IJMI

A NOVEL APPROACH FOR TEXT CLASSIFICATION BASED ON LZW COMPRESSION TECHNIQUE

GURU D.S., BHARATH BHUSHAN S.N., MANJUNATH S., HARISH B.S.

Department of Studies in Computer Science, Manasagangotri, University of Mysore, Mysore - 570 006. India,
bsharish@ymail.com
*Corresponding Author: Email- dsg@compsci.uni-mysore.ac.in, sn.bharath@gmail.com, manju_uom@yahoo.co.in,

Abstract-Internet is a pool of information, which contains billions of text documents which are stored in compressed format. In literature we can find many text classification algorithms which work on uncompressed text documents. In this paper, we propose a novel representation scheme for a given text document using compression technique. Further, proposed representation scheme is used to develop a methodology to classify the text documents. For the purpose of representation, we have used LZW compression technique and the dictionary representation obtained by LZW technique is used as representative for the text document. For classification we have used nearest neighbor method. Extensive experimentation is carried out on seven datasets, out of which three are our own datasets and remaining four are publically available datasets resulting with approximately 80% of F-measure.

Key words: Text classification, Text compression, LZW compression technique.

Introduction

Internet is the rapidly growing information gallery that contains rich textual information. This rapid growth makes it difficult for the users to locate relevant information quickly on the web. Document retrieval, categorization, routing and filtering systems are often based on text classification. Text classification problem can be stated as follows: given a set of labeled examples belonging to two or more classes, we classify a new test document to a class with the highest similarity. Text classification presents many challenges and difficulties. Firstly, it is difficult to capture high-level semantics and abstract concepts of natural languages just from a few key words and the same word can represent different meanings. Secondly, it is difficult to handle high dimensionality and variable lengths of text documents.

Text Documents are the most common type of information store house especially with the increased use of the internet. Internet web pages, e-mails, e-news feeds newsgroup messages have millions or billions of text documents. The web pages that are available in the internet are stored in the compressed format. Data mining activities such as document classification and clustering are carried out these data by decompressing the data and taking it back to the standard format. These processes of decompressing and performing mining activities consume more computational time. However to the best of our knowledge, nowhere in the literature we can find any works on classification of text documents in text compressed format. This motivated us to take up this work for design of text classification using text compression representation as a new representation method.

The rest of the paper is organized as follows. In section 2 a brief literature survey on the text classification is presented. In section 3 we present the model based on LZW compression technique. An illustrative example illustrating the proposed model is given in section 4. In section 5 we discuss about experimentation details and comparative analysis. In section 6 we present conclusion along with future work.

Related Work

In automatic text classification, it has been proved that the term is the best unit for text representation and classification [1]. Though a text document expresses vast range of information, unfortunately, it lacks the imposed structure of traditional database. Therefore, unstructured data, particularly free running text data has to be transformed into a structured data. To do this, many pre-processing techniques are proposed in literature [2, 3]. After converting an unstructured data into a structured data, we need to have an effective document representation model to build an efficient classification system. Bag of Word (BoW) is one of the basic methods of representing a document. The BoW is used to form a vector representing a document using the frequency count of each term in the document. This method of document representation is called as a Vector Space Model (VSM) [4]. The major limitation of VSM is that the correlation and context of each term is lost which is very important in understanding a document. Li and Jain [5] used binary representation for given

document. The major drawback of this model is that it results in a huge sparse matrix, which raises a problem of high dimensionality. Another approach [6] uses multi-word terms as vector components to represent a document. But this method requires a sophisticated automatic term extraction algorithms to extract the terms automatically from a document. Wei et al., (2008) proposed an approach called Latent Semantic Indexing (LSI) [7] which preserves the representative features for a document. The LSI preserves the most representative features rather than discriminating features. Thus to overcome this problem, Locality Preserving Indexing (LPI) [8] was proposed for document representation. The LPI discovers the local semantic structure of a document. Unfortunately LPI is not efficient in time and memory [9]. Choudhary and Bhattacharyya (2002) [10] used Universal Networking Language (UNL) to represent a document. The UNL represents the document in the form of a graph with words as nodes and relation between them as links. This method requires the construction of a graph for every document and hence it is unwieldy to use for an application where large numbers of documents are present.

