Image-based Document Vectors for Text Retrieval

Zhaohui Yu, Chew Lim Tan
School of Computing, National University of Singapore
Email: yuzhaohu, tancl@comp.nus.edu.sg

Abstract

We propose a method for constructing a vector for a document image to represent its content to facilitate text retrieval. The method is based on an N-Gram algorithm for text similarity measure based on the frequency of occurrence of n-character strings appearing in the electronic text. Instead of using ASCII values, the present study investigates the use of character images to obtain the document vector and has found promising results for use in our news article retrieval project.

Keywords: *Document Image, Text Retrieval, Similarity Measure, N-Gram Algorithm*

1 Introduction

This paper describes a novel approach to construct a document vector as a similarity measure using N-Gram algorithm without the recognition of the characters. The application we have in mind is the text retrieval from newspaper microfilm images [1].

Figure 1 outlines the steps in constructing vectors of document images based on contents. To identify the features, character objects in the predominant font are extracted from image documents and then character object equivalent classes are identified based on shape similarity. From several sets of classes, one unified class set can be estimated. The objects, which belong to the same class, are assumed to represent the same term. Layout analysis is performed to determine the reading order of character objects. An object sequence can be obtained from each image document to construct a vector using N-Gram algorithm. These document vectors are used to calculate the similarity between image-based document.

The remainder of this paper is organized as follows. Section 2 presents the stages to construct document vector and similarity measure. Section 3 describes experimental results that demonstrate the effect of these techniques. Finally, conclusions and future work are given in Section 4.

2 Document Vector Construction

We will discuss how character features are extracted for vector construction and how vectors are used for similarity measure.

2.1 Character Object Class

In document image, there are three kinds of character object. The first is isolated character, which has only one connected component. The second is also isolated character, but it has more than one connected component,

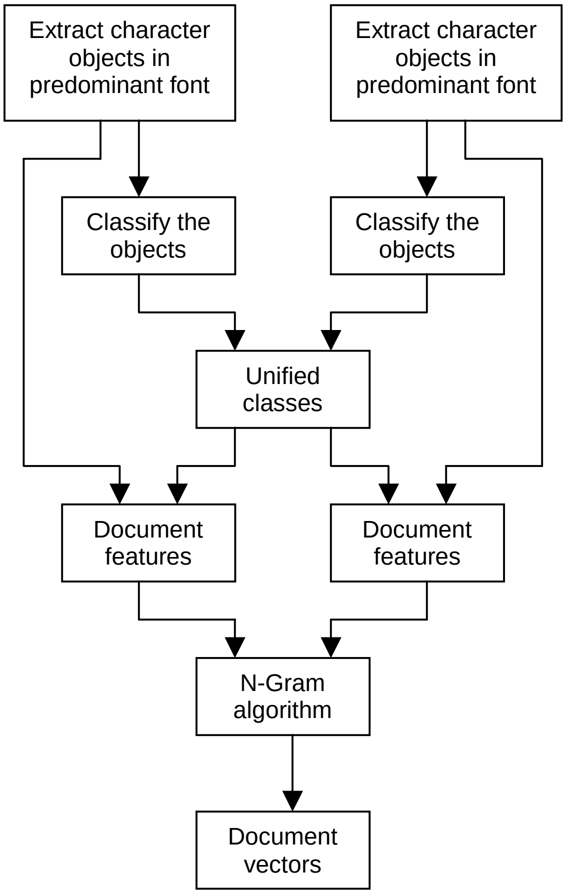


Figure 1. Construction of document vectors

such as lower character “i” and “j”. The third is several characters that are connected to each other.

Character object can be extracted by measuring the connected components of image and comparing the relative positions of adjacent components. One object includes only one connected component or several connected components, which have unambiguous relative positions.

In newspapers, the main body of text is usually printed in the one font, which is generally the predominant font, whereas headings and captions may appear in a variety of fonts. For construction of document vectors, only text in the predominant fonts is considered. To identify character object corresponding to the same term, an unsupervised classifier is used to place each character object into a set of classes. Each class is regarded as representing one unique term.

For each character object, we can create a vertical traverse density (VTD) vector and a horizontal traverse density (HTD) vector. For the character object *m* and *n* in image document, they will belong to same class, if

$$diff\,(VTD_m, VTD_n) \le Threshold * \min(w_m, w_n)$$

and

$$diff\,(HTD_m, HTD_n) \le Threshold * \min(h_m, h_n)$$

Where, $diff(V_1,V_2)$ is a function to calculate the difference between the vector V_1 and V_2, w_i and h_i is the width and height of object *i*. Otherwise, the character object m and n belong to different classes.

