A GENETIC APPROACH TO CHINESE CLASS DESCRIPTOR GENERATION

Tyne Liang and Chun-heng Kuo

Department of Computer and Information Science

National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Email: tliang@cis.nctu.edu.tw

ABSTRACT

To facilitate management of a huge amount of electronic data, efficient classification is needed. One critical issue to affect classification is the selection of class descriptors. In this paper a genetic-based class descriptor generator is proposed and investigated for Chinese document classification. It is based on the idea that with a well-designed genetic algorithm, a set of fitter descriptors can be obtained. Meanwhile, a content-based finder is incorporated with the proposed generator to discover the initial set of chromosomes. The proposed generator is verified with a real corpus in both unweighted and weighted settings. Comparisons to the traditional union-based approach are also made and the experimental results show that the generator indeed produces a better set of class descriptors than union-based model in both the cases of predefined classes and re-clustered classes.

1. INTRODUCTION

As the network technologies being widely applied to various areas of business, institutions and industries, tremendous amounts of electronic data grow more rapidly than ever before. Among textual processing techniques, efficient classification will undoubtedly facilitate document management and consequently improve retrieval satisfaction [1]. To design an efficient classification, one major step is the construction of accurate class descriptors so that the characteristic properties of a certain class can be sufficiently and accurately represented.

Unlike word or noun phrase extraction from small pieces of textual segments [2, 3, 4, 5, 6], the extraction of sufficient number of good class descriptors in Chinese textual processing has not been deeply investigated [7, 8]. One traditional solution to construct class descriptors is based on the union of all the document descriptors in the same class. Due to its simplicity, such solution is generally implemented in many classification models, such as class hierarchy and network approaches [9, 10]. To reduce the number of class descriptors generated by union-based approach, a weighting function can be applied to weight each candidate class descriptor and those descriptors with high weight will be selected as final class descriptors. On the other hand, a fixed number of representative terms can be selected by χ^2 statistic method in [11]. However it will happen that a large number of class descriptors yields large classification overhead, while a small number of class descriptors producing lower precision.

In recent years, applications of genetic algorithms to English document retrieval are discussed. In [12] a genetic algorithm and feedback method were proposed to get appropriate document descriptors. The same strategy was also used for query optimization by Kraft et al [13]. In [7] a genetic algorithm was used with a

neural network to get the representation of document set. In this paper, investigation of the genetic algorithms particularly for Chinese textual applications is concerned. A genetic approach to generate suitable class descriptors for each class of thesis documents is proposed. Moreover, a content-based finder is incorporated so as to create the set of document descriptors. Experimental results show that the proposed generator will yield better performance than the traditional union-based approach in terms of various measurements for both pre-classified case and re-clustered case.

The remainder of this paper is organized as follows. Section 2 introduces the incorporated content-based finder to produce document descriptors. Section 3 describes the proposed generator to generate class descriptors for each class. Section 4 describes and analyzes the various experiments and measurements. Section 5 gives conclusion.

2. THE CONTENT-BASED DOCUMENT DESCRIPTOR FINDER

The incorporated content-based finder is incorporated to produce document descriptors in the proposed generator. It contains tokenization, extending and appending processes. The corpus used in the experiments is collected from National Chiao Tung University Online Chinese Textual Database (http://ovid.infospring.nctu.edu.tw). It contains 20000 thesis documents, including thesis abstracts, author-given keywords, and thesis titles. The thesis corpus purposely covers 100 departments (whose names are used as class names) and for each department we extracted 200 documents.

The tokenization process identifies each Chinese noun from titles, abstracts and author-given keyword set. The identification is implemented by a word base which contains the author-given keywords and word dictionary developed by Institute of Information Science, Academia Sinica. Words occurring in titles and abstracts are extracted on the basis of maximum forward matching and their related features such as thesis number, word appearance location and occurring frequencies are recorded.

