Information Extraction from Symbolically Compressed Document Images

Dar-Shyang Lee, Jonathan J. Hull

Ricoh Silicon Valley, Inc.
2882 Sand Hill Road, Suite 115
Menlo Park, CA 94025
email: {dsl,hull}@rsv.ricoh.com

Abstract

The extraction of information from symbolically compressed document images is an increasingly important problem as the related standard (JBIG2) and commercial products become available. Symbolic compression techniques work by clustering individual connected connected components (blobs) in a document image and storing the sequence of occurrence of blobs and representative blob templates, hence the name symbolic compression. These techniques are specifically targeted to improving the compression ratio in binary document images. This paper proposes methods for extracting information from symbolically compressed document images by deciphering the sequence of occurrence of blobs. We propose a new deciphering algorithm that uses a hidden Markov model. Applications of this method to language identification, multilingual OCR, and duplicate detection are discussed. Experiments in duplicate detection are performed using the MG software package and the University of Washington database. The OCR-free and language independent nature of the algorithm suggests possible applications in a multilingual document database.

1. Introduction

The extraction of information from compressed document images is useful since the compression algorithm not only reduces the size of the image, providing less data to process, but also represents characteristics of the original image in the compressed data stream that can be used directly to compute information about original document. CCITT groups 3 and 4 compression are one example. These methods include pass codes in the compressed data stream, which are attached to connected components. The configuration of pass codes in CCITT-compressed document images has been used for skew detection [27] and duplicate detection [12, 18]

Symbolic compression has recently been proposed for inclusion in the JBIG2 standard [10]. Symbolic compression methods were first discussed by Ascher and Nagy [1]. More recent works include [9, 17, 28, 30]. In symbolic compression, images are coded with respect to a library of pattern templates. Templates in the library are typically derived by grouping (clustering) together connected components that have similar shapes. One template is chosen to represent each cluster. The connected components in the image are then stored as a sequence of template identifiers and their offsets from the previous component. In this way, an approximation of the original document is obtained without duplicating storage for similarly shaped connected components. Minor differences between individual components and their representative templates, as well as all other components which are not encoded in this manner, are optionally coded as residuals.

An example of symbolic compression is shown in Figure 1. After connected component clustering, the original document image is represented as a set of bitmap templates, “A a h i s t” in this example, their sequence of occurrence in the original image (“0 1 2 1 5 3 4 1 2 1 5 3 4 1 5 1 5”), as well as information about the relative geometric offset between adjacent connected components (e.g., (+2, 0) means the beginning of the second component in the sequence is 2 pixels to the right of the end of the first component), and a compressed residual image. The residual is the difference between the original image and the pattern templates. This data can be compressed with arithmetic coding or another technique. A lossy representation for a symbolically compressed image could be obtained by not storing the residual image.

Symbolic compression techniques improve compression efficiency by 50% to 100% in comparison to the commonly used Group 4 standard [19, 29]. A lossy version can achieve 4 to 10 times better compression efficiency than Group 4 [29]. Symbolic compression techniques are also used in some multilayered compression formats for color documents [8].

The symbolic compression format is especially useful for information extraction. Clusters of connected components that are approximately the size of characters can be assumed to be characters. Also, the sequence of cluster identifiers is a substitution cipher. This allows us to apply a deciphering algorithm to extract character interpretations. The use of a deciphering algorithm for character recognition was

Symp. on Document Image Understanding Technology, Annapolis, MD, April 14-16, 1999, pp. 176-182.

