Normalized Table Matching Algorithm for Classifying News Articles

Taeho Jo
School of Computer and Information Engineering
Inha University
Incheon, South Korea
tjo018@inha.ac.kr

***Abstract*—In this research, we propose encoding texts into normalized tables for categorizing texts, automatically. Previously, the table based approach was proposed, but the categorical scores indicating how much the text is relevant to the given category may be overestimated or underestimated by the given text length. As the solution to the problem, in this research, we encode texts into fixed sized tables, define the operation for computing the similarity between two tables as a normalized value, and characterize it mathematically. As the benefits from this research, we are able to compute category scores independently of a given text length, consider weights from both texts, and expect the more stable performance. We validate empirically the proposed approach with respect to the performance and the stability by comparing it with the traditional approaches.**

Keywords-Text Categorization; Table based Matching Algorithm

I. INTRODUCTION

As shown in Figure 1, text categorization refers to the process of assigning one or some of predefined categories to each document. In the task, an unseen document is given as the input, and one or some of the predefined categories are generated the output. The task is regarded as an instance of pattern classification where each object is classified into its own label. For the task, a list of categories should be predefined and sample documents which are manually labeled by one or some of the categories should be prepared as its preliminary tasks. Techniques of text categorization are necessary for processing and managing efficiently textual data which are growing explosively in information systems; many state of the art approaches [1][2][5] have been developed since 1990s.

In order to use a previously developed approach for text categorization, we must encode documents into numerical vectors. Encoding them so causes the two main problems: huge dimensionality and sparse distribution. The first problem, 'huge dimensionality', refers to the phenomena where documents are encoded into too many dimensional numerical vectors for preventing information loss. In spite of using feature selection methods, documents are usually encoded into several hundred dimensional vectors. Under the problem, it takes very much cost for processing each document in terms of time and system resource, and many training examples are required proportionally to the dimension for avoiding over-fitting.

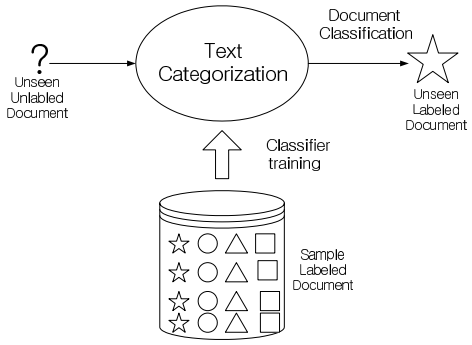

Figure 1. The Process of Indexing Corpus

The proposed version is improved over the previous one [10][11] aspects. For first, in the previous version, texts are encoded into variable sized tables, whereas, in this version, they are done into constant sized ones. For second, in the previous version, the categorical scores are computed by summing weights of tables simply, whereas in this version, they are computed by the proposed operation which is characterized mathematically. For third, in the previous version, the categorical scores are only real values, while in the current version, they are given normalized values. Therefore, in this version, we expect more stable performance as well as better performance.

We expect the three benefits from this research. For first, we overcome the overestimation and the underestimation by variable text lengths. For second, the categorical scores are given as normalized values between zero and one independently of domains; the categorical estimations are performed more stably. Compared with traditional approaches, the proposed approach is expected to have its more stable performance over corpus as well as its better performance. Together with the previous version, the proposed version also solves the main problems in encoding texts into numerical vectors.

This article consists of the five sections. In Section II, we will survey the previous research relevant to this research. In Section III, we describe the proposed version of table based matching algorithm in detail. In Section IV, we

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-217-2

validate empirically the proposed method by comparing it with the popular approaches, considering both performance and stability. In Section V, as the conclusion of this research, we mention the significances and the remaining tasks of this research.

II. Previous Works

This section is concerned with the previous research relevant to this research. Even if various kinds of approaches to text categorization are available, in this research, we count only three typical ones, KNN, Naive Bayes, and Support Vector Machine. In this section, we also survey the previous solutions to the problems in encoding texts into numerical vectors. In spite of its better performance of previous version, we will point out its demerits and mention how to improve it. Therefore, in this section, we will explore the previous research in the three directions.

Let us mention the KNN, the Naive Bayes, and the SVM as the three typical approaches to text categorization. The KNN was used for text categorization by Massand et al. and Yang in 1992 and 1999, respectively [1][2]. The Naive Bayes was used by Mladenic and Grobelink and Eyheramendy et al., in 1999 and 2003, respectively [3][4]. The SVM was used for spam mail filtering by Drucker et al. [5] and it was mentioned as typical approach to text categorization by Cristianini and Shawe-Taylor [6]. However, it requires to encode texts into numerical vectors for using one of the three approaches for the text categorization.

