International Journal of Information Management Data Insights 2 (2022) 100061

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

journal homepage: www.elsevier.com/locate/jjimei

A novel approach for dimension reduction using word embedding: An enhanced text classification approach

Ksh. Nareshkumar Singh[a,*], S. Dickeeta Devi[a], H. Mamata Devi[a], Anjana Kakoti Mahanta[b]

[a] Dept. Of Computer Science, Manipur University, India
[b] Dept. Of Computer Science Gauhati University, India

ARTICLE INFO

Keywords:
Dimension reduction
Document representation
GloVe
Term weighting

ABSTRACT

One of the challenging tasks in text classification is to reduce the dimensional feature space. This paper discusses an enhanced text classification method using Bag-of-Words representation model with term frequency-inverse document frequency (tf-idf) and word embedding technique 'GloVe' to find words with similar semantic meaning. We select the word with the highest sum of tf-idf as the most representative word with similar meanings. The performance of the proposed method is compared with other methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Latent Semantic Indexing (LSI), a hybrid approach PCA+LDA using the Naïve Bayes classifier. Experimental results on three datasets, namely BBC, Classic4, and 20-newsgroup datasets, show that the proposed algorithm gives better classification results than existing dimension reduction techniques. Lastly, we defined a new performance evaluation metric to check the classifier's performance on the reduced features.

1. Introduction

Advancement in digital technology has significantly contributed to the massive amount of data available on the internet, especially text data. Text data are easily available in news articles, academic publications, emails, governments or organizations' files, postings, and messages on social media, etc. Most of these easily available text data are unstructured, difficult to process and even require more storage to analyze (Adikari et al., 2021). An emerging field called 'text mining' is used to process and analyze such a huge amount of unstructured text data (Hearst, M., 1999). The most important aspect for evaluation of a large amount of text data is to classify them. So, we emphasize text classification, which is one of the most often used methods in text mining (Kushwaha, Kar, & Dwivedi, 2021). It classifies the text data based on their similarities. In the text classification task, a number of features represent a document, and a large collection of documents involves thousands or even millions of features for each training instance. That is, the number of features defines the dimensionality for the dataset. These extensive input features cause the training to be extremely slow and difficult. This problem is referred to as the curse of dimensionality. A general field of study called dimensionality reduction is concerned with solving this challenge.

Dimensionality reduction refers to techniques that reduce the number of input features in a dataset. It is an essential step in text mining. It transforms the set of features into a more compact form that will improve the learning algorithm's efficiency. Dimensionality reduction techniques can use both supervised and unsupervised analytical methods. However, the characteristics of a dimensionality reduction algorithm vary depending upon the type of algorithm used. For instance, in case of unsupervised learning algorithms, dimensionality reduction technique should aim to minimize the feature information loss. Whereas in case of supervised learning, it should work to maximize class information. Due to the complex nature of the dimension reduction process, there is no single method suitable to deal with all situations. Thus, many dimension reduction approaches have been developed and tested in different application domains and research communities. In general, dimension reduction techniques can be linear and non-linear (Sumithra & Surendran, 2015). Linear dimension reduction transforms the data to a low dimension space as a linear combination of the original variables. It can be broadly classified into two groups. First group refers to techniques that take advantage of class-membership information while computing the lower-dimensional space. Examples of such methods include a variety of feature selection schemes that reduce the dimensionality by selecting a subset of the original features (Hoque, Bhattacharyya, & Kalita, 2014), and techniques that derive new features by the linear combination of the terms (Dzisevic & Sesok, 2019). Second type of dimensionality reduction technique is the computational algorithms based on statistical analysis. It includes principal component analysis

* Corresponding author.
E-mail addresses: nareshksh2711@gmail.com (Ksh.N. Singh), dikitasalam@gmail.com (S.D. Devi), mamata_dh@rediffmail.com (H.M. Devi), anjana@gauhati.ac.in (A.K. Mahanta).

https://doi.org/10.1016/j.jjimei.2022.100061
Received 26 May 2021; Received in revised form 24 January 2022; Accepted 24 January 2022

(Jolliffe, 1986; Karl, 1901), latent semantic indexing (Landauer, Foltz, & Laham, 1998; Rosario, 2001), linear discriminant analysis (Martinez & Kak, 2001; Webb, 2002) etc. On the other hand, non-linear dimension reduction applies when the original high dimensional data contains non-linear relationships. Example of non-linear dimensionality reduction includes self-organizing maps (SOMs), Isomap, Kernel PCA etc. Even though the aim and objective for performing dimension reduction are clear, there are still open issues in dimension reduction methods such as no effective way to determine the minimum number of dimensions sufficient to represent the data, information loss, underlying non-linear relationships among the input features may be very complicated to determine etc. (Carreira-Perpinan, 1997).

The method proposed in this paper achieves the dimensionality reduction by removing the redundant feature by evaluating the similarity scores between words using a word embedding technique called 'GloVe' (Pennington, Socher, & Manning, 2014). GloVe (Global Vectors) is a model for distributed word representation. It is an unsupervised learning algorithm for obtaining vector representations for words. The proposed model uses Naïve Bayes classifier (Domingos & Pazzani, 1996; Tan et al., 2015) to perform classification tasks on the reduced feature set. The performance of the proposed method is measured in terms of classification accuracy and F1 measure.

1.1. Contribution

The main contributions of this paper are:

1 Development of an effective dimensionality reduction method called 'removal of Redundant Feature' (rRF) using a word embedding 'GloVe' for document classification. The method performs classification with a Naïve Bayes classifier on three existing text datasets. The effectiveness of the method is evaluated in terms of classification accuracy and F1 score.
2 Terms with high similarity scores are grouped into clusters. Then, each cluster is represented by a term within the cluster that has the maximum sum of tf-idf over the corpus. It helps to remove the redundant features (present in each cluster) without compromising the classification performance.
3 A new performance evaluation metric called *New Performance Metric (NPM)* has been defined that estimates the classifier's performance on reduced features.

The rest of the paper is organized as follows. In Section 2, we explain related work in brief and formulate the problem. Section 3 studies the proposed system design for text classification. Section 4 explains the proposed work. Details of datasets and the experimental results are discussed in Section 5. Finally, Section 6 and 7 give discussion and conclusion respectively.