In the corpus, each thesis document is assigned with one to twelve author-given keywords and there are total 42616 unique keywords whose length (in terms of number of characters in a Chinese word) varies from one to sixteen. Since these keywords are subjectively assumed to contain more information than the other words occurring in a thesis document, they become the main part of document descriptor set. There is about 18500 and 6000 words out of total 42416 keywords occurring only once or twice. It will be found that such high uniqueness may yield a poor retrieval recall if thesis retrieval is implemented on the basis of the author-given keywords only.

In fact, most of long-length words are produced by compounding process. On the other hand, most of Chinese polysyllabic words are disyllabic (two-character long) and trisyllabic (three-character long). Hence an extending process is employed in a way such that the keywords containing more than three characters will be extended into bigrams (two-character long) and trigrams (three-character long). Although there are some grams which may not be a word through the extending procedure, they are informative and useful at retrieval [14].

The extending process is implemented as follows:

Assume that $\mathbf{S} = \{s_1, s_2, ..., s_L\}$ is an order set and s_l $\{1 < l < L\}$ is a Chinese character, then $\mathbf{S}' = \{P_1, P_2, ..., P_m\}$ is an extended set, if P_i is an partial order set of $\mathbf{S}$ for $i = 1,..,m$ and, $\min\_length \le |P_i| \le \max\_length$, $P_1 \cup P_2 \cup ... \cup P_m = \mathbf{S}$.

As a result, the number of unique author-given document descriptors is reduced from 42616 to 11119. Though the extending process increases the average number of descriptors per thesis from 5.45 to 6.67, there are still 43% thesis data contain less than six descriptors. In order to enhance the similarity between documents, the appending process is employed. It first removes those descriptors whose occurrence frequencies are lower than three. Then the appending process will find those descriptors which are associated with descriptors generated after extending process. The discovering is based on the association A_{ij} between descriptors *i* and *j* is defined as follows:

$$A_{ij} = \frac{2D_{ij}}{D_i + D_j} \qquad (2.1)$$

where $D_{i,j}$*: the number of thesis documents in which both descriptors i and j occur*

$D_{i,}$*: the number of thesis documents in which descriptor i occurs*

$D_{j,}$*: the number of thesis documents in which descriptor j occurs.*

A descriptor-to-descriptor matrix is implemented to store the association values and each extended author-given keyword will have its associated descriptors list ordered by the association value. The appending process is implemented in such a way that each document will be appended with the associated descriptors in the order of their association values till it has ten document descriptors. After appending process, there is a total of 7670 unique document descriptors which will be used as candidate class descriptors for 100 classes.

3. THE GA-BASED CLASS DESCRIPTOR GENERATOR

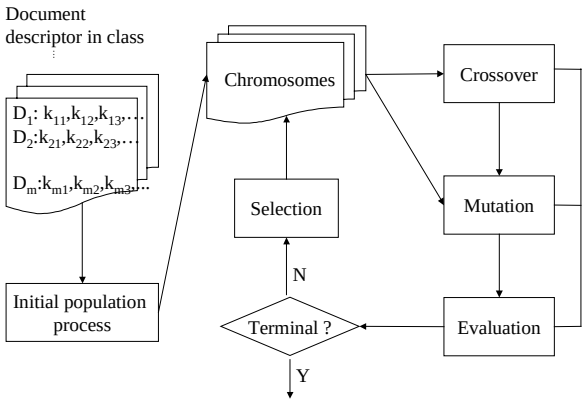

Figure 1: The flowchart of the Ga-based generator

Figure 1 illustrates the flowchart of the proposed genetic algorithm-based (abbreviated by Ga-based) generator which starts with an initial set of solutions called population. Initial population can be chosen randomly or purposely or both. The choice of population size is critical since small-size population may make genetic algorithm converge too fast, yielding a local optimal solution while a large size population will take more computation. Each individual in the population is known as a chromosome. The initial set of chromosomes in the proposed generator is composed of two parts. The first part is purposely selected from those documents with high similarity values so that those potential descriptors will be in promising regions of search space. The similarity for document d_i is calculated on the basis of *Dice* function as follows [15]:

$$S(d_i) = \frac{\sum_{j=1}^{n} dice(d_i, d_j)}{n} \qquad (3.1)$$

$$dice(d_i, d_j) = \frac{2 \times (d_i \cap d_j)}{|d_i| + |d_j|}$$

where n: the number of documents in a certain class.

