Journal of Korea Multimedia Society Vol. 11, No. 12, DECEMBER 2008(pp. 1749-1757)

Single Pass Algorithm for Text Clustering by Encoding Documents into Tables

Taeho Jo†

ABSTRACT

This research proposes a modified version of single pass algorithm specialized for text clustering. Encoding documents into numerical vectors for using the traditional version of single pass algorithm causes the two main problems: huge dimensionality and sparse distribution. Therefore, in order to address the two problems, this research modifies the single pass algorithm into its version where documents are encoded into not numerical vectors but other forms. In the proposed version, documents are mapped into tables and the operation on two tables is defined for using the single pass algorithm. The goal of this research is to improve the performance of single pass algorithm for text clustering by modifying it into the specialized version.

Key words: Text Clustering, Single Pass Algorithm, String Vector

1. INTRODUCTION

Text clustering refers to the process of segmenting a particular group of documents into subgroups each of which contains content-based similar documents. A collection or group of documents is given as the input of the task. Several smaller groups of content-based similar documents are generated from the task as its output. Although there are many heuristic approaches to the task, unsupervised learning algorithms have been used as state of the art approaches to it. As an instance of text mining, text clustering is necessary for organizing documents automatically.

The process of encoding documents into numerical vectors for using traditional unsupervised learning algorithms for text clustering causes the two main problems. The first problem is huge dimensionality where documents must be encoded into very large dimensional numerical vectors for preventing information loss. In general, documents must be encoded at least into several hundreds dimensional numerical vectors in previous literatures. This problem causes very expensive cost for processing each numerical vector representing a document in terms of time and system resources. Furthermore, much more training examples are required proportionally to the dimension for avoiding over-fitting.

The second problem is sparse distribution where each numerical vector has zero values dominantly. In other words, more than 90% of its elements are zero values in each numerical vector. This phenomenon degrades the discrimination among numerical vectors. This causes poor performance of text categorization or text clustering. In order to improve performance of both tasks, the two problems should be solved.

Figure 1 illustrates an original document or documents and its or their surrogate given as a table. The table consists of entries of words and their weights indicating their content based importance in the original document. This research adopts the strategy of encoding documents illustrated in figure

※ Corresponding Author : Taeho Jo, Address : (402-751) 1212 Hitech Inha University, Yonghyundong Namgu Incheon, Korea, TEL : +82-32-860-8984, FAX : N/A, E-mail : tjo018@inha.ac.kr

Reccipt date : Dec. 29 2007, Approval date : Sep. 16, 2008

† School of Computer and Information Engineering Inha University

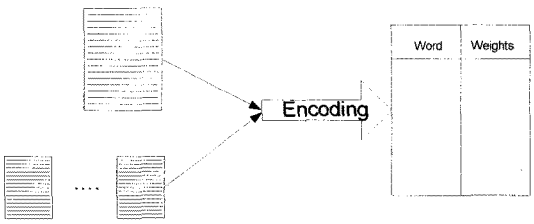

Fig. 1. Original Document or Documents and its or their Table as a Surrogate

1 and applies single pass algorithm under the strategy. A semantic similarity between two documents is computed based on words shared by both tables. The computation will be described in detail in section 4.

This research proposes another version of single pass algorithm where documents are encoded into tables. By doing that, it offers three contributions. The first contribution is to avoid the two problems, huge dimensionality and sparse distribution, by encoding documents into another form which is completely different from numerical vectors. The second contribution is to open a way of developing a new class of approaches to text clustering. The third contribution is to make it easy to generate symbolic clustering rules for tracing why a particular document should belong to a cluster, because the table is close to symbolic data rather than numerical data.

This paper consists of six sections, including this section. In section 2, we will explore previous approaches to text clustering and a previous solution to the two problems. In section 3 and 4, the process of encoding documents into tables and the proposed text clustering system are described in detail, respectively. In section 5, the traditional and proposed versions of single pass algorithm are compared with each other in terms of their clustering performance, in order to validate that the proposed version is more desirable. In section 6, the significance of this research and further research will be mentioned as the conclusion of this article.

