Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 6, Issue. 7, July 2017, pg.189 – 193

LZW Compressed Text Classification using Nearest Neighbor Classifier

Ronnie Merin George

Assistant Professor, Department of Computer Science & Engineering

Sahyadri College of Engineering & Management, Mangaluru - 575 007

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.

Keywords: 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.

2. 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, K-NN classifier, Centroid Classifier, Decision Trees, Rocchio classifier, Support Vector Machines[11-13].

3. 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 ω ← Read first input character.
3. K← 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.

4. 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. Sixth dataset is vehicles Wikipedia 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.

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.

	40 : 60			**60:40**		
Dataset	**Precision**	**Recall**	**F Measure**	**Precision**	**Recall**	**F Measure**
DATASET 1	0.8055	0.7777	0.775	0.8888	0.8333	0.8222
DATASET 2	0.9	0.8666	0.863	0.9333	0.9	0.8933
DATASET 3	0.7743	0.7983	0.7854	0.7953	0.7976	0.7959
DATASET 4	0.7876	0.7767	0.7819	0.7901	0.7775	0.7831
DATASET 5	0.7983	0.8	0.7984	0.7803	0.8	0.7894
DATASET 6	0.7754	0.7955	0.7828	0.7944	0.7898	0.7888
DATASET 7	0.7876	0.7758	0.7797	0.7985	0.77	0.7829

5. 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.: Ph.D. Thesis, University of Siena(2004)
2. Porter, M.F. (1980), 130—137
3. Hotho, A., Nürnberger, A., Paaß, G. (2005) Journal for forComputational 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. S.N. Bharath Bhushan and Ajit Danti. Classification of text documents based on score level fusion approach. Pattern Recognition Letters 94 (2017) 118–126.
12. AjitDanti and S. N. Bharath Bhushan. Document Vector Space Representation Model for Automatic Text Classification. In Proceedings of International Conference on Multimedia Processing, Communication and Information Technology (MPCIT), Shimoga. pp. 338-344. 2013.
13. Ajit Danti and S. N. Bharath Bhushan. Classification of Text Documents Using Integer Representation and Regression: An Integrated Approach. Special Issue of The IIOAB Scopus Indexed Journal.Vol. 7, No.2, pp. 45–50. 2015.

