International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 2 (2017), pp. 175-182

http://www.ripublication.com

A Machine Learning Approach For Emotion Classification Using Document Semantics

Lydia DL
Christ University, Bengaluru,
Karnataka, India.

Abstract

In the past few years, a lot of research has been done to figure out the concepts in the documents. These documents would consist of verbs, adverbs, prepositions etc. The current application is not limited by identifying keywords to understand the document concept but aims to gain an exact understanding of concepts through correlation of words and to classify the documents. In our application we use the Latent Semantic Analysis (LSA) algorithm for text emotion. The training dataset is trained using the algorithm and a matrix is generated. This matrix gives us the correlation of words within documents. By finding the similarity of test dataset with the training dataset, the emotion of the test data is classified.

Keywords: Latent Semantic Analysis; Vector Space Model; Text Emotion Classification;

I. INTRODUCTION

The emotion classification within documents is done manually by going through the documents. This manual method is impossible when there is a need to classify numerous documents in a short period of time. Hence we develop an application which would classify the emotions among several documents simultaneously. In this application there is no need to read the documents to classify the category of the document. Our application would be more efficient comparatively.

The application is a text mining application used to classify the emotions in the documents. An input data is extracted and it is divided into train dataset and test dataset. The input data is preprocessed for further processing. The preprocessing includes stop words removal and stemming of the words. Stop words are the words which does not change the meaning of the sentence. For eg; prepositions. Stemming is the process of finding the route word. For eg; explain is the route word of explaining.

The term frequency matrix which is obtained by finding the count of the preprocessed unique words is used as the vector space model (VSM). This particular matrix is used to find the correlation of words. LSA includes two steps, namely Singular Value Decomposition (SVD) and dimensionality reduction. The train dataset is trained with the LSA algorithm and a model is constructed i.e. a singular value matrix with reduced dimensions. The test dataset is classified by obtaining the similarity with the train dataset. The emotion of the closest training dataset will be the emotion of the test dataset.

II. LITERATURE SURVEY

Nowadays, vector space model (VSM) is used to express information in text classification. This model can be applied to any language theoretically which can split into words.

Besides, documents are represented as vectors that the vectors consist of several keywords [1]. The basic idea of traditional vector space model is that text is represented as words elements of vector. Similarity can be generated by calculating cosine value between the two vectors and that also used to text classification [1]. In a document retrieval, or other pattern matching environment where stored entities (documents) are compared with each other or with incoming patterns (search requests), it appears that the best indexing (property) space is one where each entity lies as far away from the others as possible; in these circumstances the value of an indexing system may be expressible as a function of the density of the object space; in particular, retrieval performance may correlate inversely with space density.

An approach based on space density computations is used to choose an optimum indexing vocabulary for a collection of documents [2]. Among various approaches found in the document analysis literature, LSA is one technique that captures the semantic structure of documents based on word co occurrences within them. In spite of being completely independent of any external sources of semantics, it performs quite well. However, any extra information included in LSA influences the model's ability to capture the semantic structure of documents [2]. There are several extensions of LSA that were empirically shown to perform better in classification problems. Relevant prior work is that of Wiemer-Hastings et al [3] in which surface parsing is employed in LSA by replacing pronouns in the text with their antecedents. The model was evaluated as a cognitive model. Serafin et al. [4] suggested that an LSA semantic space can be built from the co occurrence of arbitrary textual features which can be used for dialogue act classification. Kanejiya et al. [5] attempted to capture syntactic context in a shallow manner by enhancing target words with the parts-of-speech of their immediately preceding words. The syntactically enhanced LSA model is used in the context of an intelligent tutoring system. The results reported an increased ability to evaluate more student answers. Rishel et al. [6] achieved a significant improvement in classification accuracy of LSA by using part of speech tags to augment the term by document matrix and then applying SVD. The results of the work showed that the addition of parts of speech tags can decrease word ambiguities significantly. Eugenio et al. [7] used LSA in a text classification application to capture the higher order structure of dialogue contexts by adding richer linguistic features to LSA.

III. DATA PRE PROCESSING

A data set (or dataset, although this spelling as one word is not present in many contemporary dictionaries) is the collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set.

