg in our case) independent of the slideware used to create the presentations. The second is that the use of an OCR process allows one to index the text (mostly not accessible to APIs) embedded in figures.
The system presented in this paper can be the starting point for several directions of future

work. The first is to use the slides to index the talks where they have been used. In fact they can be synchronized (through the framegrabber) to video segments recorded with a video-camera. By retrieving the slides it will be thus possible to retrieve the corresponding video segments. The second is to enrich the slide indexing with information sources like layout (bullett lists, position of the text with respect to images, etc.), presence of visual elements (images, plots, diagrams, etc.), animations, videos, etc.. Moreover, the slides can be used together with other information streams (e.g. the speech recording) to index the presentations they are extracted from.
The above possibilities for future work are far from being exhaustive and the investigation of the problem can lead to new applications not considered so far. This is, in our opinion, one of the most interesting aspect of our work.

References

[1] G. Abowd. Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, 38(4):508–530, 1999.

[2] A. Amir, G. Ashour, and S. Srinivasan. Toward automatic real time preparation of online video proceedings for conference talks and presentations. In *Proceedings of 34th Hawaii International Conference on System Sciences*, pages 1662–1669, 2001.

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

[4] H. Bourlard and S. Bengio, editors. *Machine Learning for Multimodal Interaction: First International Workshop, MLMI'2004*, volume 3361 of *Lecture Notes in Computer Science*. Springer Verlag, 2005.

[5] D. Chen, J-M. Odobez, and H. Bourlard. Text Detection and Recognition in Images and Videos. *Pattern Recognition*, 37(3):595–609, March 2004.

[6] S.G. Deshpande and J.N. Hwang. A real-time interactive virtual classroom multimedia distance learning system. *IEEE Transactions on Multimedia*, 3(4):432–444, 2001.

[7] S.T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments & Computers*, 23:229–236, 1991.

[8] 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.

[9] 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.

[10] C.K. Gan and R.W. and Donaldson. Adaptive silence deletion for speech storage and voicemail application. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 36(6):924–927, 1988.

[11] L. He, E. Sanocki, A. Gupta, and J. Grudin. Audio-summarization of audio-video presentations. In *Proceedings of ACM International Conference on Multimedia*, pages 489–498, 1999.

[12] M. Hearst. TextTiling: segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[13] A. K. Jain and B. Yu. Automatic text localisation in images and video frames. *Pattern Recognition*, 12(31):2055–2076, 1998.

[14] S.X. Ju, M.J. Black, S. Minneman, and D. Kimber. Summarization of videotaped presentations: automatic analysis of motion and gesture. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):686–696, 1998.

[15] T. Kawahara, M. Hasagawa, K. Shitaoka, T. Kitade, and H. Nanjo. Automatic indexing of lecture presentations using unsupervised learning of presumed discourse markers. *IEEE Transactions on Speech and Audio Processing*, 12(4):409–419, 2004.

[16] H. Li, D. Doermann, and O. Kia. Automatic text detection and tracking in digital videos. *IEEE Trans. Image Processing*, 9(1):47–156, 2000.

[17] W Li, H. Tang, and Z. Zhu. Automated registration of high-resolution images from slide presentations and whiteboard handwritings via a low-cost digital video camera. In *The 2^{nd} IEEE International Workshop on Image and Video Registration*, 2004.

[18] R. Lienhart and A. Wernicke. Localizing and segmenting text in images and videos. *IEEE Trans. on Circuits and Systems for Video Technology*, 12(4):256–268, 2002.

[19] S. Mukhopadhyay and B. Smith. Passive capture and structuring of lectures. In *Proceedings of ACM International Conference on Multimedia*, pages 477–487, 1999.

[20] C.W. Ngo, T.C. Pong, and T.S. Huang. Detection of slide transition for topic indexing. In *Proceedings of IEEE Conference on Multimedia and Expo*, pages 533–536, 2002.

[21] W. Niblack. Slidefinder: a tool for browsing presentation graphics using content-based retrieval. In *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 114–118, 1999.

[22] D.B. Paul and J.M. Baker. The design for the wall street journal-based CSR corpus. In *Proceedings of DARPA SLS Workshop*, 1992.

[23] M.G. Pimentel, Y. Ishiguro, G. Abowd, B. Kerimbaev, and M. Guzdial. Supporting educational activities through dynamic web interfaces. *Interacting with Computers Journal*, 13(3):353–374, 2001.

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

[25] S.E. Robertson, S. Walker, and M. Beaulieu. Experimentation as a way of life: Okapi at TREC. *Information Processing and Management*, 36:95–108, 2000.

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

[27] T. Sato, T. Kanade, E. K. Hughes, M. A. Smith, and S. Satoh. Video OCR: indexing digital news libraries by recognition of superimposed caption. *Multimedia Systems*, 7(5):385–395, September 1999.

[28] A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document length normalization. *Information Processing and Management*, 32:619–633, 1996.

[29] T. Syeda-Mahmood and S. Srinivasan. Detecting topical events in digital video. In *Proceedings of ACM International Conference on Multimedia*, pages 85–94, 2000.

[30] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2001.

[31] F. Wang, C.W. Ngo, and T.C. Pong. Synchronization of lecture videos and electronic slides by video text analysis. In *Proceedings of ACM International Conference on Multimedia*, pages 315–318, 2003.

[32] Christian Wolf, Jean-Michel Jolion, and Francoise Chassaing. Text localization, enhancement and binarization in multimedia documents. In *Proc. Int. Conf. on Pattern recognition*, pages 1037–1040, Quebec City, Canada., August 2002.

[33] V. Wu, R. Manmatha, and E. M. Riseman. Textfinder: An automatic system to detect and recognize text in images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(20):1224–1229, 1999.

[34] D. Zhang and J.F. Nunamaker. A natural language approach to content based video indexing and retrieval for interactive e-learning. *IEEE Transactions on Multimedia*, 6(3):450–458, 2004.

[35] Z. Zhu, C. McKittrick, and W. Li. Virtualized classroom - automated production, media integration and user-customized presentation. In *The 4th International Workshop on Multimedia Data and Document Engineering*, 2004.