2. Related work

The curse of dimensionality is a common problem in text mining. Many researchers and scientists have been working to alleviate this problem using different approaches such as computation based on statistical methods, synonym words merge, word embedding, etc. Many dimensionality reduction methods based on statistical (Jollife, 1986; Karl, 1901; Karypis & Han, 2000; Martinez & Kak, 2001; Webb, 2002) and hybrid approaches (Pechenizkiy, Tsymbal, & Puuronen, 2006; Tang, Peng, Bi, Shan, & Hu, 2014; Zhao, Mio, & Liu, 2011) have been proposed for text classification. Pechenizkiy et al. (2006) and Zhao et al. (2011) proposed a new hybrid dimensionality reduction model combining PCA and LDA. The integrated approach could achieve the reduced dimension with good classification accuracy. Tang et al. (2014) defined a new hybrid approach that combines the Partial Least Squares (PLS) method with LDA. The LDA-PLS amends the projection direction of LDA by using the information of the PLS latent variable to the optimal direction. Changes in the projection direction make more conducive in the classification results.

Some researchers have also started to propose a text classification based on synonym words merging while selecting the feature word. It helps to reduce data's feature dimensions and also improve classification accuracy. Computing the word similarity to merge the synonym words using the lexical database – WordNet (Chunxiu, Pengwei, & Huailiang, 2007; Miller, Beckwith, Fellbaum, & Miller, 1991), Tongyici Cilin (Mei, Zhu, Gao, & Hongxiang, 1983) and HowNet (Dong & Dong, 2001). Yao, Liu, Zhang, and Wang (2017) proposed a feature selection algorithm based on synonym merging named SM-CHI. This method first selects features based on an improved CHI formula and then merges synonyms to re-select those features that can represent the categories better and reduce dimension. Fan, Zhang, and Li (2015) proposed an algorithm of word similarity computation based on HowNet. According to the definition of words in HowNet, a word is composed of a plurality of concepts. So, the word similarity is computed based on the concept similarity, and computing of concept similarity is based on the sememe similarity. Word similarity computation is essential to merge the synonym words. Zhang, Sun, and Wang (2004) use WordNet to extract more semantically related features. It can help to reduce the feature dimension and improve classification accuracy.

Recently, the popularity of word embedding techniques such as Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski, Grave, Joulin, & Mikolov, 2017; Joulin, Grave, Bojanowski, & Mikolov, 2017) etc. have been increasing in various applications because of capturing word semantics and syntactic. Word embedding techniques can also help in reducing feature dimensionality. Jin, Zhang, and Liu (2018) propose a method combining semantic dictionary and large-scale corpus statistics and weighting the strategy to calculate the semantic similarity of words. Xiaolin et al. solved the problem of poor universality and the absence of contextual information in word similarity calculation. Kim, Howland, and Park (2005) adopt a novel dimension reduction method to reduce the dimension of the document vectors. They used a centroid-based algorithm for dimension reduction of clustered data. Lilleberg, Zhu, and Zhang (2015) work focuses on using the term frequency-inverse document frequency (tf-idf) in conjunction with Word2Vec. Since Word2Vec treats each word equally in the document, it cannot distinguish the importance of each word to the document being classified. Therefore, the weighting scheme tf-idf is used along with Word2Vec and can also improve the classification accuracy. Rui, Liu, and Jia (2016) proposed an unsupervised feature selection method that utilizes word embedding to find the words with similar semantic meaning. The word embedding approach evaluates the similarity score between any words but does not answer why similarity. Zhang, Jatowt, and Tanaka (2016) defined several criteria to get effective shreds of evidence to support similarity justification between any two words.

2.1. Motivation

High dimensionality is one of the challenging problems for text data classification. High-dimensionality might mean hundreds, thousands, or even millions of input features. Datasets with high dimensionality can be much more complicated than low-dimensional ones, and that those complications are harder to discern. Processing high dimensional data is very expensive in terms of execution time and storage of data. Many of the problems encountered in machine learning involve thousands or even millions of features for each training instance. Not all these features will contribute well to the models; some of them are irrelevant and correlated features. The main idea behind dimensionality reduction is to reduce the number of features while preserving the original information content as much as possible without compromising the classification accuracy. Therefore, it is desired to remove irrelevant and redundant features to reduce the feature space. It motivates us to develop a new dimensionality reduction method that uses the word embedding

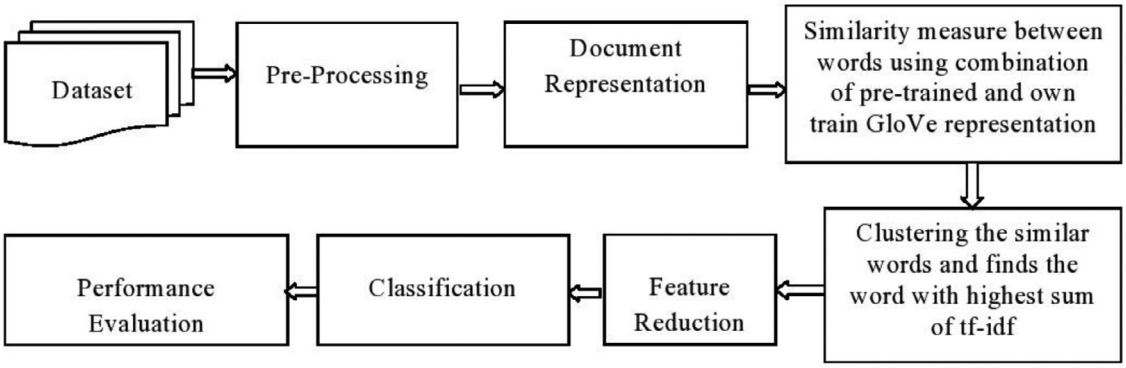

Fig. 1. Proposed system design for classification.

method 'GloVe' to remove the redundant features by evaluating the similarity score between the word vectors.

2.2. Problem formulation

The corpus of samples is mathematically represented by a $d \times n$ matrix $X \in R^{d \times n}$, where d is the number of documents and n is the number of features. Each document is denoted by a row vector x_i, $i = 1,2,\ldots,d$ and the k^{th} entry of x_i is denoted by x_{ik}, $k = 1,2,\ldots,n$. The dimensionality reduction problem can be stated as the problem of finding a function $f : R^n \rightarrow R^m$ where m is the dimension of data after dimensionality reduction and the value of m is smaller than n (might be $m<<n$). Each document $x_i \in R^d$ is transformed into $y_i = f(x_i) \in R^m$. We can formulate our framework of dimensionality reduction problem as follows: Given a set of collection of (words or) features from the documents, create a matrix W called term-document matrix. Learn a transformation of matrix W into W' such that W' is optimal according to some objective function J(W) in the given solution space where the function J is the process of reducing dimensionality.

3. Study of methodology

The workflow of the proposed method to perform text classification is shown in Fig. 1. The following sub-sections consider the steps involved in the proposed system design.