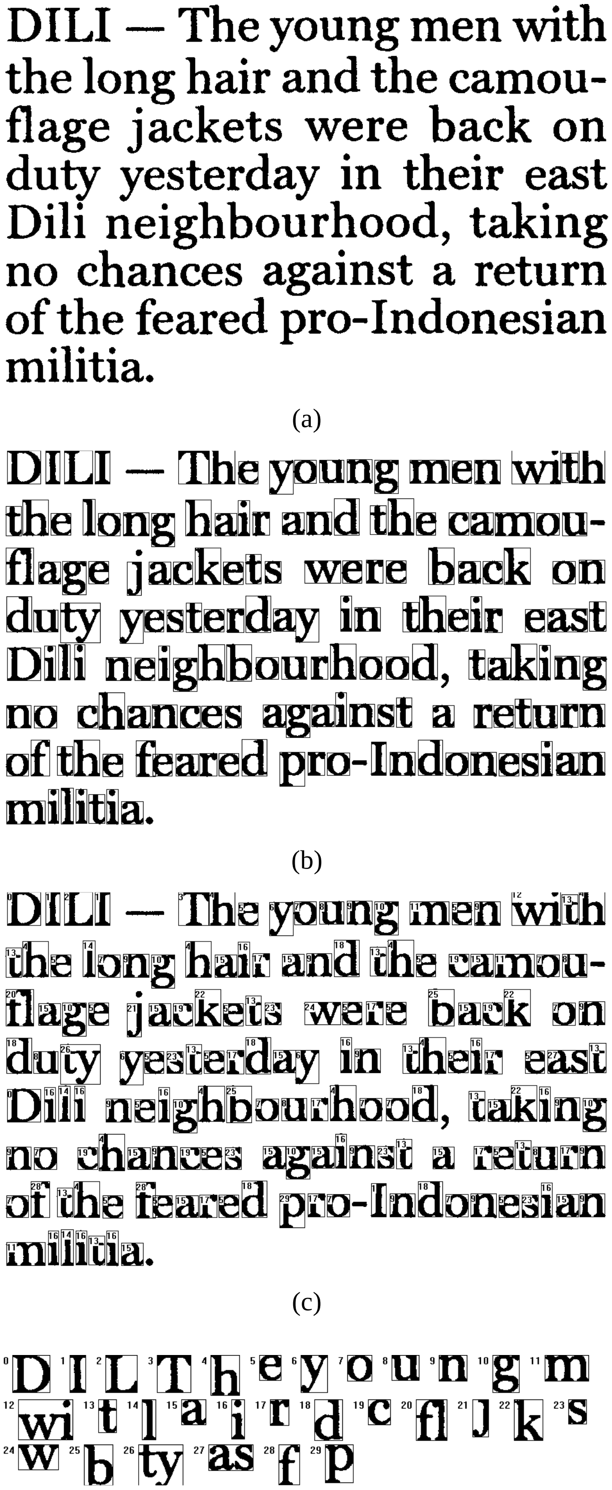
The result of classification is shown in Figure 2. Item (a) is the original image and item (b) show the extracted character objects. In item (c), each rectangle outlines a character object extracted and the number expresses the sequence number of class that the object belongs to. The objects that have same number represent same term. Item (d) list the total class set created from this image.

2.2 Unifying Character Object Class

One set of character object classes can be obtained from each image document. The predominant fonts of different images may have different font sizes. And the numbers of different sets of classes maybe also different. To find a unified way to express the feature of image documents, we must build a unified object classes.

Firstly, the VTD vector and HTD vector of all character object classes are normalised. The number of permitted dimension of vectors is set sufficiently large. All the features of VTD and HTD will be preserved. The normalised classes are then classified again to create a set of unified classes. Finally, a look-up table from the original class set to the unified class set is built for each image document. Using these tables, all character objects in these image documents can be mapped to the unified character object classes. The objects corresponding to the same class will be regarded as representing the same term.

After all character objects have been expressed by a set of classes, the layout analysis is performed to determine the reading order of character objects and the space



DILI — The young men with the long hair and the camouflage jackets were back on duty yesterday in their east Dili neighbourhood, taking no chances against a return of the feared pro-Indonesian militia.