After giving an effective representation for a document, the task of text classification is to classify the documents to the predefined categories. In order to do so, many statistical and computational models have been developed based on Naïve Bayes classifier [11], K-NN classifier [12], Centroid Classifier [13], Decision Trees [14], Rocchio classifier [15], Support Vector Machines [16].

Although many text document representation models are available in literature, frequency-based BoW model gives effective results in text classification task. Indeed, till date the best multi-class, multi-labelled categorization results   for   well   known datasets are based   on BoW representation [17]. Unfortunately, BoW representation scheme has its own limitations. Some of them are: high dimensionality of the representation,   loss   of correlation   with   adjacent   words and   loss   of semantic relationship that exist among the terms in a document [18]. Also the main problem with the frequency based approach is that given a term, with lesser frequency   of occurrence   may be appropriate in describing a document, whereas, a term with the higher frequency may have a less importance. Unfortunately, frequency-based BoW methods do not take this into account [10].

All the mentioned algorithms works on uncompressed documents. Whereas the challenging and required is to classify documents at compression level.

In literature we can find many compression techniques which are used for the effective representation of data in compressed format. In this paper we consider only the lossless compression schemes. Run Length Encoding (RLE) [19] is a simple and popular data compression algorithm. It is based on the idea to replace a long sequence of the same symbol by a shorter sequence. Huffman compression [20] it is a statistical lossless compression method that converts characters into

variable length bit strings. Huffman compression technique works on frequency of individual symbol. The Huffman algorithm is a so-called "greedy" approach to solving this problem in the sense that at each step, the algorithm chooses the best available option. Lempel–Ziv–Welch (LZW) is a universal lossless data compression algorithm created by Abraham Lempel, Jacob Ziv, and Terry Welch. The LZW compression algorithm organized around a translation table or string table, that maps input characters into the fixed length codes [21].   Among different compression techniques, we have used LZW compression technique. LZW compression is used as the foremost technique, mainly because of its versatility and simplicity. Typically, the LZW compression can compress executable code, text, and similar data files to almost one-half of their original size. It usually uses single codes to replace strings of characters, thereby compressing the data. LZW also gives a good performance when extremely redundant data files are presented to it like computer source code, tabulated numbers and acquired signals. The common compression ratio for these cases is almost in the range of 5:1.Though RLE and Huffman compression techniques are also very simple; they are not suitable for text documents and also these two methods does not provide good compression ratio like LZW method.

Proposed Method
In this paper we are presenting a novel method used for classification of compressed text documents. Normally text documents are available in several formats such as html, xhtml, pdf, plain text etc. The first step is to pre-process the text document, hence to bring them to a common format before processing the text. In the literature we have stop word elimination, stemming, pruning etc as pre-processing steps.[27]. In this work we have used only stop word elimination technique. Once the pre-processing is done on training data, the text documents are compressed using LZW compression scheme and a compressed training document library is created.   The working principle of LZW compression technique is given as follows.

LZW is a universal lossless compression algorithm which is organized around string table. String table contains strings that have been encountered previously in the text being compressed. It consists of a running sample of strings in the text, so the available strings reflect the statistics of the text. It uses greedy parsing algorithm, where the input string is examined character-serially on one pass, and the longest recognized input string is parsed off each time. A recognized string is one that exists in the string table. Each such added string is assigned a uniquely identified by code value.

The proposed model is of two stages, in which stage one is of creation of database in which all pre-processed text data are compressed and stored in the database, stage two is classification stage in which given unknown sample is classified into its corresponding class label using compression technique.

Algorithm: LZW text compression.

Input: Pool of text data
Output: Pool of compressed text data, String table.
Method:

1. Initialize table to contain single character strings.
2. Prefix string $\omega \leftarrow$ Read first input character.
3. K$\leftarrow$ Read next input character
   If no such K (input exhausted) : code (ω) – output; EXIT
4. If ωK exists in string table : ωK- ω; repeat 3;
5. else ωK not in string table : code (ω) – output;
6. ωK – string table;
7. K – ω; repeat Step.

Algorithm end.