There were previous attempts to solve the problems in encoding texts into numerical vectors. In 2000, Jo initially encoded texts into string vectors instead of numerical vectors as the alternative representations of texts [7]. In 2002, Lodhi et al. proposed the string kernel as a kernel function in using the SVM for the text categorization [8]. In 2007, Lee and K. Kageura tried to solve the problems where many examples are required from the huge dimensionality by generating the virtual documents [9]. The trials show that the problems in encoding texts into numerical vectors were realized.

We started to encode texts into tables instead of numerical vectors and string vectors. In 2008, Jo and Cho created initially the table based matching algorithm as the approach to the text categorization [10]. In 2008, Jo applied it to soft text categorization where more than one category may be assigned to each text [11]. In 2008, Jo proposed the table based approach to the text clustering as well as the text categorization [12]. The previous version of the table based algorithm solved the problems in encoding texts into numerical vectors, but it has its own demerit where the categorical scores are overestimated or underestimated by variable sized texts.

We need to consider the demerits of the previous version, even if it was applied successfully to text categorization. Even more, the string kernel proposed by Lodhi et al. failed to improve the text categorization performance. It is not easy to implement the text categorization algorithms where texts are encoded into string vectors, because operations on string vectors are not defined systematically, mathematically, and theoretically. The previous version of the table based matching algorithm was very unstable because of the bias by text lengths. Therefore, the task of this research is to improve the table based approach into the more stable version.

III. Normalized Table Matching Algorithm

This section describes a table based matching approach to text categorization. Figure 2 illustrates conceptually the architecture of the proposed text categorization system. The part, 'Encoding' encodes a document into a table as the interface of the system, and will be described in detail in Section III-A. In Section III-B, we will describe the process of computing a similarity between two tables; the computation is used for classification of unseen documents. In Section III-C, we will describe the process of learning sample labeled documents and classifying unseen documents using the proposed approach.

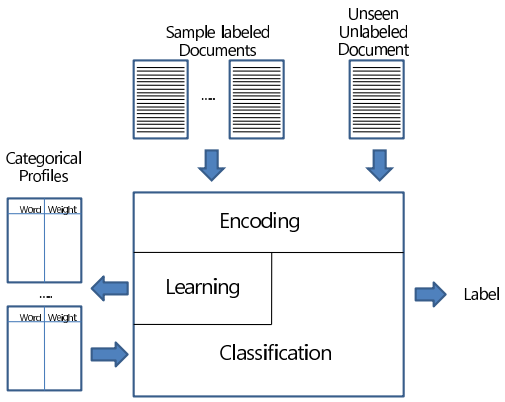

Figure 2. The Process of Indexing Corpus

A. Document Encoding

This section is concerned with the part, 'Encoding' of the architecture of the text categorization system which is illustrated in Figure 2. Here, *document encoding* is defined as the process of mapping a document into a table. Figure 3 illustrates the process of encoding a document so through the five steps. As illustrated in Figure 3, a particular document is given as the input and its corresponding table is generated as the output. In this section, we will describe in detail each of the five steps involved in the document encoding.

The first step of document encoding is tokenization as shown in Figure3. A full text in a document which is wirtten

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-217-2

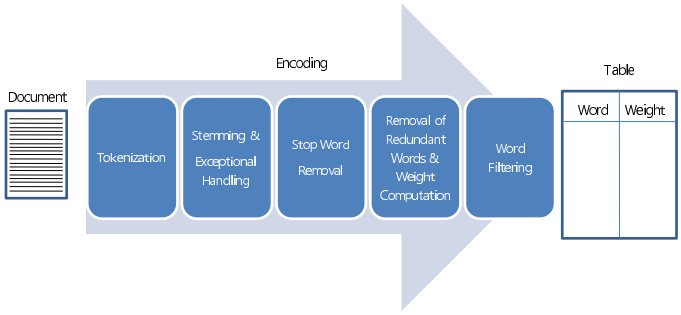

Figure 3. The Process of Encoding a Document into a Table

in a natural language is given as the input of this step. This step, 'tokenization', segments a full text into tokens by white space or punctuation mark. The step generates a list of tokens as the output. A token refers to a word in its raw form.

The second step of document encoding refers to stemming & exception handling. The list of tokens which is generated from the previous step is given as the input of this step. This step converts each token into its root form by stemming it or applying an exception rule to it. This step is carried out by loading stemming & exception rules each of which specifies conversion of each word into its root. Therefore, a list of words in their root forms is generated as the output of this step.

