Document Analysis Techniques for the Infinite Memory Multifunction Machine

Jonathan J. Hull, Dar-Shyang Lee, John Cullen, and Peter Hart
Ricoh California Research Center
Menlo Park, CA
hull@crc.ricoh.com

Abstract

A system that saves a digital copy of every document that users copy, print, or fax, without asking the user, has recently been proposed. Referred to as the Infinite Memory Multifunction Machine (IM^3), this system solves most of the problem of lost documents. However, because of the indiscriminate way it captures data, it is important that users have easy-to-use retrieval tools.

Two document analysis techniques are described that simplify retrieval from large collections like the IM^3. One technique detects duplicates or versions of a document. Another method automatically files a document in a hierarchy familiar to a user. Experimental results are presented that illustrate the performance of each method.

1. Introduction

The Infinite Memory Multifunction Machine (IM^3) is a document storage and retrieval system that solves a large portion of the problem of lost documents [6]. It captures a copy of every printed, copied, or faxed document generated in an office. This guarantees that almost any document a user needs will be available when they need it. This concept was developed after it was observed that even though the obvious method for document capture, namely scanners, were commonly available, they were not commonly used.

The IM^3 makes document capture effortless by saving an electronic copy of *every* document that users copy, print, or fax. Furthermore, users are not asked whether any particular document should be captured -- no conscious decision is required at capture time. Thus, every person in an office that copies, prints, or faxes a document automatically contributes data to the IM^3.

The design for the IM^3 prototype system that was implemented and tested at the authors' laboratory is shown in Figure 1. It is based on a typical office environment in which PC's, Mac's, Unix workstations, digital copiers and printers are interconnected on a local area network. When users print a document, it is first sent to the print server. In addition to sending it to the appropriate printer, an electronic copy is transferred to the IM^3 server. OCR is automatically performed and the document is indexed for later retrieval. Copiers and fax machines work similarly.

Documents stored in the IM^3 are accessed with a web browser. Each user has a home page that provides a portal to their document collection. Users can restrict access to their document collections, if they desire. Also, individual documents can be encrypted. A number of techniques are provided for search and retrieval. These include full text search and various methods for browsing based on the dates when documents were captured.

The rest of this paper describes two document analysis methods that provide additional retrieval techniques for users of the IM^3 system. One algorithm detects duplicates and versions of a given document.

Another algorithm automatically files a document in multiple locations in an existing hierarchy of categories. Later, when users search for a document, they can browse this hierarchy to find it. Such an automatic filing technique is especially useful in the IM^3 because it captures documents indiscriminately. Users cannot categorize a document before it is saved, as they might do in other document storage and retrieval systems.

2. Duplicate Detection

Duplicate documents can be a significant problem in large collections. Ideally, a duplicate detection algorithm can find both exact duplicates, which have exactly the same content, and partial duplicates which have a large percentage of their text in common. Locating exact duplicates could reduce the storage required for an IM^3 database. Finding partial duplicates would allow users to easily find other versions of a given document.

We assume that all document images are compressed with a "symbolic" technique (e.g., JBIG2 [5]). Features are extracted directly from the compressed version of two document images. A comparison procedure determines

10th Int. Workshop on Database & Expert Systems Applications, Florence, Italy, pp. 561-565, Sept. 1-3, 1999.

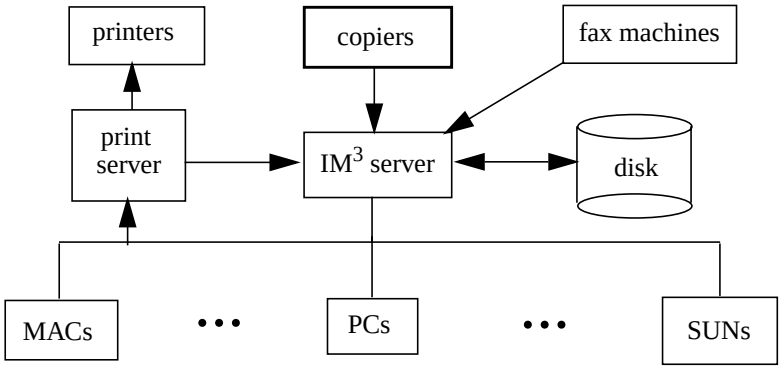

Figure 1. IM^3 system design.

whether the feature descriptions are similar enough for the original documents to be duplicates.

Symbolic compression (for binary document images) first clusters connected components (that often correspond to isolated characters). A unique identifier is then assigned to each cluster. The compressed document contains one image for each cluster and the sequence of identifiers for the connected components in the original image. This sequence of identifiers corresponds to the sequence of occurrence of characters in the original document.