(a)

(b)

(c)

0 D 1 I 2 L 3 T 4 h 5 e 6 y 7 o 8 u 9 n 10 g 11 m
12 wi 13 t 14 l 15 a 16 i 17 r 18 d 19 c 20 fl 21 j 22 k 23 s
24 w 25 b 26 ty 27 as 28 f 29 p

(d)

Figure 2. Character objects and classification

between two adjacent objects. One list is built for each document. The item of list is the sequence number of class that the character object belongs to. This list will be used to construct the document vector.

2.3 Similarity between Vectors

The use of N-Gram algorithm in various text processing has been reported in [2,3,4]. On the other hand, instead of text processing, the use of character images has been attempted by researchers [5-9] for summary extraction, similarity measurement and document retrieval. The present method attempted to use image-processing approach to the text-based N-Gram method. The algorithm is adapted for image-based similarity measure.

First, the class number list is converted to a set of n-gram slices. An n-gram is an n-item slice of a stream. N-grams, which are sequences of n consecutive items, are copied out of the list using a window of n-item length, which is moved over the list one item forward at a time.

Secondly, every possible n-gram is given a number, so called hash key. How the n-grams are numbered is not important, as long as each instance of a certain n-gram is always given the same number, and that two different n-grams are always assigned different numbers.

Then, a hash table is created to keep track of how many times each n-gram has been found in the list being studied. Every time an n-gram is picked, the element of the hash table given to the n-gram is increased by one. When all n-grams have been put into a hash table, the occurrence numbers of the hash table is divided by total number of extracted n-grams. This means that the absolute number of occurrence will be replaced with the relative frequencies of corresponding n-grams. The reason for doing this is that similar texts of different lengths after this normalisation will have similar hash tables.

Lastly, the hash tables are used to calculate the similarity. The hash tables can be treated as document vectors. Document vectors from the similar text point in almost the same direction. The similarity score between two texts is defined as their scalar product divided by their lengths. A scalar product is calculated through summing up the products of the corresponding elements. This is the same thing as the cosine of the angle between two documents seen from the origin. So, the similarity between image document *m* and *n* will be

$$Similarity(X_m, X_n) = \frac{\sum_{j=1}^{J} x_{mj} x_{nj}}{\sqrt{\sum_{j=1}^{J} x_{mj}^2 \sum_{j=1}^{J} x_{nj}^2}}$$

Where, X_m and X_n are the document vector of image *m* and *n*, *J* is the dimension number of document vector, and $X_i = x_{i1} x_{i2} \cdots x_{iJ}$.

3 Experimental results

To verify the validity of our method, the corpus was selected from recent international news in *The Strait Times*, a major local newspaper. In this corpus, four articles talk about Indonesia. And the other four talk about the news in Japan, Kampuchea, Thailand and Russia respectively. The paper articles were scanned at 600 pixels/inch (ppi). To make the process simple, some preprocessing is done. First, the images are de-skewed. Then, noise such as small dirty spots is removed from the image. Next, headings and captions are removed from the imaged documents. As a standard of evaluating, ASCII version articles of original documents are created. An OCR system is used to extract the text in the documents and the text was then hand-corrected.

The processed images are used as the input to our method. As a comparison, the ASCII version of articles is used to extract the n-gram slices and build the document vectors. From the document vectors, we also can obtain a group result of the similarity of each pair documents. The two group results are shown in Table 1, in which *n* of N-Gram algorithm is equal to six.

From the result, we can see that the similarities of documents measured from both methods share some resemblance though not entirely equivalent to each other. The result of ASCII version of documents provides more distinguishable similarity measure. This is because that the character objects we extracted are not equivalent to characters and the objects corresponding to same character may be classified into different object classes.

But the two results exhibit similar trend. When two ASCII version documents have large similarity, their corresponding image documents also have large similarity. Figure 3 shows the comparison of the similarities between news N04 to the other news. Where, the square points show the result of image documents, and the diamond points show the result of ASCII version documents.

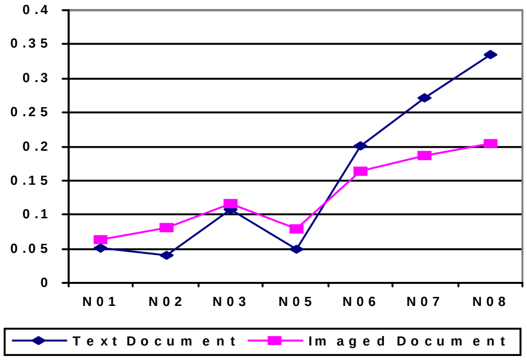


Figure 3. The similarity between news N04 and the other news

4 Conclusion and Future Work

We proposed a document vector construction process based on an N-Gram algorithm without the use of OCR. We extract the features of document images by obtaining and classifying the character objects. Then, N-Grams algorithm is used to measure their similarity. This method is suited for gauging the similarity of image documents that have the same font style. It is our future research direction to examine for those documents of different font styles. The final objective of our method is to be used for microfilm images in a news retrieval project. Microfilm images are noisier than the images studied in the present project. So, how to deal with noise will also be another future research.