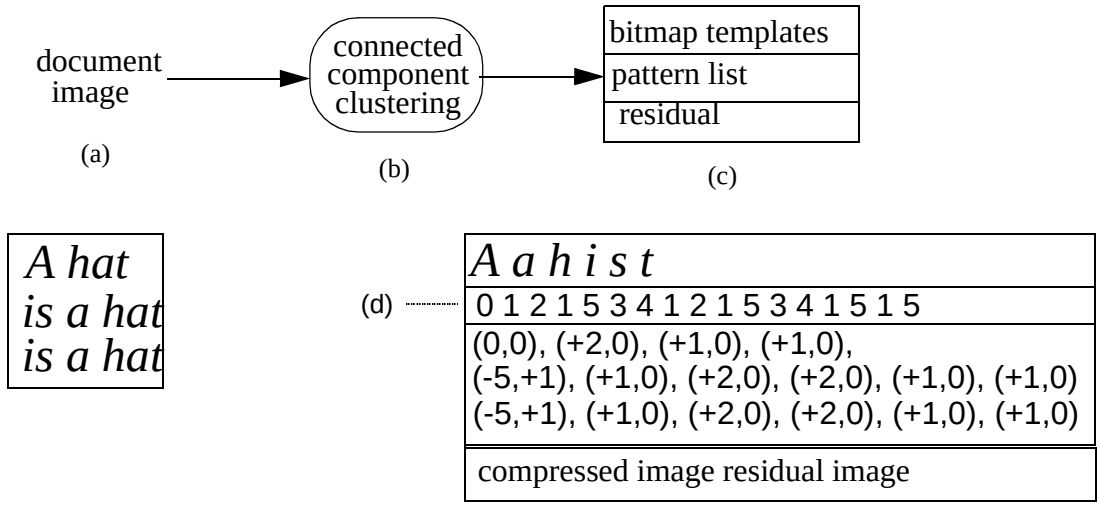

Figure 1 - Example of symbolic compression. The connected components in an original document image (a) are grouped into clusters (b). A bitmap template is chosen to represent each cluster and stored in the compressed file (c) together with their sequence of occurrence in the original image (d). Information about the geometric offset between adjacent components as well as image residual data are also stored in the compressed format.

proposed by Nagy and Casey [3]. Both character-level [2, 20] and word level [6] deciphering techniques have been proposed for text recognition.

This paper describes the application of a novel deciphering algorithm to the extraction of information from symbolically compressed document images. The deciphering algorithm can be configured to perform language identification and OCR in multiple languages. The accuracy of the OCR results may not be as high as that of a commercial OCR process. However, the accuracy is often high enough to be useful for various applications. The use of such character recognition results for document duplicate detection is described in this paper. An n-gram method for document matching is proposed. Experimental results demonstrate the utility of the character recognition technique and the accuracy of the document matching method.

2. Character Interpretation Deciphering

In the example shown in Figure 1, with the exception of the capital "A", there is a one-to-one correspondence between bitmap templates and English alphabetic characters. This is an ideal case known as a *simple substitution cipher*. If we were to replace each template identifier by its corresponding alphabetic character, "a" for 1, "h" for 2, and so on, the original message could be recovered from the sequence of component identifiers.

In practice, however, multiple templates can be formed for a single alphabetic symbol, as in the case of upper and lower case "a". This results in a many-to-one *homophonic substitution cipher* [17]. In an even more realistic scenario, a single pattern could correspond to a partial symbol or multiple symbols due to image fragmentation and segmentation errors.

A deciphering algorithm is available for simple substitution ciphers. By exploiting the redundancy in a language, the plain text message can be recovered from a sequence of cipher symbols of sufficient length [23]. Numerous algorithmic solutions have been proposed for simple substitution ciphers, including relaxation techniques [14, 16, 21] dictionary-based pattern matching [20, 25] and optimization techniques [7, 26]. We propose a deciphering algorithm that uses a Hidden Markov Model (HMM) [24].

Considering the Markov process of state traversal as a language source from which a particular plain text message can be generated with some probability, then the added symbol production at the traversed states in an HMM describes the enciphering process of a substitution cipher, where each letter in plain text is replaced with a cipher symbol one at a time. This analogy between the source language modeling as a Markov process and the representation of the enciphering function by symbol probabilities is the basis for our solution. The state probabilities are initialized with language statistics, and the symbol probabilities are estimated with the EM algorithm.