2. PREVIOUS WORKS

This article concerns the exploration of previous research on text clustering. As mentioned in section 1, there exist various kinds of approaches to text clustering. However, in exploring previous research, we restrict the scope of approaches only to unsupervised learning algorithms. Among unsupervised learning algorithms, based on their popularities, we select only three representative ones: single pass algorithm, Kohonen Networks, and EM algorithm. In this section, we explore previous cases of using one of the three unsupervised learning algorithms.

A simple and popular clustering algorithm is single pass algorithm. When a number of clusters is far less than a number of objects, this algorithm runs in an almost linear complexity to the number of objects. The algorithm has been popularly used for clustering objects especially in industrial worlds, since it is fast enough to implement a real time clustering system. However, note that quality of clustering objects in this algorithm is not as good as that in other clustering algorithms. In 2000, Hatzivassiloglou et al used this algorithm as an approach to text clustering where documents are encoded into numerical vectors together with linguistic features and compared it with complete pair-wise algorithm [1].

Kohonen Networks is an unsupervised neural network and was used as a popular approach to text clustering [2-4]. WEBSOM was a typical text clustering system where Kohonen Networks was adopted as the approach to text clustering [3-4]. In 1998, its initial version was developed by Kaski et al in 1998 [3]. Each cluster of documents is identified with a group of relevant words. In the system, not only documents, but also words are clustered using Kohonen Networks.

K means algorithm is also a typical approach to not only text clustering but also any other pattern clustering. It is the simplest version of EM algo-

rithm consisting of E-step and M-step [5,6]. In 2000, Vinokourov and Girolami proposed five probabilistic models of hierarchical text clustering as specific versions of the EM algorithm [7]. In 2003, Banerjee et al proposed two variants of the EM algorithm for soft clustering, where each object is allowed to belong to more than one cluster, and applied them to text clustering and gene expression clustering [8].

The previous solutions to the two problems are illustrated in table 2. As the rst attempt, in 2002, Lodhi et al proposed the string kernel for using SVM for text categorization [9]. The string kernel refers to a kernel function of two raw texts which returns their syntactical similarity. Owing to the string kernel, documents did not need to be represented into numerical vectors. However, they attempted to solve the two problems not for text clustering but for text categorization. Furthermore, it took very much time to perform the string kernel and the performance of text categorization failed to be improved.

When using one of the most three popular approaches, documents should be encoded into numerical vectors. Although a previous literature on text mining mentioned the two problems, it was regarded as natural and unavoidable task to encode documents so. However, this research attempts to find solutions to the two problems without accepting it naturally. The solution proposed in this research is to encode documents into another form. After doing that, this research modifies the single pass algorithm to be able to process the form of data.

3. DOCUMENT ENCODING

This section concerns the process of encoding a document or documents into a table. Figure 2 illustrates the process with three steps. A document or documents is given as input of the process, and a list of words and their frequencies is generated

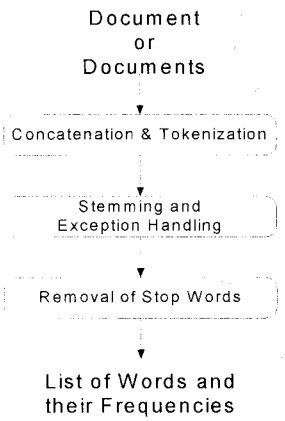

Fig. 2. The Process of Mapping Document or Documents into a Table

from the process. The three steps illustrated in figure 2 will be explained. After that, the three schemes of weighting words will be also mentioned.

As illustrated in figure 2, a document or documents may be given as input of this stage. If more than two documents are given as the input, their full texts are concatenated into an integrated full text. The integrated full text becomes the target for the tokenization. The full text is tokenized into tokens by a white space or a punctuation mark. Therefore, the output of this step is a list of tokens.

The next step to the concatenation & tokenization is the stemming & exception handling, as illustrated in figure 2. In this step, each token is converted into its root form. Before doing that, rules of stemming and exception handling are saved into a file. When the program encoding documents is activated, all rules are loaded into memory and the corresponding one of them is applied to each token. The output of this step is a list of tokens converted into their root forms.