An idealized example of a symbolically compressed (similar to JBIG2) document image is shown in Figure 2. An original document image is shown (a) as well as its compressed form (b). The unique letters in the original image are represented as individual sub-images and numeric identifiers at the top of (b). The sequence of identifiers shown in Figure 2 (c) encodes the order in which the corresponding sub-images occurred in the original image (a). For example, "0 1 2 3" are the first four sub-images in this sequence. They correspond to the first four letters in the image, "*P A L O*". The x-y locations of the sub-images and image residual data are also encoded in the compressed format.

The characteristic of symbolic compression that we use for duplicate detection is the sequence of cluster identifiers ("0 1 2 3 1 2 4 3 0 3 2 5 6 7" in Figure 2 (c)). This sequence encodes a representation for the text in the original document. Since each cluster (for the most part) corresponds to a single character, we can treat the sequence of cluster identifiers as a substitution cipher.

A substitution cipher replaces one character with another to produce an enciphered message. The original plain text can be recovered by a deciphering algorithm that computes the pairwise correspondence between symbols in the enciphered message and plain text characters. For the example shown in Figure 2, this correspondence is {(0,P), (1,A), (2,L), (3,O), (4,T), (5,I), (6,C), (7,Y)}.

We apply a deciphering algorithm to the sequence of cluster identifiers. It computes a pairwise correspondence between cluster identifiers and letters. This correspondence is used to recover the text in the original document image. This is essentially OCR'ing the document without actually applying any OCR techniques. A similar idea was first proposed in [2]. Our method takes advantage of the image preprocessing done by the symbolic compression technique. Also, we developed a new algorithm for substitution cipher decoding that takes into account characteristics of symbolic clustering in document images [7].

The deciphering algorithm reads the sequence of cluster identifiers from a symbolically compressed document image and uses character transition probabilities and a hidden Markov model to estimate the text that appeared in the original document. There might not be a decision for every character and all the decisions might not be correct. However, enough of the text is usually output that accurate duplicate detection can be performed.

The text strings extracted from two documents are compared by calculating a weighted sum of the frequencies of the *conditional* n-grams they have in common. A conditional n-gram is a sequence of n characters where each character satisfies a predicate. For example, we use a predicate that every character must follow a space. Conditional trigrams are composed from characters that follow three consecutive spaces. If the weighted sum of conditional ngrams that occur in two documents exceeds a threshold, then the documents are declared to be duplicates.

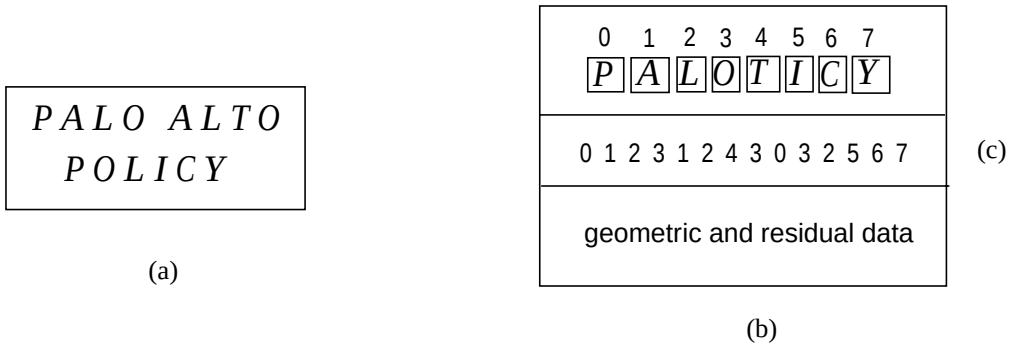

Figure 2. Depiction of symbolic compression, showing an original image (a) and the compressed image (b).

The experimental performance of the deciphering algorithm and the conditional n-gram method for text string comparison were investigated. The deciphering algorithm was trained with character transition probabilities calculated from a corpus of over 100,000 words of English. It was tested on sequences of cluster identifiers extracted from three all-text images in the University of Washington (UW) database [9]. The mgtic algorithm [10] was used as the symbolic compression algorithm. Between 80% and 95% of the characters in the test documents were deciphered correctly.

The performance of the conditional n-gram method for text string comparison was tested on the 979 documents in the UW database. This database contains 146 pairs of duplicate documents. Each member of a pair had been scanned from a different generation photocopy of the same document. Test data was constructed by adding noise to the ASCII truth files that simulated a 90% correct decode rate by the deciphering algorithm.

Each document was compared to the other 978 documents by calculating a similarity score using a weighted sum of the frequencies of the conditonal n-grams they have in common. A sorted list of the 10 documents with the highest similarity scores was output. The most similar document is at the top of the list. Ideally, this is a duplicate of the original document, if it exists in the database.