3.1. Pre-processing

Once the dataset has been imported, the next step is to pre-process the text data. Text pre-processing is the process of cleaning the raw text data. A robust text pre-processing system is essential for any application on NLP (Natural Language Processing) tasks. Because all of the textual components obtained after pre-processing serve as the fundamental building blocks of input that are fed into further text data applications. Pre-processing involves various techniques to convert the raw text into a well-defined structure: *lexical analysis* (word tokenization, removal of punctuations and special characters/symbol, ignore case sensitivity), *removal of stopwords, lemmatization*.

3.2. Document representation

The successful performance of text processing applications is dependent on the effective representations of documents. In the Vector Space Model (VSM), each document in a corpus is represented as a vector of real numbers. The Bag-of word (BoW) model is one of the popular models to represent the document (Harris, 1954). We used the BoW model to represent the documents in n-dimensional vector space where n is the number of unique words, and its weight is calculated by term frequency-inverse document frequency (tf-idf) weighting scheme (Manning, Raghavan, & Schutze, 2009) by setting two hyperparameters: *minimum document frequency* of 2 and *maximum document frequency* of 0.7. That means those terms whose document frequency is more than two and but not more than 70% of their appearance on the whole dataset will become the candidate features in the vocabulary. The hyperparameters help us to reduce the feature space. But still BoW model requires high computational cost due to massive vocabulary size (Neogi, Garg, Mishra, & Dwivedi, 2021). So, we further need to eliminate the irrelevant and redundancy features from the vocabulary (Blum & Langley, 1997).

3.3. Similarity detection

The pre-trained word embedding of the GloVe model (trained on Wikipedia 2014 and Gigaword 5 corpus) contains 400 K vocabulary used to evaluate the similarity between word vectors in our algorithm. In GloVe, each word is represented as a vector where words with similar meanings have similar representation (Mikolov et al., 2013; Pennington et al., 2014). Each word vector can be represented in various dimensions - 50d, 100d, 200d and 300d (Pennington et al., 2014). The proposed method used word vectors of small dimensions, i.e., 50d, because there is an issue with word embedding size (Ling, Song, & Roth, 2016). For example, it can take up to 6 GB of memory to load a word embedding matrix of 2.5 M tokens which have 300 dimension vectors. So, the practical use of word embedding should impose significant constraints to handle such large memory requirements. Even (Andrews, 2016) tried to compress the word embeddings using different compression algorithms. Our approach differs from this work as we try to reduce the feature dimensionality by eliminating redundant features. To measure the similarity of the word vector, only the pre-trained word embedding of GloVe is insufficient because many words do not include in the list of GloVe pre-trained word embedding. For example, the word 'thanx' is the selected feature for our experiment, but its vector representation is not present in the list of pre-trained word vectors of GloVe. Hence out-of-vocabulary problem occurs while measuring the similarity between word vectors. To solve this problem, our method performed training on the created vocabulary of our experimental dataset by using source code published by the GloVe authors to generate the word vector of each word in the vocabulary. Then we added extra 387 new words (selected words in the vocabulary but not in the pre-trained word embedding of GloVe) to the GloVe pre-trained word vector (400 K) list. So, we have a total of 400,387 words vector of 50d, which will be used in the word similarity measure. Now, we need to define the similarity threshold α ($0 \leq \alpha \leq 1$); choosing the value of α is difficult, but the value close to 1 can be taken so that it can capture the word-level synonymy. The experimental result shows that classification can achieve higher accuracy results when α=0.8 value. If the similarity value between word vectors is more than α, then words are grouped together into the same cluster. In most literature on clustering, each cluster is

represented by its mean or centroid (Rui et al., 2016). The worth of the sum of tf-idf over the corpus is not considered to represent the cluster. This paper considers the representation of each cluster with the term whose sum of tf-idf value is the highest.

3.4. Dimension reduction

We define dimension reduction as any operation that maps high dimensional data into a lower-dimensional space while preserving characteristics and relationships in the raw data. To reduce the dimensionality of input data features, we could use dimensionality reduction techniques based on statistical methods – PCA, LDA, LSI, and a hybrid approach of PCA+LDA. Both LSI and PCA use Singular Value Decomposition (SVD) techniques to perform dimensionality reduction. LSI applied SVD to a term-document matrix ((Dumais, 2005); Landauer et al., 1998), while PCA applied SVD to a term-covariance matrix (or symmetric correlation matrix). That means both should have a numeric and standardized data matrix, but only the difference is starting matrix. LDA finds the directions (or linear discriminants) representing the axes that maximize the separation between multiple classes. We experiment on combining two dimension reduction methods PCA and LDA (Pechenizkiy et al., 2006; Zhao et al., 2011). PCA is unsupervised learning, and it reduces the features by maximizing the variance of features in the lower dimension without considering the class label. Then apply LDA to find the components that maximize the separation between multiple classes. Lastly, a classifier model is applied over these modified features.

3.5. Classifier algorithm

From a few years back, it has been observed that many scholars are actively working on the task of automatic text classification as the internet usage rate is rapidly on the rise. Many text classifiers using machine learning techniques such as Decision Tree, Support Vector Machine, Naïve Bayes, K-Nearest Neighbour, etc., are commonly found in the literature of text classification (Jindal, Malhotra, & Jain, 2015; Khan, Baharudin, Lee, & Khairullah, 2010). One classifier gives better performance than others on some datasets. In some other datasets, another classifier may provide a better result. So, it depends on the type of applications, corpora characteristics, the different requirements for categorization, etc. This paper chooses the Naïve Bayes classifier because redundancy features cancel each other without affecting classification results. It works very well in a domain where lots of equally important features are there.

Naïve Bayes Classifier is a probabilistic classifier based on Bayes' theorem with strong assumptions that all features are independent. The sample (attributes) assigns to the class with the highest calculated conditional probability (Han, Kamber, & Pei, 2011; Tan et al., 2015).

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{1}$$

where A is the class and B is the feature vector. The probabilities of P(B|A), P(A) and P(B) are calculated using previously known instances, i.e. training data (Domingos & Pazzani, 1996). In general, the Naïve Bayes classifier is a good dependable baseline for text classification.

3.6. Performance evaluation

A confusion matrix (also called a contingency table or error matrix) is used to visualize the performance of the classifier. Accuracy, precision, recall and F-measure are calculated based on the confusion matrix (Olson & Delen, 2008). These four performance metrics are used to analyze the results of our work. Basically, a confusion matrix is a table consisting of two rows and columns that report the number of false positives (FP), false negatives (FN), true positives (TP) and true negatives (TN). Table 1 shows the confusion matrix.

Table 1
Confusion matrix.