A. Dataset

For the experiments, we use ISEAR (International Survey on Emotion Antecedents and Reactions) dataset. This dataset consists of 7666 textual pieces tagged with the most appropriate of seven major emotions (joy, fear, anger, sadness, disgust, shame, and guilt).

We use only 700 textual pieces in our application. For the experiments, we randomly select 100 for each category of emotions. The training set consists of 560 records with 7 emotions. After constructing latent semantic space, we randomly select 20 for each category of emotions that consists of test dataset.

TABLE I. DATASET

Document attributes	**Values**
Number of documents in our dataset	700
Number of categories	7
Number of documents per category	100
Number of documents in training set	560
Number of documents in test set	140

B. Pre Processing

Data pre-processing is an important step in the data mining process. If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data preparation and filtering steps can take considerable amount of processing time. Data pre-processing includes cleaning, normalization, transformation, feature extraction and selection, etc. The product of data pre-processing is the final training set. In our application the data pre-processing is done in two steps. They are stop words removal and stemming.

Stop words are words which do not contain important significance to be used in Search Queries. Usually these words are filtered out from search queries because they return vast amount of unnecessary information. Hence we remove such words from the dataset for further processing.

Stemming is the process of reducing inflected or sometimes derived words to their word stem, base or root form, generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. We use the porter stemmer algorithm in our application. The Porter stemming algorithm or ‘Porter stemmer’ is a process for removing the commoner morphological and in flexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems.

IV. METHODOLOGY AND IMPLEMENTATION

LSA uses SVD followed by dimensionality reduction to capture all correlations latent within documents by modeling interrelationships among words so that it can semantically cluster words and documents that occur in similar contexts. SVD works by taking the conventional VSM of text representation with term frequencies in the input term by document matrix. Various other weighting measures apart from term-frequency also exist. According to the theorem stated by Baker, the input matrix A_{mn} of order$_{m*n}$ is constructed as a product of three matrices obtained upon its eigen decomposition:

$$A_{mn} = U_{mm}S_{mn}V^{T}_{nn} \qquad (1)$$

where $U^TU = I$, $V^TV = I$; I being an identity matrix, the columns of U and V are orthonormal eigenvectors of AA^T and A^TA respectively, and S is a diagonal matrix containing the square roots of eigen values from U or V, known as singular values, sorted in descending order.

The underlying principle of LSA is that the original matrix is not perfectly reconstructed. Rather, a representation that approximates the original matrix is reconstructed based on reduced number of dimensions of the original component matrices. Mathematically, the original representation of data in matrix Amn is reconstructed as an approximately equal matrix Ak_{mn} from the product of three matrices U_{mk}, S_{kk} and and V_{kn} based on just k dimensions of the component matrices U_{mm}, S_{mn} and V_{nn} of the original matrix A. The diagonal elements of matrix S are non-negative descending values. If S is reduced to a k*k order diagonal matrix S_{kk}, then the first k columns of U and V form matrices U_{mk} and V_{nk} respectively. The reduced model is:

$$Ak_{mn} = U_{mk}S_{kk}V^{T}_{kn} \tag{2}$$

This approximate representation of the original documents after dimensionality reduction reflects all the underlying word correlations. Word correlations that occurred in some context prior to dimensionality reduction now become more or less frequent, and some word correlations that did not appear at all originally may now appear significantly or at least fractionally. This lower-dimensional matrix representation of the linguistic texts is termed as "Semantic structure" or "LSA space" or "Semantic space" in the literature.

The quality of LSA space directly determines the performance of LSA applications. Factors that could affect LSA space quality include the kind and size of corpus, the dimensions, and the term-weighting measures.

Fixing an optimal dimensionality to be retained in LSA is an empirical issue. Retaining larger dimensions reconstructs closer approximations to the original matrix but may span many unessential relationships. On the other hand, retaining smaller dimensions saves much of computation but with a compromise on the essential relationships. Typically, the number of dimensions retained should be large enough to capture the semantic structure in the text, and small enough to omit trivial correlations. The proper way to make such choices is an open issue in the factor analytic literature.

