SIGSPACE-TEXT: PARALLEL AND DISTRIBUTED SIGNATURE LEARNING IN TEXT ANALYTICS

A THESIS IN
Computer Science

Presented to the Faculty of the University
Of Missouri-Kansas City in partial fulfillment
Of the requirements for the degree

MASTER OF SCIENCE

By
RAKESH REDDY BANDI

B.Tech, Jawaharlal Nehru Technological University – Hyderabad, India, 2015

Kansas City, Missouri
2016

SIGSPACE-TEXT: PARALLEL AND DISTRIBUTED SIGNATURE LEARNING IN TEXT ANALYTICS

Rakesh Reddy Bandi, Candidate for the Master of Science Degree

University of Missouri-Kansas City, 2016

ABSTRACT

Big data analytics uncover hidden patterns and useful information from big data. It is a complex and time-consuming process. Recent advancements in parallel and distributed approaches have led to the evolution of big data analytics. It also claimed bigger data may not always be better data. Toward scalable solutions for big data analytics, it is highly demanded to have a scalable and dynamic process with more representative and relevant sets of data. We envision that if the condensed and representative sample can be drawn from very large-scale datasets in a parallel and distributed manner and this can be defined as signature learning, this approach can provide more accurate results in an efficient manner. Using signature learning with relevant datasets in a parallel and distributed manner, the complexity of big data problems can be reduced.

In this thesis, we propose the SigSpace-Text framework that is an extension of our previous model of signature-based learning (SigSpace) that proved the effectiveness of signature-based classification with image signatures and audio signatures. SigSpace was not feasible with text data due to the inherent problems in the text domain such as a high-dimensional feature space and sparse feature vectors. In order to handle these issues, we explore using Natural Language Processing, that features extraction and feature selection techniques (TFIDF, Word2Vec). Signature learning in SigSpace-Text is based on a class-level

clustering approach, in which a generic pattern is identified for a given category using state-of-the-art clustering algorithms, i.e., K-Means, Self-Organizing Maps (SOM), and Gaussian Mixture Models (GMM). These signatures are used (instead of raw data) as a feature set to the classification. Through extension, the proposed SigSpace-Text approach brings vital, practical information to signature learning approaches on several text classification tasks. The SigSpace-Text model supports incremental, distributed, and parallel learning using big data analytics including Apache Spark and the Machine Learning library such as Spark MLlib. In experiments with the SigSpace-Text framework, the effectiveness of the proposed signature learning model was evaluated for various parameters (such as the signature size, classification algorithms, local signatures/global signatures) and was also validated with a number of classification algorithms (i.e., Naïve Bayes, Decision Trees, and Random Forests) using 20 newsgroup dataset. Based on these observations, we identify that SigSpace-Text outperforms state-of-the-art performance results on the dataset.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled “SigSpace-Text: Parallel and Distributed Signature Learning in Text Analytics” presented by Rakesh Reddy Bandi, candidate for the Master of Science degree, and hereby certify that in their opinion, it is worthy of acceptance.

Supervisory Committee

Yugyung Lee, Ph.D., Committee Chair
Department of Computer Science Electrical Engineering

Sejun Song, Ph.D.
Department of Computer Science Electrical Engineering

Yongjie Zheng, Ph.D.
Department of Computer Science Electrical Engineering

TABLE OF CONTENTS

ILLUSTRATIONS

Figure Page

ACKNOWLEDGEMENTS

I would like to thank Dr. Yugyung Lee for the constant and endearing support which has helped me in fulfilling my thesis. She has provided me with an opportunity to realize my potential in the field of data science. Her encouragement and inputs were elements of vital guidance in my thesis. She has been a constant source of motivation and challenged me with deadlines, that have contributed to me acquiring inspiration and ideology. Her expertise and innovative insights have been phenomenal in completing my thesis.

I would like to thank the University of Missouri- Kansas City for providing me with an opportunity to continue my research and supporting me in this regard. The computer labs in the university had provided me with fantastic opportunities to pursue my research. Special thanks to fellow students, in particularly Mayanka, Sudhakar who helped and encouraged me in during each module of my time in the lab.

Finally, I would like to thank my family and friends who have been supportive and have helped me complete my thesis, without whom this accomplishment would not have been possible.

CHAPTER 1

INTRODUCTION

1.1 Motivation

We are in the phase of technology capable of building machines that can perform cognitive computing. There are many such cognitive services: Microsoft's Oxford [18], Baidu's Minwa, IBM's Watson [19] and Boston Dynamics' robots [20]. Although each agent works mostly on Artificial Intelligence [26], each machine is designed to achieve specific goals like Natural Language processing, accurate Image recognition etc., with less or no human involvement. In all these machines, the very common process is the ability to extract information from what humans can perceive - interpret, and cluster them with unsupervised learning, so that machines can also understand and interact using human understandable content. Advantages are immense and the application possibilities are enormous.

Datasets which are obtained from images and audio systems are encoded in the form of vectors, for both the audio and images. Vectors would possess information of the pixel intensities in case of images, and the density coefficients for the power spectrum are handled in case of audio data. These datasets have a rich set of data and huge dimensionality to enhance the information lying within. The encoding of data would contain information required for performing the tasks such as recognition of objects or patterns pertaining to images and audio. In the case of natural language processing systems, there are identifiers to describe the different words but it is difficult to bring about a relationship. Hence, it would be more complex to identify patterns and objects, in text data.

TF-IDF [32] and Word2Vec [2] is used to represent the text data in vector space. TF-IDF is used to get the most significant words and this can be represented in vector space using the Word2Vec model built on the text corpus. SigSpace-text model is used to identify objects in the text data using the vector representation of text data and these objects would consist of related words and data patterns, which are similar to image and audio data objects.

SigSpace model [5] is developed using SOM clustering algorithm which is aimed at generating feature representation of input data known as signatures. Signatures being smaller in size, carry essential information only and then can be used as input for further stages like building knowledge graphs or ontology models. In this thesis we have conducted experiments to evaluate the applicability of SigSpace on text domain by training the classification models with signatures generated on 20 news groups data and classify the test documents to evaluate the accuracy of the training models.

1.2 Problem Statements

Machine learning algorithms [27] perform better when inclined with an increased number of features, is a fallacy. There have been many constraints in considering the “Curse of Dimensionality”, which is an approach dealing with the performance of a model based on the exponential increase of features and dimensions. Big data is a trendsetter in the field of technology. It is radical that an update in the existing algorithmic implementation of classification and regression occur. This is mainly because they work well under given data, but with dynamism and high speeds of incoming data, the time taken in building a model should also be considered.

Some of the key observations can be considered to be the modules of:

1. Independent learning
2. Distributed learning
3. Lightweight models
4. Incremental learning.

Each of these fields have been crucial in the field of data science. The process of independent learning usually incurs the principles of bag of words, classification and PCA to cluster data. In the training phase, the class features are in the process of continuous comparison and contrast. Technological improvements have led to an increase in distributed computing, where there are many nodes for performance of learning and recognition. Algorithms must be framed to support the distribution and transfer of chunks of data modules, in order to construct a single model.

The size of machines has come down significantly over a time period, and it is necessary to implement the machine learning algorithms to the smaller sized device, such as smartphones. Algorithms to handle and process this scalability with magnanimous speeds is a key requirement. The process of Fuzzy classification is aimed at obtaining the label that is predicted at a relatively high frequency, provide a test data point. The shallow machine learning algorithms do not consider the prediction of multiple labels based upon the confidence score. The data operations are executed again, after learning the data points, when a model is to be trained. This can lead to a high complexity. Online learning can be considered

to be an implementation of the incremental learning process that would consider each step in framing the algorithmic techniques.

1.3 Summary

The rest of the paper contents are organized as following; background and summary of related work is discussed in chapter 2. We have presented the concept and model of SigSpace that will be used for text document classification model in chapter 3. And in chapter 4 we have discussed about the implementation and different datasets used for experimenting. And in chapter 5, we have presented a series of evaluation results conducted on different models. Finally, in chapter 6, we have concluded the applicability of signatures to the text data and documented the future scope.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Introduction

This chapter provides the relatively substantial back ground information and introduces the key terms related to Machine Learning and defines their interpretations in the context.

2.2 Machine Learning

Machine learning [27] can be inferred to be that specific subsection of computer science which deals with artificial intelligence. It provides the computers with the ability to take decisions without being programmer in an explicit fashion. This has been evolved as a result of the study bound within the frameworks of pattern recognition and the computational learning theory, that are based on the domain of artificial intelligence.

This is a field which deals with exploration of the study and constructing algorithms which can predict data and make decision. This is modeled around building a model based upon inputs, which can be termed to be sample data. Machine learning is an important concept that can be implemented in various computational tasks, such as design and program of algorithms. In the field of data analytics, the machine learning model provides techniques to devise models with complexity and predict the algorithmic values and implementation schemes.

Based on grouping of algorithms by the learning style we have three different learning styles in machine learning algorithms. They are:

a. Supervised Learning

b. Unsupervised Learning

c. Semi-Supervised Learning

Figure 1 shows the difference between supervised learning and unsupervised learning where former is performed on labelled data and later is performed on unlabeled data.

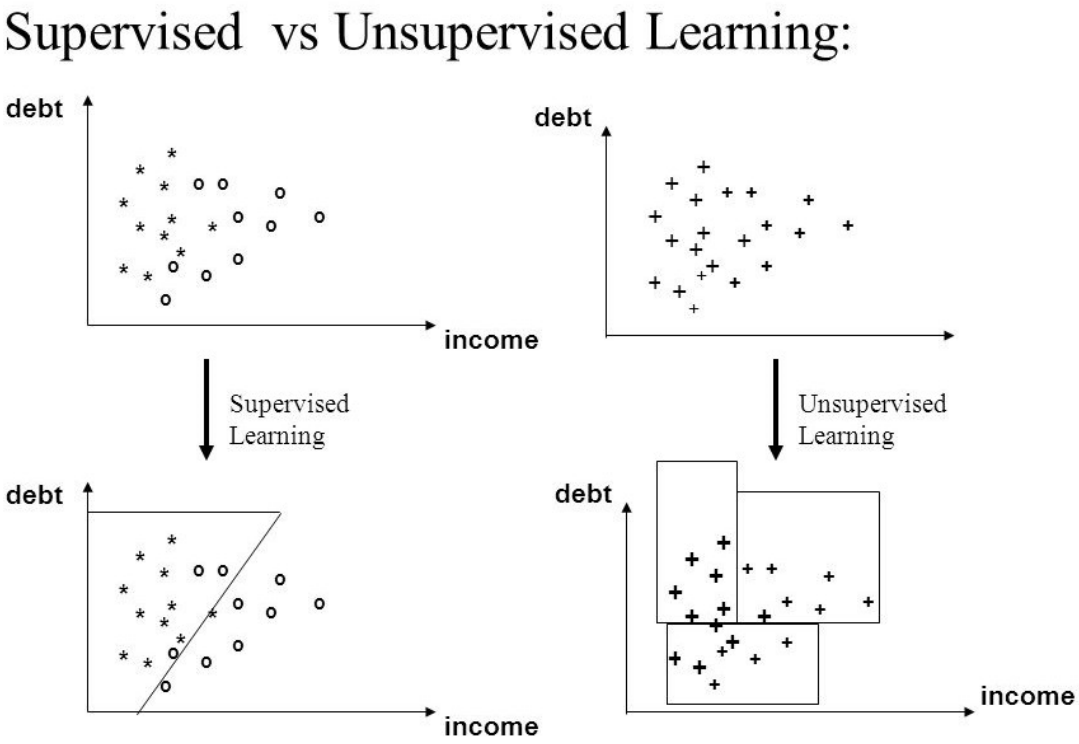


Figure 1: Difference between Supervised and Unsupervised Learning [28]

2.2.1 Supervised Learning

Supervised learning is another branch of machine learning where the inference of the function is from training data that is labeled. The training data would mainly compose of training examples. The most important components are vector (for input objects) and supervisory signal (the output value). This is the algorithmic categories that analyze the training data to

provide a function, which can determine the class labels. One of the branches, such as concept learning is a parallel task.

2.2.2 Unsupervised Learning

Unsupervised learning is that specific category of machine learning, where the tasks included are the inference of functions to get a description of the structure from data that is not labeled. The main focus of the unsupervised learning would be to solve the problem of density estimation in statistics. There are other key requirements also such as summarization and explanation of the data features. Some of the unsupervised learning techniques such as K-means and Gaussian mixture models (GMM) which aims to cluster the n-observations into k-clusters in which each observation belongs to the nearest mean cluster. This results in the partitioning of data that is not labeled into similar and related groups.

2.2.3 Semi-Supervised Learning

Semi-supervised learning [46] is a specific category in which input data is a combination of both labeled and unlabeled. Generally training data is composed of less observations of labeled data with more observations of unlabeled data. There is considerable amount of increase in learning accuracy when unlabeled data is used along with labeled data. This type of learning is used when collection of fully labeled data is infeasible. Examples algorithms are classification and regression.

2.3 Features

Features are the root and individual measurable field of an event being observed in any machine learning tasks. The main step for any machine learning algorithms such as clustering and classification is choosing the informative, independent and distinctive features. As part of

thesis, we have worked on 5 kinds of features, three belonging to the text domain and two from image domain.

2.3.1 Word2Vec

Word2Vec [2] can be approached as a group of models that are interrelated and produce embodies of words. The models are not in depth, and are a dual layer of neural networks. They are trained to perform a reconstruction of the words in order to obtain a linguistic context. The input taken by the Word2Vec model would be a large text corpus, that necessitates the assignment of word to a vector in the space. The word vectors are located such that the context is shared in common as well, as close proximity in space. Figure 2 shows the two different Word2Vec architecture models Continuous Bag of Words model(CBOW) and continuous Skip-gram model.

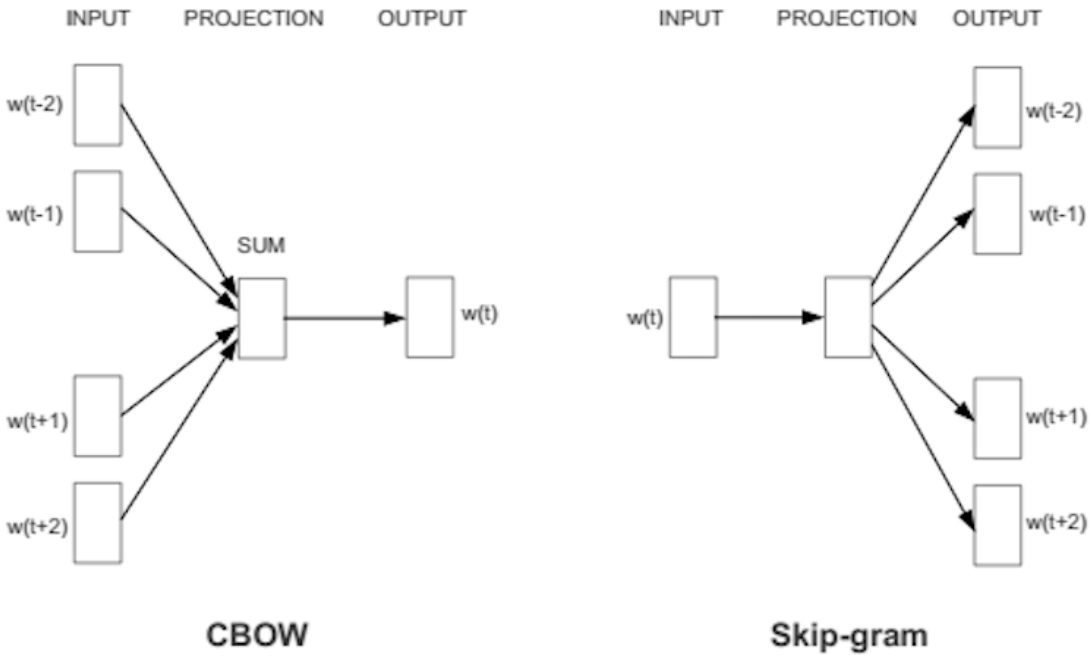


Figure 2: Word2Vec architectures CBOW and Skip-gram

2.3.2 TF-IDF

This TF-IDF [32] is a technique used specifically in the retrieval of information. It can also be summarized as a statistic of numeric measures that would most certainly identify the reflection of a word's importance in a document or a collection, termed as a corpus. The most important use of tf-idf is to calculate the weighting factor when it comes to mining text or retrieve information. This is an increase in proportionality of the number of times a word would appear in a specific document.

