

Indexing and retrieval of words in old documents

Simone Marinai Emanuele Marino Giovanni Soda
DSI - University of Florence (Italy)
E-mail: marinai@dsi.unifi.it

Abstract

This paper describes a system for efficient indexing and retrieval of words in collections of document images. The proposed method is based on two main principles: unsupervised prototype clustering, and string encoding for efficient string matching. During indexing, a self organizing map (SOM) is trained so as to cluster together similar symbols (character-like objects) in a sub-set of the documents to be stored. By using the trained SOM the words in the whole collection can be stored and represented with a fixed-length description, that can be easily compared in order to score most similar words in response to a user query.

The system can be automatically adapted to different languages and font styles. The most appropriate applications are for the processing of old documents (18th and 19th Centuries) where current OCRs have more difficulties. Experimental results describe three application scenarios having various levels of difficulty for current OCR systems.

1. Introduction

Document image retrieval aims at finding relevant document images from a corpus of digitized pages. One application field of this research are Digital Libraries, where large collections of scanned documents already exist and are available on the Internet. The basic idea of document image retrieval is to find documents relying on document image features only. Relevant sub-tasks include the retrieval of documents on the basis of layout similarity, and the retrieval considering the textual content [2]. When dealing with text the approaches can be clustered on two main categories on the basis of the use of OCR.

The first stream of methods avoids the use of OCR and is usually based on two steps. In the indexing step the textual content of the document is encoded with some “ad-hoc” method, and the characters are represented with features *without explicitly assigning a*

character class to individual objects. In the retrieval step the relevant documents are extracted from the database by encoding the query with the same algorithm used during indexing, and matching the query representation with the encoded documents. Some queries are based on a simple word matching approach, whereas methods closer to *Information Retrieval* identify the documents on the basis of distributions of word occurrences, and rank the fetched documents with similarity measures.

When using OCR the approaches rely on the processing of the converted text to extract the interesting information. In this case the systems must cope with the recognition errors that are inevitably introduced by the OCR. The classical approach relies on the *string edit distance* between the query and the strings in the database. This approach has been adapted by introducing “ad hoc” edit costs for most common OCR errors (e.g. [4]).

String edit distance can be used also when dealing with symbolic encoding of text in OCR-free approaches. The main problem of string edit distance based methods is the computational cost when dealing with large collections of documents. One solution is the use of *approximate string matching* (e.g. [6]), however from an user point of view *ranking queries* appear more appropriate. The ranking query is a generalization of *k*-nearest-neighbor query with a previously unknown result set size *k*. The results of this query are ordered on the basis of the proximity with the query (similarly to *k*-nn query), but there is no need to define in advance the number of points to be found.

Unfortunately, approaching ranking queries for strings is computationally expensive, since all the words in the collection should be compared with the query and sorted. On the other hand the ranking query has been efficiently solved in vectorial spaces where points are stored with multi-dimensional access methods (e.g. R-tree and X-tree [1]). In order to use techniques used in vectorial spaces in the domain of strings we need to formulate the string matching problem in

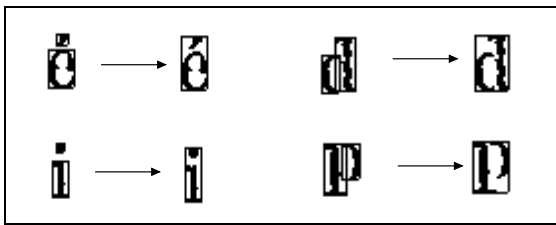


Figure 1. Extraction of character objects from connected components.

vector spaces.

In this paper we propose a method for the retrieval of words encoded by describing the character-like objects of the words with classes obtained by the unsupervised learning of a Self Organizing Map (SOM [3]). The encoded words are afterwards converted into a fixed-size representation that can be compared with the query by means of the standard Euclidean distance.