The results showed that conditional trigrams provide a 100% correct rate in duplicate detection. This compares to the 81.85% correct rate achieved by non-conditional trigrams, in the first choice, and 97.95% in the top 10 choices. Non-conditional 5-grams also produced a 100% correct duplicate detection rate. However, this was at the cost of an almost 40:1 increase in storage.

3. Automatic Document Filing

An automatic document filing technique assigns a given document to a category that characterizes its content [1, 4]. This improves retrieval by allowing users to browse the documents in a small number of categories. They only need recall some general characteristics of a document, not necessarily any of the particular keywords it contains. Also, the categories should be familiar to the user so that at retrieval time they can easily guess where they will find a particular document.

We hypothesize two characteristics that are useful for automatic filing and retrieval. One is the text in the document. This describes its message or meaning. The other is the visual appearance of the document. There is ample evidence for the utility of both descriptions. For example, the text-based categorization of yahoo.com is popular for browsing the world wide web. Also, image-based techniques have been frequently proposed for the retrieval of photos. Furthermore, it is well known that people can remember large numbers of images.

We propose an algorithm for automatic filing that combines text-based and image-based methods. This allows us to categorize a document both by the topics it discusses and its physical layout. This algorithmic design was chosen because of the observation that a text-based feature set is useful for grouping documents that are about similar topics. For example, all documents that concern “alchemy” could share a common set of keywords. However, a text-based feature set might not be able to distinguish a scientific paper about alchemy from a business letter about alchemy. This characteristic is often apparent from the physical layout of the document image.

An outline of the proposed algorithm for automatic filing is shown in Figure 3. An input document (Figure 3 (a)) is passed to both a text-based technique and an image-based technique. The text-based technique applies OCR to the image. A naive Bayes classifier uses the words of text to rank the categories that it most likely belongs to [8]. In our algorithm, the categories are directories in an existing hierarchy of documents. Each directory is represented by a feature vector composed of the words and their frequencies in the directory's documents. The naive Bayes classifier uses these feature vectors for its training data.

The image-based technique computes a feature vector that describes the visual appearance of the input document. The "interest points" are located and a vector is composed from their geometric layout (in a fixed grid) on the page. Interest points are locations in an image where there is a high degree of local variation, e.g., the edges of characters [3]. A nearest neighbor classifier uses this feature vector to rank the directories in the existing hierarchy. It uses training data composed of one such feature vector for each directory where an input document could potentially be assigned.

The rankings of destination directories produced by the text-based and image-based methods are combined with the Borda Count technique. Its N highest ranked choices are the categories assigned to the document.

We also adopted a novel method for generating the hierarchy in which documents are filed. Our objective was to provide a hierarchy that was familiar to users but did not require any extensive customization. This was done by using the directory structure from a user's PC hard disk. This takes advantage of the effort that users already employ to organize their computer files and uses it for automatic filing.

The automatic filing system creates a "mirror" hierarchy (Figure 3 (d)) that reflects the structure of the user's hard disk hierarchy. At retrieval time, users browse the mirror hierarchy, which is already familiar to them since it looks very similar to the hard disk hierarchy they work with on a regular basis.

The mirror hierarchy is generated by recursively descending a user's file system hierarchy. Files comprised of text, including ASCII text, postscript, PDF, etc. are labeled as having text features. Files that can be rendered as images, such as postscript, tiff, etc. are labeled as having image features. A directory in the hard disk hierarchy is retained in the mirror hierarchy only if it contains at least five files labeled as containing either text or image features.

Experiments were conducted to test the performance of the automatic filing system. A mirror hierarchy and training data were derived from one user's hard disk. The mirror hierarchy consisted of 20 directories that corresponded to 20 different classes of document. Testing was carried out on 781 documents. These documents all came from known locations in the user's hard disk. Therefore, the location where the document should be