However, it is important to consider the frequency of all words since there might be multiple occurrences of the general terms. It would be necessary to accommodate that. In such cases, the stop words filtering mechanism is implemented. It would ensure that the text summarization and classification could occur based upon these rules. The main usage of tf-idf is in search engines where the score and rank pertaining to a document are analyzed thereby bringing the relevance of the document to a user's query to the fore. Figure 3 shows the formula to calculate the tf-idf of word.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

Figure 3: TF-IDF of word $w_{i,j}$ [30]

2.3.3 LDA

Latent Drichlet Allocation(LDA) is one of the main aspects of natural language processing where a statistical model is computed based on the observations which delve into the similarity of the parts of data. It is a model where the topics of the data can be fetched. The position of words in a document is attributed to the topic and the respective context.

In LDA, the document can be interpreted to be a combination of several topics. It is more or less similar to a latent semantic analysis, but there exists a Drichlet prior which would help sustain the distribution of the topic in the document. The main assumption over here is that a document can be characteristic of a set of topics, pretty much relevant to the bag of words assumption, existing in natural language processing terminology as well as machine learning algorithms, to infer the dictionary as well as context of words. Figure 4 shows the Plate notation representing the LDA model where the α and β are the parameters of the Dirichlet prior on the per-document topic distributions and per-topic word distributions respectively.

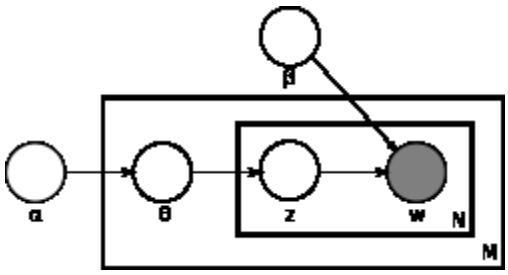


Figure 4: Plate notation representing the LDA model

2.4 Clustering

The main impact of the clustering algorithms is data binding. In this aspect, the set of data points are analyzed and based on similarity they are bound to a certain cluster, if the contrary then they are termed to be dissimilar. They are grouped based on certain constraints. One of

the main specifics of clustering is to have a foray into the field of data mining. It is a very common technique when it comes to analysis of statistical data. The usage has been expanded to several fields such as pattern recognition, retrieval of information, machine learning as well as data compression.

Clustering can be conceived to be one of the important unsupervised learning problems, as it most certainly deals with a collection of data which is unlabeled. The main focus would be on grouping members that are similar.

Some of the key examples that can be listed include:

1. K-means- exclusive clustering algorithm
2. Fuzzy C-means- overlapping clustering algorithm
3. Hierarchical clustering- mix of Gaussian – Probabilistic clustering algorithm

2.4.1 K-Means

This specific algorithm is a simple unsupervised learning method of forming clusters based on the given data. After fixing a priori, the given data set would be classified based upon the clusters created. The key aspect over here is that, k centroids get defined, where each cluster has one. The location of this is dependent upon the data surrounding it. The observations are partitioned and the nearest mean is used to put the data item into that cluster. It is a difficult problem to compute, but it is more or less equivalent to maximizing the expected output based on the mixtures of Gaussian distribution. The algorithm is an expansion of the k-nearest neighbor classifier techniques.

2.4.2 E-M Clustering

This clustering method involves finding the maximum probability or rather maximize the probability of finding parameters involves in a statistical model. This model would however be dependent upon the latent variables, whose behavior may have not been observed. The EM iteration would alternate between performance of an expectation(E) step and a maximization(M) step. The E step would involve creating a function to obtain the expectation of the probability evaluated using the parametric values at hand. The M step would compute the parameters which would maximize the values obtained in the first step. The obtained estimates for each of the parameters are then combined to obtain a determination of the distribution properties the latent variables possess.

2.4.3 SOM Clustering

A self-organizing map (SOM) can be explained to be a type of artificial neural network which would be trained under the basis of unsupervised learning. It would produce a 2D representation of the input set of training data, that is called as a map. SOMs apply the techniques of competitive learning and use a function in order to contain the topological properties inclined to an input space. SOMs are useful in performing visualization of high dimensional data in a low dimensional view. This is pretty much similar to the process of multidimensional scaling. The SOM can be used in order to cluster the input data and then perform feature extraction on it. Figure 5 shows the input vector represented on the 2D lattice map.

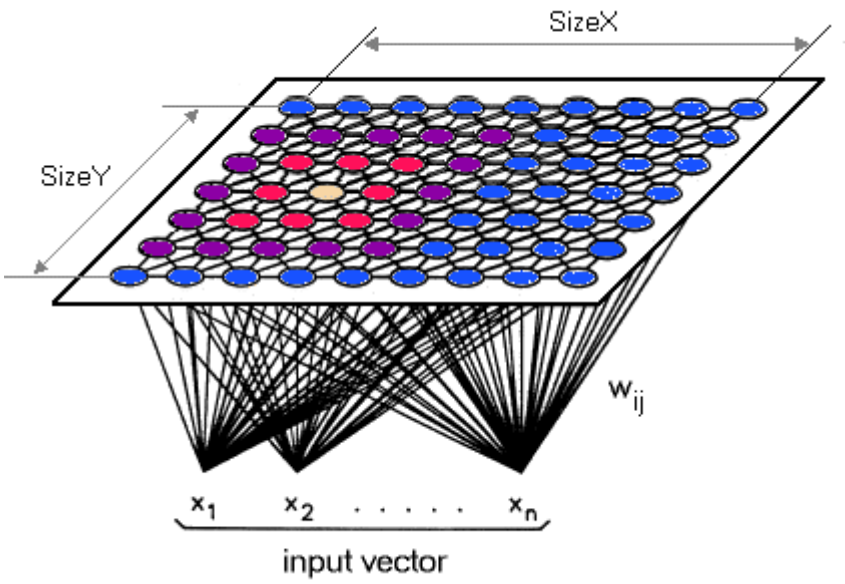


Figure 5: SOM Map [29]

The main advantages of SOM clustering includes the ability of it being easy to understand. The interpretation of data mapping is less complex and there is a strong potential capability of organizing voluminous complex data sets. SOM clustering works well with data that has high dimensionality.

However, there are certain drawbacks as the process is expensive to compute. The input weights to be used are difficult to determine. Each of the SOM would vary from the other, which is another disadvantage.

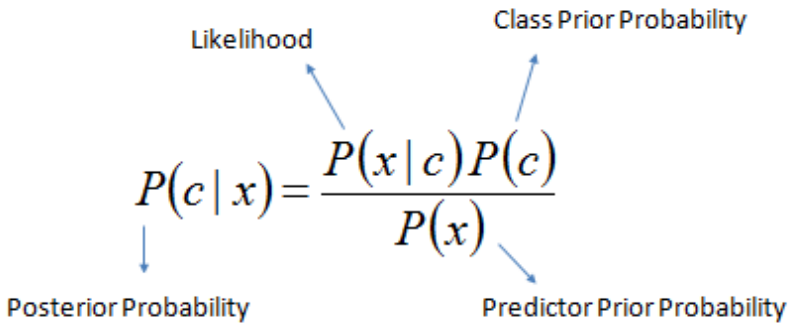
2.5 Classification Algorithms

2.5.1 Naïve Bayes

In the context of machine learning, Naïve Bayes classification [41] is a family of probability classifiers that would apply Bayes' theorem with independent assumptions in

regard to the features. They are scalable and would basically require many parameters such as features and predictors in a learning problem.

The maximum likelihood is obtained by performing evaluation upon the closed form expression, which can be considered to take linear time. It is a basic method for construction of classifiers and models which would assign labels of class to the instances of the problems. A key advantage that can be obtained from this method is that the number of training data is relatively less and parameters necessary for classification are reduced. Figure 6 shows the calculation of posterior probability of event.



$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 6: Naïve Bayes Formula

2.5.2 Decision Trees

A decision tree [16] is used to calculate decisions having a set of outcomes. This uses a graph model which is based upon how trees work. There are outcomes which can demonstrate how the event may occur. The algorithm would also consist of parameters dealing with the cost of resources used and the chance of an event occurring. They are used in fields pertaining

to machine learning in order to perform operations of decision analysis, which would help in achieving a specific goal.

2.5.3 Random Forest

Random forests [17] are learning methods which can been used in order to perform those tasks such as classification and regression. It operates by construction of a multitude of decision trees which exists at training time and performs the output of the class, which can be the most existing frequency when it comes to the classes. This is the mode of classification. There can also be the mean prediction or regression of each of the trees. Random decision forests usually overtake the habit of fitting to the respective training set. The key steps involved are:

1. Decision Tree learning: Decision tree learning uses the decision trees for mapping the observations of an item to its target value. Decision trees are used take the decisions for the new test observations.
2. Tree Bagging: The training of random forest applies to aggregating the all tree learners. After training, predictions for unseen samples can be made by averaging the predictions from all the separate regression trees on the unobserved sample.
3. Bagging to random forests: This is also called as feature bagging which is randomly selecting the subset of features and training the trees with those selected features. For example in general classification of f features observations, square root of f features are used in every split.
4. Extra Trees: Instead of selecting the locally optimal split combination, a random value is considered to split which is selected from the range of empirical feature range.

The properties that can be exhibited are the importance of variables as well as the relationship to the nearest neighbors. Figure 7 shows the aggregation of n decision trees to form an instance of Random Forest.

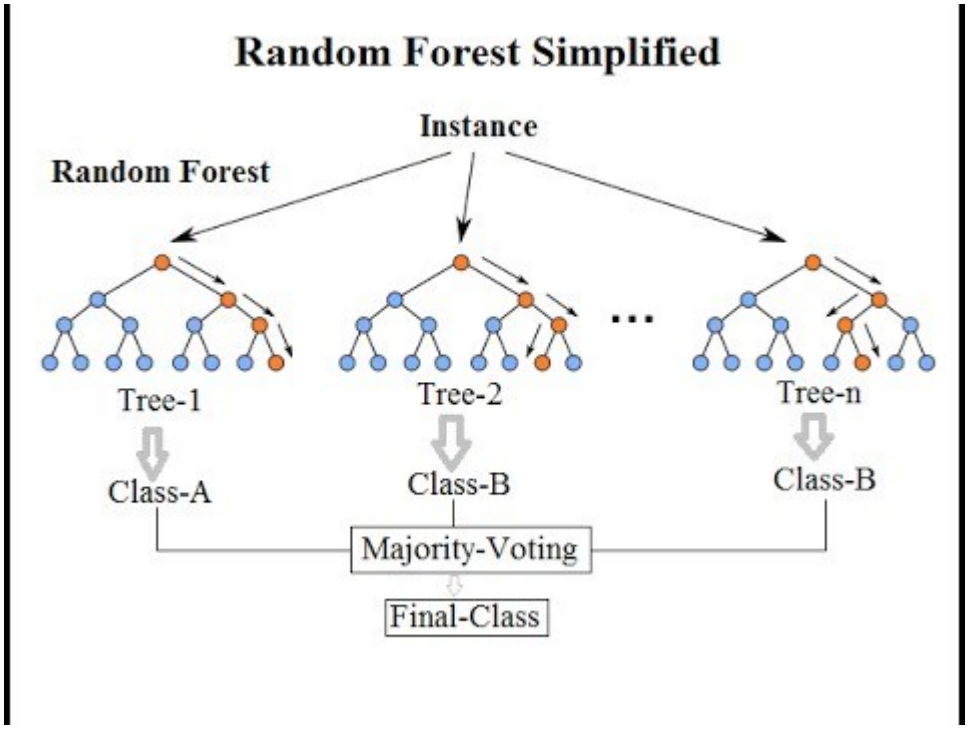


Figure 7: Random Forest Example [31]

2.6 Related Work

Document analysis for classification purpose is to find the document content using the most significant terms by the process of TF-IDF. This approach gives a simple way of numeric representation and many have used TF-IDF to represent the text into vector form and perform computations in order to classify the documents. Only disadvantage with this approach is that it requires whole data to be given at one single time in order to compute correct term weights. Also it cannot support individual class learning and hence incremental learning is not possible. Figure 8 shows the comparison of Sigspace-Text and text document clustering using Bag of Words approach on below factors.

	[1] Text Document Clustering(BoW)	SigSpace - Text
TF-IDF	✓	✓
Word2Vec		✓
Clustering	✓	✓
Data Reduction		✓
Ease of Model Update		✓
Distributed learning		✓
Independent learning		✓
Incremental learning		✓

Figure 8: Comparison of SigSpace-Text and Text Document Clustering on Several Factors

Juha Vesanto and Esa Alhoniemi presented a paper titled, 'Clustering of the Self-Organizing Map' [1] and it discusses about the contribution of the SOM in data mining. The input space has been projected in a way such that the visualization and properties of data have been explained. The main prospective approach of the authors was to find the clustering of the

SOM when the number of units are large and it was important to provide facilities to include the quantitative analysis of the data existing within the map. The use of hierarchical agglomerative clustering and partitive clustering by implementing the k-means algorithm was checked. This was a two stage process, where the first one was to use SOM to produce the prototypes which are again clustered in the next stage. The conclusions obtained were that the two stage process yielded a better performance compared to the direct clustering. Moreover, the computation time was reduced.

The earlier work done by Vesanto and Alhoniemi [1] showed that SOM was an effective platform to perform the visualization for data which would have high dimensions. The drawback was that the contents of a data set were not able to be perceived. The cluster structure was significant and it required the extraction of the data from these clusters in order to produce the information based on the summary. This specific paper evaluated the mechanisms of data abstraction in context with the creation of SOM that was used in the data clustering.

The clustering results revealed through the experiments found out that the intermediate steps were compared with the data based results.

Matharage, Ganegedara and Alahakoon have scientific observations [15] in the regard of the Self-Organizing Map has explained about how the text data has been clustered. The main context of their findings is that the volume is really large. The techniques which are used most in text mining are Self Organizing Map(SOM) and Growing SOMs. The basic ideology is that the intensity of the processor is a requisite for data mining to occur on large sets of text data. Fast GSOM was considered to be an improvement to the GSOM as it would cluster the text data in

a more efficient manner. However, the huge sizes of text corpuses can make it take a lot of time as there is a constraint of turnaround time during the analytic phase. Only disadvantage with this approach is that it requires whole data to be given at one single time in order to compute correct term weights. Also it cannot support individual class learning and hence incremental learning is not possible.