The third step of document encoding is to remove stop words from the list of words. A stop word refers to a grammatical word which do only grammatical functions, irrelevantly to the content of the original document. In English, conjunctions, pronouns, prepositions, and so on belong to this kind of words. Removing the kind of words is necessary for processing documents more efficiently in context of text mining and information retrieval. This step usually remains verbs or nouns as its output.

The fourth step is to remove redundant words and compute weights of each of remaining words. A list of words in their root forms except stop words is given as the input of this step and redundant words are removed among them. The weight of each word indicates how much important it is in terms of the relevancy to the content of the given document. The weight is computed using 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 the weight of word, w_k, relevantly to the content of document, i; $tf_i(w_k)$ indicates the frequency of the word, w_k, in the document, i; D means the total number of documents in the referenced corpus; and $df(w_k)$ indicates the number of documents of the corpus including the word, w_k. A particular corpus is required for computing weights of words using equation (1), and a list of pairs of a word and its weight is generated as the output of this step.

Although stop words and redundant words are removed, we need to filter out additionally words with lower weights for more efficient processing. The previous version which Jo and Cho proposed in 2008 [10], omitted the word filtering, so it took very much time for processing documents for tasks of text categorization. Especially when computing a similarity between two tables, its complexity is quadratic $O(n^2)$, so we need to cut down the size of tables as much as possible, minimizing information loss. We can consider two kinds of schemes for filtering words with their lower weights. One is called rank filtering, where a fixed number of words with their higher weights is selected, and the other is called threshold filtering where weights of words are normalized as continuous values between zero and one, and words with their weights higher than the threshold are selected. In this research, the former is adopted.

B. Similarity between two Tables

This section is concerned with the computation of a similarity between two tables. Two tables each of which represents a document or a group of documents are given as the input. A table which consists of words shared by the two tables is derived from the two tables. A similarity between the two tables is computed based on the shared words in the derived table. Whether weights of words are given as normalized or unnormalized values, the similarity is always generated as a normalized value.

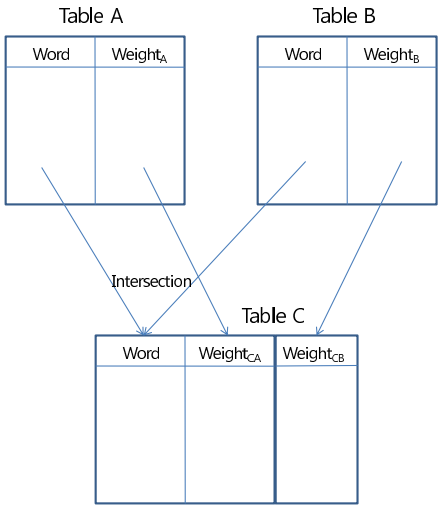

Figure 4. The Process of Deriving a Table from the Two Input Tables

The process of deriving a table from the two input tables is illustrated in Figure 4. Let table A and table B in Figure 4 be the source tables. Let table C be the destination table which is derived by extracting shared words from the source tables. Table C consists of words shared by both source tables. Each

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-217-2

entry of the destination table consists of a shared word and its two weights: one is from table A and the other is from table B.

A similarity between two tables is computed using equation (2)

$$similarity = \frac{weight_{CA} + weight_{CB}}{weight_A + weight_B} \quad (2)$$

where $weight_{CA}$ and $weight_{CB}$ indicate sums of weights of common words from table A and B, respectively, and $weight_A$ and $weight_B$ indicate sums of weights of total words in table A and B, respectively. A similarity computed by equation (2) is bound from 0 to one as a normalized value. If there is no shared word between the two tables, the similarity becomes zero. If the two source tables are exactly same as each other, the similarity becomes one. Therefore, even if weights of words are given as non-normalized values, it is guaranteed that the similarity is given as a normalized value.

We demonstrate the computation of the similarity through a simple example. Two source tables are given in Table I. The destination table is derived from the two source tables as illustrated in Table II. The similarity between the two source tables is computed based on the destination table using equation (2) as follows:

$$\frac{1.5}{1.2 + 1.7}$$

Therefore, the similarity in this example becomes 0.51.

Table I
TWO SOURCE TABLES: TABLE A (LEFT) AND TABLE B (RIGHT)

Table A		Table B	
computer	0.3	computer	0.6
system	0.2	system	0.4
hardware	0.5	information	0.5
CPU	0.2	data	0.2

Table II
DESTINATION TABLE: TABLE C

computer	0.3	0.6
system	0.2	0.4

C. Learning & Classification

This section is concerned with the process of learning sample labeled documents and classifying an unseen document. There exist two functions in the text categorization system: learning and classification. Learning refers to the process of building rules or equations of classification using sample labeled documents in context of text categorization. Classification refers to the process of classifying an unseen document based on the defined rules or equations. Note that learning is prerequisite for classification.