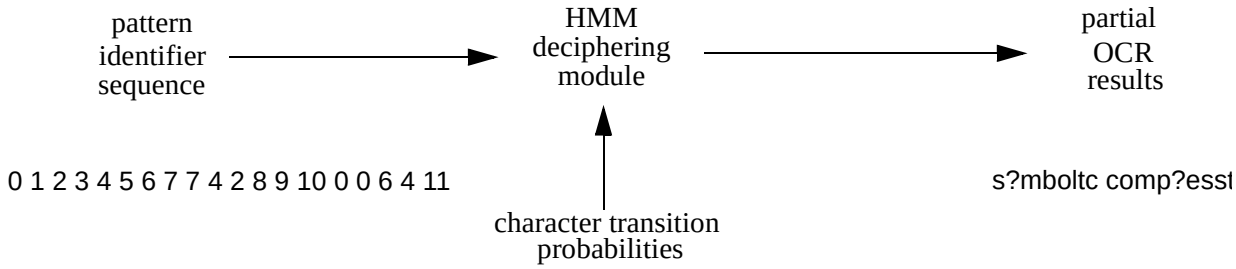

Figure 2 - Deciphering a symbolic compressed image produces partial OCR results.

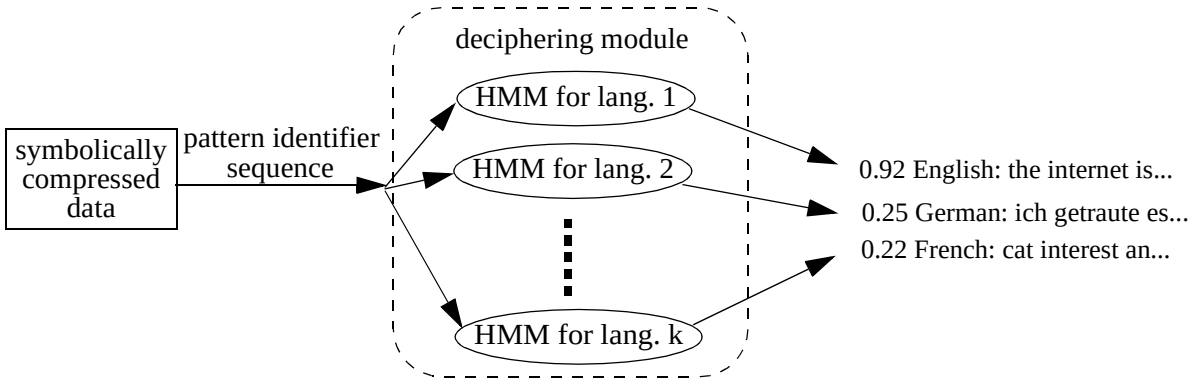

Figure 3 - Simultaneous language identification and deciphering. The deciphering module is given a compressed document of unknown language. It produces a set of possible interpretations in various languages.

Information extraction from symbolically compressed documents can be viewed as a deciphering problem. The objective is the recovery of the association between character interpretations and pattern templates from a sequence of template identifiers. In symbolic compression schemes, image components are grouped to improve clustering and they are also roughly sorted in reading order to the reduce entropy in their relative offsets. The objective of both measures is to improve compression performance. However, they also facilitate the application of deciphering techniques for information extraction.

Figure 2 shows an outline of the proposed HMM deciphering algorithm. It reads the pattern identifier sequence from a symbolically compressed document image and uses character transition probabilities to produce partial OCR results. These results may not be completely correct. However, they are often adequate for various tasks which will be described in the next section.

There are several reasons why the deciphered results will be less than perfect. First of all, it is obvious that the problem is never truly a simple substitution. The use of upper and lower case letters and multiple typefaces always lead to more than one template per alphabetic symbol. Imaging defects and segmentation problems further complicate the template-to-symbol mapping. In addition, short sequences and rare patterns do not posses sufficient statistics for deciphering. Even with ample exemplars, certain contents such as numeric strings can not be deciphered due to lack of context. Nevertheless, we believe sufficient information can be recovered for language identification, duplicate detection or document classification.