The proposition by Sumith, Hiran and Damminda is a scalable algorithm which would use distributed and parallel computing. The output was that the proposal yielded similar or rather a better phase of accuracy as and when compared to GSOM. Sammon's projection may not have a lattice structure but the map which is produced as a result would still satisfy most of the data analytic features of SOM. The projection would provide a platform for visualization of the dataset to describe topological framework. The Vornoi diagram highlights the inter node distance in order to inspect the boundary of neurons. The algorithm proposed also provided the scalable FastGSOM algorithm to have a better accuracy.

The future work would include the partitioning of data into the required number as merging can still be taking some time. Moreover, the key consideration would be to utilize the optimal partitions for parallel processing.

The paper "Classification of Documents Using Kohonen's Self-Organizing Map" [34] discusses about the prevalence of methods that are efficient and easy for the user to retrieve information which would be based upon text that is readily made available on the World Wide Web. The most widely used algorithm pertaining to neural networks is the self-organizing map (SOM). The paper explored the visualization of the Self Organizing Map and classification of the

documents comprised of text. The capabilities of SOM in the classification of text discussed. The dataset used was the 20 news group. Documents are classifies based on the learning done in either supervised or unsupervised. The SOM was used as a methodology for clustering the vectors obtained as input from the documents which had been processed. Only disadvantage with this approach is that it requires whole data to be given at one single time in order to compute correct term weights. Additionally, it is not distributive and also it cannot support individual class learning and hence incremental learning is not possible.

The paper "Representation and Classification of Text Documents: A Brief Review" [33] explained about text classification and the approaches taken towards classifying documents. Text mining has been discussed where the classification of documents takes places based upon supervised sets of knowledge. Categorization was done on text documents, in this case literature. There were different schemes used to represent texts and the classification was compared depending on the classes that were predefined. All the approaches discussed in this review paper requires whole data to be given at one single time in order to compute correct term weights. Also it cannot support individual class learning and hence incremental learning is not possible.

The paper "Exploiting Wikipedia as External Knowledge for Document Clustering" [35] discusses about the clustering of text documents with consideration of semantic information of every text document rather than traditional text document clustering which have been discussed as a representation where documents were considered to be bags of words that were drafted without taking into account the semantic knowledge of each document. The main basis that has been used here is the collection of words for documents where there may be

core words which are shared, in different forms or even having the same meaning. The methodology used in this format is creating knowledge graphs and form an ontology. There were two major issues discussed: one being the limited scope for ontologies and the other being loss of information, or introduction of noise parameters. The paper looked at solving these issues by using the concept of Wikipedia. The documents were categorized based upon the match of text documents by mapping them to Wikipedia categorizes as well as concepts. There were three different datasets used that measured the similarity of clustering.

"Generalized Independence Subspace Clustering" [37] - Data had been used to wrap up all the grouping of objects into spaces designed for the dimensionality and orientation of arbitrary datasets. The subspace clustering was the main phenomenon discussed which included finding these spaces and the respective groups designed. The paper presented a method that was used to find multiple clustering of data that is not repeated. The technique of Independent Subspace Analysis (ISA) was used to search the collection of subspace which would decrease the probability of dependency between the clusters. They were then clustered into the subspaces. The algorithm used the principle of minimum description length to select attributes which were not easy to set. The demonstration was done on both synthetic as well as real data.

"L-EnsNMF: Boosted Local Topic Discovery via Ensemble of Nonnegative Matrix Factorization" [38] - The process of nonnegative matrix factorization(NMF) has been used in applications pertaining to modeling the topics, which involved a set of topics that are obtained through the matrix based on the rank. the main drawback which usually exists is that there is not much information conveyed through this. The method proposed by the author was to

devise a model to discover topics that are based on local topics and have information of high quality. This model helps in transformation of a contextual supervised learning set into unsupervised learning. There is a sequence of topics that are generated through the model. The input is updates based on utilization of residual matrix and the weights of local data has been used. The evaluation was done by comparing with the LDA and different forms of NMF.

The paper "Self-Organizing Maps in Document Classification: A Comparison with Six Machine Learning Methods" [36] focused on the usage of self-organizing maps for classifying the text documents, and are known as Kohonen maps. The main motive of the paper was to separate and classify documents into classes which are based on the categories. The classification was done based on six techniques: k-means clustering, k nearest neighbor search, Ward's clustering method, Naïve Bayes classification algorithm, classification tree, discriminant analysis. There were three different data sets. The SOM was tested to find out the high accuracy of methods which are unsupervised, in the collection of Reuter news. The comparison was made against the methods that were supervised. The key concepts discussed included implementing concepts based on machine learning and neural networks.

The proposal of this paper "Efficient Estimation of Word Representations in Vector Space" [39] was to develop architectural models, which would help in continual vector computation of representing words from data sets that are large in size. The qualitative analysis is done based upon the similarity of words and the results are on comparison with the techniques that were used previously on the various types of neural networks which are made available. There was an observation made where the accuracy could be improved at a faster pace with respect to a decrease in the amount of computational cost. An example which was

cited showcased the learning of word vectors in a period of less than a day, from a dataset as large as 1.6 billion. These vectors helped in the measurement of syntax as well as semantics of words based on similarity.

"A Fused Multi-Feature Based Co-Training Approach for Document Clustering" [40]: Information retrieval and mining of data are topics that have been considered for the clustering of documents. Most of the models and methods are based on computation of similarity index between two documents which are used to frame models in a space, or models which have been applied techniques on, from a corpus. The paper considered these specific concepts as clustering in domains based on term features and semantic features. The approach proposed was for clustering of spectra based on training both models. The experiments conducted have shown an efficiency and feasible constraints on comparison with the already existing methods. The main features that were used in the approach included the TFIDF and LSI. The word property and the properties based on semantics have been considered respectively. The given document corpus was performed the operations on. Only disadvantage with this approach is that it requires whole data to be given at one single time in order to compute correct term weights. Also it cannot support individual class learning and hence incremental learning is not possible. The other features are the kernel addition, kernel product, clustering techniques involved were k-means and agglomerative, and concatenation of features.

2.7 Summary

Classification in the fields of text analysis is on the forefront as it has many applications in real time. It can be improved from simply using text features data to classify the incoming data to machine performing highly complex tasks.

Machine learning decisions have moved from high processing machine to commodity hardware and then to smart phones which are low processing devices. The decisions should be taken in a very less time, hence we propose the distributive, independent, incremental and scalable architecture i.e. SigSpace architecture to support the properties mentioned above and validate its performance with respective to the other machine learning algorithms. Even though the experiments are done on using Word2Vec and TF-IDF features for text and 3 kinds of clustering algorithms, we leave the other evaluations for the future work to get a detailed report.

CHAPTER 3

PROPOSED SOLUTION

3.1 Introduction

In the beginning of this chapter, we will discuss SigSpace architecture for text data and the SigSpace models formed using different number of significant words of each text category. We will also discuss how we trained the different classification models using the signatures formed from the text data and fuzzy matching works using these signatures.

The architecture of this system contains the following components: Text dataset, Word2vec models, SOM, Training models, classifiers. Figure 9 shows the Sigspace- Text Architecture for the Text Domain.

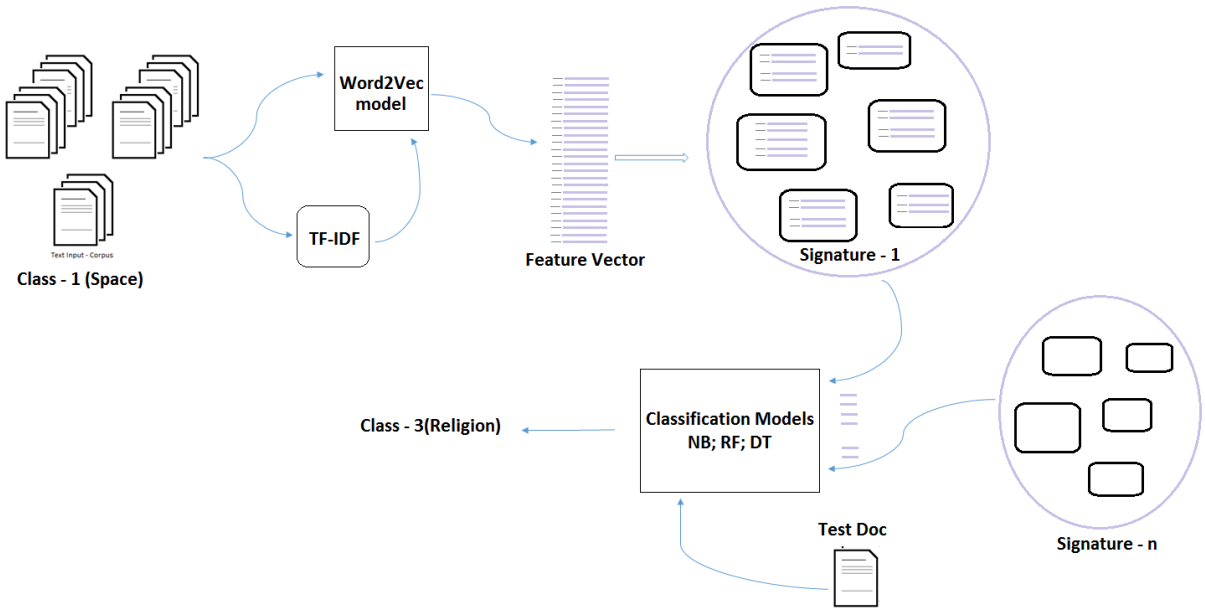


Figure 9: SigSpace-Text Architecture for Text Domain

3.2 SigSpace-Text Model

SigSpace-Text architecture is the heart of the system. It is a novel approach of representing common features of data in its condensed manner. Whole text data is divided in to multiple classes based on the individual object representation and then text corpus from each class is represented in its vector form by using word2vec model and then similar data is clustered using any clustering algorithms like K-means [45] or Self Organizing Maps (SOM) [44] and then summarizing the grouped data will produce signature of the grouped similar data.

The advantages of representing data in the form of signature gives us the following advantages:

- Data size reduction is one of the most prominent advantage, in our experiments, one class contained over 8000 observations. This data gets reduced to almost 3000 observations which is significant reduction in data size.
- Time complexity reduction is another advantage. Since the data size is very less the time required to train the model is very less.
- Distributed learning is achieved by using class signature. Classes are independent of each other, so there is no need of having input data all at one place.
- Incremental learning is also possible where in whole data is not required to be given at once.

The data reduced is then given for training the machine learning models. To test the applicability of this signature approach we have conducted experiments to classify the test documents to find which category they are classified to and evaluated the predictions to obtain the overall results.

The architecture proposed is shown in the figure. The SigSpace-Text model is divided into two phases. The first phase discusses about the feature extraction and the representation of text data in vector space and significant words selection. Second phase discusses about the Signature learning.

3.3 Phase-1: Feature Extraction

In this phase we will discuss about the representation of text data in vector space and significant words selection. The below figure shows the overall procedure to extract the significant words using TF-IDF and represent those words in vector space using Word2Vec models of each class. Figure 10 shows the workflow of feature extraction of text data.

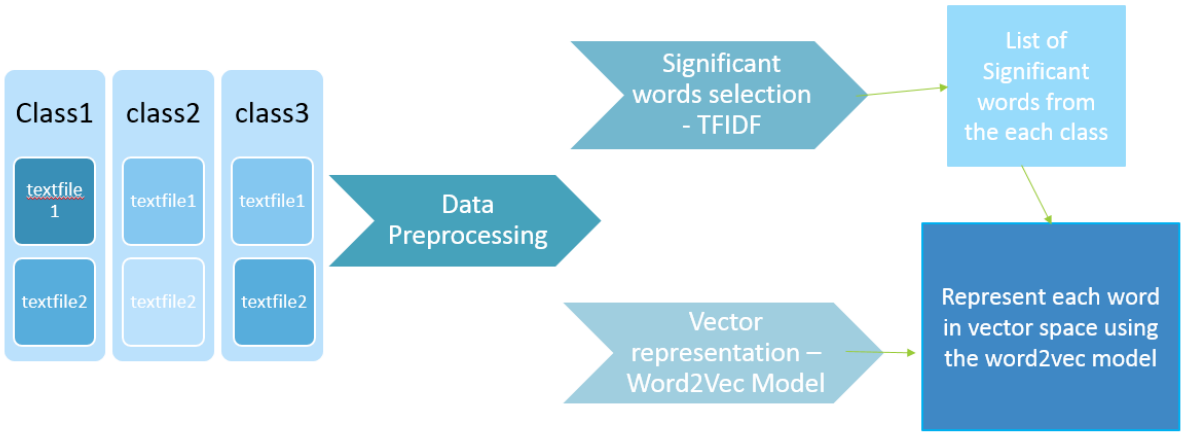


Figure 10: Vector Representation of Text Data-Feature Extraction

Data preprocessing

The training data is preprocessed using lemmatization to replace the root form of word. Also the stop words are removed and coreNLP is used to represent the data in proper format. Below are the preprocessing steps and are discussed individually. Figure 11 shows the preprocessing of text data.

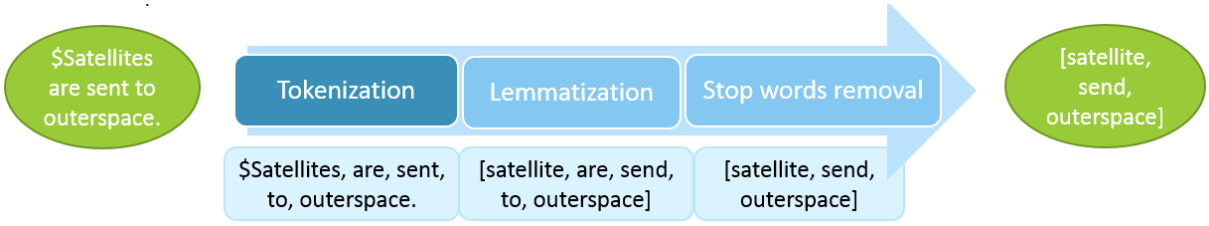


Figure 11: Preprocessing of Text Data

1. Tokenization

   The process of decomposing streams of text into blocks of words, symbols, phrasal combinations, or other elements which provide a contextual meaning, are called tokens. The tokens serve as input for further processes such as parsing of data and mining of text.

   Example: Original Sentence - $Satellites are sent to outerspace.

   After tokenization - $Satellites, are, sent, to, outerspace

2. Lemmatization

   This involves the conversion of a word to its base form and also the conversion of case to lower case alphabets. The special characters are also removed. ($, [], ., & etc.,)

   Example: Input - $Satellites, are, sent, to, outerspace

   After lemmatization - [satellite, are, send, to, outerspace]

3. Stop words removal

   Removal of common terms is stop words removal. The stop words are the common words which would appear to be of little value in helping to select documents matching a user's need and are excluded from the vocabulary entirely.

Example: Input - [satellite, are, send, to, outerspace]

After stop words removal - [satellite, send, outerspace]

Word2Vec Model building

The training data after preprocessing is projected into a vector space using word2vec model. Word2Vec is a neural network based language model that represents words in terms of feature vectors. It is trained to perform a reconstruction of the words in order to obtain a linguistic context. Figure 12 shows word2vec model built on the input document.

Input: The input taken by the Word2Vec model would be a large text corpus, that necessitates the assignment of word to a vector in the space.

Output: Representation of word in n- dimensional vector space.