The semantic space obtained after dimensionality reduction through LSA can be used for document classification. In this context, LSA is viewed from a geometrical perspective where words and documents are considered as points in space. The combination of SVD and dimensionality reduction establishes a k-dimensional orthogonal semantic space where the words and documents are distributed according to their common usage patterns. The semantic space reflects those words that have been used in the document to give information about the concepts (the axes) to which the words are closer. Essentially, LSA is a proximity model that spatially groups similar points together. As the dimensional space is reduced, related points draw closer to one another. The relative distances between these points in the reduced vector space show the semantic similarity between documents and is used as a basis for document classification. A test document (a set of words) is mapped as a pseudo-document into the semantic space by the process of "Folding-in". To fold-in an m*1 test document vector d into the LSA space of lower dimensions k, a pseudo-document representation ds based on the span of the existing term vectors (the rows of U_{mk}) is calculated as:

$$ds = d^{T}U_{mk}S^{-1} \tag{3}$$

Then the pseudo-document's closeness with all other documents is measured using any of the standard measures of similarity like Cosine measure, Euclidean distance, etc. The category of the document that is located in its nearest proximity in space is the category of the test document. One of the standard approaches for document classification like k-Nearest-Neighbor (kNN), Decision Trees, Naive Bayes, Support Vector Machines (SVM), etc. is applied for classification purposes.

In contrast to many other methods of text classification, LSA is categorizes semantically related texts as similar even when they do not share a single term. This is because in the reduced semantic space, the closeness of documents is determined by the overall patterns of term usage. So documents are classified as similar regardless of the precise terms that are used to describe them. As a result, terms that did not actually appear in a document may still end up close to it if that is consistent with the major patterns of association in the data.

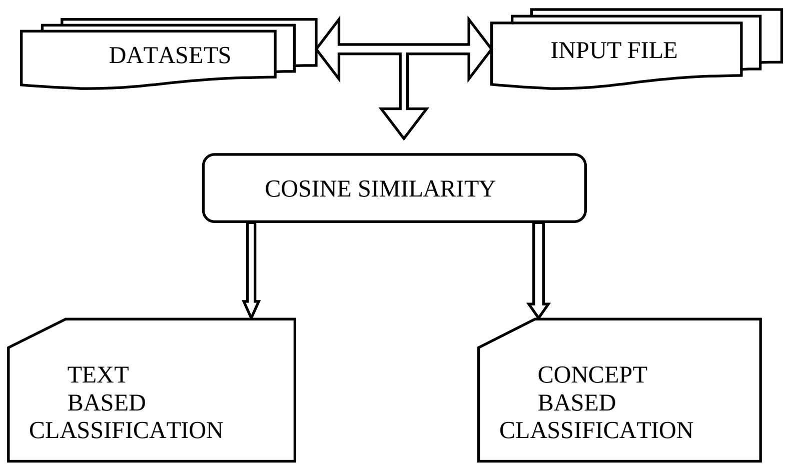

Fig. 1. System Architecture

The system architecture demonstrates that the datasets are divided into train and test data. After the model is constructed for the train data, the model is used to find the similarity of the test data. The similarity is found using the cosine similarity. Although we obtain both text based and concept based classification, our application focuses only on the concept based classification.