Identification of the language of the text in the original image can also be performed by a version of the HMM deciphering algorithm. The pattern sequence extracted from a compressed document is simultaneously deciphered with various language models. Each result includes the partial OCR results as well as a score that measures the confidence of the model. The language can be identified by selecting the model that produced the maximum score. This process is depicted in Figure 3

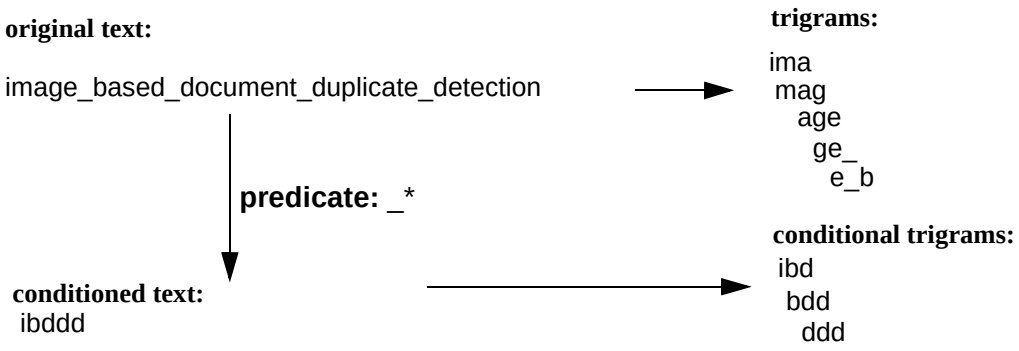

Figure 4 - Conditional n-grams are generated from consecutive characters satisifying a predicate.

3. Information Extraction from Partial OCR Results

We use an n-gram method to extract information from the partial OCR results output by the deciphering algorithm. N-gram based methods have been used for various information extraction tasks [11, 13, 30]. Their error-tolerant and language-independent characteristics are particularly suitable for information extraction from partially deciphered character interpretations. Cavnar and Trenkle [4] and Damashek [5] showed that an n-gram based document categorization algorithm is resistant to garbled input text.

An n-gram method for measuring the similarity of two documents typically extracts all sequences of n consecutive characters from each document. Their similarity is represented by a function of the number of n-grams they have in common.

We use a modified n-gram method to detect whether two documents are duplicates of each other. We use *conditional* n-grams that are generated from consecutive characters that satisfy a predicate. For example, a predicate of "the character following the space character" would form n-grams from the first character of consecutive words, as illustrated in Figure 4. Since conditional n-grams are formed on a string of filtered text, they generate fewer terms per document than the more typically used non-conditional n-grams described above.

The similarity of two documents is measured by calculating the dot product of their n-gram frequency vectors. Each entry in such a vector is the number of occurrences of an -gram in the document. Documents that have dot products above a threshold are duplicates.

4. Experimental Results

The experimental performance of the HMM deciphering algorithm and the conditional n-gram method for text string comparison were investigated. The HMM 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 database [22]. The mgtic algorithm [29] was used as the symbolic compression algorithm.

The character deciphering rate (number of characters deciphered in a test document) for a given amount of text was investigated. The results in Table 1 show that a 99% deciphering rate is achieved with only 1200 characters of test data, using character bigram statistics. Similar performance is achieved with 800 characters of test data and a trigram statistics. This illustrates the value of the additional contextual information present in trigrams. The accuracy of the deciphered results was also investigated. Between 80% and 95% of the characters in each of the cases shown in Table 1 were decoded correctly.

The performance of the conditional n-gram method for text string comparison was tested on the 979 documents in the University of Washington (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. Approximately 10% of the characters in the ground truth files for the UW database were corrupted to simulate a 90% correct decode rate by the HMM.