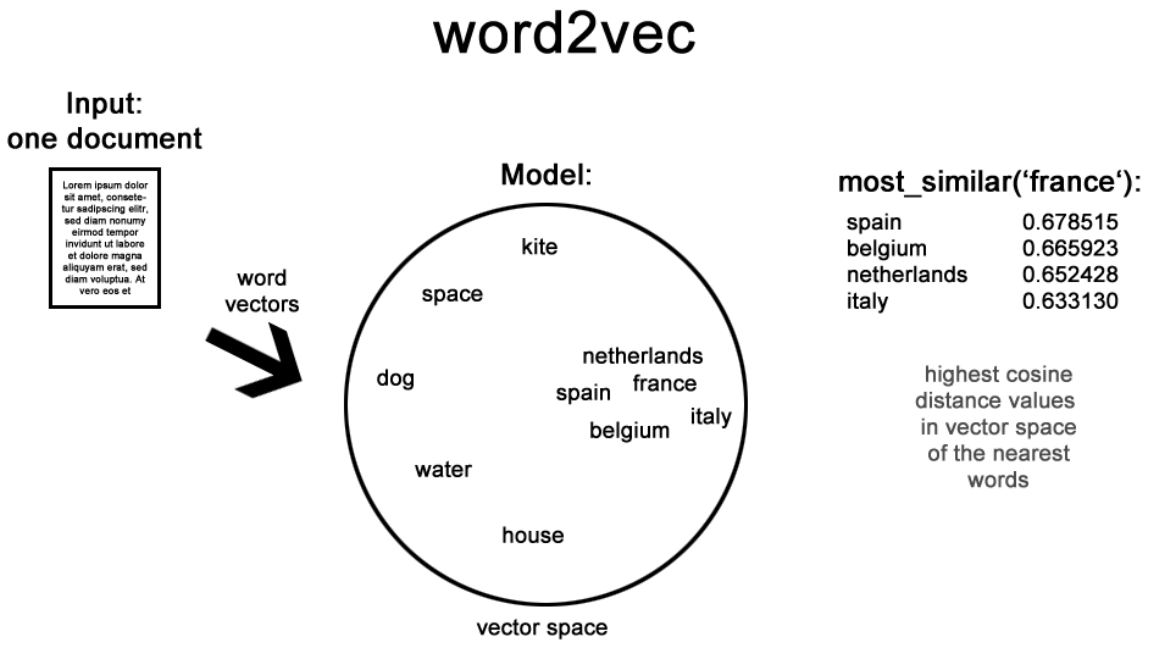


Figure 12: Word2Vec Model

Word2vec model is built using the entire corpus and each word in the text is included in the word2vec vocabulary. The word is represented in terms of its vector space resembles the point with respect to every other word from the vocabulary in the same vector dimension.

Below are some of the reasons for using word2vec model over other feature representation models.

- Word2vec model is the simplest and easiest form of representing text data in terms of vector form.
- Other models like LDA depends on the probabilistic distribution and may give different values each time it is executed.
- Word2vec model allows incremental learning. In our case, incremental learning is achieved by updating the word2vec model vocabulary with the incoming data.

For classification purpose, the test document has to be classified and the vector representation of test document words using the same word2vec model gives the word position from the same vector space.

Significant word selection – TFIDF

Term frequency – Inverse document frequency is a

Statistical numeration of the importance a word possesses in a document and a clear reflection of its significance in a corpus. The TF-IDF values is directly proportional to the occurrence of word in a document, which is however offset by the rate of recurrence the word would yield in the corpus. This is an indication of the repetition the word may have in a document and is not considered to have a high frequency. Significant words are the very relevant and high frequency words in the text document. These significant words alone add more weight to each

group of documents i.e., the most significant words for each class of text documents are selected from the high tf-idf value words of the entire class text corpus. TF-IDF value is calculated using below formula. $W_{i,j}$ is the weight of word i in the document j. Figure 14 shows the sample TF-IDF feature vector. Figure 13 shows the collection significant words using TF-IDF.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

TOP TF IDF Words

(Space,0.6931471805599453)
(Rocket,0.6931471805599453)
(moon,0.6931471805599453)
(human,0.6931471805599453)
(planet,0.6931471805599453)
(earth,0.6931471805599453)
(neat,0.6931471805599453)

TF IDF FEATURE VECTOR

(1048576,[73,2337,336781,393917,585782],[0.28768207245178085,0.6931471805599453,0.6931471805599453,0.6931471805599453,0.6931471805599453])

(1048576,[96852,231466,491585,748138,906880],[0.6931471805599453,0.6931471805599453,0.6931471805599453,0.6931471805599453,0.6931471805599453])

(1048576,[73,79910,116103,204354,479425,503975,949040],[0.28768207245178085,0.6931471805599453,0.6931471805599453,0.6931471805599453,0.6931471805599453,0.6931471805599453,0.6931471805599453])

Figure 13: Significant words - TFIDF

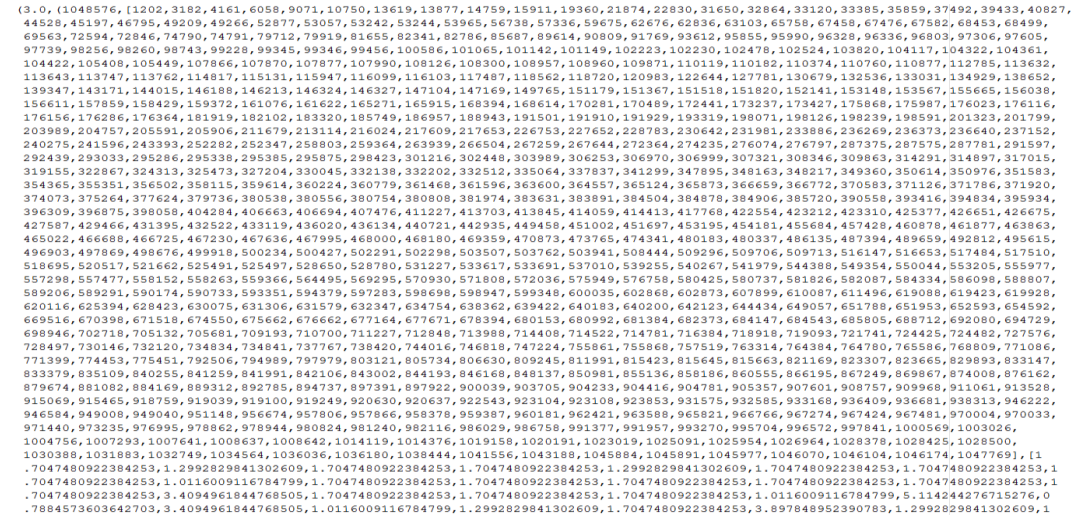


Figure 14: TFIDF Feature Vector

3.4 Phase-2: Signature Learning

Signature learning is based on class level approach. A generic pattern is identified for a given object using the existing clustering algorithms i.e. GMM, K-Means and Self Organizing Maps. Clustering is primarily used when there are groups of items having similarity, and are effective in capturing the centric distributive tendency. The clustering of features come together to form a more effective summary for specific group of elements, which would comprise the ”SigSpace”. The following are the some of the SigSpace-Text objectives:

1. Independent Learning: Signature learning provides the class to class independence while learning. Signature learning does not need all the data at one place to build the model.
2. Incremental Learning: SigSpaces will use incremental clustering algorithms: K-means and SOM for learning data.

3. Distributed Learning: The SigSpace architecture is implanted on a platform of distributive big data processing tool, as classes are not dependent on its peers, and has the possibility of generation and learning the SigSpaces, in distributive nature.
4. Feature space reduction: The SigSpace is a condensation of data space, whose occupancy would negate to less than 30% of the data space.

Algorithm: Signature Learning

Step 1: Read the text features of text corpus

Step 2: For each class of text features

Signature-Generation-Algorithm()

1. Cluster-model-train with Cluster(K).fit(text features of class)
2. Find the Cluster-centers of Cluster-Model-trained
3. Represent Signature=(label, Cluster Center)
4. return Signature

Step 3: Each processor finds the signature of each class and save the signatures

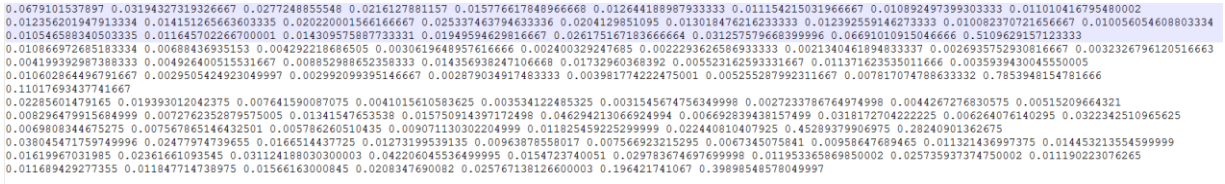


Figure 15: Signature

Local Signature Representation:

For each class or category of text documents features, clustering these features results the cluster objects, aggregate the features of each cluster using some measure of central tendency

like mean. Collect the aggregated features as vectors per class, which becomes the SigSpace per class. Figure 16 shows the vector representation of significant words in a class where the similar words are closer and maintain the relative distance with other similar words. After clustering, similar words from the vector space are mapped to local map as shown. This is represented as Local Signature for the class.

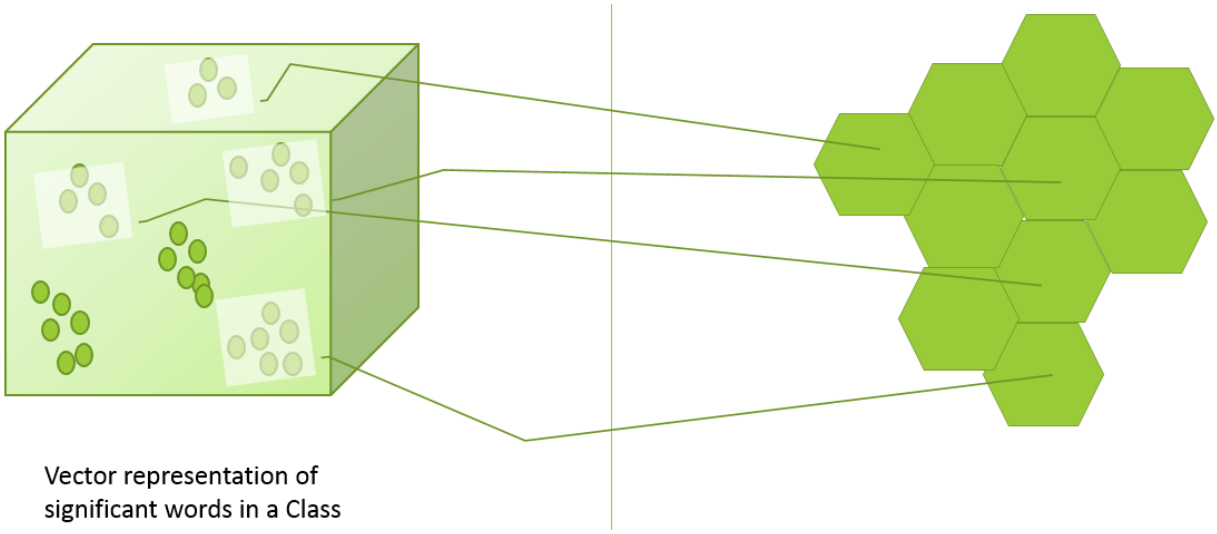


Figure 16: Local Signature Representation

Incremental learning:

SigSpaces – Text will learn the data using incremental clustering algorithms: K-means and SOM. Figure 17 shows that the SOM weight matrix is used to obtain the cluster object for the new feature vector and maps the new feature vector to component. After assigning the component to the new feature, update the signature. The new signature is used for training the classification models.

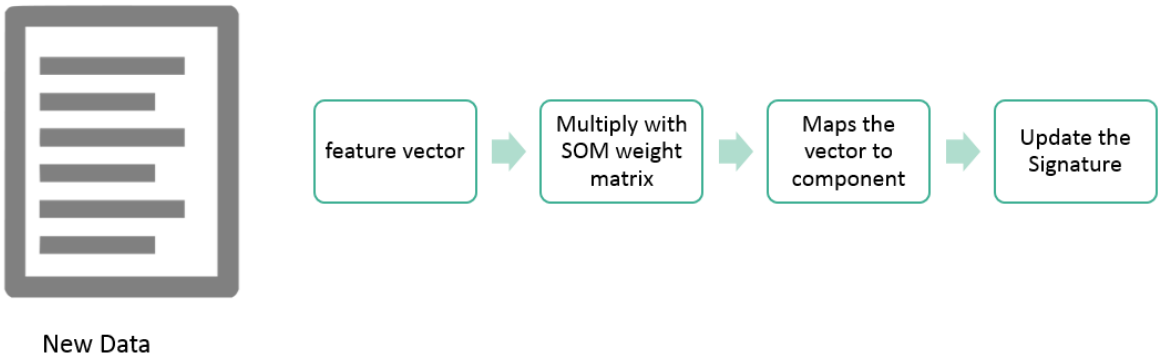


Figure 17: Incremental Learning Workflow

Global Signature Representation:

The workflow of Global Signature generation using the local signature components is shown in figure 18. Global signatures are generated from the local signatures by performing the clustering on the local signature components. The clustering algorithms K-Means and Gaussian mixture models are used to cluster these local components from all classes. The words of local components have been collected by mapping the values corresponding to the components in the signature and they are grouped together to form a document for each respective signature-components.

For all signatures, the signature component documents are generated. TF-IDF is performed on these documents and these vectors are used to perform clustering on the documents. The TF-IDF of the documents is generated and then clustered using k-means and GMM algorithms. The results have been retrieved and represented through histograms.

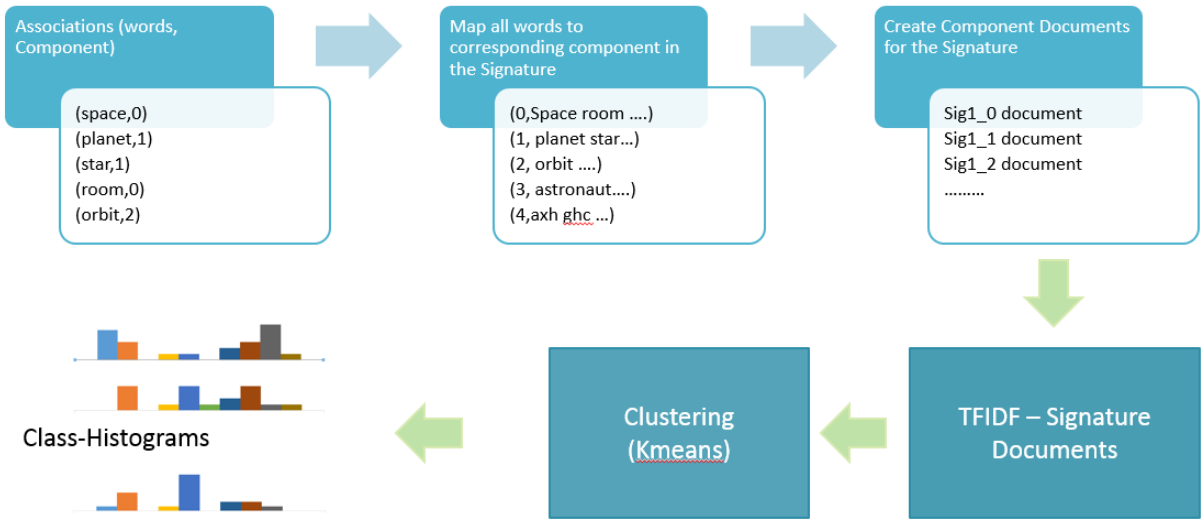


Figure 18: Global Signature Generation Workflow

Global Feature Map:

Each of the Class signature are represented through the outer hexagonal structure and those underlying within are components of the signature. The global representation of all classes has been represented through the outermost hexagonal structure has been shown as the result. The underlying notation is the components of the global signatures.