At each execution of the basic step an acceptable input string ω has been parsed off. The next character K is read and the extended string ωK is tested to see if it exists in the string table. For each training document we obtain a string table which is referred as dictionary representation and stored in the library. Further, given a test document we obtain dictionary representation and during classification we use string matching based on nearest neighbor technique. We classify the test document into first nearest neighbor class label. The block diagram of the proposed model is as shown in fig 1.

Illustrative Example:
For illustrative purpose, we consider three classes of documents as shown in Fig 2(a). For each class we have created a knowledge base as follows. Given a set of training documents of an individual class, stop words from each training documents are eliminated and the terms are pooled to form a knowledge base as shown in Fig 2(b). Then LZW compression technique is applied, compressed dictionary will be created. Documents after compression look like as shown in Fig 2(c).

Similarly, document data is also processed and compressed as shown in fig 2(d), 2(e) and 2(f) respectively. Similarity is calculated between training and testing dictionaries and the given test document will be assigned to a class label with highest similarity value. Table 4 shows the similarity value between test document and document present in the database. From table 4 it is clear that given test document belongs to class 2 as the similarity value between test document and D2 of class 2 has highest similarity value.

Experimentation
In this section, we present the details of the experiments conducted to represent the effectiveness of the proposed method on seven datasets. We have created three datasets of our own and four publically available datasets to evaluate the performance of the proposed model. First dataset consists of three classes and each class consists of five documents. Second dataset consists of five classes and each class consists of ten documents. Third dataset consist of 1000 documents from 10 different classes.

Fourth dataset is Google news group dataset which contains one thousand documents from ten different classes and fifth dataset is research article dataset. The research article abstracts are downloaded from scientific web portals like ieeexplore.org, portal.acm.org and sciencedirect.com [27] Sixth dataset is vehicles Wikipedia [22] used to evaluate a prototype system used for the evaluation of classification performance.

Seventh dataset is 20mini newsgroup dataset. The 20 mini newsgroups dataset is a publically available dataset consisting collection of approximately 2,000 newsgroup documents, partitioned evenly across 20 different classes. The first three datasets consists of documents which do not have overlap compared to other publically available datasets. This is considered to study the performance of the proposed model in case of less overlap and large overlap.
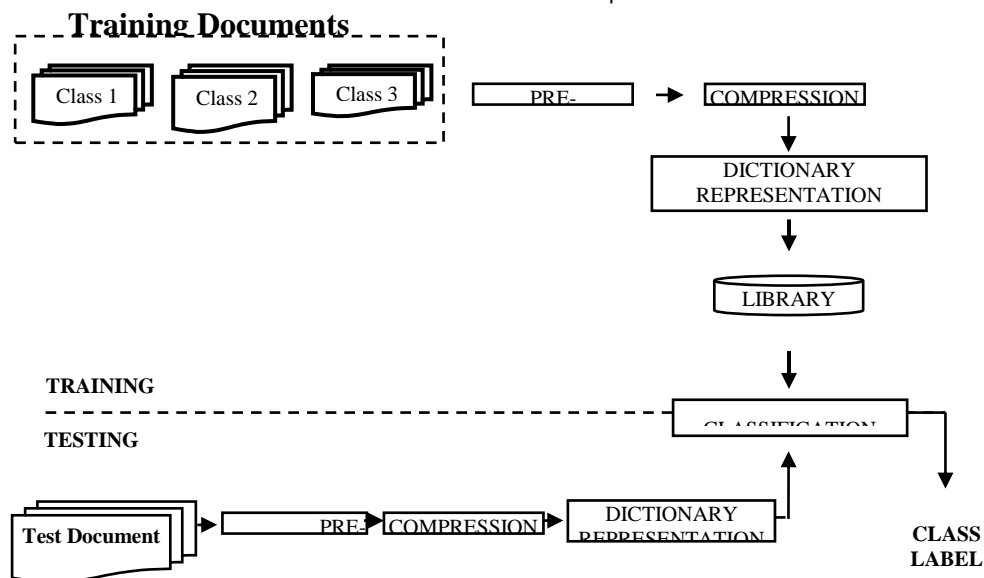

Figure 1: Block Diagram of the proposed model.