The second part is generated randomly in order to prevent the genetic algorithm from falling into local optimal solution and converging quickly if the

chromosomes selected from document identifiers are very similar. Under the constraint of our computers, the population size in the experiments is 200.

Since the proposed generator will be implemented in both the weighted and unweighted settings, a chromosome will be represented with a binary string for an unweighted model or a real string for a weighted model. The weight $w(t_{i,j})$of document descriptor t_i in document j is calculated as follows:

$$w(t_{i,j}) = 1 \quad \text{if } t_{i,j} \text{ is selected from author-given keyword set;} \qquad (3.2)$$
$$= A_{ij} \quad \text{otherwise.}$$

During generation, a chromosome will evolves through selection, crossover, mutation and evaluation components in order to produce much fitter offspring. An appropriate fitness function to evaluate chromosomes is needed and it is critical issue in designing the Ga-based generator. The closer a chromosome to an optimal solution, the higher its fitness value will be. In the proposed generator, the fitness function *Sim(X, Y)* is based on Jaccard function [16] as follows:

$$Sim(X,Y) = \frac{\sum_{i=1}^{t} x_i y_i}{\sum_{i=1}^{t} x_i^2 + \sum_{i=1}^{t} y_i^2 - \sum_{i=1}^{t} x_i y_i} \qquad (3.3)$$

where $X = (x_1, x_2, \ldots, x_t)$,

$Y = (y_1, y_2 \ldots, y_t)$,

x_i, y_i: *weight of the i-th descriptor.*

It will be used to measure the similarity between a chromosome *X* (class descriptors) and document descriptors *Y* in a class. A chromosome with higher fitness value will be good set of class descriptors to represent class content.

In the proposed Ga-based generator, the selection component will select chromosomes from the population for reproduction. The fitter the chromosome is, the more likely it will be selected to reproduce. The selection operator will be employed to select some chromosomes from parents as well as offspring according to their fitness values. The sample space in selection is the enlarged sample space. A deterministic sample method is used to select the best population-size chromosomes from the sample space as new population in the next generation.

The crossover component is to explore all search space, and exploit the promising region in each space. In the proposed model, the crossover is a simple one-point crossover, choosing a point at random and exchanging the segments to the right of this point. Meanwhile the crossover rate is set to be 0.7 in order to reduce the exploring time spent in unpromising search.

As to the mutation component, it can provide new genes which are not present in the initial population, and improve the genes lost from the population during selection process. The mutation operator flips one gene from 1 to 0, and vice versa for the binary string type of chromosomes in the unweighted model. On the other hand, the mutation operator will add a random value to a gene for the real value type of chromosomes in the weighted model. The value will be restricted to [0..1]. Meanwhile a mutation rate is practically set to be 0.001 in the implementation. It is expected that a good set of document descriptors will be extracted as class descriptors when the proposed generator converges or it runs 100 iterations.

4. EXPERIMENTS AND COMPARISONS

Two kinds of experiments are investigated in this paper. One is between the Ga-based generator and the traditional union-based generator. The other is to explore the applicability of the proposed model in which the thesis documents are classified by an

artificial neuro network (abbreviated as ANN) classifier.