The smallest modules are the grouping of the local signature components denoted as the input to form a hierarchical blending of signatures, such that all related multiple class signature components are grouped together to form the single component in the global signature. Figure 19 shows the example of global feature map.

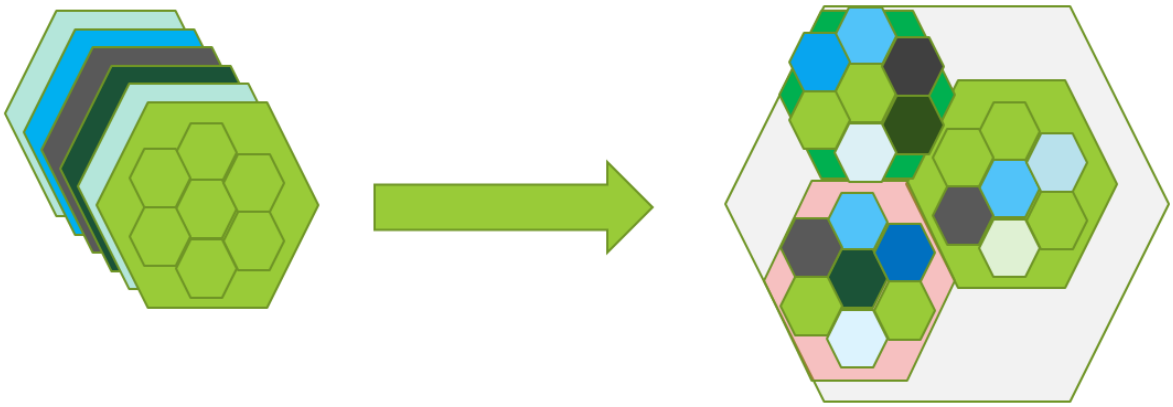


Figure 19: Global Feature Map

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

In this chapter we will discuss about the implementation and experimentation setup i.e. System configurations that were used to build the SigSpace-Text Architecture. All the experiments were conducted on Hadoop file system (HDFS) using Apache Spark framework. Figure 20 shows implementation and experimental setup details that are used to build the SigSpace-Text Architecture.

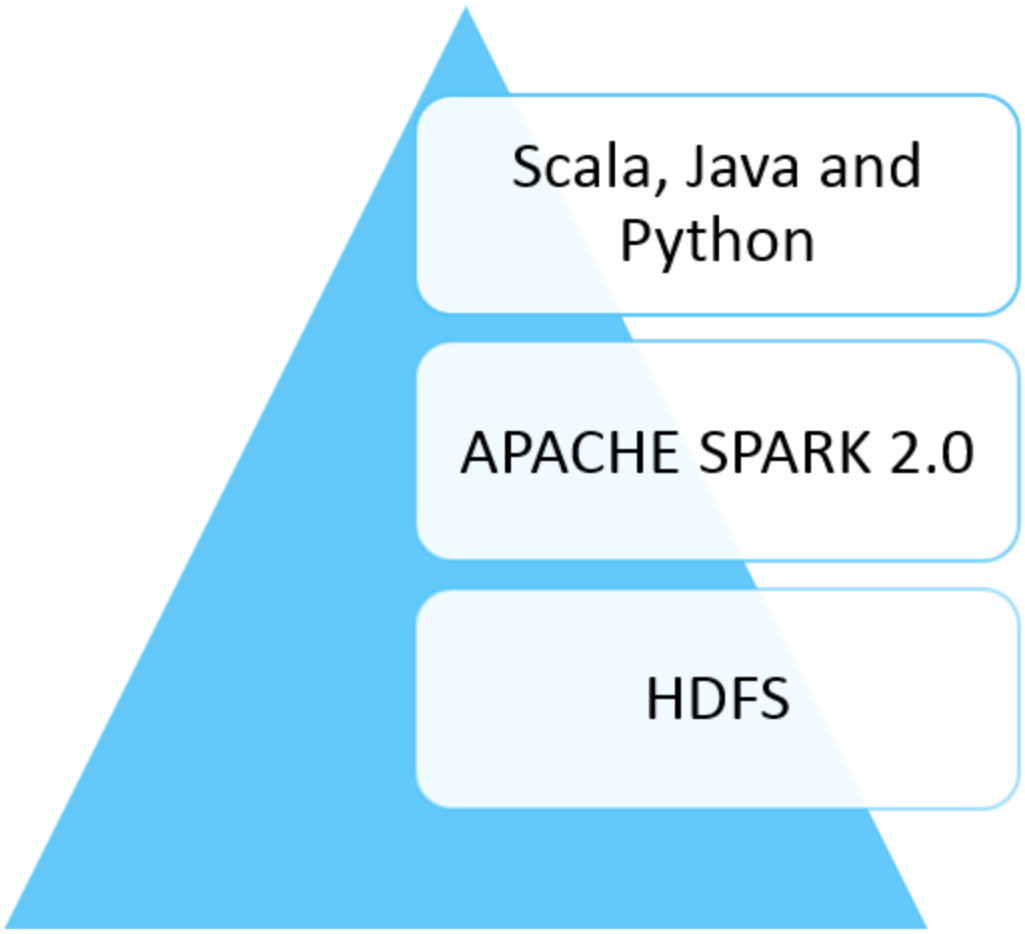


Figure 20: Implementation

4.2 Apache Spark

Apache Spark [43] is an open source cluster computing framework and it is platform which provides programmers with an interface that is focused on the resilient distributed dataset(RDD), that is a data structure consisting of sets of data, and are of read-only type. It is distributed across a cluster of machines. This was a technology developed in accordance with the limitations proposed by the MapReduce programming paradigm. This paradigm mainly insists that the dataflow occurs in a parallel manner, especially the structure of data flow. The MapReduce program reads the data as input from the disk, after which it maps the function across the data, this is followed by the reduction of data which gets stored on the disk. Figure 21 shows the architecture of Spark.

Apache spark comes with Machine Learning Library(MLlib) which is a library that is hugely scalable and the most important components are algorithms and utilities. Some of the machine learning components that can be listed are:

1. Classification

   Spark MLlib supports for the various methods for the multiclassification such as Decision Trees, Random Forest, and Naïve Bayes.

2. Clustering

   Spark MLlib supports the various clustering models such as K-Means, Gaussian Mixture Models (GMM), and Latent Dirichlet allocation (LDA).

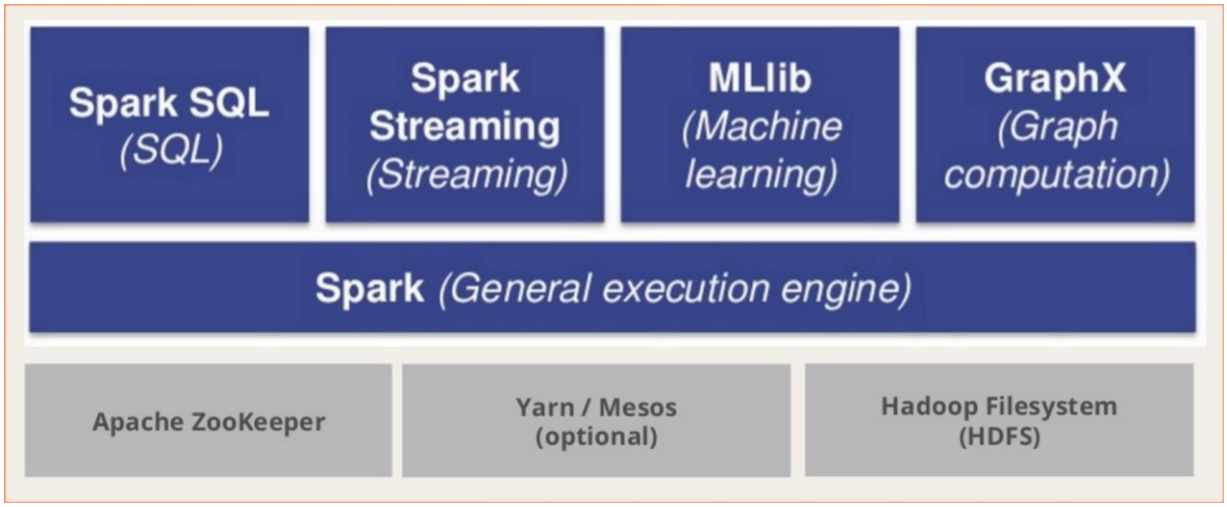


Figure 21: Spark Architecture

4.3 Standford CoreNLP

The NLP (Natural Language Processing) group provides programmers and students with an opportunity of working on algorithms which would allow computers to understand and process the human language. The group mainly involves the computational linguistics and application in human language technology and covers certain key aspects such as Sentence Understanding, Automatic Question Answering, Machine Translation, Syntactic Parsing and Tagging, Sentiment Analysis, and Models of text and visual scenes.

The key components of Stanford NLP [42] include such as:

1. Stop Words:

   The words which are obtained after filters on the processing of NLP (natural language processing). There are tools that avoid stop words to support the search of phrases.

2. Tokenizer:

The tokenization is the process of breaking down a stream of text into modules such as words, phrases and symbols which can be collectively called as tokens. The tokens become inputs for processing such as text mining. It is a process of lexical analysis.

3. Lemmatization:

   In linguistics, the process of grouping the words such that an analysis of lemma, or dictionary form. It is the algorithmic process of determination of lemma of a words based upon the intention of meaning.

Figure 22 shows the SigSpace-Text architecture implementation. As shown in the figure, SisSpace-Text architecture is divided into two major phases - Feature Extraction and Signature Learning. As shown, the signatures generated are used to train the classification models and these trained models are used to predict the category of unlabeled test data file.

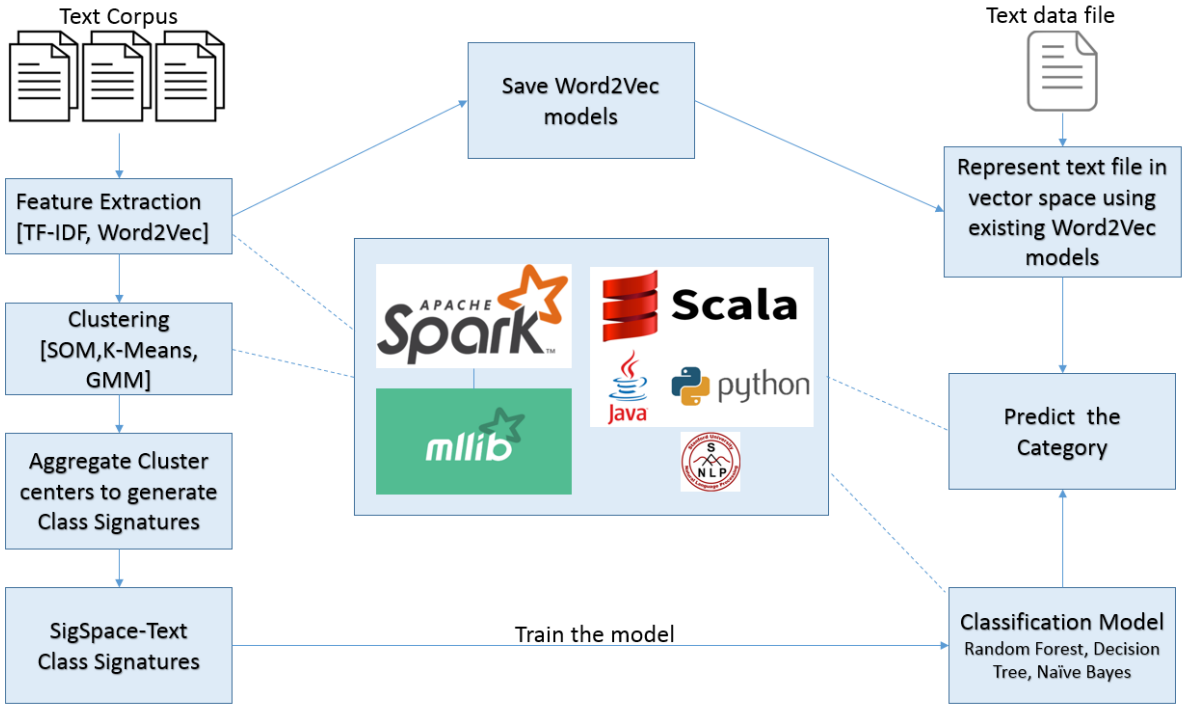


Figure 22: SigSpace-Text Architecture Implementation

4.4 Dataset

20 newsgroups dataset is being used for initial experiments to observe the functioning aspects and to determine the accuracy levels of SigSpace signature generation over text data. The overall count of documents present is almost equivalent to 20,000 documents, and they are partitioned into 20 different categories. Training Dataset size: 60% (12000 documents). Figure 23 shows the list of categories in 20 newsgroups. Further 20 newsgroups are divided into 6 divisions as shown in figure 23. The major 5 divisions text documents are grouped together to form the 5 groups namely comp, rec, sci, talk and religion categories. These 5 groups are used in 5 classes classification in the evaluations.

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Figure 23: Data sets

4.5 Pseudocode

The pseudocode of SigSpace-Text consists of two major algorithms. They are feature_extraction and Signature_generation. The algorithm feature_extraction performs the preprocessing of text data, significant word collection for each class using TF-IDF, building Word2Vec models for each classes and representing the significant words in vector space using Word2Vec models.

```
algorithm feature_extraction is
    input: input filepath fin,
           output filepath out
    output: write the word, feature vector

    (Note that each directory is category of documents)

    for each directory (class) in fin do
       //read all files in directory
       (filename, content) ← read(fin/directory)
       t ← tokenization(content)
       l ← lemmatization(t)
       s ← stopwords_removal(l)

       //build word2vec model for entire text corpus of the class
       Word2Vec(s)

       //get significant words from the entire text corpus of the class
       //using tfidf
       significant_words ← Significant_words(s)

       //represent the each significant word in Word2Vec vector space
       for each word in significant_words do
              (word, vector) ← Word2Vec.getVector(word)

       features ← (words, vectors)
       //write all the features in output file path
       write(features,out/directory/)

    return (features)
```

The algorithm Signature_generation performs the generation of signature for each class using K-means [45] and SOM [44].

```
algorithm Signature_generation is
    input: input filepath fin,
```

```
        output filepath out
output: write the word

(Note that each directory is category of documents)

for each directory (class) in fin do
    //read all features vectors of the category
    (words, feature_vector) ← read(fin/directory)

    initialize clusters ← Array(1,2,3,5,10,15,20,30)
    for each cluster in clusters do
            generate_k-means_signatures(cluster,data,directory,out)
            generate_som_signatures(cluster,data,directory,out)
    //save signatures
return signatures
```

CHAPTER 5

RESULTS AND EVALUATION

5.1 Introduction

In this chapter, we will go through the several sets of evaluations that were conducted using SigSpace-Text Architecture. The experiments were designed to verify factors like training time taken by SigSpace-Text compared to the traditional machine learning architecture, space occupied by normal features vs SigSpace, the accuracy of the traditional methods compared to SigSpace. Additionally, the experiments also showed the results between the different clustering algorithm such as SOM vs K-Means in the SigSpace Architecture.