The last step of extracting feature candidates from a corpus is to remove stop words as illustrated in figure 2. Here, stop words are defined as words which function only grammatically without their relevance to content of their document; articles (a an, or the), prepositions (in, on, into, or

at), pronoun (he, she, I, or me), and conjunctions (and, or, but, and so on) belong to this kind of words. It is necessary to remove this kind of words for more efficient processing. After removing stop words, frequencies of remaining words are counted. Therefore, a list of the remaining words and their frequencies is generated as the final output from the stage illustrated in figure 1.

Although there are other schemes of weighting words, we will mention only three schemes as representative ones. For first, we can assign frequencies themselves to words as their weights. For second, we may assign normalized frequencies generated from dividing their frequency by the maximum frequency. For third, we can weights words using equation by equation (1),

$$weight_i(w_k) = tf_i(w_k)(\log_2 D - \log_2 df(w_k) + 1) \quad (1)$$

where $weight_i(w_k)$ indicates a weight of the word, w_k, which indicates its content based importance in the document, i, $tf_i(w_k)$ indicates the frequency of the word, w_k in the document, i, $df(w_k)$ is the number of documents including the word, w_k, and D is the total number of documents in a given corpus. Among the three schemes, we adopt the third for weighting words in this research.

4. PROPOSED TEXT CLUSTERING SYSTEM

This section concerns the proposed version of single pass algorithm and the text clustering system which adopts the proposed version. Figure 3 illustrates the modules involved in implementing the proposed text clustering system. The first module is document encoder given as the interface of the system and encodes documents into tables. The second module is similarity measurer and computes a semantic similarity between two documents. The third module is document arranger and arranges documents into their content based corresponding clusters or creates a new cluster.

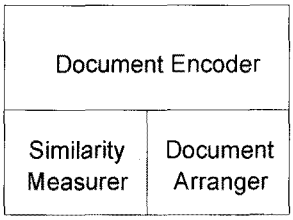

Fig. 3. The Modules inolved in Implementing the Proposed Text Clustering System

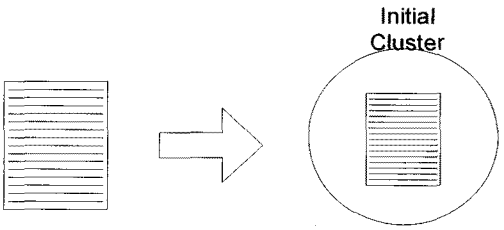

Fig. 4. The Initialization of the Single Pass Algorithm

Figure 4 illustrates the initialization of the single pass algorithm as its first step applicable to the first document. The initialization refers to the process of creating the first cluster and containing the first document in the cluster. The first document is given as the input of the step. The first document contained in the cluster becomes its prototype which represents it[1]. Therefore, from the initialization, a cluster with a document is generated as the output as illustrated in figure 4.

Figure 5 illustrates the process of generating a normalized value as a similarity between two documents. The role of document encoder was already mentioned above. The process of encoding documents into tables was already described in detail in section 3. The module, similarity measurer, computes a similarity between two tables based on words shared by both tables. The process illustrated in figure 5 generates a normalized real value

1) In other literatures on single pass algorithm, average over similarities of a document with contained ones in a cluster is used as a similarity between the document and the cluster. However, in this research, a similarity between a document and the first document in the cluster is used as the similarity between the document and the cluster for fast clustering.

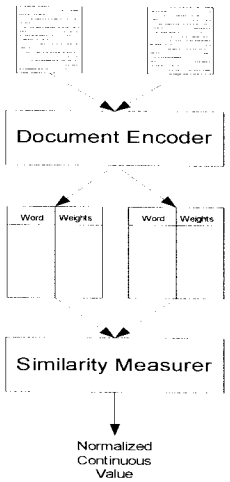

Fig. 5. The Process of Generating a Similarity between Two Documents

as the output.

Figure 6 illustrates the process of generating an output table from two input tables for computing a semantic similarity between two tables. Let the two tables be 'Table 1' and 'Table 2'. By getting words shared by Table 1 and Table 2, the output table, Table 3, is built, and each word in Table 3 has its two weights: one from Table 1 and the other from Table 2. From the three table, we can define four sums of weights as follows.