In the view of the proposed text categorization system, learning is defined as the process of building tables corresponding to categories using sample labeled documents. Categories are predefined, and sample documents are allocated to their corresponding categories. Figure 5 illustrates the part, 'Learning', in the proposed text categorization system which is illustrated in Figure 2. From a collection of documents labeled identically, as the learning process, a table is built and called categorical profile in this paper; learning is carried out by attaching the concatenation which concatenates full texts of documents into a full text, to the process of encoding which is illustrated in Figure 3. Therefore, learning generates categorical profiles as many as categories as its output as shown in Figure 5.

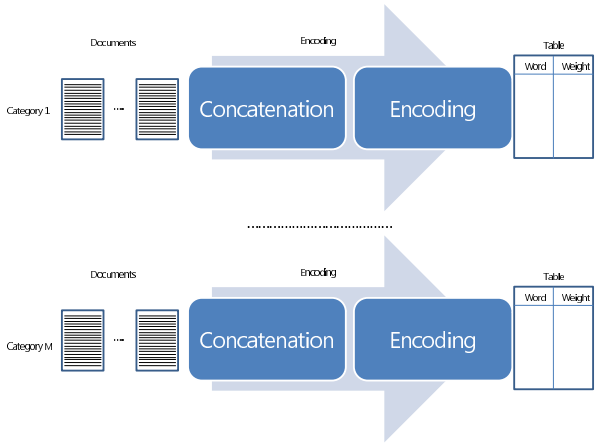

Figure 5. The Process of Learning Sample Labeled Documents in the Proposed Text Categorization System

Classification is defined as the process of deciding one of the predefine categories to each unseen document. The process of classifying an unseen document is illustrated in Figure 6. An unseen document is encoded into a table by the process illustrated in Figure 3, as shown the left part of Figure 6. As shown in the middle part of Figure 6, similarities of the table with categorical profiles given as tables are computed; the computation was already described in Section III. Therefore, the unseen document is classified into the category corresponding to the maximum similarity between its table and the corresponding categorical profile.

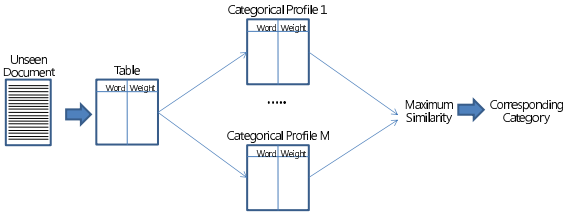

Figure 6. The Process of Classifying a particular Unseen Document

IV. EXPERIMENTS AND RESULTS

This section is concerned with the empirical results of the first set of experiments. The test data used in this set of experiments is a collection of news articles called 'NewsPage.com'. The texts in the collection encoded into numerical vectors for using the machine learning based approaches, and tables for using the proposed one. We selected the four categories and decompose the tasks of categorizing news articles into the four binary tasks. In this section, we describe the test data and the experimental process, present the empirical results, and discuss on them.

In this set of experiments, we use the collection of news articles 'called NewsPage.com'. The collection was built by copying and pasting news articles available in the web site, "newspage.com", manually into plain text files. The collection is partitioned into the training set and test set as shown in table 1. The four categories are selected among the five, and the task is decomposed into the four binary classifications as many as the selected categories.

The configurations of the approaches participating in the experiments are presented in Table III. In using the KNN, the number of nearest neighbors is set three. In using the SVM, the kernel function, the capacity, and the maximum iteration are set the inner product, 4.0, and 1,000, respectively. In using the MLP, the learning rate, the number of hidden nodes, the iterations, are set 0.1, 10, and 1000, respectively. Texts are encoded into 100 dimensional numerical vectors and ten sized table for using the three machine learning algorithms and the proposed approach, respectively.

Table III
THE CONFIGURATIONS OF THE APPROACHES PARTICIPATING IN EXPERIMENTS

Approaches	Configurations	Document Encoding
Naive Bayes	N/A	100 dimensional numerical vectors
KNN	$K = 1, 3$	
NNBP	#Hidden Nodes=10 #Epochs=500 Learning Rate = 0.1 Sigmoid Function	
SVM	Capacity=4 Inner Product	
Proposed Approach	10 sized Tables	

The results from this set of experiments are illustrated in Figure 7. The y-axis indicates the value of F1 measure and the white bar indicates the result of the proposed approach. As shown in Figure 7, the proposed approach is better outstandingly in the category, 'business', than the three approaches. The approaches involved in this set of experiments are comparable to each other in the rest categories. When considering the simplicity and input size, the proposed is more recommendable than the others, even if its performance is comparable to those of the others.