5.2 Evaluations

Four case studies have been performed to effectively evaluate the SigSpace-Text.

Case 1: Signature Learning with 5 classes (Dataset – 20 newsgroup)

Case 2: Signature Learning with 20 classes (Dataset – 20 newsgroup)

Case 3: Performance Comparison

- Different Classification Algorithms
- Runtime Performance between two clustering algorithms

Case 4: Signature Learning over varying different number of categories

5.2.1 Case 1: Signature Learning with 5 classes

The objective of this case is to evaluate the accuracy, space reduction and significant word selection in Signature Learning with 5 classes.

a) Accuracy comparison of K-Means vs SOM clustering – Signature Learning

b) Accuracy change using SOM over the different number of Significant words

c) Comparison Signature Learning with Classification (traditional) over varying feature size

d) Comparison Signature Learning with Classification over varying space reduction

a) Accuracy comparison of K-Means vs SOM clustering – Signature Learning

The Classification Algorithm used was Random Forest and the number of Significant words chosen for each class was 3000.

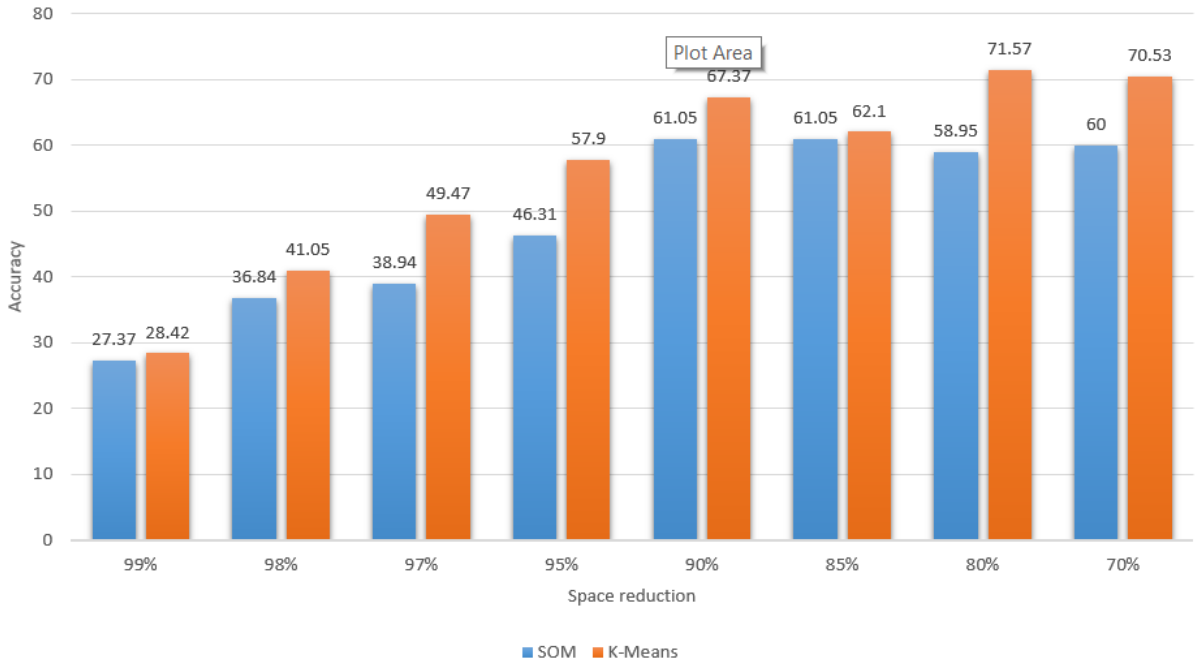


Figure 24: Accuracy comparison of K-Means vs SOM clustering – Signature Learning

There is an exponential increase on an average, when we consider the SOM clusters as well as the k-means clusters, to infer some sort of accurate measures. There is an increase in the accuracy when the space reduction is decreased. When the SOM technique is used, the idealistic output can be contained to be the fluctuation of accuracy when the number of clusters reaches 300 i.e., space reduction obtained is 80%. Comparing both techniques, we can infer that the k-means clustering yielded a better accuracy in almost all clusters, and the variance would make us prefer the k-means technique of clustering.

b) Accuracy change using SOM over the different number of Significant words

The Classification Algorithm used was Random Forest with a Space Reduction of 70%.

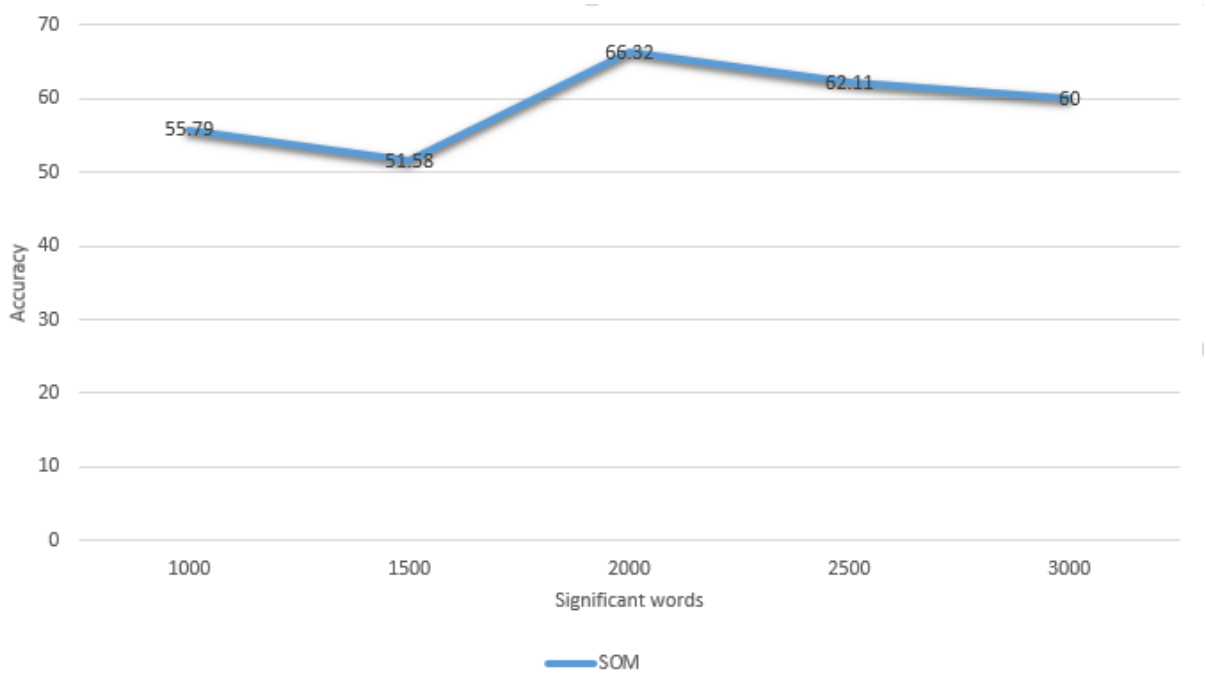


Figure 25: Accuracy change using SOM over the different number of Significant words

Considering the approach of random forest algorithm, the result that can be inferred is that the top TF-IDF words, which serve to be the significant words are realization of the accuracy.

This is low, in the case of 1000 words. It dips by about 4% when the number of words used are 1500. The highest proportion of accuracy can be attributed to the level of 2000 words at 66.32%. When the words are decreased by 500, there is a dip of 4% and the percentage of variance differs as there is a downward progression.

c) Comparison Signature Learning with Classification (traditional) over varying feature size

The Classification Algorithm used was Random Forest with a Space Reduction of 70%.

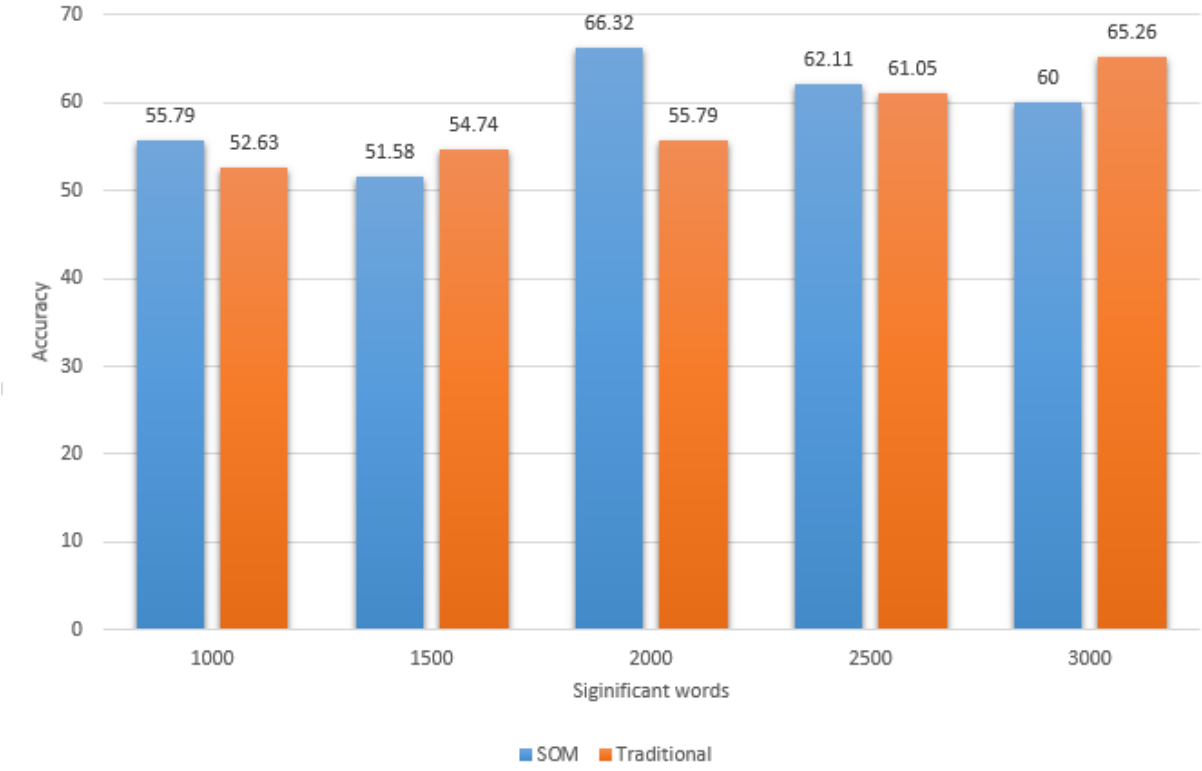


Figure 26: Comparison Signature Learning with Classification (traditional) over varying feature size

The traditional approach, on contrasting with the properties inferenced by the SOM-900 clusters, brought about the ratio to be varying especially when the significant words ranged from a number of 1000 to 3000. When the significant words were 2000, SOM brought out the

highest accuracy rate of 66.32%. The fluctuation was both positive and negative over an increase in cluster size. There was a constant increase when the traditional approach was considered. The percentage of accuracy was 51.59 for the SOM method, which was lowest at 1500 words. In this case, there is a varying trend and the pattern cannot be brought out properly.

d) Comparison Signature Learning with Classification over varying space reduction

The Classification Algorithm used was Random Forest for comparing the signature learning with traditional classification approaches.

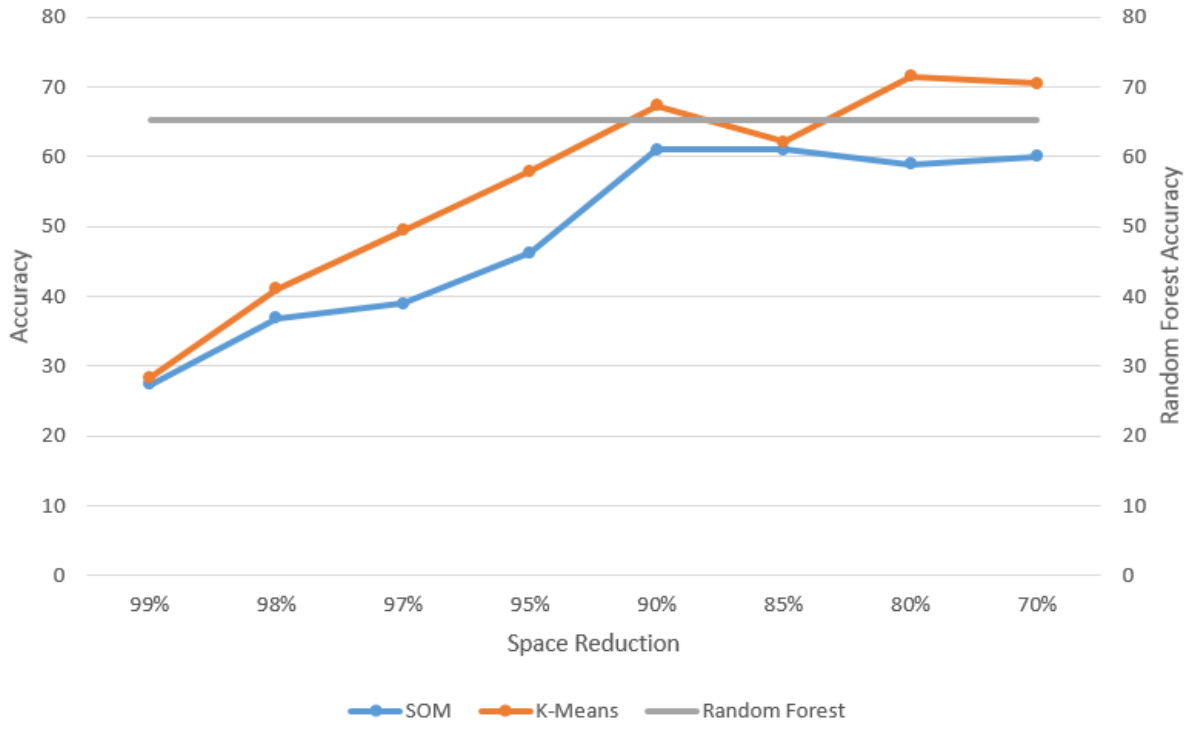


Figure 27: Comparison Signature Learning with Classification over varying space reduction

Case 1: Evaluations:

Accuracy: Compared to the traditional classification using Random Forest, there is a marginal reduction of 5 % using signatures generated by SOM.

Space and runtime (Training):

The training set is reduced to 10%. (90% space reduction). Training the classification model yielded a reduction of training time to 87%. By Using Signatures, it was 26 seconds against the case, where training using all features, was used with a result of 192 seconds.

5.2.2 Case 2: Signature Learning with 20 Classes

The objective of this case is to evaluate the accuracy, space reduction and significant word selection in Signature Learning with 20 classes.

e) Accuracy comparison of K-Means vs SOM clustering – Signature Learning
f) Accuracy change using SOM over the different number of Significant words
g) Comparison Signature Learning with Classification (traditional) over varying feature size
h) Comparison Signature Learning with Classification over varying space reduction

a) Accuracy comparison of K-Means vs SOM clustering – Signature Learning

The Classification Algorithm used was Random Forest and the number of Significant words was 3000.