4.1 Evaluation Measurements

There are several measurements used to evaluate the proposed generator. The first one is *classification accuracy* as defined in equation 4.1.

$$Classification accuracy = \frac{no.\ of\ documents\ correctly\ classified}{no.\ of\ documents\ to\ be\ classified} \quad (4.1)$$

During classification a thesis document d_i will be assigned into a class C_j which has the highest similarity $Sim(C_j, d_i)$(defined as equation 3.3). Similarly, measurements can be also done from the viewpoint of retrieval. Since each class is represented with a set of class descriptors, a whole class of documents will be retrieved during retrieval whenever the C_j has enough similarity with an incoming query q_i. So those documents with top two-hundred similarity values w.r.t. class C_j will be treated as the documents in class C_j. Hence the average retrieval *precision* and *recall* for 100 classes can be calculated as follows:

$$precision = \frac{\sum_{i=1}^{100} p_i}{100} \quad (4.2)$$

$$where \quad p_i = \frac{retrieved\ and\ correct\ documents}{no.\ of\ retrieved\ documents}$$

$$recall = \frac{\sum_{i=1}^{100} R_i}{100} \quad (4.3)$$

$$where \quad R_i = \frac{retrieved\ and\ correct\ documents}{no.\ of\ relevant\ documents}$$

A weighted precision-recall formula is proposed in Equation 4.4 where both w_1 and w_2 are 0.5.

$$Weighted\ Precision\_Recall(WPR) = w_1 \times pecision + w_2 \times recall \quad (4.4)$$

Meanwhile a good set of class descriptors will have more similarity with the documents in the same class than the similarity of other classes. Hence the average similarity as defined in Equation 4.5 can be used to measure the quality of class descriptors for each class. Higher similarity will imply a better set of class descriptors.

$$Average\_sim = \frac{\sum_{j=1}^{100} SM(C_j)}{100} \quad (4.5)$$

where

$$SM(C_j) = \frac{\sum_{i=1}^{200} Sim(C_j, d_i)}{200}$$

$Sim(C_j, d_i)$: calculated as Equation 3.3.

In addition, evaluation can be done in terms of the average space overhead required for a class descriptor generator. It can be defined as Equation 4.6.

$$Average\_space\_overhead = \frac{\sum no.\ of\ class\ descriptors\ per\ class}{100} \quad (4.6)$$

On the other hand, a set of class descriptor can be used as seed to retrieve documents during retrieval, so a generator can be also evaluated in terms of hit ratio which is defined as follows:

$$Average\ hit\ ratio = \frac{\sum_{i=1}^{100} h_i}{100} \quad where$$

$$h_i = \frac{no.\ of\ relevant\ and\ retrived\ documents\ for\ class\ i}{no.\ of\ retrieved\ documents\ for\ class\ i} \quad (4.7)$$

Those *Top K* documents w.r.t. the class descriptors will be verified whether they are in the corresponding class or not. A good generator of class descriptors will certainly yield a higher hit ratio.

4.2 Comparisons Between Weighted And Unweighted Models

In the verification, the proposed class descriptor will be

implemented in both unweighted and weighted cases for the 20000 thesis documents collected from 100 departments (used as classes), and each class contain 200 thesis documents. From Tables 4.1 and 4.2 it is found that the weighted model yield better results than the unweighted model in terms of various measurements.

	Unweighted model	Weighted model
Classification Accuracy	0.617	0.760
Average retrieval precision	0.610	0.752
Average retrieval recall	0.616	0.760
WPR	0.613	0.756
Average_sim	0.377	0.390
Average_space_overhead	7.465	9.202

Table 4.1: weighted vs. unweighted models at class size=200

	Top 40	Top 80	Top 120	Top 160	Top 200
Unweighted Model	0.717	0.658	0.602	0.541	0.491
Weighted Model	0.713	0.671	0.636	0.601	0.563

Table 4.2: hit ratio for weighted and unweighted models at class size=200

4.3 Comparisons Between Ga-based and Union-based Models

The proposed Ga-based constructor is compared with the general union-based one in both unweighted and weighted settings. In union-based model. each set of class descriptor, in an unweighted case, is produced by unioning all document descriptors in the same class. On the other hand, Equation 4.8 will be used to get weight for a descriptor in the weighted case.

$$W_U(t_{ij}) = \frac{\sum_{j=1}^{200} w(t_{ij})}{D_i} \qquad (4.8)$$

where $w(t_{i,j})$: calculated as Equation 3.2

D_i: the number of documents in which t_i is the document descriptor.