- Sum_Weight 1: The sum of weights of words contained in Table 1
- Sum_Weight 2: The sum of weights of words contained in Table 2

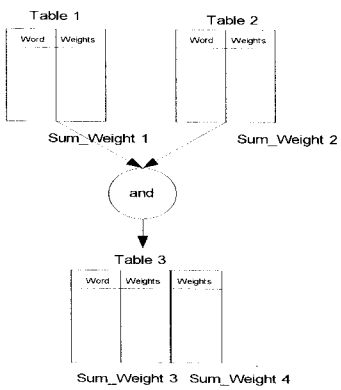

Fig. 6. The Process of computing a Similarity between Two Tables

- Sum_Weight 3: The sum of weights from table 1 of words contained in Table 3
- Sum_Weight 4: The sum of weights from table 2 of words contained in Table 3

Therefore, the similarity between Table 1 and Table 2 is computed using equation (2).

$$similarity = \frac{Sum\_Weight3 + Sum\_Weight4}{Sum\_Weight1 + Sum\_Weight2} \quad (2)$$

Equation (2) indicates that more shared words between two tables, higher the value of similarity is. The output of equation (2) is given as a normalized value between 0 and 1. When there is no shared words between two tables, zero value is generated, since both Sum_Weight 3 and Sum_Weight 4 are zeros. If both tables are identical to each other, equation (2) generates 1 value, since Sum_Weight 3 is identical to Sum_Weight 1 and Sum_Weight 4 is identical to Sum_Weight 2; the addition of Sum_Weight 3 and Sum_Weight 4 is same to that of Sum_Weight 1 and Sum_Weight 2.

Figure 7 illustrates the process of arranging documents or creating one more cluster. The threshold of similarity is given as the parameter of the single pass algorithm. For each successive

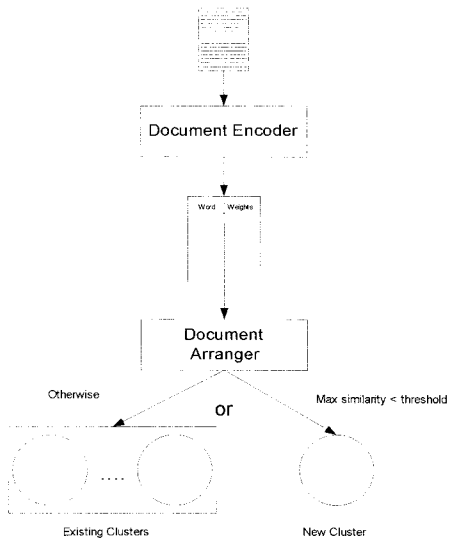

Fig. 7. The Process of arranging Documents or creating one more Cluster

Table 1. The Different and Shared Points between Traditional and Proposed Version

	Traditional Version	Modified Version
Clustering Process	Initialization and Arrangement	
Encoding Documents	Numerical Vectors	Tables
Semantic Similarity	Inner Product Cosine Similarity Euclidean Distance	Equation (2)

document, its similarities with prototypes of clusters are computed using equation (4). If its maximum similarity is less than the threshold, one more cluster is created and it is contained in the cluster. Otherwise, the document is arranged into the cluster corresponding to the maximum similarity.

Table 1 summarized the difference and shared between the traditional and proposed versions of single pass algorithm. Basically, in the both versions, single pass algorithm consists of the two steps: initialization and arrangement. In the traditional version, documents are encoded into numerical vectors, while in the proposed one, they are encoded into tables. In the traditional version, a similarity between two documents is computed based on Euclidean distance or inner product between their corresponding numerical vectors, while in the proposed version, it is computed based on words shared by their two corresponding tables. Single pass algorithm is a very fast clustering algorithm, but its reliability is very poor since prototypes of clusters are fixed while clustering objects.

5. EMPIRICAL RESULTS

This section concerns another set of experiments for comparing the two versions of single pass algorithm one more time. The test bed used in this set of experiments is another collection of electronic news articles called 20NewsGroups. In this set of experiments, documents are encoded according to the rules illustrated in table 3. The parameter of both versions of single pass algorithm is set identically to that in the previous set of experiments. The goal of this set of experiments is to observe the comparison of two versions one more time on another test bed.