We have conducted two sets of experiments; where each set contain three different trails. In first set of experiments, we have used 40% of the database for training and remaining 60 % is used for testing. In second set of experiments, we have used 60 % training and 40 % for testing . Each set of experiments contain three different trials. In each trail we have selected training and testing document randomly. For the purpose of evaluation of results, we have calculated precision, recall and f-measure for each trail. The details of the experiments are shown in the following table 1.

In table 2, we have listed, Max, Mean and standard deviation of the results of each data set and it can be seen that mean is comparatively high and standard deviation is very less. This indicates that the proposed method works well even in case of different set of training and testing sets. Also, Tab 2 indicates that, the proposed model performs equally well in case of both large overlap and less overlap cases.

The quantitative evaluation of the proposed method is carried out with existing different methods of text classifiers. The proposed method with different type classification techniques are analyzed in qualitative comparative analysis is also presented. Tabble 3 reveals that, all the existing works in the literature are done on the uncompressed text documents. But the proposed model classifies the documents in compressed format also.

6. Conclusion
We have proposed a novel method to classify text documents. The proposed method uses LZW compression scheme. Using string matching and nearest neighbor method we have proposed text classification technique. To check the efficiency and the robustness of the proposed models, an extensive experiment is carried out on all the seven dataset. The performance evaluation of the proposed method is carried out by performance measures such as precision, recall and f-measure. Even though, the results are not better than other uncompressed based techniques, they are comparatively equal to them, i.e., approximately 80% of classification accuracy. In this paper we have put forward a new representation model for text classification using compression technique, which is first of its kind. Further, we explore novel proximity measures for comparing text in compressed format which may improve the classification accuracy.

References
[1] Rigutini L. (2004) *Ph.D. Thesis*, University of Siena.
[2] Porter M.F. (1980) 130—137.
[3] Hotho A., Nürnberger A., Paaß G. (2005) *Journal for Computational Linguistics and Language Technology*. 19—62.
[4] Salton G., Wang A., Yang C. S. (1975) *A Communications of the ACM*, 613—620.
[5] Li Y.H., Jain A.K. (1998) *The Computer Journal*. 537--546 .
[6] Milios E., Zhang Y., He B., Dong L. (2003) *Sixth Conference of the Pacific Association for Computational Linguistics*. 275—284.
[7] Wei C.P., Yang C.C., Lin C.M. (2008) *Journal of Decision Support System*. 606—620
[8] He X., Cai D., Liu H., Ma W.Y. (2004) *In: SIGIR*, 96—103
[9] Cai D., He X., Zhang W.V., Han J. (2007) *In: ACM International Conference on Information and Knowledge Management* 741—750.
[10] Choudhary B., Bhattacharyya P. (2002) *Eleventh International World Wide Web Conference.*
[11] McCallum A., Nigam K. (2003) *Journal of Machine Learning Research*, 1265—1287.
[12] Tan S. (2006) *Journal of Expert Systems with Applications*, 290—298.
[13] Tan S. (2008) *Journal of Expert Systems with Applications*, 279—285.
[14] Wang L. M., Li X. L., Cao C. H., Yuan S. M. (2006) 511—515.
[15] Lewis D. D., Schapire R. E., Callan J. P., Papka R. (1997) *Nineteenth International conference on Research and Development in Information Retrieval*, 289—297.
[16] Mitra V., Wang C.J., Banerjee S. (2007) *Journal of Applied Soft Computing*. 908—914.
[17] Bekkerman R., Allan J. (2004) *CIIR Technical Report*, 408.
[18] Bernotas M., Karklius K., Laurutis R., Slotkiene A. (2007) *Journal of Information Technology and Control*. 217—220.
[19] Franaszek P. A. (1972), *U.S. Patent 3,689,899.*
[20] Huffman D. A. (1952) *Proceedings of the I.R.E.*, 1098–1102.
[21] Terry A. Welch, (1984) *IEEE Computer*, 17(6):8-19.
[22] Isa D., Lee L.H., Kallimani V. P. and Rajkumar R. (2008) *IEEE Transactions on Knowledge and Data Engineering*, 23 – 31
[23] Harish B.S., Guru D. S. and Manjunath S. (2010) *IJCA Special Issue RTIPPR.*
[24] Xue X. B. and Zhou Z. H. (2009) *IEEE Transactions on Knowledge and Data Engineering*, 428 – 442.
[25] Guru D.S., Harish B.S. and Manjunath S.(2010) *Proceedings of the International ACM Conference*, Compute, Bangalore.
[26] Dinesh R., Harish B. S., Guru D.S. and Manjunath S. (2009) *In the Proceedings of Indian International Conference on Artificial Intelligence*, Tumkur, India, 2071 – 2079.
[27] Harish B.S. (2011)*Ph.D.Thesis*, University of Mysore.