The paper is organized as follows. In Section 2 we describe the processing steps performed during word indexing, whereas the operations performed during word retrieval are analyzed in Section 3. Experimental results are analyzed in Section 4, and conclusions are drawn in Section 5.

2. Word indexing

The textual content of the document image is segmented into words with an RLSA-based algorithm, and *character objects* (*CO*) are afterwards extracted by locating connected components and grouping together overlapping components. This process is not error-free and depending on document quality broken or touching characters happen to be considered as *CO*. However, this is not a major problem, as we will see later. The *COs* of each word are labeled according to a clustering algorithm, so as to obtain a string representing the word. This string is afterwards encoded into a fixed-size representation to be used in word retrieval.

2.1. SOM-based clustering

Character-like coding is a well known approach for performing text retrieval without OCR. In its most general form the indexing (compression) algorithm is made by two main parts. First, character objects (*CO*) are extracted from the document image. Second, each *CO* is described with a symbolic (a code) or sub-symbolic (a feature vector) representation. Character coding is based on the grouping of similar characters with a clustering-based approach, without attempting to assign the “right” class to characters (as in OCR). In its simplest form the encoding can be based on *character shape coding* that has been successfully applied for

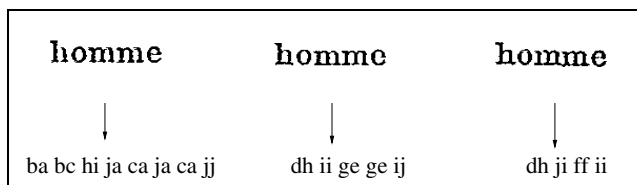


Figure 2. Different word encodings due to broken and touching characters.

Information Retrieval without OCR [5]. With a more expensive approach a feature vector can be computed for each *CO*, as described in a recent application to text retrieval [7]. Lastly, it is worth to mention that a very similar philosophy is the basis of some methods for document image compression [9].

In this framework we use Self Organizing Maps (SOM [3]) as a method for template matching and character object clustering. As discussed later, SOM have several advantages compared to other clustering methods. The SOM neurons are arranged in a two dimensional lattice (feature map). Each neuron receives inputs from the input layer and from the other neurons in the map. During the learning the network performs clustering by means of a competitive learning mechanism. Moreover neurons are moved in the lattice so as to reflect cluster similarity by means of distance in the map.

One advantage of the use of SOM for *CO* clustering is the spatial organization of feature maps that is achieved after the learning process. Basically, more similar clusters are closer than more different ones. Consequently the distance among prototypes in the output layer of the SOM can be considered as a measure of similarity between characters in the clusters.

2.2. Normalized word representation

Two instances of a given word are not constrained to have the same number of *COs* due to the presence of broken or touching characters. In our system each *CO* of a word is assigned to one output neuron of the SOM, and identified with a pair of characters that describe the position of the neuron in SOM feature map. For instance the label 'aa' represent the upper-left neuron, whereas (in a 9 by 9 grid) the bottom-right neuron is represented by 'ii'. A word is therefore represented with a string corresponding to *CO* labels as shown for example in Figure 2.

This variation in string length is an obstacle to fast retrieval of words, and a string edit distance algorithm should be considered for word retrieval. However, the use of string edit distance for comparing word encod-

ings is somehow inappropriate, since we lose one of the main features of original words: regardless of touching or broken characters the overall word size is nearly fixed in a given document. This feature is considered in holistic keyword spotting, where entire words are described with features that are afterwards compared with a word representation during retrieval (e.g. [8]).

One simple approach in holistic word recognition is based on zoning of the input pattern (the word). Zoning consists of overlapping the word with a fixed-size grid, and computing some features (e.g. the density of black pixels) in each grid region. One advantage of this approach is the property that individual characters are located “roughly” in the same position in the grid regardless of touching or broken characters in the word.