From Tables 4.3 and 4.4, one can find that for both weighted and unweighted cases, the proposed Ga-based generator yields better performance than the traditional Union-based generator in terms of various measurements.

	Ga	Union
Classification accuracy	0.617	0.449
Average retrieval precision	0.610	0.703
Average retrieval recall	0.643	0.449
WPR	0.613	0.579
Average_sim	0.377	0.023
Average_space_overhead	7.465	474.727

Table 4.3: Ga-based Model vs. Union-based Model in unweighted case

	Ga	Union
Classification accuracy	0.760	0.581
Average retrieved precision	0.752	0.764
Average retrieved recall	0.760	0.581
WPR	0.756	0.672
Average_sim	0.390	0.018
Average_space_overhead	9.202	474.727

Table 4.4: Ga-based Model vs. Union-based Model for weighted case

As described before, the tested thesis documents are collected from 100 departments, so each document is classified into its predefined class. However, there exist documents even collected from the same department name may be not similar to each other, thus affecting the performance of the purposed model. Also there exists the case in our test corpus that some departments are similar to each other for example, *Computer and Information Science Department* and *Computer Engineering and Information Science Department*. So even some thesis documents contain high similarity they have to be classified into their predefined classes. Hence an Adaptive Resonance Theory network (ART) is applied to re-cluster the documents. The reason to use ART as a classifier is because it can retain useful information in memory, and meanwhile it can learn new important facts or information. In our experiment, an input vector is the set of document descriptors. Each node in input layer

indicates whether a descriptor appear in a document or not. The output vector will indicate which class a document should belong to and each node in output layer represents a class. The vigilance value in the experiments is set to 0.4.

Figures 4. 3 throughout 4.5 show that the Ga-based model (indicated with dark straight line) is better than union-based model (indicated with light dots) w.r.t. different measurements in both predefined (class size=200) and re-clustered, weighted and unweighted cases .

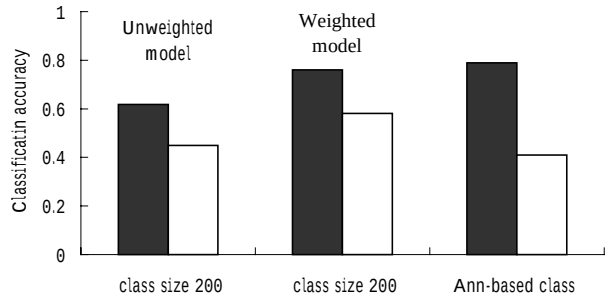

Figure 4.3: Ga-based model vs. union-based model (I)

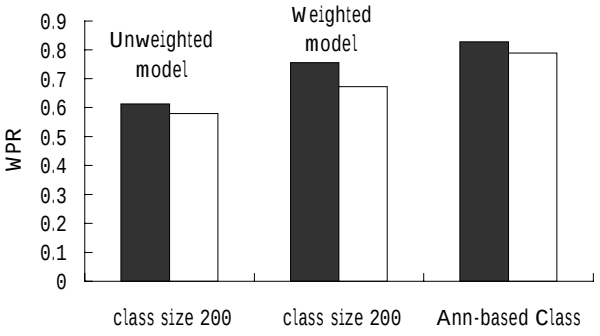

Figure 4.4: Ga-based model vs. union-based model (II)

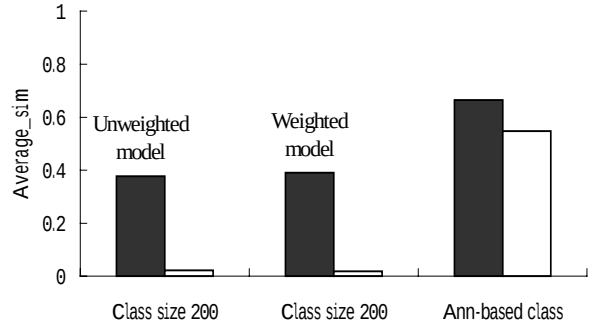

Figure 4.5: Ga-based model vs. union-based model (III)

5. CONCLUSION

In this paper, a genetic approach to generate class descriptors especially for Chinese textual retrieval is concerned and investigated. Meanwhile, a content-based finder is incorporated with the proposed generator to generate the appropriate document descriptors. In addition, various evaluation measurements are proposed and employed to evaluate the proposed generator. Experimental results show that the proposed Ga-based generator indeed produces a better set of class descriptors than union-based model in both the cases of predefined and re-clustered classes and the cases of weighted and unweighted model.