V. EVALUATION AND RESULT

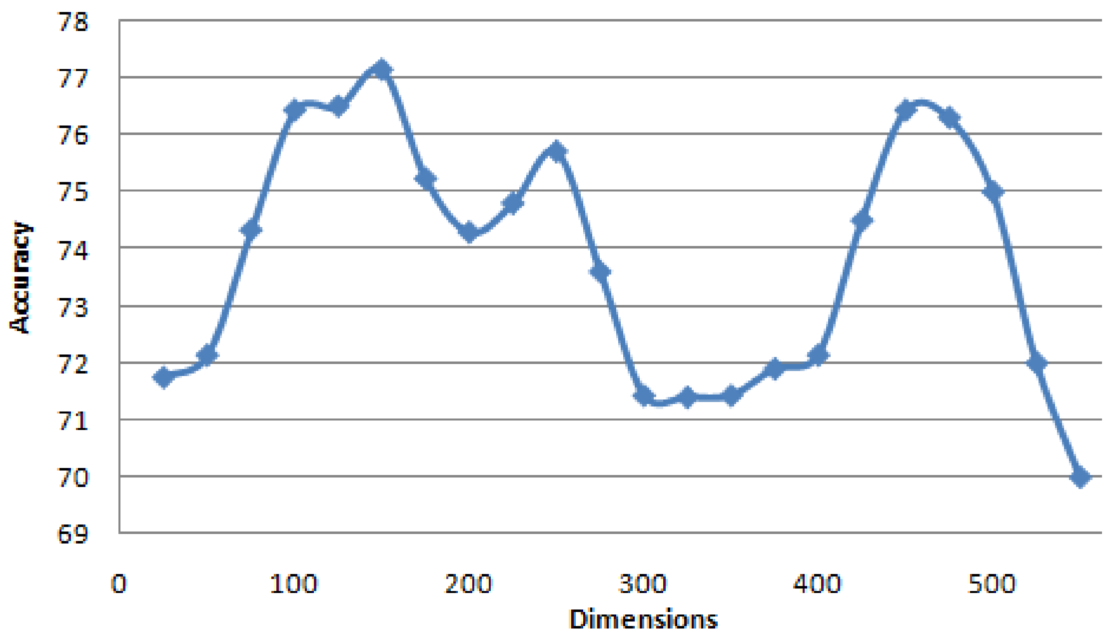

Fig 2. Classification Accuracy

The above figure is a graph which shows the accuracy of document classification. The dimensions on the x axis and the accuracy rate on the y axis. The accuracy rate differs with the change of dimension. Yet, it differs only by a few percent. Hence we can conclude that the accuracy rate is more or less the same for varied dimensions.

VI. CONCLUSIONS AND FUTURE SCOPE

We have presented our proposed LSA algorithm for emotion classification of text. LSA algorithm improves the efficiency of text emotion classification. This method could be further improved by taking more emotional information. Besides, we will also learn more efficiently from a spare training dataset. An analysis of LSA is carried from a coordinate geometrical perspective which gives an understanding of how LSA's behavior is influenced when extra information is provided. It is shown that the modified LSA model captures reasonably stronger correlations than LSA in the semantic space. It is concluded that supplementing LSA with extra information indeed increases its performance and therefore the modified LSA can be used as an efficient model to analyze word correlations.

REFERENCES

[1] Wang, Xuren, and Qiuhui Zheng. "Text Emotion Classification Research Based on Improved Latent Semantic Analysis Algorithm." Proceedings of the 2nd International Conference on Computer Science nd Electronics Engineering (ICCSEE 2013), number Iccsee. 2013.

[2] Salton G, Wong A, Yang CS. A Vector Space Model for Automatic Indexing [J] .Communications of the ACM, 1995,18(11) : 613-620.

[3] Karthik Krishnamurthi1, Vijayapal Reddy Panuganti2, Vishnu Vardhan Bulusu3, Influence of Supplementary Information on the Semantic Structure of Documents, International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 7, July 2015.

[4] Wiemer-Hastings and I. Zipitria, "Rules for Syntax, Vectors for Semantics", Annual Conference of the Cognitive Science Society, 1112–1117, 2001.

[5] R. Serafin, B. D. Eugenio and M. Glass, "Latent semantic analysis for dialogue act classification", North American Chapter of the ACL on Human Language Technology, 94–96, 2003.

[6] D. Kanejiya, A. Kumar and S. Prasad, "Automatic Evaluation of Students Answers using Syntactically Enhanced LSA", Workshop on Building Educational Applications using NLP, 53–60, 2003.

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science 1989.

[8] T. Rishel, A. L. Perkins and S. Yenduri, "Augmentation of a Term- Document Matrix with Part-of-Speech Tags to Improve Accuracy of LSA", International Conference on Applied Computer Science, 573–578, 2006.

[9] B. D. Eugenio and R. Serafin, "Dialogue Act Classification, Higher Order Dialogue Structure and Instance-Based Learning", Dialogue and Discourse, 1–24, 2010.