We extended this approach to convert a variable length word encoding (obtained by *CO* extraction and SOM clustering) into a fixed-size symbolic description (the *expanded string*). The basic algorithm is the following one (see Figure 3):

1. The *CO*s in the word are located.
2. Each *CO* is labeled with the output neuron of the trained SOM.
3. The word image is partitioned into a fixed number of vertical slices.
4. Each slice gets the label of the *CO* with the largest overlap with it.

After this step a fixed-length string is assigned to each word, though the length of the new string is larger than the original one (e.g. four tokens marked with **eg** correspond to the ‘p’ in Figure 3). Obviously it is not possible to use a unique number of slices for all the words. Actually we compute the aspect-ratio of the word (N), and define the number of slices (NS) on the basis of the N , ranging from $NS = 25$ when $N < 0.15$ to $NS = 6$ when $N > 0.55$.

3. Word retrieval

In this Section we describe the retrieval of words matching a given query. Two main operations are performed (see Figure 4). First, a textual query is translated into one or more word images that are encoded with the method described in Section 2. Second, each query is compared with the word descriptions that are stored in the database.

3.1. Query translation

From a user point of view the queries are made to the system with a simple text-based interface. Starting from the ASCII word one “clean” word image is produced with L^AT_EX software. The clean image is

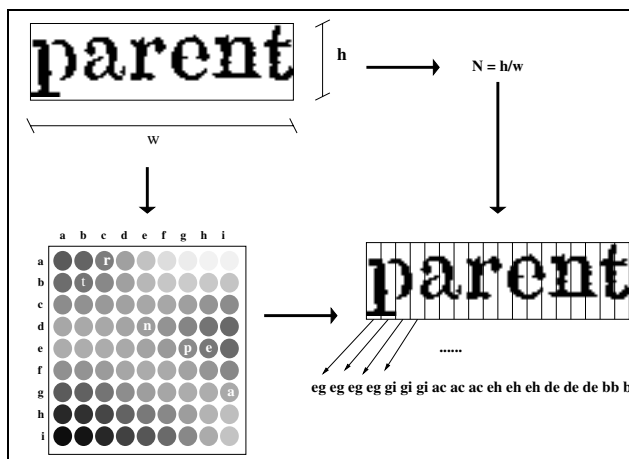


Figure 3. Word encoding in the expanded string.

corrupted with synthetic noise (we use programs based on the Baird’s noise model) in order to simulate actual distortions occurring in real world. Note that this step is the unique point in the overall system that needs to be customized when changing the set of documents considered, since the predominant font in the text must be indicated.

Each word image is mapped to a fixed-size vector (as described in Section 2.2) and compared with vectors in the database having the same length. Words with different aspect ratios are unlikely to correspond to the query, and consequently are not considered for the comparison.

3.2. Similarity of expanded strings

Words in the database are ranked on the basis of the similarity with one query word by comparing the corresponding expanded strings. The comparison is made with the Euclidean distance, and the distance between symbols of expanded strings is computed by using the actual distance of the corresponding neurons in the SOM. We can observe that the Euclidean distance between expanded strings approximates the similarity between the corresponding words for two reasons. First, also in presence of broken or touching characters the strings have the same length, and codes corresponding to well separated characters are roughly in the same position in the string. Second, the topology preservation feature of SOM maps allows us to evaluate also the similarity between close clusters in the output layer of the SOM, since closer nodes have lower distances.

The limitations of this similarity evaluation are only reached when very noisy words are encountered. In this case for each couple of words only a few (if any)