References

[1] C.L. Tan, S.Y. Sung, D. Shi, B. Yuan, Y.T. Lim, Y. Xu, “News articles retrieval from microfilm images” , IJCAI'99 Workshop:Text Mining: Foundations, Techniques and Application, Stockholm, Sweden, August 2, 1999, pages 110-116

[2] Marc Damashek, “Gauging similarity via n-grams: Language-independent sorting, categorization, and retrieval of text”, Science, 267, 1995, pages 843-848

[3] Cavnar, William B., and Trenkle, John M., "N-Gram-Based text categorization", Proceedings of the 1994 Symposium On Document Analysis and Information Retrieval, University of Nevada, Las Vegas, April 1994.

[4] C.Y. Suen, “N-gram statistics for natural language understanding and text processing” IEEE Trans. on Pattern Analysis & Machine Intelligence. Vol. PAMI-1, No. 2, April 1979, pages 164-172

[5] F.R. Chen and D.S. Bloomberg, “Extraction of thematically relevant text from images”, Proceeding of the Symposium on Document Analysis and Information Retrieval, 1996, pages 163-178

[6] F.R. Chen, D.S. Bloomberg, “Extraction of indicative summary sentences from imaged documents”, Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR’97), Volume 1, 1997, pages 227-232

[7] J.J. Hull; J.F. Cullen, “Document image similarity and equivalence detection”, Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR’97), Volume 1, 1997, pages 308-312

[8] D. Doermann, Li Huiping, O. Kia, “The detection of duplicates in document image databases”, Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR’97), Volume 1, 1997, pages 314-318

[9] Yaodong He; Zao Jiang; Bing Liu; Hong Zhao, “Content-based indexing and retrieval method of Chinese document images”, Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR’99), 1999, pages 685-688

[10]A.F. Smeaton, A.L. Spitz, “Using character shape coding for information retrieval”, Proceeding of the Fourth International Conference on Document Analysis and Recognition (ICDAR’97), Volume 2, 1997, pages 974-978

Table 1. Comparison of document similarity between N-Gram algorithm (text mode) and our method

		N01	N02	N03	N04	N05	N06	N07	N08	News Title
N01	*	1.000	0.067	0.060	0.051	0.049	0.050	0.038	0.057	Wanted A Japanese Bill Gates
	**	1.000	0.089	0.070	0.064	0.093	0.100	0.093	0.103	
N02	*	0.067	1.000	0.117	0.041	0.058	0.087	0.027	0.066	Bangkok wants thais to holiday at home
	**	0.089	1.000	0.113	0.081	0.090	0.111	0.097	0.105	
N03	*	0.060	0.117	1.000	0.108	0.187	0.107	0.064	0.070	Hun Sen set to meet Annan over tribunal
	**	0.070	0.113	1.000	0.117	0.161	0.135	0.116	0.120	
N04	*	0.051	0.041	0.108	1.000	0.050	0.201	0.271	0.335	Plan to send more police to Dili
	**	0.064	0.081	0.117	1.000	0.080	0.164	0.187	0.205	
N05	*	0.049	0.058	0.187	0.050	1.000	0.113	0.058	0.075	Russian graft probe moves to Switzerland
	**	0.093	0.090	0.161	0.080	1.000	0.156	0.140	0.117	
N06	*	0.050	0.087	0.107	0.201	0.113	1.000	0.252	0.370	Tensions rise after vote in East Timer
	**	0.100	0.111	0.135	0.164	0.156	1.000	0.213	0.278	
N07	*	0.038	0.027	0.064	0.271	0.058	0.252	1.000	0.347	Vigilantes taking no chances
	**	0.093	0.097	0.116	0.187	0.140	0.213	1.000	0.256	
N08	*	0.057	0.066	0.070	0.335	0.075	0.370	0.347	1.000	Jakarta rushes troops to E. Timor
	**	0.103	0.105	0.120	0.205	0.117	0.278	0.256	1.000	

Notes:

*: The result of N-Gram algorithm based on text-mode documents.

**: The result of out method.