	Predict as positive	Predict as negative
Actual Positve	TP	FN
Actual Negative	FP	TN

- *Accuracy:* it is the ratio of the classifier's correct predictions to the sum of the classifier's predictions. It is defined as:

$$\text{Acc} = \frac{TP + TN}{(TP + FN + FP + TN)} \tag{2}$$

- *Precision:* it is the ratio of the accurate data among the retrieved data. Its formula based on the confusion matrix is defined as:

$$\text{P} = \frac{TP}{(TP + FP)} \tag{3}$$

- *Recall:* it is the ratio of relevant data among the retrieved data. Its formula is defined as:

$$R = \frac{TP}{(TP + FN)} \tag{4}$$

- *F-measure:* it is the harmonic mean of precision and recall. It is defined as:

$$F - \text{measure} = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{5}$$

Receiver Operator Characteristic (ROC) curve: is also used to evaluate the classifier output quality. It is a probability curve that plots the TPR (True Positive Rate) against FPR (False Positive Rate). The AUC (Area Under the Curve) is a measure of the classifier's ability to distinguish between classes. ROC curve is mainly used for the binary classification problem, but we are considering a multi-class classification problem here. So, we used One-vs-One (OvO) and One-vs-Rest (OvR) schemes. In OvO scheme, compare every unique pairwise combination of classes. In OvR scheme, the comparison is made between one class and the rest of the classes.

Feature Reduction Rate (FRR): the purpose of feature selection methods is to select the most relevant and important features. It helps to reduce the feature space. This calculation is shown in Eq. (6) (Hsin-Yu, Chen, & Min, 2015).

$$\text{FRR} = \frac{(OF - FS)}{OF} \tag{6}$$

where OF = number of original features and FS = number of remaining features after applying the feature selection/reduction method. The value of FRR lies between 0 and 1. When FRR is closed to 1, it means high performance to reduce the feature space. When FRR is closed to 0, it indicates low performance.

4. Proposed feature reduction method

In this paper, we propose a feature reduction method that utilizes the word embedding method 'GloVe'. In English, there are several words that express a similar meaning, and these are called synonyms. This causes redundancy in BoW features that enlarge the feature space. Our work is to identify synonym words based on the similarity between word vectors. The words within one cluster are considered similar words, and they are redundant features in text classification. In most literature on clustering, clusters are represented by their centroid or mean value. Here, each cluster is represented by a term that has the highest sum of tf-idf over the corpus. It helps to remove the redundant features without decreasing the classification performance. Thus, the proposed work reduces the feature dimensionality. The proposed reduction method is described below.

4.1. removal of redundant feature (rRF)

Algorithm:

Inputs: a) $D = \{d_1, d_2, d_3, \ldots, d_N\}$ be a set of N training documents
b) $V = \{t_1, t_2, t_3, \ldots, t_n\}$ is the set of n terms in the vocabulary
c) Use the GloVe model to train each term in V to get a vector representation for each term. Then, put those terms which are not in the list of pre-trained GloVe representation so that out of vocabulary problem is solved during similarity measurement between word vectors (wv).
d) Empirically, we take the value of α, the predetermined similarity threshold equals 0.8, which gives the highest classification accuracy.
Output: A set of reduced term, FS
Steps:
1. for each $t_i \in V$ do
2. $D \leftarrow \phi$
3. for each $t_{i+1} \in V$ do
4. sim_val = model.similarity (wv[t_i],wv[t_{i+1}])
5. if sim_val > α do
6. C_k = set []
7. $C_k \leftarrow$ set([t_i, t_{i+1}])
8. goto step 3; t_i is compared with next element and so on.
9. $D \leftarrow C_k$
10. End for
11. if t_i not in D do
12. $D \leftarrow t_i$
13. End for
14. $D = [t_1, t_2, t_3, t_6 \ldots t_j, C_k[t_4,t_7,t_8,t_9], \ldots C_p[t_{11},\ldots t_{12},t_{99}],..,C_s[\ldots]]$ where t_i = single independent term (uncorrelated to other features) & C_z = set of clusters
15. Each Cluster C_z is represented by a term with the highest sum of tf-idf over the corpus.
16. Selected reduced feature, FS = $[t_1,t_2,t_3,t_6,\ldots,t_i,\ldots t_{k4},..t_{p99},..t_{sj}]$ where t_{sj} = select the term 't_j' to represent the C_s cluster. (t_{k4} = select the 't_4' term with highest sum of tf-idf among others $\{t_7,t_8,t_9\}$ in the C_k cluster)

4.2. New performance metric

In Feature Reduction Rate (FRR), only the rate of feature reduction is calculated. FRR values range from 0 to 1 and may have very high or low, but it doesn't give any information regarding how well the classifier model works on the reduced features. So, we defined a new performance evaluation metric, the so-called New Performance Metric (NPM), that estimates the performance of the classifier on reduced features. To evaluate NPM, it depends on two parameters- F1 measure and a factor 'χ'. 'χ' is the ratio of the number of remaining features or selected features (FS) after applying the feature reduction method to the original number of features (OF), that is χ = FS/OF. The relationship between χ and FRR is

$$\mathrm{FRR} = 1 - \chi \tag{7}$$

The goal of NPM is to obtain maximum feature reduction with a high F1 score value. We defined the new performance evaluation matric as:

$$New\ Performance\ Metric\ (NPM) = \frac{1}{(1+\chi)} \times \mathrm{F1} \tag{8}$$

where $\frac{1}{1+\chi}$ indicates the importance of each selected feature and the NPM value lies between 0 and 1. NPM can be used in determining which dimension reduction methods perform well on reduced features. While comparing, the dimension reduction method with a high NPM value will be considered as the best reduction method among them. We could also elaborate the factor which is multiplied to F1 score as:

$$\frac{1}{1+\chi} = \frac{OF}{OF+FS} \tag{9}$$

In this factor, the value of OF will remain constant; only the value of FS is varied. When FS value is small (that is, high FRR), which leads increase the value of $1/(1+\chi)$. So, the overall value of NPM will also increase. When the FS value is large, the term $1/(1+\chi)$ will decrease which leads to decrease the NPM value. As an example, suppose we have 1000 numbers of features. There are two feature reduction methods 'A' and 'B'. Both are applied to it, and the method 'A' can able to reduce the original 1000 features to 10 features. It can obtain a 99 percent feature reduction rate with an F1 score of, say, 70 percent. Another method 'B' can reduce to 200 features and get 80 percent FRR with the same F1 value. According to the F1 score value, both classifiers results are the same, so we can't distinguish which reduction methods precisely select the most important features. Actually, one classifier can achieve this F1 value using only ten features, whereas another can get the same F1 value using 200 features. So, the effectiveness (or contribution) of each selected feature by method 'A' toward the classification tasks is higher than features selected by method 'B'. The classifier works only on the reduced features. The role of parameter 'χ' comes here to identify the number of important features out of original features. According to NPM measurement, the classifier using the feature reduction method 'A' has an NPM value of 69.30 percent, while another classifier with feature reduction method 'B' has an NPM value of 58.33. Thus, it can conclude that the classifier with method 'A' gives better performance on reduced features than the classifier with method 'B'. We also used NPM to evaluate the classification performance on the reduced features to analyze our experimental results.