6. ACKNOWLEDGEMENT

This paper is partly supported by the grant from the National Science Council under the contract No. NSC89-2213-E009-029.

7. REFERENCE

[1] W. B. Croft, “A Comparison of Text Retrieval Models,” The Computer Journal, Vol. 35, No. 3, pp. 279-290, 1992.

[2] M.-W. Wu and K.-Y. Su, “Corpus-based Automatic Compound Extraction with Mutual Information and Relative Frequency Count,” Proceedings of ROCLING VI, Nantou, Taiwan, ROC, pp. 207-216, Sep. 1993.

[3] J. Y. Nie, M. L. Hannan, and W. Jin, “Combining Dictionary, Rules and Statistical Information in Segmentation of Chinese,” Computer Processing of Chinese and Oriental Languages, Vol. 9, No. 2, pp. 125-143, 1995.

[4] H. H. Chen, Y. W. Ding and S. C. Tsai, “Named Entity Extraction for Information Retrieval,”

Computer Processing of Oriental Languages, Special Issue on Information Retrieval on Oriental Languages, 12(1), pp. 75-85, 1998.

[5] M. Fuketa, S. Mizofuchi, Y. Hayashi and J. I. Aoe, "A Fast Method of Determining Weighted Compound Keywords from text Databases," Information Processing & Management, Vol. 34, No. 4, pp. 431-442, 1998.

[6] L. F. Chien, "PAT-tree-based adaptive keyphrase extraction for intelligent Chinese information retrieval," Information Processing & Management, Vol. 35, No. 4, July 1999.

[7] Hsinchun Chen and Jinwoo Kim "GANNET: A Machine Learning Approach to Document Retrieval," Journal of Management Information Systems, Vol. 11, No. 3, pp. 7-41, 1995.

[8] T. Liang and S. T. Tseng, "Classification based on multilayer feedforward with back propagation neural network," Proceedings of Workshop on Artificial Intelligence, 1998 International Computer Symposium, Tainan, Taiwan, pp. 62-67.

[9] C. H. Lin and H. Chen, "An Automatic Indexing and Neural Network Approach to Concept Retrieval and Classification of Multilingual (Chinese-English) Documents," IEEE Transactions on System, Man and Cybernetics-Part B : Cybernetics, Vol. 26, No. 1, pp. 75-88, 1996.

[10] T. Liang and C. C. Chang, "Chinese textual retrieval based on fuzzy concept network," Proceedings of National Computer Symposium Taipei 1999.

[11] J. J. Tsay, J. D. Wang, "Comparing classifiers for automatic Chinese textual categorization," Proceedings of National Computer Symposium 99, Tankang, Taiwan.

[12] Michael Gordon "Probabilistic and Genetic Algorithms for Document Retrieval," Communication of ACM, Vol. 31, No. 10, pp. 1208-1218, 1988.

[13] D. Kraft, E. F. Petry, F. B. Buckles, and T. Sadasivan, "Genetic Algorithm for Query Optimization in Information Retrieval: Relevance Feedback," Advances in Fuzzy Systems—Application and Theory Vol.7, pp. 155-173, 1996.

[14] K. L. Kwok, "Comparing representation in Chinese information retrieval," Proceedings of SIGIR 1997, Philadelphia, PA., U.S.A., pp. 34-41.

[15] G. Salton, "Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by computer, Reading", MA : Addison-Wesley, 1989.

[16] Melanie Mitchell "An Introduction to Genetic Algorithms," 1996.