Table 1 : Classification result table on different dataset using proposed model

Dataset	Trails	40 : 60			Avg F-Measure	60:40			Avg F-Measure
		Precision	Recall	F Measure		Precision	Recall	F Measure	
DATASET 1 (3 CLASS)	1	0.8055	0.7777	0.775	0.768	0.8888	0.8333	0.8222	0.767
	2	0.8333	0.7777	0.74		0.8888	0.8333	0.8222	
	3	0.8333	0.7777	0.79		0.7222	0.6666	0.6555	
DATASET 2 (5 CLASS)	1	0.9	0.8666	0.863	0.8	0.9333	0.9	0.8933	0.822
	2	0.8333	0.8	0.796		0.8666	0.8	0.7866	
	3	0.7833	0.7333	0.74		0.8666	0.8	0.7866	
DATASET 3 (10 CLASS)	1	0.7743	0.7983	0.7854	0.774	0.7953	0.7976	0.7959	0.786
	2	0.7882	0.7717	0.7773		0.7762	0.7925	0.7837	
	3	0.7576	0.7633	0.7598		0.7667	0.7925	0.7777	
DATASET 4 Google Newsgroup	1	0.7876	0.7767	0.7819	0.783	0.7901	0.7775	0.7831	0.782
	2	0.7717	0.7333	0.7801		0.7714	0.79	0.78	
	3	0.7929	0.7929	0.7866		0.7941	0.775	0.7839	
DATASET 5 Research Article	1	0.7983	0.8	0.7984	0.798	0.7803	0.8	0.7894	0.784
	2	0.8089	0.7883	0.7975		0.7714	0.79	0.78	
	3	0.7889	0.81	0.7978		0.7763	0.7925	0.7834	
DATASET 6 Vehicles Wikipedia	1	0.7754	0.7955	0.7828	0.784	0.7944	0.7898	0.7888	0.788
	2	0.787	0.7689	0.775		0.7762	0.7841	0.7799	
	3	0.7994	0.7841	0.7928		0.8044	0.7898	0.7954	
DATASET 7 20mini news group	1	0.7876	0.7758	0.7797	0.776	0.7985	0.77	0.7829	0.776
	2	0.7526	0.7816	0.7473		0.7863	0.7725	0.7787	
	3	0.8124	0.7925	0.8		0.7722	0.76	0.7649	

Table 2 : Max, Mean and Standard deviation of F-Measure

	40 : 60			60 : 40		
	Max	Mean	Standard deviation	Max	Mean	Standard deviation
D1	0.79	0.775	0.025658	0.8222	0.8222	0.096244
D2	0.863	0.796	0.061582	0.8933	0.7866	0.061603
D3	0.7854	0.7773	0.013084	0.7959	0.7837	0.009274
D4	0.7866	0.7819	0.003356	0.7839	0.7831	0.00206
D5	0.7984	0.7978	0.000458	0.7894	0.7834	0.00476
D6	0.7928	0.7828	0.008923	0.7954	0.7888	0.007778
D7	0.8	0.7797	0.026581	0.7829	0.7787	0.009417