Conditional trigrams, as well as non-conditional trigrams and non-conditional 5-grams were extracted from each of the 979 UW documents. Each document was compared to the other 978 documents by calculating a similarity score using a weighted sum of the frequencies of the 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 top of the list. Ideally, this is a duplicate for the original document, if it exists in the database.

Table 2 compares the performance of conditional and non-conditional n-grams in duplicate detection. The Top 1 correct rate is the percentage of the 292 test documents with the highest similarity scores that are duplicates. This shows how often the correct match is the first choice output by the comparison algorithm. The Top 10 correct rate is the percentage of documents with duplicates for which the duplicate was contained in the 10 documents with the highest similarity score. The storage space for this technique is indicated by the total number of n-grams indexed.

The results in Table 2 show 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 almost a 40:1 increase in storage requirement in comparison to conditional trigrams.

.

# of chars	100	200	400	800	1200	1600	2000
bigram	57.55	72.73	93.19	96.74	99.13	99.13	99.56
trigram	66.47	90.17	98.80	99.01	99.44	99.54	99.76

Table 1. Character deciphering rates for various lengths of text.

criterion	**non-conditional trigrams**	**non-conditional 5-grams**	**conditional trigrams**
Top 1 correct rate	81.85%	100%	100%
Top 10 correct rate	97.95%	100%	100%
Total number of n-grams indexed	19,098	712,460	16,180

Table 2. Comparison of duplicate detection rates and storage required for various conditional and non-conditional n-grams.

5. Conclusions

A method for information extraction from symbolically compressed document images was presented. The technique is based on a novel deciphering approach that uses Hidden Markov Models. Although the error rate in the text recovered by deciphering is normally higher than that by a conventional OCR system, we demonstrated that there is sufficient information for certain document processing tasks. An n-gram based method for duplicate detection was proposed here.

The deciphering algorithm is limited to documents composed mostly of text. Also, the success of deciphering depends on redundancy in the language and the original document. Therefore, it would be difficult to adapt it to ideographic languages such as Chinese or to apply it to very short documents.

Experimental results showed that the HMM can successfully decipher over 98% of the text in English language document images that contain as little as 400 characters. The proposed technique for duplicate detection was also investigated experimentally. Duplicates were successfully detected in a database of about 979 images.

Future work includes investigation of adaptation to new languages. Only gathering of statistics for the HMM should be required.

References

[1] R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized printed text," *IEEE Transactions on Computers*, Vol. C-23, No. 11, pp. 1174-1179, Nov. 1974.

[2] L. R. Bahl and J. Cocke, "Font-independent character recognition by cryptanalysis," *IBM Technical Disclosure Bulletin*, Vol. 23, No. 3, pp. 1588-1589, August 1981.

[3] R. Casey and G. Nagy, "Autonomous reading machine," *IEEE Transactions on Computers*, vol. C-17, No. 5, May 1968.

[4] W. B. Cavnar and J. M. Trenkle, "N-gram-based text categorization," *Proceedings of the Symposium on Document Analysis and Information Retrieval*, pp. 161-175, Las Vegas, Nevada, 1994.

[5] M. Damashek, "Gauging similarity with n-grams: language-independent categorization of text", *Science*, pp. 843-848, February, 1995.

[6] C. Fang and J. J. Hull, "A word-level deciphering algorithm for degraded document recognition," *Fourth Symposium on Document Analysis and Information Retrieval*, University of Nevada at Las Vegas, Las Vegas, Nevada, April 24-26, 1995, pp. 191-202.

[7] W. S. Forsyth and R. Safavi-Naini, "Automated cryptanalysis of substitution ciphers," *Cryptologia*, vol. 17, no. 4, pp. 407-418, 1993.

[8] P. Haffner, L. Bottou, P. G. Howard, P. Simard, Y. Bengio and Y. Le Cun, "Browsing through high quality document images with DjVu," *Proceedings of IEEE Advances in Digital Libraries*, Santa Barbara, California, April, 1998.