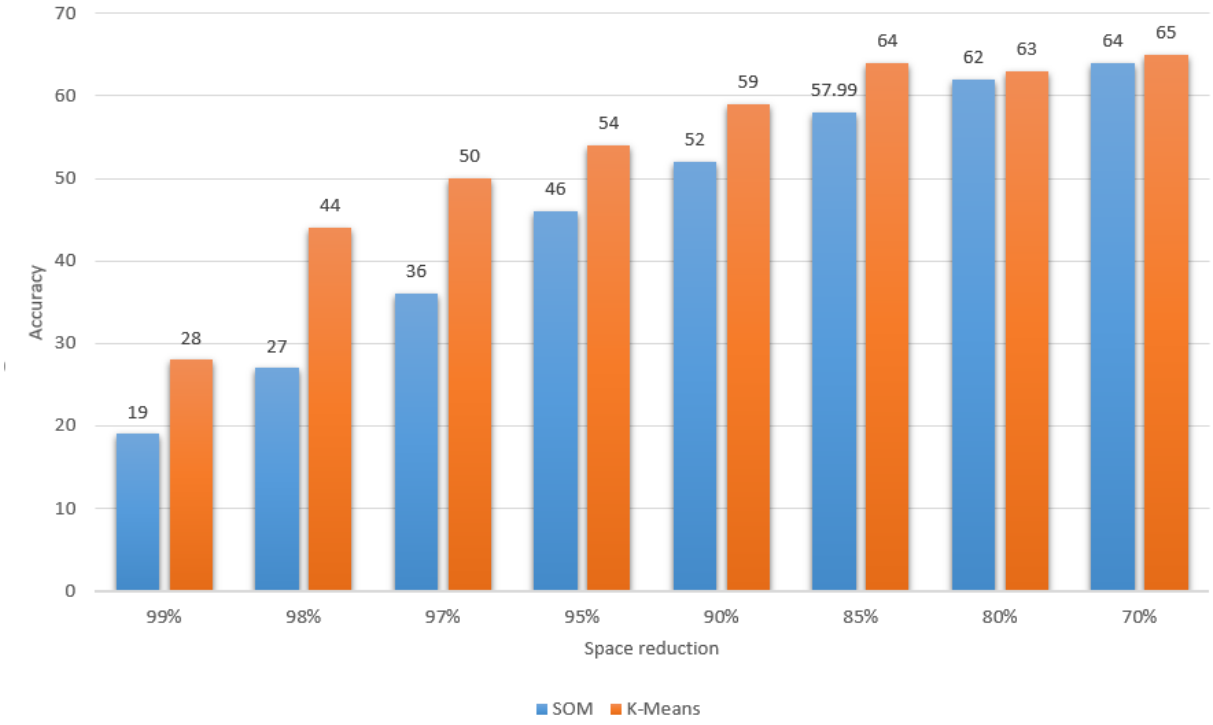


Figure 28: Accuracy comparison of K-Means vs SOM clustering – Signature Learning -20 classes

There is an exponential increase on an average, when we consider the SOM clusters as well as the k-means clusters, to infer some sort of accurate measures. There is a constant increase in the accuracy when the number of clusters are changed. When the SOM technique is used, the idealistic output can be contained to be the fluctuation of accuracy when the space reduction is 85%. Comparing both techniques, we can infer that the k-means clustering yielded a better accuracy in almost all cluster sizes, and the variance would make us prefer the k-means technique of clustering. There is however a dip when the range value changes to 600 clusters i.e., 80% space reduction using k-means.

b) Accuracy change using SOM over the different number of Significant words

The classification algorithm used was Random Forest with a Space Reduction of 70%

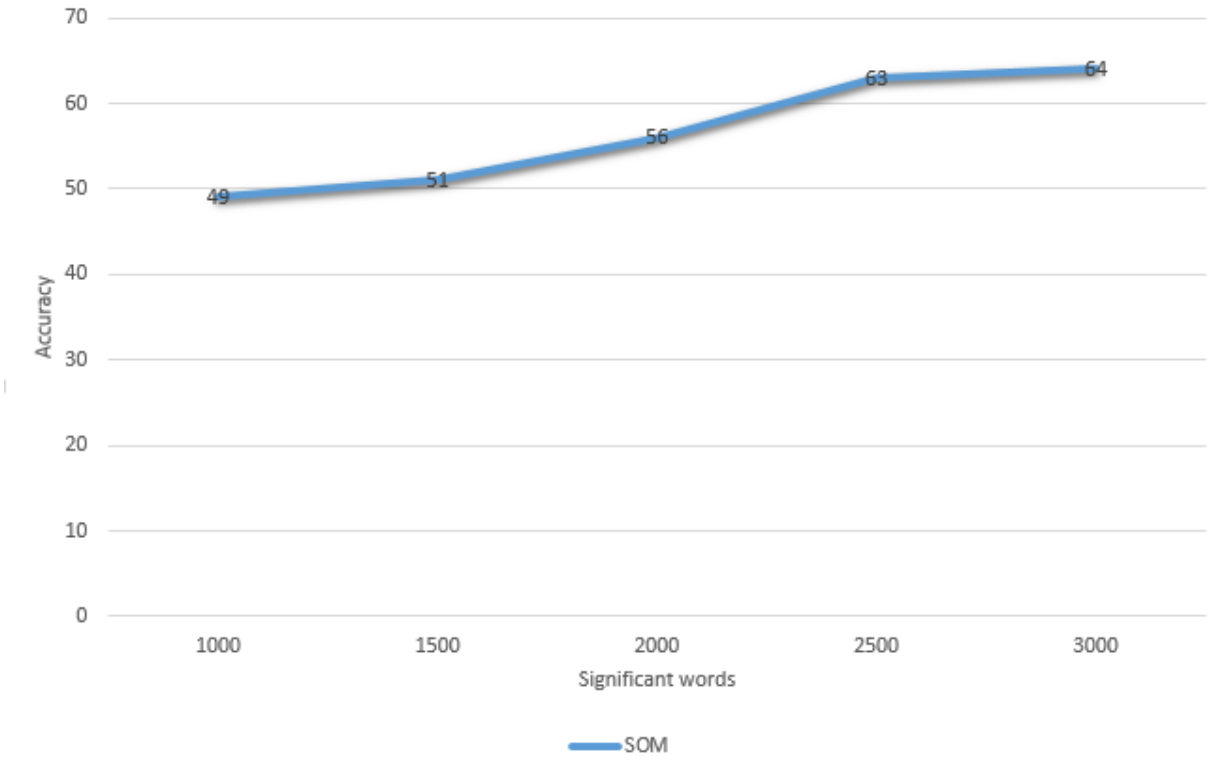


Figure 29: Accuracy change using SOM over the different number of Significant words-20 classes

From the above figure, we can make some significant conclusions. This representation can yield better accuracy when the number of words increases from 1000 to 3000. This is low, in the case of 1000 words. It increases by about 2% when the number of words used are 1500. There is a progressive increase in the accuracy, the highest of which can be observed from the number for words being changed from 2000 to 2500, as it can attribute to a percentage of 7.

c) Comparison Signature Learning with Classification (traditional) over varying feature size

The classification algorithm used was Random Forest with a Space Reduction of 70%.

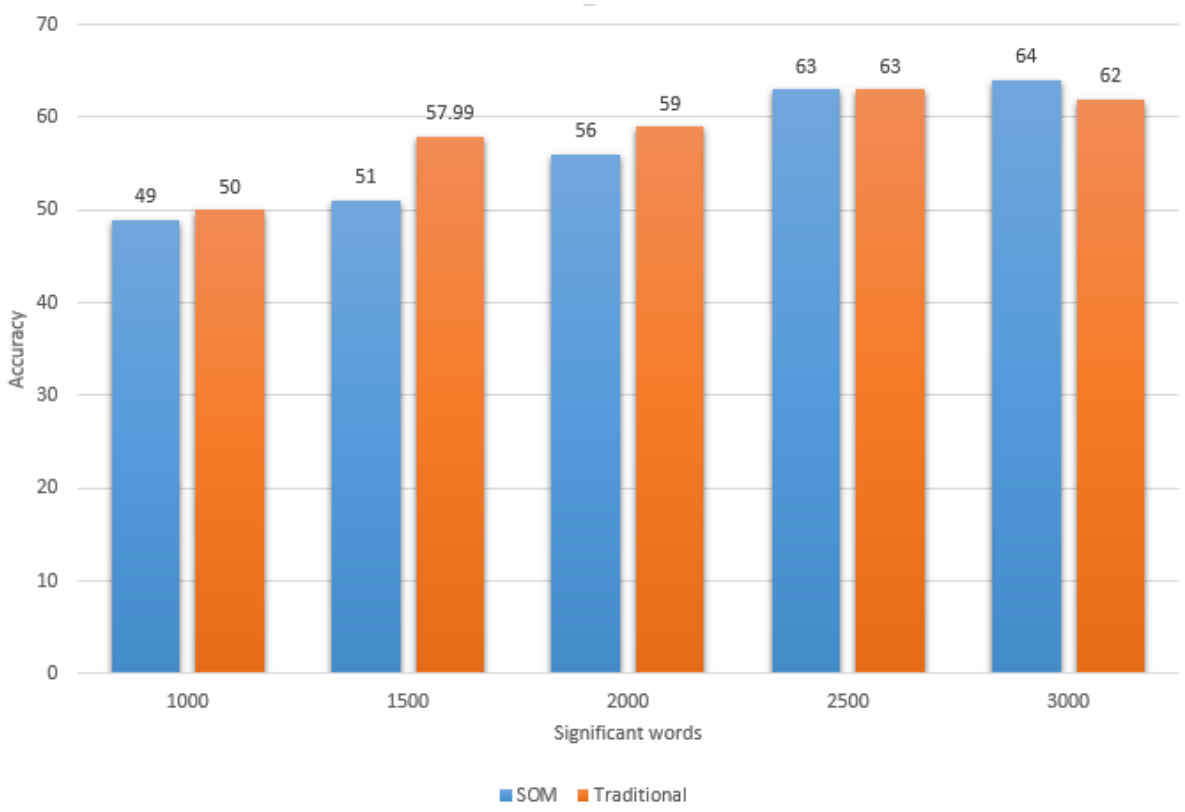


Figure 30: Comparison Signature Learning with Classification over varying feature size-20 classes

The traditional approach, on contrasting with the properties inferenced by the SOM-900 clusters i.e., 70% space reduction, brought about the ratio to be varying especially when the significant words ranged from a number of 1000 to 3000. When the words were 3000, SOM brought out the highest accuracy rate of 64%. The fluctuation was positive over an increase in cluster size. There was a constant increase when the traditional approach was considered. The inference that can be made is that the highest accuracy for 2500 words is by the traditional approach, which saw an increase of around 8% when there was a change in the size from 1000 to 1500 words.

d) Comparison Signature Learning with Classification over varying space reduction

The Classification Algorithm used was Random Forest for comparing the signature learning with traditional classification approaches.

Figure 31: Comparison Signature Learning with Classification over varying space reduction-20 classes

Case 2: Evaluations:

Accuracy: Compared to the traditional classification using Random forest, there is a marginal increase of 2 % using signatures generated by SOM.

Space and runtime (Training):

The training set is reduced to 20%. (80% space reduction). Training the classification model reduced the training time to 83%. When using Signatures, time taken was 62 seconds against the training where all features was used, of 357.2 seconds.

5.2.3 Case 3: Performance Comparison

The main objective of this evaluations to determine the effectiveness of using different classification algorithms and also the runtime performance between two clustering algorithms SOM and K-Means.

a) Performance between different classification algorithms:

The number of categories are 5 and 20 for the two divisions shown below, on which classification algorithms were performed. The clustering algorithm used was SOM which had a space reduction of 70% with the number of significant words being 3000.

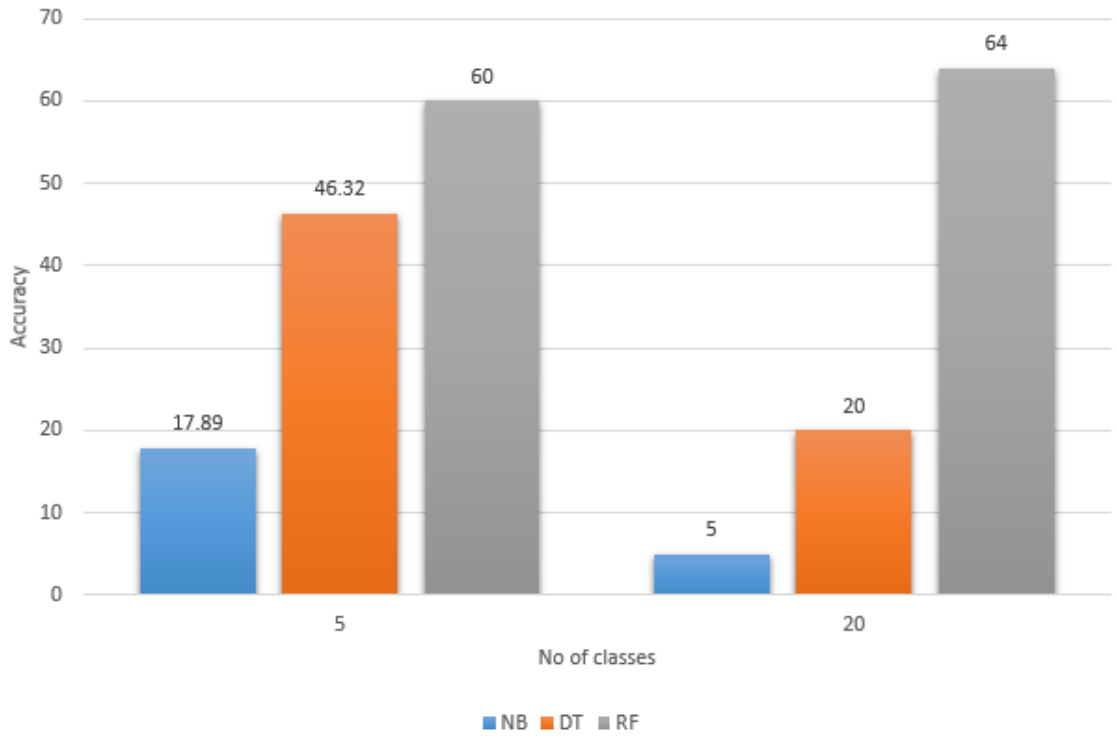


Figure 32: Performance between different classification algorithms

b) Runtime performance between K-Means and SOM clustering algorithms

The number of categories is 5 with the significant words having a count of 3000.