4.3. How does the proposed method differ from others?

- Like PCA, LDA, LSA and PCA+LDA, our proposed method also helps in data features reduction and hence reduces storage space. They also help remove redundant features, if any. So they can reduce the computation cost.
- All the considered dimension reduction techniques except the proposed method are based on statistical analysis. They are appropriate to use in situations when the relationships among the dimensions are linear. But the proposed method uses the word embedding technique to reduce the dimensionality.
- When applying dimension reduction methods PCA, LDA, LSA and PCA+LDA, it is important to indicate the following parameter: how many dimensions should be reduced. Experimentally, we can evaluate how many dimensions should be in the new vector space for each task. The number of dimensions needed must be determined for each collection. But in the case of rRF method, it is not required to determine how many dimensions should be in the new vector space.
- rRF method calculates the similarity score between words and removes the redundancy features, if any. It needs simple mathematical calculation as compared to statistical-based approaches.
- There are non-linear relationships among data. Understanding the underlying relationships among the input features may be very complicated. But the proposed method used word vectors representation to measure the relationship score among words. So, there is not much important whether the data is linear or not.
- When mean and covariance are not enough to define datasets, in that case, PCA fails. We may not know how many principal components to keep in practice; some thumb rules are applied. LDA fails when the mean of the distributions are shared, as it becomes impossible for LDA to find a new axis that makes both the classes linearly separable. In such cases, rRF can work on it because mean and covariance are not important parameters of it.

5. Experimental results

5.1. Data collections

Three corpora, whose sizes vary from small to large: BBC corpus (Greene & Cunningham, 2006), Classic4 dataset (Tunali, 2010) and 20-newsgroups dataset (Mitchell, 1997), are used to implement our experiments.

The BBC news corpus is a small dataset consisting of 2225 documents collected from 2004 to 05 in five topical areas. Each document is labelled with one of the following five classes: Business (510 documents), Entertainment (386 documents), Politics (417 documents), Sport (511

Table 2
Performance analysis for BBC Dataset.

Dimension Reduction Methods	Accuracy	Precision	Recall	F1 Score	No. of features/ Components	FRR	NPM
PCA	90.56	90.06	90.57	90.31	1600	0.89	80.60
LDA	52.81	52.24	53.44	53.44	4	0.99	53.42
LSI	72.81	70.86	70.87	70.87	1580	0.89	63.33
PCA+LDA	74.16	74.28	74.49	74.39	1600+4	0.89	66.37
rRF	**96.18**	**96.25**	**95.98**	**96.11**	**358**	**0.97**	**93.58**

Table 3
Performance analysis for Classic4 Dataset.

Dimension Reduction Methods	Accuracy	Precision	Recall	F1 Score	No. of features/ Components	FRR	NPM
PCA	89.36	90.54	89.00	89.76	3800	0.62	64.90
LDA	48.56	47.03	49.52	48.25	3	0.99	48.23
LSI	72.52	70.15	79.30	74.45	3249	0.67	56.08
PCA+LDA	70.40	70.21	69.68	69.94	3800+3	0.62	50.56
rRF	**91.12**	**90.88**	**91.98**	**91.43**	**459**	**0.95**	**87.38**

documents) and Technology (401 documents). It has 1780 training samples and 445 test samples. There are a totally of 13,290 candidate features in the training part.

Classic4 dataset, a well-known benchmark dataset, is used in text mining. It is a medium-size corpus consisting of 7095 documents and has four different categories: CACM (3204 documents), CISI (1460 documents), CRAN (1398 documents) and MED (1033 documents). It has 5676 training samples and 1419 test samples. A total of 9923 candidate features appear in the training part.

20 Newsgroups dataset, one of the most popular benchmark datasets, is used as a large dataset in our experiment. It is the collection of nearly 20 thousand documents with 20 different class labels. It has 15,997 training samples and 4000 test samples. The vocabulary size of the dataset is quite large, which contains 50,908 candidate features in the training part.

The above three datasets exhibit the different levels of imbalanced over categories: 20 Newsgroup dataset is well balanced; BBC dataset is moderately imbalanced, but Classic4 dataset is quite an imbalance.

5.2. Results

The experiment has been carried out to validate the proposed method in terms of classification accuracy and F1 measure on the above three datasets. The metric 'accuracy' is considered a good performance evaluation metric when the classes are balanced. But in our case, all three dataset classes exhibit different levels of imbalance over their categories. So, only accuracy can't determine the performance of the classifier. Along with good accuracy, we try to achieve the high F1 score value with a minimum number of features as well as a large AUC (area under curve) score of ROC. The following Tables no. 2, 3, and 4 show the performance results of the Naïve Bayes classifier on BBC corpus, Classic4 and 20-Newsgroups datasets, respectively.

The above tables indicate the comparative analysis of the results between the proposed method rRF and different standard dimension reduction techniques. The rRF method with the minimum number of features gives better performance than other individual dimension reduction techniques.

We can also tune hyperparameters of dimension reduction techniques to increase classification performance. For PCA, the following graph of Figs. 2, 3 and 4 have the ability to estimate how many components are needed to explain at least 90% of our feature variation. This can be determined by looking at the cumulative explained variance ratio as a function of the number of components. Here, 1600, 3800 and 4000 components (to describe more than 90% of the variance) were chosen as the optimum number of components for BBC, Classic4 and 20 newsgroups, respectively. Likewise, for LDA, the number of components that can cover more than 90% of essential characteristics of the data for BBC corpus, Classic4 dataset and 20 newsgroups dataset are 4, 3 and 9, respectively. For LSA, the number of components to get at least 90% variance are 1580, 3249 and 3266 for BBC corpus, Classic4 dataset and 20 newsgroups, respectively.

Table 4
Performance analysis for 20 newsgroups dataset.