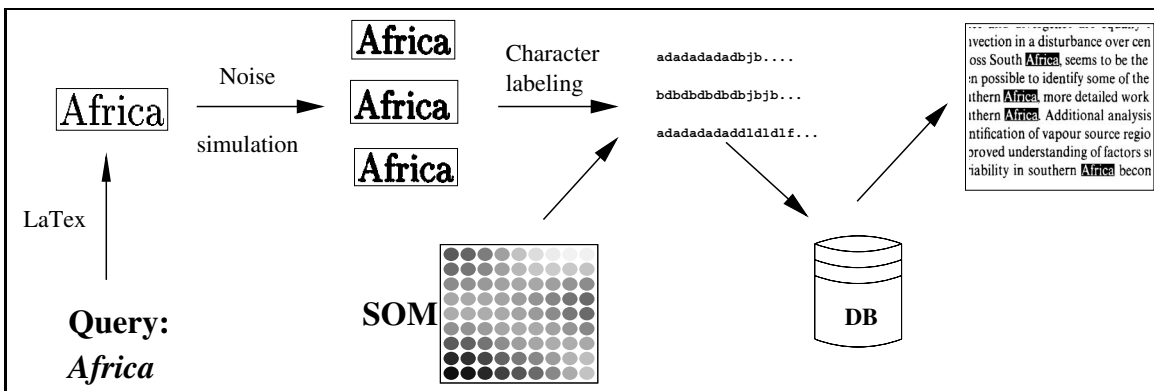


Figure 4. Processing steps during word retrieval.

characters are aligned. In this case the method can efficiently work only if an appropriate noise model is used so as to enable to simulate realistic perturbations in the word.

4. Experimental results

In this Section we report the most relevant experimental results carried out on three heterogeneous sets of documents. The tests have been made by comparing the results that can be achieved with the proposed method to results of a commercial OCR. We considered three groups of scanned documents belonging to uniform collections 1) Technical papers from the UW-III CD-ROM. 2) Old (18th Century) Italian books. 3) German documents printed in Gothic. It is important to point out that, despite of the large difference in language and fonts of the three experimental tests, the system has not been modified when changing the data-set. The only change is the latex font used in query generation for Gothic documents.

In each of the three experiments we use a fraction of pages for SOM training and we encode all the pages with the trained network. Afterwards, we tested the retrieval with some queries. For each query we ranked all the stored words with respect to their similarity with the query. In this list we identify the position (i) of the first wrong word, and we consider that the system identified $i - 1$ correct words. The comparison with the OCR is made computing the number of words in the output of the OCR that exactly match the query. When there is no exact match, we check the edit distance between the query and “true” words.

4.1. UW-III documents

The documents from this standard data-set are not too difficult for good OCR-packages. In this test we used 6 papers with a total of 19 pages. For each paper we trained a new SOM with at most two pages. All the pages in the paper are afterwards encoded with the

trained SOM, and some test queries are made on all the pages. For this set of documents the performance of the proposed method is similar to OCR. In general the OCR works very well, and the virtually only problems it encounters are related to words in languages different from the main language.

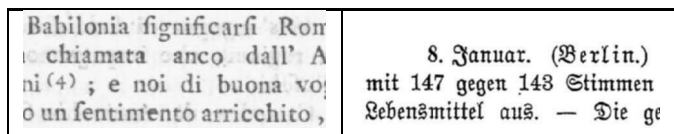


Figure 5. Examples of documents of data sets 2 and 3, respectively.

4.2. Old documents

On the average OCR systems work very well on UWASH documents. The automatic reading of old documents is a more difficult task for two main reasons: differences in font and in language. Documents of the 18th Century have the peculiarity that old 's' are very similar to 'f' (Figure 5), and this property is a severe problem for the OCR since recognized words are not in the dictionary. On the opposite when using our system the user can formulate the queries directly with the “spelling error”, and this allows it to correctly retrieve most words. In Table 1 we report the recognition results for some typical words. In this experiment we used 12 pages from an Italian journal, and four of them are used for SOM training. It is interesting to see how good the OCR identifies the words that are in contemporary Italian (without the 's'↔'f' problem). For other words more difficulties are encountered. For instance two of the four occurrences of the word “effere” have an edit distance of 1 with respect to the query, one is at distance 2 and one is at distance 3. Unfortunately, there are some other (wrong) words with distance 2 to the word “effere”.