Table IV presents the average F1 measures and the variances of the approaches over the four categories. The

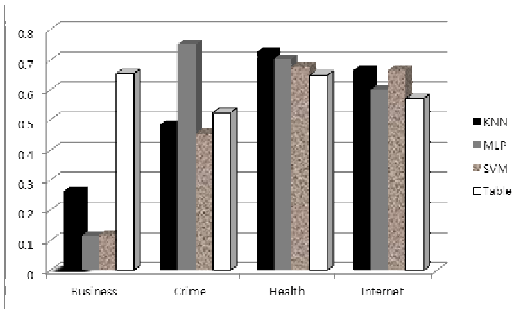

Figure 7. The Results from the Set of Experiments in NewsPage.com

variance of F1 measures indicates the stability of the approaches to the categories; the smaller it is, the more stable. The proposed approaches have largest averaged F1 measure since it is outstandingly better in the category, 'business', as shown in Figure 7. It has smallest variance of its F1 measures; it indicates that it has the best stability in addition. Therefore, from this set of experiments, we conclude that the proposed approach works better and more stable than the three approaches based on Table 2.

Table IV
THE OVERALL PERFORMANCE AND STABILITY OF APPROACHES IN NEWSPAGE.COM

	KNN	MLP	SVM	Table Matching
F1 Average	0.5349	0.5426	0.4796	0.5998
F1 Variance	0.0327	0.0636	0.0511	0.0030

V. CONCLUSION

Let us consider the significances of this research. Like the previous version of the table based algorithm, we are free from the three main problems in encoding texts into numerical vectors: huge dimensionality, sparse distribution, and poor transparency. Because the tables representing texts are symbolic, we trace the classification more easily, in order to provide the evidences. In the proposed version, the categorical scores of the given text are independent of its length. The table based approach is improved to reach more stable performance as shown in the set of experiments presented in Section V.

In order to reinforce the current research, we may consider the four directions of further research. In the first direction, we need to validate the categorization performance of the proposed approach in multiple labels categorization as well as single label one. In the second direction, we may consider that a document or documents are encoded into a committee of tables rather than a table by using multiple schemes for weighting words. In the third direction, in order to keep efficiency and reliability of the proposed approach, we may build the text categorization system in evolutionary fashion by incrementing tables gradually. In the last direction, we

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-217-2

may implement a text categorization system as a prototype program where the proposed approach is adopted.

REFERENCES

[1] B. Massand, G. Linoff, and D. Waltz, “Classifying News Stories using Memory based Reasoning”, pp. 59-65, The Proceedings of 15th ACM International Conference on Research and Development in Information Retrieval, 1992.

[2] Y. Yang, ”An evaluation of statistical approaches to text categorization”, pp. 67-88, Information Retrieval, vol 1, no 1-2, 1999.

[3] D. Mladenic and M. Grobelink, “Feature Selection for unbalanced class distribution and Naive Bayes”, pp. 256-267, The Proceedings of International Conference on Machine Learning, 1999.

[4] S. Eyheramendy and D. Lewis and D. Madigan, “On the Naive Bayes Model for Text Categorization ”,pp. 165-171, The Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 2003.

[5] H. Drucker, D. Wu, and V. N. Vapnik, “Support Vector Machines for Spam Categorization”, pp. 1048-1054, IEEE Transaction on Neural Networks, vol 10, no 5, 1999.

[6] N. Cristianini and J. Shawe-Taylor, “Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.

[7] T. Jo, “NeuroTextCategorizer: A New Model of Neural Network for Text Categorization”, pp. 280-285, The Proceedings of ICONIP 2000, 2000.

[8] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text Classification with String Kernels”, pp. 419-444, Journal of Machine Learning Research, vol 2, no 2, 2002.

[9] K. Lee and K. Kageura, “Virtual relevant documents in text categorization with support vector machines”, pp. 902-913, Information Processing and Management, vol 43, no 4, 2007.

[10] Taeho Jo and Dongho Cho, ”Index Based Approach for Text Categorization”, pp. 127-132, International Journal of Mathematics and Computers in Simulation, vol 2, no 1, 2008.

[11] Taeho Jo, ”Table based Matching Algorithm for Soft Categorization of News Articles in Reuter 21578”, pp. 875-882, Journal of Korea Multimedia Society, vol 11, no 6, 2008.

[12] Taeho Jo, ”Single Pass Algorithm for Text Clustering by Encoding Documents into Tables”, pp. 1749-1757, Journal of Korea Multimedia Society, vol 11, no 12, 2008.

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-217-2