Dimension Reduction Methods	Accuracy	Precision	Recall	F1 Score	No. of features/ Components	FRR	NPM
PCA	89.53	89.48	89.59	89.54	4000	0.92	83.01
LDA	63.38	63.32	63.58	63.58	9	0.99	63.56
LSI	84.67	84.61	85.58	85.09	3266	0.93	79.96
PCA+LDA	92.67	92.61	92.66	92.64	4000+9	0.92	85.87
rRF	**90.25**	**90.24**	**90.40**	**90.32**	**2135**	**0.96**	**86.68**

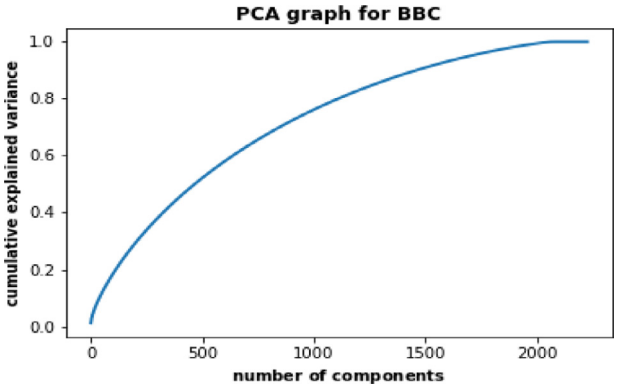

Fig. 2. PCA graph for BBC.

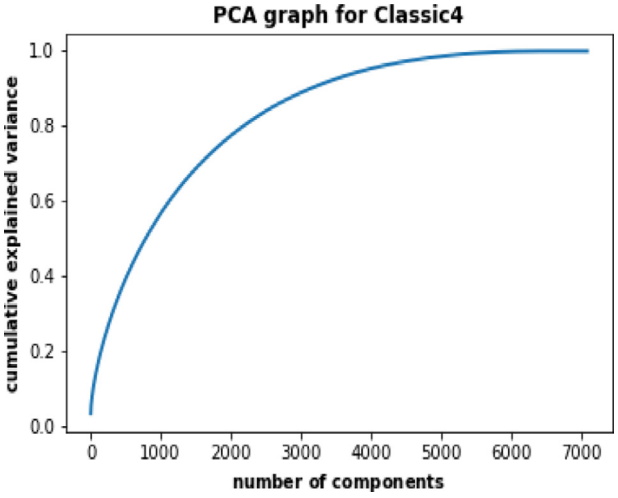

Fig. 3. PCA graph for Classic4.

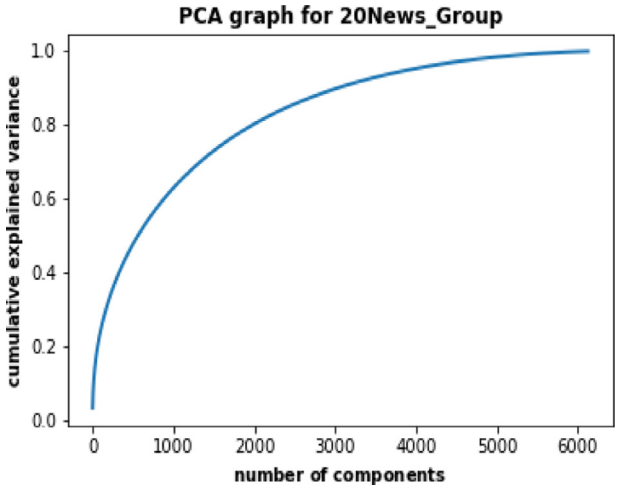

Fig. 4. PCA graph for 20-newsgroup.

6. Discussion

The performance analysis of the Naïve Bayes classifier in terms of F1 score on three datasets against the four standard dimension reduction methods and the proposed method is shown in Fig. 5. The X-axis represents the dimension reduction methods applied on the three datasets, and Y-axis indicates the classifier's F1 score on a specific dimension reduction method.

Next, the calculated ROC AUC scores of three datasets using OvO and OvR schemes are shown in Tables No. 5, 6, and 7, respectively. We report the macro average and weighted by prevalence for each method. The higher the AUC score, better the performance of the model to distinguish the positive and negative classes. The AUC score for rRF method is higher than all other individual methods in all three datasets. The hybrid method PCA+LDA for 20 newsgroups gave the highest AUC score. That means the hybrid method works well for large datasets.

The two key aspects of the developed system model that we intended to achieve together are i) high feature reduction rate and ii) good F1 score. The performance results show that F1 scores of PCA (in Classic4 and 20-NewsGroup) are closed to the rRF results. But rRF achieved those results with less number of features. So, the rRF is more superior to PCA in terms of feature reduction rate and classification accuracy. In the 20-NewsGroup dataset, the hybrid approach PCA+LDA outperforms the proposed method rRF in terms of classification accuracy and F1 score but not the feature reduction rate. But evaluation based on both feature reduction rate and F1 score i.e. NPM, rRF gives better results than PCA+LDA. LDA approach could achieve a high feature reduction rate but gives low classification accuracy. This is not the system model wanted to achieve. Similarly, the AUC scores of PCA, PCA+LDA, and rRF are very similar in Table 5 & 7. That means, all the methods perform very well to distinguish the positive and negative classes. But the proposed rRF method is able to distinguish the two classes using lesser

Table 5
ROC AUC score for BBC Dataset.

	One-vs-One		One –vs – Rest	
Methods	macro average	weighted by prevalence	macro average	weighted by prevalence
PCA	99.01	99.01	99.01	99.03
LDA	76.15	75.97	76.08	75.73
LSI	91.87	92.30	92.24	92.84
PCA+LDA	93.61	93.53	93.58	93.45
rRF	**99.76**	**99.77**	**99.77**	**99.78**

Table 6
ROC AUC score for Classic4 Dataset.

	One-vs-One		One-vs-Rest	
Methods	macro average	weighted by prevalence	macro average	weighted by prevalence
PCA	98.68	98.60	98.50	98.51
LDA	71.16	71.33	71.56	71.34
LSI	92.93	91.93	92.15	90.08
PCA+LDA	92.60	92.80	92.82	92.90
rRF	**98.99**	**98.83**	**98.87**	**98.59**

Table 7
ROC AUC score for 20-News Groups Dataset.