Table 3 : **Quantitative Evaluation Table**

Authors	Dataset	Copressed /Uncompressed	Representation Scheme	Classifiers Used	Min F- Measure	Max F – Measure
Xue and Zhou, 2009	Reuters 21578	Uncompressed	Distributional Words	K-NN	0.4950	0.8440
	20Newsgroup	Uncompressed	Distributional Words	AVM	0.8860	0.8870
Guru et al 2010	Vehicles Wikipedia	Uncompressed	Symbolic Representation	SVM	0.8850	0.9050
	20 Mini Newsgroup	Uncompressed			0.8650	0.8950
	Google Newsgroup	Uncompressed			0.8750	0.8900
	Research Article Abstracts	Uncompressed			0.8620	0.9070
Harish et al 2010	Google Newsgroup	Uncompressed	Symbolic Representation		0.8440	0.9220
	Research Article Abstracts	Uncompressed	Symbolic Representation		0.8460	0.9750
Dinesh et al 2009	Vehicles Wikipedia	Uncompressed	Status Matrix		0.8850	0.9050
	20 Mini Newsgroup	Uncompressed	Status Matrix		0.8650	0.8950
	Google Newsgroup	Uncompressed	Status Matrix		0.8800	0.9200
	Research article Abstracts	Uncompressed	Status Matrix		0.9180	0.9260
Proposed Method	Dataset 1	**Compressed**	Dictionary Representation	NN Classifier	0.6977	0.8061
	Dataset 2			NN Classifier	0.7400	0.8933
	Dataset 3			NN Classifier	0.7598	0.7959
	Dataset 4			NN Classifier	0.7800	0.7866
	Dataset 5			NN Classifier	0.7975	0.7834
	Dataset 6			NN Classifier	0.7750	0.7829
	Dataset 7			NN Classifier	0.7473	0.8000

C 1: **D1 :** Business is an art of making money and managing money.
D2 : Operating a business in the health care area carries special challenges in today's marketplace managing money.

C 2: **D1 :** Heart is a muscular tissue. The word heart is derived by Greek word cardiac.
D2 : Heart beats 72 times per minute.

C 3: **D1:** Cricket is a ball and bat sport played between two teams.
D2: Each team consists of eleven players and it is played on a pitch.

Fig: 2 (a) : Unstructured data.

C 1: **D1 :** Business _____ managing   money.
D2 : Operating __ business ____ health _____marketplace _____.

C 2: **D1 :** Heart _____ muscular tissue. _____ cardiac.
D2 : Heart _____.

C 3: **D1:** Cricket _____ ball _____ bat sport _____ teams.
D2: _____ team _____ players _____pitch.

CLASS 1											
D 1 'Bu'	'us'	'si'	'in'	'ne'	'es'	'ss'	'sM'	'Ma'	'an'	'na'	'ag'
'gi"ing'	'gM'	'Mo'	'on'	'ney'							
D 2 'Op'	'pe'	'er'	'ra'	'at'	'ti'	'in'	'ng'	'gB'	'Bu'	'us'	'si"ine'
'es'	'ss'	'sH'	'He'	'ea'	'al'	'lt'	'th'	'hM'	'Ma'	'ar"rk'	'ke'
'et'	'tp'	'pl'	'la'	'ac'	'ce'						

CLASS 2										
D 1 'He'	'ea'	'ar'	'rt'	'tM'	'Mu'	'us'	'sc'	'cu'	'ul'	'la'
'arT' 'Ti'	'is'	'ss'	'su'	'ue'	'eC'	'Ca'	'ard'	'di'	'iac'	
D 2 'He'	'ea'	'ar'	'rt'							

CLASS 3											
D 1 'Cr'	'ri'	'ic'	'ck'	'ke'	'et'	'tB'	'Ba'	'al'	'll'	'lB' 'Bat' 'tS'	
'Sp'	'po'	'or"rt'	'tT'	'Te'	'ea'	'am'	'ms'				
D 2 'Te'	'ea'	'am'	'mP'	'Pl'	'la'	'ay'	'ye'	'er'	'rs'	'sP'	'Pi'
'it'	'tc'	'ch'									

TEST SAMPLE: Heart has four chambers Fig: 2 (d) : Test Data.

Heart _____ _____ chambers **Fig: 2 (e) :** Test Data after pre-processing.

'He' 'ea' 'ar' 'rt' 'tC' 'Ch' 'ha' 'am' 'mb' 'be' 'er' **Fig: 2 (f):** Test sample after compression using LZW algorithm.

Fig 2 : An illustrative example

Table 4 : Similarity matrix between test and train documents					
	Class 1		**Class 2**		**Class 3**
	D1	D2	D1	D2	D1	D2
Test Document	0	12	17	28	14	18
Highest similarity is for class 2. Hence test document belongs to class 2.					