[9] P. Howard, "Lossless and lossy compression of text images by soft pattern matching," *Proceedings of the IEEE Data Compression Conference (DCC'96)*, Snowbird, pp. 210-219, 1996.

[10] P. 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, pp. 838-848, November 1998.

[11] S. Huffman, "Acquaintance: language-independent document categorization by n-grams," *Proceedings of the 4th Text REtrieval Conference*, 1996,

[12] J. J. Hull, "Document image similarity and equivalence detection," *International Journal on Document Analysis and Recognition*, vol. 1 no. 1, February, 1998, 37-42.

[13] J. J. Hull and S. N. Srihari, "Experiments in text recognition with binary n-gram and viterbi algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-4, no. 5, pp. 520-530, September 1982.

[14] D. G. N. Hunter and A. R. McKenzie, "Experiments with relaxation algorithms for breaking simple substitution ciphers," *The Computer Journal*, vol. 26, no. 1, pp. 68-71, 1983.

[15] O. Kia and D. Doermann, "Symbolic Compression for document analysis," *Proceedings of International Conference on Pattern Recognition*, Volume III, 1996, pp. 664-668.

[16] J. King and D. Bahler, "An implementation of probabilistic relaxation in the cryptanalysis of simple substitution ciphers," *Cryptologia*, vol. 16, no. 3, pp. 215-225, 1992.

[17] J. King and D. Bahler, "An algorithmic solution of sequential homophonic ciphers," *Cryptologia*, vol. 17, no. 2, pp. 148-165, 1993.

[18] D. S. Lee and J. J. Hull, "Group 4 Compressed Document Matching," *Proceedings the the Third IAPR Symposium on Document Analysis Systems*, Nagano, Japan, Nov. 4-6, 1998, pp. 29-38.

[19] K. Mohiuddin, J. Rissanen and R. Arps, "Lossless binary image compression based on pattern matching," *Proceedings of International Confer-*

ence on Computers, Systems & Signal Processing, December, 1984.

[20] G. Nagy, S. Seth and K. Einspahr, "Decoding substitution ciphers by means of word matching with application to OCR," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 710-715, 1987.

[21] S. Peleg and A. Rosenfeld, "Breaking substitution ciphers using a relaxation algorithm," *Communications of the ACM*, vol.22, no.11, pp. 598-605, November 1979.

[22] I. T. Phillips, S. Chen, R. M. Haralick, "CD-ROM document database standard," *Proceedings of the 2nd ICDAR*, pp. 478-483, 1993.

[23] F. Pratt, *Secret and Urgent: the story of codes and ciphers*, Blue Ribbon Books, Garden City, New York, 1942.

[24] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[25] R. S. Ramesh, G. Athithan and K. Thiruvengadam, "An automated approach to solve simple substitution ciphers," *Cryptologia*, vol. 17, no. 2, pp. 202-218, 1993.

[26] R. Spillman, M. Janssen, B. Nelson and M. Kepner, "Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers," *Cryptologia*, vol. 17, no. 1, pp. 31-44, 1993.

[27] A. Lawrence Spitz, "Skew determination in CCITT group 4 compressed document images," *Proceedings of the Symposium on Document Analysis and Information Retrieval*, Las Vegas, Nevada, March 16-18, 1992, pp. 11-25.

[28] I. Witten, T. Bell, H. Emberson, S. Inglis, and A. Moffat, "Textual Image Compression: two stage lossy/lossless encoding of textual images," *Proceedings of the IEEE*, vol. 82, no. 6, pp. 878-888, June 1994.

[29] I. Witten, A. Moffat and T. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images*, Van Nostrand Reinhold, New York, 1994.

[30] Q. Zhang and J. Danskin, "Entropy-based pattern matching for document image compression," *Proceedings of the International Conference on Image Processing*, pp. 221-224, Lausanne, Switzerland, Sept. 1996.