	One-vs-One		One-vs-Rest	
Methods	macro average	weighted by prevalence	macro average	weighted by prevalence
PCA	99.26	99.26	99.26	99.26
LDA	96.38	96.39	96.39	96.40
LSI	98.50	98.50	98.50	98.51
PCA+LDA	99.66	99.66	99.66	99.66
rRF	**99.58**	**99.58**	**99.59**	**99.58**

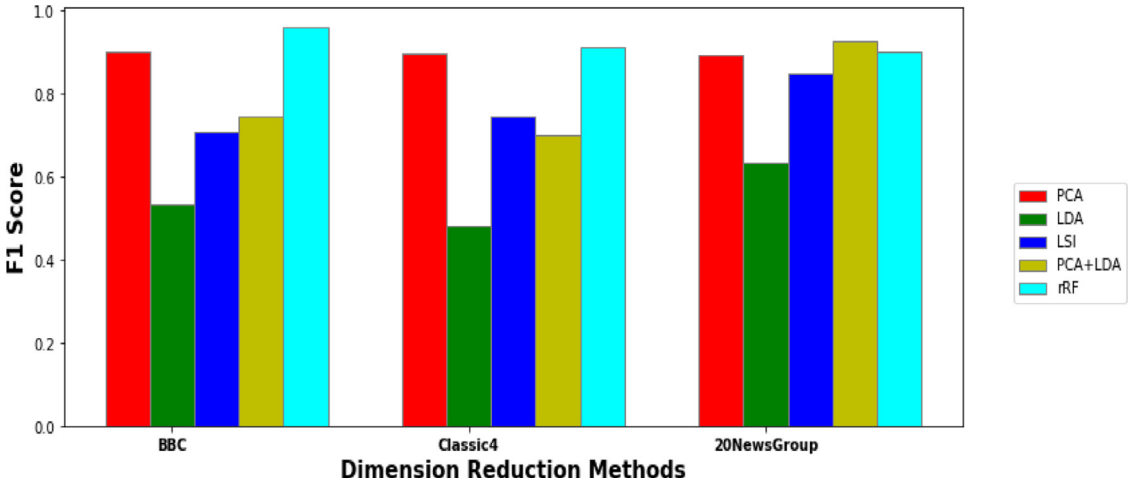

Fig. 5. Shows the F1 score results of different dimension reduction methods on given three datasets.

number of features as compared to other competent methods (i.e. PCA and PCA+LDA). This will help in reducing the processing time.

6.1. Contributions to literature

The number of features represent the document, and a large collection of documents involve thousands or even millions of features. This high dimensional dataset is at risk of being sparse, making training slower and complicated to find good solutions. Our focus is to represent the document with lesser number of features in the vector space. In this study we proposed a dimension reduction method based on GloVe that helps to identify the redundant feature. The proposed method rRF removes all the redundant features. The rRF gives better performance than other competing methods in the given three datasets except for the result of PCA+LDA in 20-NewsGroup. The numbers of considered features are also less in the proposed method. That is, using rRF method, all documents can be represented with fewer dimensions in the new vector space. LDA can also represent the data in fewer dimensions in the vector space, but its classification performance is not good. Overall, it can say that the rRF could achieve good classification accuracy with less number of features. For determining the performance of classifier model on the reduced features, we also define the NPM. Due to the different characteristics and complex nature of the data, there is no single best method of dimensionality reduction to deal with all situations. The best approach is to use systematic controlled experiments to discover what dimensionality reduction techniques is paired with the developed model that gives the best performance result on the given dataset.

6.2. Implication to practice

With the ever-increasing social media data in today's age, especially text data, there is a dire need to analyze such huge text data systematically and effectively. It is observed that dimensionality reduction method plays an important role in classifying the text data with thousands of text features. Using dimension reduction technique is indeed very beneficial in classification algorithms not only in term of accuracy but also in reducing the overfitting of data and computational time. This research will be valuable in the current trend of big data world.

7. Conclusion

In this paper, we proposed a method for removing redundant features based on word embedding 'GloVe'. Our method groups the terms with similar semantic meaning by evaluating the similarity between words using GloVe. Then we selected the most representative term in each cluster to reduce the number of features in the BoW collection. Experimental results show that f-measure of the proposed method rRF performs better compared to individual dimension reduction techniques except in the case of the hybrid PCA+LDA approach in 20 newsgroups. But based on the proposed performance metric, i.e. NPM, our method outperforms all other existing methods. For example, for 20-newsgroups, the F1 score of PCA+LDA is higher than rRF, but it requires more feature space than rRF. On considering both feature reduction rate and f-measure, rRF gives even better results than PCA+LDA. The advantage of NPM is that it uses the feature reduction rate along with F1 score. Performance evaluation metrics built from the confusion metric does not consider the features reduction rate. The rRF method has the advantage of not determining the number of dimensions to be reduced, and it can be used when the dataset's mean and covariance are unknown. We could conclude that when the dataset becomes large, the performance of the hybrid PCA+LDA approach works better. In general, PCA also gave good classification results as compared to LDA and LSI. LDA gave the worst performance among them. While experimenting, we could also observe that classification performance depends on many factors: choosing classification algorithms, using different textual representation methods, feature selection methods, dimension reduction techniques, various parameters factor on the chosen methods, etc.

References

Adikari, A., Burnett, D., Sedera, D., Silva, D., & Alahakoon, D. (2021). Value co-creation for open innovation: An evidence-based study of the data driven paradigm of social media using machine learning. *International Journal of Information Management Data Insights, 1*, Article 100022. 10.1016/j.jjimei.2021.100022.

Andrews, M. (2016). Compressing Word Embeddings. The 23rd International Conference on Neural Information Processing (ICONIP), Japan. http://arxiv.org/abs/1511.06397

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*(1–2), 245–271. 10.1016/S0004-3702(97)00063-5.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. arXiv:1607. 04606v2[cs.CL].

Carreira-Perpinan, M. A. (.1997). A review of dimension reduction techniques. technical report cs 96-09, Dept. of Computer Science, University of Sheffield.

Chunxiu, Q., Pengwei, Z., & Huailiang, L. (2007). Research of Word Similarity Computation. *Information studies: Theory & Application, 6*(01), 22–26.

Domingos, P. M., & Pazzani, M. (1996). Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 105–112). http://www.ics.uci.edu/~pazzani/Publications/mlc96-pedro.pdf.

Dong, Zhendong, & Dong, Qiang (2001). Construction of a Knowledge System and its Impact on Chinese Research. *Comtemporary Linguistics, 3*, 33–44.

Dumais, Susan T. (2005). Latent Semantic Analysis. *Annual Review of Information Science and Technology, 38*(1), 188–230. 10.1002/aris.1440380105.

Dzisevic, R., & Sesok, D. (2019). Text Classification using Different Feature Extraction Approaches. *Open Conference of Electrical, Electronic and Information Sciences (eStream)*. 10.1109/estream.2019.8732167.

Fan, M., Zhang, Y., & Li, J. (2015). Word similarity computation based on HowNet. In *12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)* (pp. 1487–1492). 10.1109/FSKD.2015.7382164.

Greene, D., & Cunningham, P. (2006). Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering. In *Proceedings of 23rd International Conference on Machine learning (ICML) [Data set]*. ACM Press http://mlg.ucd.ie/datasets/bbc.html.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining concepts and techniques* (3rd ed.). Morgan Kaufmann Publishers. 10.1016/C2009-0-61819-5.

Harris, Z. (1954). Distributional Structure. *Word, 10*(2–3), 146–162. 10.1080/00437956.1954.11659520.