Keyword	SOM Map	OCR
Babilonia (*)	1	1
Autore (*)	3	3
rifleffioni	2	0
Germania (*)	2	2
Greca (*)	2	2
effere	4	0
Regno (*)	6	7
Roma (*)	5	4
Vienna (*)	1	1
folamente	1	0
Cefare	1	0

Table 1. Experiments on 18th Century documents. Words marked with a (*) are also in modern Italian vocabulary.

The recognition of Gothic documents of the 19th Century is a difficult task for current OCR systems. The main reason is the different font with respect to contemporary ones. To have an idea of what are the typical characters in these documents we show in Figure 6 the prototypes corresponding to output neurons in the trained SOM. Note that close prototypes are usually most similar than distant ones. The data set was composed by 14 pages. We used 4 pages for SOM training and the whole data set for word retrieval. In this case the recognition results for OCR are very low (see Table 2). In the case of “Januar”, the best recognized word, only 13 words are correctly recognized. However, only 9 words are at a distance 2, whereas the other ones cannot be retrieved since in the data set there are 154 words with an edit distance of 3 with respect to “Januar”.

Keyword	SOM Map	OCR
Januar	34	13
Kaifer	9	0
Finanzlage	5	0
Eine	5	0
Monarchie	8	0
Stettin	2	3
Landwirtschaft	11	0

Table 2. Experiments on Gothic documents.

5. Conclusions

In this paper we described a system for indexing and retrieval of words in printed documents. The system is language-independent and can be easily adapted to different printing qualities. The evaluation of word similarity can be computed with a simple Euclidean dis-

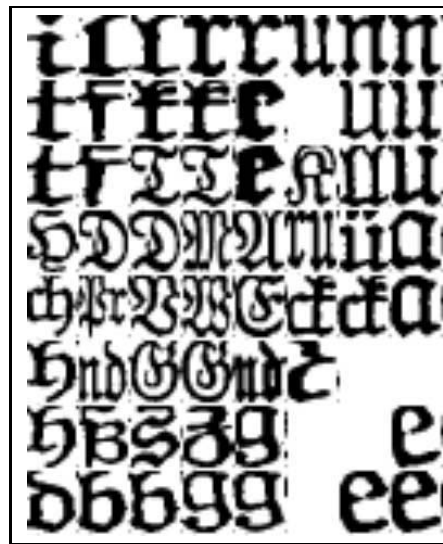


Figure 6. Prototypes in a 8x8 SOM trained on Gothic documents.

tance in vector spaces, and this property can be considered for improving the performance by using standard methods for indexing objects in vector spaces. The experimental results show that the method is particularly effective when dealing with old documents where commercial OCRs have severe difficulties in text recognition.

References

- [1] S. Berchtold, D. A. Keim, and H.-P. Kriegel. The X-tree: an index structure for high-dimensional data. In *Proc. Proc. 22nd VLDB*, pages 28–39, 1996.
- [2] D. Doermann. The indexing and retrieval of document images: A survey. *Computer Vision and Image Understanding*, 70(3):287–298, June 1998.
- [3] T. Kohonen. *Self-organizing maps*. Springer Series in Information Sciences, 2001.
- [4] D. P. Lopresti. Robust retrieval of noisy text. In *Proc. of ADL’96*, pages 76–85, 1996.
- [5] A. F. Smeaton and A. L. Spitz. Using character shape coding for information retrieval. In *Proc. 4th ICDAR*, pages 974–978, 1997.
- [6] A. Takasu. Document filtering for fast approximate string matching of erroneous text. In *Proc. 6th ICDAR*, pages 916–920, 2001.
- [7] C. L. Tan, W. Huang, Z. Yu, and Y. Xu. Imaged document text retrieval without OCR. In *IEEE Trans. PAMI*, volume 24, pages 838–844, June 2002.
- [8] J. Trenkle and R. Vogt. Word recognition for information retrieval in the image domain. In *SDAIR*, pages 105–122, 1993.
- [9] I. H. Witten, A. Moffat, and T. C. Bell. *Managing gigabytes: compressing and indexing documents and images*. International Thomson Publishing, 1994.