The test bed is 20NewsGroup which is a large collection of news articles. The test bed was used in previous literatures [10-12]. The collection consists of 20 categories and 20,000 news articles. The test bed was obtained by downloading the collection from the web site, kdd.ics.uci.edu as an integrated compressed file. Each news article is exclusively labeled with one of the twenty categories. This fact is the reason for adopting the collection as the test bed for evaluating the performance of text clustering, instead of the most standard test bed, Reuter21578.

Ten subgroups are built from the test bed for evaluating the approaches to text clustering. The twenty predefined categories are grouped into two groups of ten categories. Each subgroup consists of ten categories and 500 news articles; each category in the subgroup contains 50 news articles. Among ten subgroups of documents, five subgroups span over ten categories of twenty categories. The other five subgroups span over the other ten categories.

Differently from k means algorithm and Kohonen Networks, the similarity threshold is given as the parameter of the single pass algorithm, instead of the number of clusters. In the clustering algorithm, the number and the size of clusters are determined automatically, depending on the similarity threshold. The parameter is given as a continuous normalized value between zero and one, and if it is close to zero, the small number of large clusters is resulted in. If it is close to one, the large number of small clusters is resulted in. Since the test bed has a small number of target categories, the parameter is set as 10^{-6} close to zero.

Table 2 illustrates the three groups by input size,

and within each group the two versions of single pass algorithm are compared with each other. In the first group, documents are encoded into 100 dimensional numerical vectors in the traditional version, and they are encoded into 10 entries tables in the proposed version. In the second group, they are encoded into 250 dimensional numerical vectors in the traditional version, while they are encoded into 25 entries tables in the proposed version. In the third group, they are encoded into 500 dimensional numerical vectors and 50 entries tables in the traditional and proposed version, respectively. Note that we set the dimensions of numerical vectors based on previous dimensions in the previous literatures.

Since the number of categories or clusters as the results of document clustering may be different from the desired number, we adopt another measure for evaluating the results, instead of F1 measure. The adopted evaluation measure consists of two factors. The first factor is intra-cluster similarity and it should be maximized closely to one for the desirable clustering. The second factor is inter-cluster similarity and it should be minimized closely to zero for the desirable clustering. The measure is called clustering index, combining the two factors with each other, and it is described in detail in the literature [13].

Figure 8 illustrates the results of comparing the two versions of the single pass algorithm on this test bed. The x-axis of figure 1 contains the three groups of bars by input size as shown in table 3. Within each group, the black bar indicates the performance of the previous version, while the white bar does that of the proposed version. The y-axis indicates the logarithmic clustering index [10] computed by equation (3),

$$\frac{1}{-\log_{10} CI} \tag{3}$$

where the base of the logarithm is ten. The reason of rescaling the clustering index logarithmically is that the performance difference between

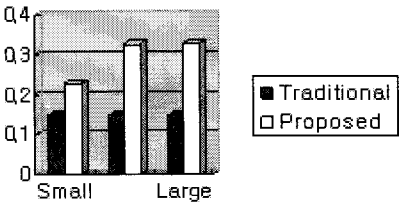

Fig. 8. The Results of Comparing Two Versions on Subgroups of 20NewsGroups

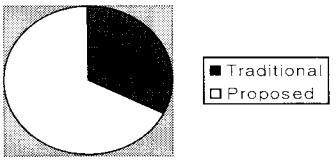

Fig. 9. Visualization of Comparison of Two Versions on 20NewsGroups

the two versions in the original scale is too big to display with a bar-graph.

Figure 9 visualizes the comparison of the two versions of single pass algorithm on the test bed, 20NewsGroups. The portion of traditional version indicates 0.1481 by the averaged logarithmic clustering index. That of proposed version indicates 0.2944. The ratio of the proposed version to the traditional version becomes 66.34; the ratio is similar as that in the previous test bed. The comparison of the two versions on this test bed characterizes almost identically to that on the previous test bed.

6. CONCLUSIONS

The significance of this research is to specialize the single pass algorithm to be more suitable for text clustering, solving the two problems completely. We used a more suitable measure for evaluating approaches to text clustering, rather than F1 measure. In section 5, the proposed version worked better than the traditional version through the two sets of experiments. The reason of the better performance of the proposed version is that the two main problems were addressed by encoding documents into another form different from numerical vectors. From the empirical validation in

section 5, we may conclude that the proposed version is more desirable than the traditional version.