Hearst, M. (1999). Untangling Text Data Mining. In *Proceedings of ACL'99: The 37th Annual Meeting of the Association for Computational Linguistics*. University of Maryland (invited paper).

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2014). MIFS-ND: A mutual information-based feature selection method. *Expert Systems with Applications* http://dx.doi.org/10.1016/j.eswa.2014.04.019.

Hsin-Yu, H., Chen, Sh., & Min, C. (2015). FC-MST: Feature Correlation Maximum Spanning Tree for Multimedia Concept Classification. In *Proceedings of the IEEE 9th International Conference on Semantic Computing, USA* (pp. 276–283). http://dx.doi.org/10.1109/ICOSC.2015.7050820.

Jin, X., Zhang, S., & Liu, J. (2018). Word Semantic Similarity Calculation Based on Word2vec. In *International Conference on Control, Automation and Information Sciences (ICCAIS)* (pp. 12–16). 10.1109/ICCAIS.2018.8570612.

Jindal, R., Malhotra, R., & Jain, A. (2015). Techniques for text classification: Literature review and current trends. *Webology, 12*. https://www.webology.org/2015/v12n2/a139.pdf.

Jollife, I. T. (1986). *Principal component analysis*. Springer-Verlag.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2*, 427–431.

Karl, P. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series, 2*(11), 559–572.

Karypis, G. (2000). Fast Supervised Dimensionality Reduction Algorithm with Applications to Document Categorization & Retrieval. In *CIKM '00: Proceedings of the ninth international conference on Information and knowledge management* (pp. 12–19). 10.1145/354756.354772.

Khan, A., Baharudin, B., Lee, L. H., & Khairullah, K. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. *Journal of Advances in Information Technology, 1*(1), 4–20 http://dx.doi.org/10.4304/jait.1.1.4-20.

Kim, H., Howland, P., & Park, H. (2005). Dimension Reduction in Text Classification with Support Vector Machines. *Journal of Machine Learning Research, 6*, 37–53. http://jmlr.org/papers/v6/kim05a.html.

Kushwaha, A. K., Kar, A. K., & Dwivedi, Y. K. (2021). Applications of big data in emerging management disciplines: A literature review using text mining. *International Journal of Information Management Data Insights, 1*(2). doi.org/10.1016/j.jjimei.2021.100017.

Landauer, K. T. h., Foltz, P., & Laham, D. (1998). An Introduction to Latent Semantic Analysis. *Discourse Processes, 25*, 259–284. 10.1080/01638539809545028.

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support Vector Machines and Word2Vec for Text Classification with Semantic Features. In *Proceeding on IEEE 14th International Conference on Cognitive Informatics and Cognitive Computing* (pp. 136–140). 10.1109/ICCI-CC.2015.7259377.

Ling, S., Song, Y., & Roth, D. (2016). Word embeddings with limited memory. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Germany, 387–392. https://doi.org/10.18653/v1/P16-2063

Manning, C. D., Raghavan, P., & Schutze, H. (2009). *An introduction to information retrieval*. Cambridge University Press. 10.1017/CBO9780511809071.

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Trans. Pattern Analysis and Machine Intelligence, 23*, 228–233.

Mei, J., Zhu, Y., Gao, Y., & Hongxiang, Y. (1983). *Tongyici cilin (Dictionary of synonymous words)*. Shanghai Cishu Publisher.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR)USA* http://arxiv.org/abs/1301.3781.

Miller, G. A., Beckwith, R., Fellbaum, C., & Miller, K. J. (1991). Introduction to WordNet: An On-line Lexical Database. *International Journal of Lexicography, 3*(4), 235–244. 10.1093/ijl/3.4.235.

Mitchell, T. (1997). 20 Newsgroups [Data set]. *UCI Machine Learning Repository*. https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups.

Neogi, A. S., Garg, K. A., Mishra, R. K., & Dwivedi, Y. K. (2021). Sentiment analysis and classification of Indian farmers' protest using twitter data. *International Journal of Information Management Data Insights, 1*(2), 1–11. doi.org/10.1016/j.jjimei.2021.100019.

Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques* (1st ed.). Springer http://dx.doi.org/10.1007/978-3-540-76917-0.

Pechenizkiy, M., Tsymbal, A., & Puuronen, S. (2006). On combining principal components with Fisher's linear discriminants for supervised learning. *Foundations of Computing and Decision Sciences, 31*, 59–73. http://www.win.tue.nl/~mpechen/publications/FCDS_Pechenizkiy_et_al.pdf.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Qatar* (pp. 1532–1543). 10.3115/v1/D14-1162.

Rosario, B. (2001). Latent Semantic Indexing: An Overview. *INFOSYS*, 240.

Rui, W., Liu, J., & Jia, Y. (2016). Unsupervised feature selection for text classification via word embedding. In *Proceedings of IEEE International Conference on Big Data Analysis (ICBDA), China* (pp. 1–5). 10.1109/ICBDA.2016.7509787.

Sumithra, V. S., & Surendran, S. (2015). A Review of Various Linear and Non Linear Dimensionality Reduction Techniques. *International Journal of Computer Science and Information Technologies (IJCSIT), 6*(3), 2354–2360.

Tan, P., Steinbach, M., & Kumar, V. (2015). *Introduction to data mining (3rd imp.)*. Dorling Kindersley: Pearson.

Tang, L., Peng, S., Bi, Y., Shan, P., & Hu, X. (2014). A New Method Combining LDA and PLS for Dimension Reduction. *PLoS One, 9*(5), E96944 12. 10.1371/journal.pone.0096944.

Tunali, V. (2010). Classic4 [Data set]. http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/

Webb, A. R. (2002). *Statistical pattern recognition* (2nd ed.). John Wiley.

Yao, H., Liu, C., Zhang, P., & Wang, L. (2017). A feature selection method based on synonym merging in text classification system. *EURASIP Journal on Wireless Communications and Networking, 166*, 1–8. 10.1186/s13638-017-0950-z.

Zhang, K., Sun, J., & Wang, B. (2004). A WordNet-Based approach to feature selection in text categorization. *Intelligent Information Processing, II*, 475–484.

Zhang, Y., Jatowt, A., & Tanaka, K. (2016). Towards Understanding Word Embeddings: Automatically Explaining Similarity of Terms. In *IEEE International Conference on Big Data, USA* (pp. 823–832). 10.1109/BigData.2016.7840675.

Zhao, N., Mio, W., & Liu, X. (2011). A Hybrid PCA-LDA Model for Dimension Reduction. In *Proceedings on International Joint Conference on Neural Networks, USA* (pp. 2184–2190). 10.1109/IJCNN.2011.6033499.