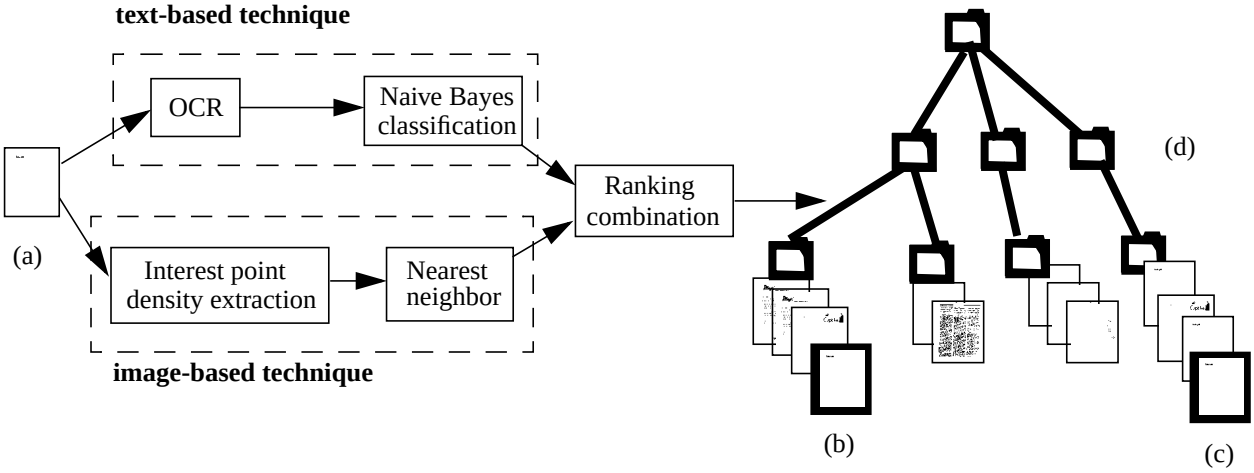

Figure 3. The automatic filing algorithm. A new document (a) is assigned to one or more locations (b) and (c), in a hierarchy (d). A text-based technique applies OCR and uses a naive Bayes classifier to rank the directories in the hierarchy. This is combined with the ranking produced by an image-based technique that extracts features called "interest points" and uses nearest neighbor matching. The hierarchy is a representation for a user's PC hard disk.

number of directories	text features	image features	combination of text and image features
1	85%	48%	86.7%
3	95%	78.5%	95.6%

Table 1. Correct classification rate for the automatic filing algorithm applied to 781 test documents.

stored in the mirror hierarchy was known. It should be noted that the test documents were not included in the training data.

The results of the experiments are shown in Table 1. When documents were filed in one location in the mirror hierarchy, text features alone gave an 85% correct rate. Image features gave a correct rate of 48%. The combination of text and image features gave an 86.7% correct classification rate. This shows only a modest improvement over using the text-based features alone.

The improvement in performance that could be obtained by filing each test document in three directories was also investigated. The text-based feature set gave a 95% correct classification rate. The image-based feature set gave a 78.5% correct rate and the combination a 95.6% correct rate. Most of the error cases were documents that reasonably could have been filed in several different directories. Examples included C source files.

The modest improvement in correct classification rate that was obtained with the image-based feature set is still encouraging. The fact that it was so small is more a reflection of the test set than the technique itself. There are many document types, e.g., scientific papers, that would be useful to file together and are easily distinguished by their physical layout. It is expected that the value of this approach will become more apparent in actual practice.

4. Conclusions

The design of the Infinite Memory Multifunction Machine (IM^3) was described. This system solves a large portion of the problem of lost documents by capturing an electronic copy of every document that users copy, print, or fax. This effectively reduces the effort expended by the user at capture time to zero. However, this increases the effort that must be expended at retrieval time since users must filter through large numbers of documents.

Two document analysis techniques were described that help users retrieve documents in such a system. One technique detects duplicate documents. Another method automatically files a document in an existing hierarchy. Experimental results demonstrated the efficacy of both methods.

References

[1] M.J. Blosseville, G. Hebrail, M.G. Montei, N. Penot, “Automatic Document Classification: Natural Language Processing, Statistical Analysis...”, 15th Annual International SIGIR’92, Denmark June 1992

[2] R. Casey and G. Nagy, “Autonomous reading machine,” IEEE Transactions on Computers, vol. C-7, no. 4, May, 1968.

[3] J.F. Cullen, J.J. Hull, and P.E. Hart, “Document Image Database Retrieval and Browsing using Texture Analysis”, Fourth International Conference on Document Analysis and Recognition, Ulm, Germany, August 1997, 718-721.

[4] R. Hoch, “Using IR Techniques for Text Classification in Document Analysis”, Proc. of 17th International Conference on Research and Development in Information Retrieval, SIGIR’94, Ireland, July 1994.

[5] P.G. Howard, F. Kossentini, B. Martins, S. Forchhammer, and W.J. Rucklidge, “The emerging JBIG2 standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 7, November, 1998, 838-848.

[6] J.J. Hull and P. Hart, “The infinite memory multifunction machine (IM^3),” Proceedings of the Third International Workshop on Document Analysis Systems, Nagano, Japan, November 4-6, 1998, 49-58.

[7] D. S. Lee and J.J. Hull, “Information extraction from symbolically compressed document images,” Symposium on Document Image Understanding Technology, Annapolis, MD, April 14-16, 1999, 176-182.

[8] T.M. Mitchell., “Machine Learning.” McGraw-Hill, 1997.

[9] I.T. Phillips, S. Chen, and R.M. Haralick, “CD-ROM document database standard,” Proceeding of the Second International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, 478-483.

[10] I. Witten, A. Moffat and T. Bell, “Managing Gigabytes: Compressing and indexing documents and images,” Van Nostrand Reinhold, New York, 1994.