There may be many ways of computing weights of words. In this research, we computed weights of words using equation (1), because of the popularity in the information retrieval. Note that the weights do not reflect exactly the relevancy of words to a given category or a content of a document. We need to develop several state of the art schemes for computing weights. In further research, we will compute weights of words using by combining multiple schemes with each other.

If we could develop various schemes for computing weights of words, we may define multiple tables to a document or corpus. There are two ways for treating multiple tables. The first way is to integrate multiple tables corresponding to a document or a corpus into a table. The second way is to treat the multiple tables as a committee. In further research, we will evolve the proposed approach by encoding a document or corpus into multiple tables.

Note that there is another clustering algorithm, k means algorithm, as well as single pass algorithm. Like the single pass algorithm, we can modify the k means algorithm so. The difference of the k means algorithm from the single pass algorithm is that a number of clusters is given as the parameter instead of the similarity threshold and prototypes of clusters change continually during clustering objects. In order to modify the k means algorithm, we must define one more operation where a table representing a group of tables should be defined. By building a table consisting of words spanning over tables, we can do that.

In this research, documents were encoded into tables with their fixed size. Note that the optimal size of tables depends on their corresponding document. We must optimize the size of each table for satisfying the two factors; reliability and efficiency. In other words, too large tables cause poor efficiency and too small one cause poor reliability. In further research, we will develop a scheme for sizing tables differently.

REFERENCES

[1] V. Hatzivassiloglou, L. Gravano, and A. Maganti, "An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering," *The Proceedings of 23rd SIGIR*, pp. 224-231. 2000.

[2] G. Bote, P. Vincent, M. A. Felix, and V. H. Solana, "Document Organization using Kohonen's Algorithm," *Information Processing and Management*, Vol.38, No.1, pp. 79-89. 2002.

[3] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, "WEBSOM-Self Organizing Maps of Document Collections," *Neurocomputing*, Vol.21, pp. 101-117. 1998.

[4] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, V. Paatero, and A. Saarela, "Self Organization of a Massive Document Collection," *IEEE Transaction on Neural Networks*, Vol.11, No.3, pp. 574-585. 2000.

[5] C. Ambroise and G. Govaert, "Convergence of an EM-type algorithm for spatial clustering," *Pattern Recognition Letters*, Vol.19, No.10, pp. 919-927. 1998.

[6] T. M. Mitchell, Machine Learning, McGraw-Hill, 1997.

[7] A. Vinokourov and M. Girolami, "A Probabilistic Hierarchical Clustering Method for Organizing Collections of Text Documents," *The Proceedings of 15th International Conference on Pattern Recognition*, pp. 182-185. 2000.

[8] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Generative model-based clustering of directional data," *The Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 19-28. 2003.

[9] H. Lodhi, C. Saunders, J. Shawe-Taylor, N.

Cristianini, and C.Watkins, "Text Classication with String Kernels," *Journal of Machine Learning Research*, Vol.2, No.2, pp. 419-444. 2002.

[10] LD Baker, AK McCallum, "Distributional clustering of words for text classification," The Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp96-103. 1998.

[11] N Slonim, N Tishby, "Agglomerative information bottleneck," *Advances in Neural Information Processing Systems*, 2000.

[12] G Siolas and Fd'Alche-Buc, "Support Vector Machines based on a semantic kernel for text-categorization," The Proceedings of IEEE International Joint Conference on Neural Networks, pp. 205-209. 2000.

[13] T. Jo and M. Lee, "The Evaluation Measure of Text Clustering for the Variable Number of Clusters," *Lecture Notes in Computer Science*, Vol.4492, pp. 871-879. 2007.

Taeho Jo

Taeho Jo received PhD degree from University of Ottawa in 2006. Currently, he works for Inha University as a professor. He has submitted and published more than 100 research papers to journals and proceedings since 1996. Previously he has ever worked for industrial organizations: Samsung, ETRI, KISTI, and KAIST Institute for IT Convergence. His research interests are text mining, neural networks, machine learning, and information retrieval.