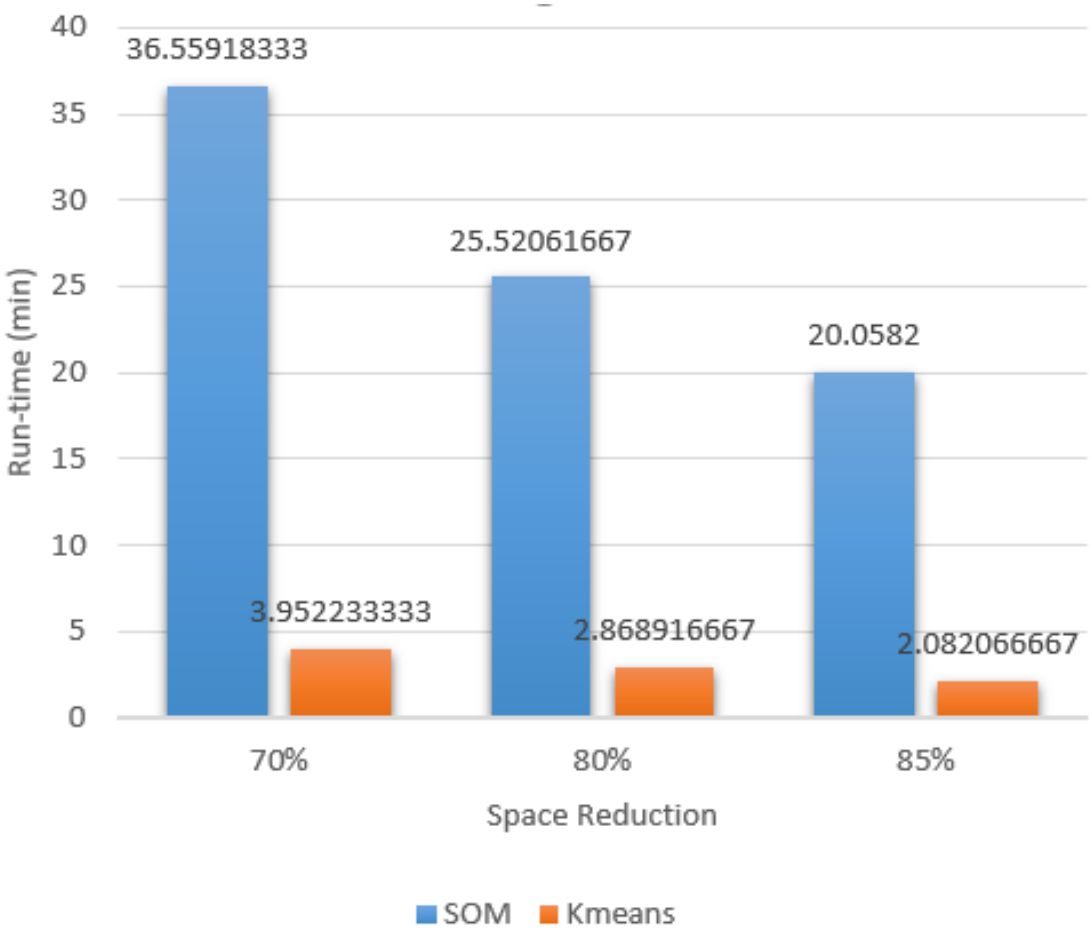


Figure 33: Runtime performance between K-Means and SOM

5.2.4 Case 4: Signature Learning on varying number of categories

The main objective of this evaluation is to identify the effectiveness of signature learning for lower number of categories. The number of significant words is 3000 with a space reduction of 70%. The following figure shows the categories divided, with the first being two categories – alt.atheism and talk.religion.misc. The next division included three categories talk.religion.misc, alt.atheism and soc.religion.Christian. For completely two different categories- alt.atheism and sci.crypt , we achieved 90% accuracy using SOM.

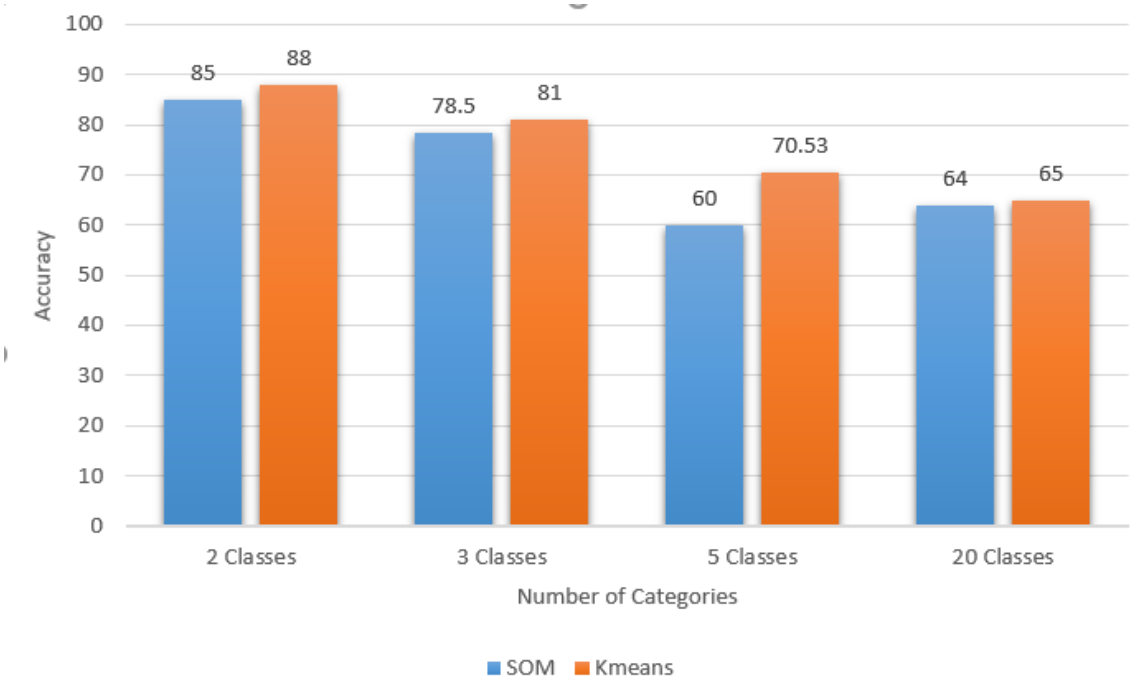


Figure 34: Accuracy – Signature Learning Over Varying Number Of Categories

5.2.5 Signature Components

Figure 35 shows the words that are grouped into components of a signature. Below is the Comp class signature which has component-words grouped together.

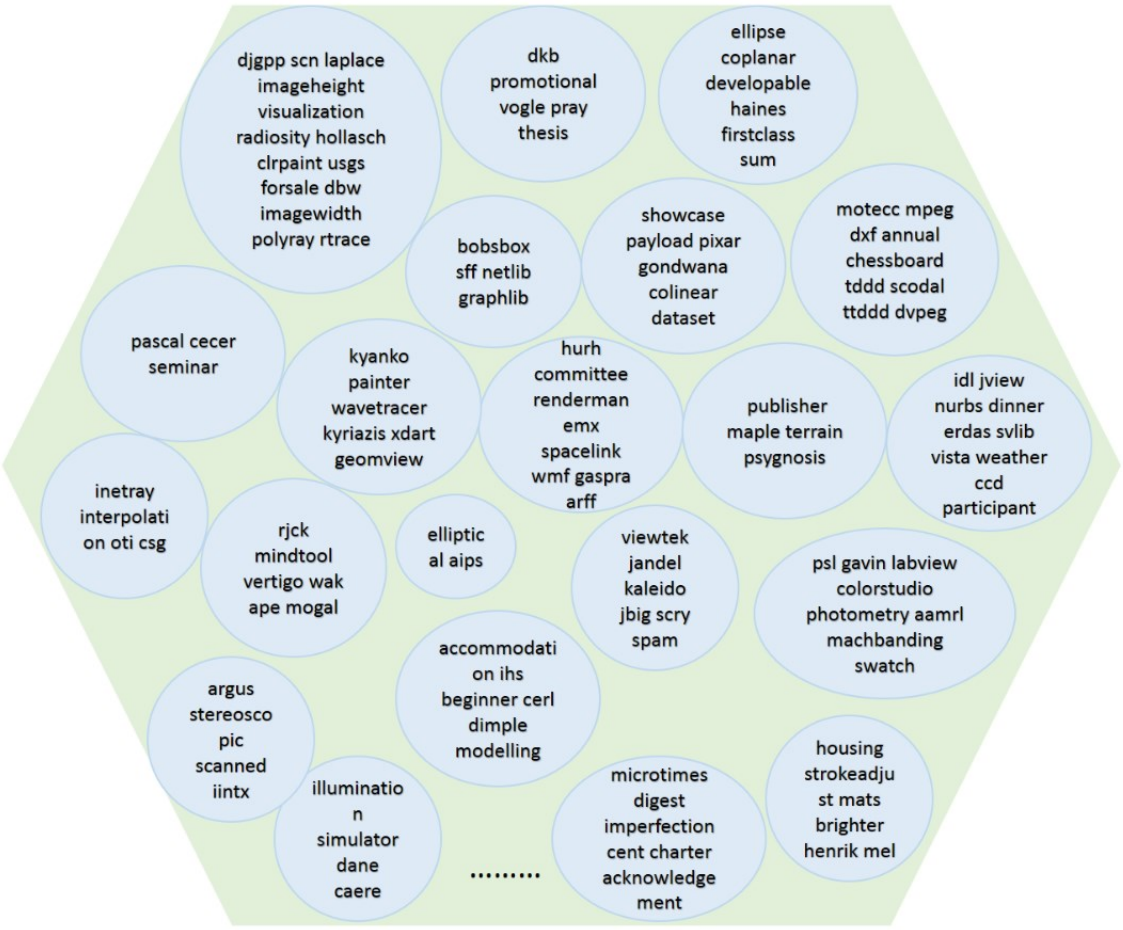


Figure 35: Signature-Components for comp class in 20 Newsgroup

5.3 Summary

From the results of the experiments we can clearly state that even though there is a slight decrease in accuracy, SigSpace-Text is better than the traditional approaches in space and runtime factors. Figure 36 shows the table which discusses about the other different approaches accuracy and performance.

Model	Accuracy	Significant words	Space reduction
SigSpace-Text	65	3000	70% of feature size
Random Forest Classification	64	3000	No Space reduction (trained with 3000 features for each class)
Hybrid Discriminative RBM[21]	76	5000	No Space reduction (trained with 5000 features)
SVM[22]	80.8	No significant words selection	Complete data is used to train the model
ECOC Naïve Bayes[23]	81.8	No significant words selection	Complete data is used to train the model
Regularized Least Squares classifier[24]	84.86	No significant words selection	Complete data is used to train the model
K-NN[25]	69.1	No significant words selection	Complete data is used to train the model

Figure 36: Comparative Study

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The thesis conformed to idealistic approaches of presenting intuitive capabilities in fields of text analytics. The distinction of signature learning from the traditional learning lies in the form of independent learning, distributive learning, incremental learning and scalability. However, there is a marginal reduction of accuracy, on comparing the performance of this approach to the traditional machine learning approach, it is a very good trade-off to consider a reduced data size for applications where handling data sizes is critical.

The present evaluation thus confirms SigSpace-Text as an important approach for distributed and scalable machine learning in Text Analytics.

6.2 Limitations

The Limitation of our approach is that the word2vec models of all classes are required to vectorize the test document, thus the word2vec model has to be updated with incoming new data, for effective representation of words in vector space.

6.3 Future Work

The future scope is to develop the Multilevel signature components and use these components to organize the classes. There can be different mechanisms for building and using Global feature map. There exist different measures to calculate the cluster center other than mean. The Effective use of EM clustering in Signature learning can be studied further and the usage of Global signatures to categorize the text documents at local level can be a prospective field

of research. The future scope of this project also includes building of global vector space initially using the Word2Vec model on whole text corpus, such that the representation of all words is consistent across all the classes.

REFERENCES

[1] Vesanto, Juha, and Esa Alhoniemi. "Clustering of the self-organizing map." IEEE Transactions on Neural Networks 11, no. 3 (2000): 586-600.

[2] Goldberg, Yoav, and Omer Levy. "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method." arXiv preprint arXiv:1402.3722 (2014).

[3] Gharib, Tarek F., Mohammed M. Fouad, Abdulfattah Mashat, and Ibrahim Bidawi. "Self organizing map-based document clustering using WordNet ontologies." IJCSI International Journal of Computer Science Issues 9, no. 1 (2012): 1694-0814.

[4] Kohonen, Teuvo, Samuel Kaski, Krista Lagus, Jarkko Salojarvi, Jukka Honkela, Vesa Paatero, and Antti Saarela. "Self organization of a massive document collection." IEEE Transactions on Neural Networks 11, no. 3 (2000): 574-585.

[5] Doddala, Seetha Rama Pradyumna “SigSpace – Class-Based Feature Representation for Scalable and Distributed Machine Learning” (University of Missouri–Kansas City, 2016)

[6] Bakus, J., M. F. Hussin, and M. Kamel. "A SOM-based document clustering using phrases." In Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on, vol. 5, pp. 2212-2216. IEEE, 2002.

[7] Amine, Abdelmalek, Zakaria Elberrichi, Ladjel Bellatreche, Michel Simonet, and Mimoun Malki. "Concept-based clustering of textual documents using SOM." In 2008 IEEE/ACS International Conference on Computer Systems and Applications, pp. 156-163. IEEE, 2008.

[8] Poinçot, Phillipe, Soizick Lesteven, and Fionn Murtagh. "A spatial user interface to the astronomical literature." Astronomy and Astrophysics Supplement Series 130, no. 1 (1998): 183-191.

[9] Ma, Jian, Wei Xu, Yong-hong Sun, Efraim Turban, Shouyang Wang, and Ou Liu. "An ontology-based text-mining method to cluster proposals for research project selection." IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 42, no. 3 (2012): 784-790.

[10] Matharage, Sumith, Hiran Ganegedara, and Damminda Alahakoon. "A scalable and dynamic self-organizing map for clustering large volumes of text data." In Neural Networks (IJCNN), The 2013 International Joint Conference on, pp. 1-8. IEEE, 2013.

[11] Kohonen, Teuvo, Samuel Kaski, Krista Lagus, Jarkko Salojarvi, Jukka Honkela, Vesa Paatero, and Antti Saarela. "Self organization of a massive document collection." IEEE transactions on Neural Networks 11, no. 3 (2000): 574-585.

[12] Mingoti, Sueli A., and Joab O. Lima. "Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms." European Journal of Operational Research 174, no. 3 (2006): 1742-1759.

[13] Joachims, Thorsten. "A Statistical Learning Model of Text Classification for SVMs." In Learning to Classify Text Using Support Vector Machines, pp. 45-74. Springer US, 2002.

[14] Mingoti, Sueli A., and Joab O. Lima. "Comparing SOM neural network with Fuzzy c-means, K-means and traditional hierarchical clustering algorithms." European Journal of Operational Research 174, no. 3 (2006): 1742-1759.

[15] Matharage, Sumith, Hiran Ganegedara, and Damminda Alahakoon. "A scalable and dynamic self-organizing map for clustering large volumes of text data." In Neural Networks (IJCNN), The 2013 International Joint Conference on, pp. 1-8. IEEE, 2013.

[16] Decision tree - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Decision_tree. (Accessed on 11/04/2016).

[17] Random Forest - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Random_forest. (Accessed on 11/04/2016).

[18] Microsoft Cognitive Services - Microsoft, Documentation and Support. https://www.microsoft.com/cognitive-services/en-us/documentation

[19] IBM Watson - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Watson_(computer)

[20] Boston Dynamics - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Boston_Dynamics

[21] Larochelle, Hugo, and Yoshua Bengio. "Classification using discriminative restricted Boltzmann machines." In Proceedings of the 25th International Conference on Machine Learning, pp. 536-543. ACM, 2008.

[22] Lan, Man, Chew Lim Tan, and Hwee-Boon Low. "Proposing a new term weighting scheme for text categorization." In AAAI American Association for Artificial Intelligence, vol. 6, pp. 763-768. 2006.

[23] Li, Baoli, and Carl Vogel. "Improving multiclass text classification with error-correcting output coding and sub-class partitions." In Canadian Conference on Artificial Intelligence, pp. 4-15. Springer Berlin Heidelberg, 2010.

[24] Rennie, Jason DM. "On The Value of Leave-One-Out Cross-Validation Bounds." http://www.ai.mit.edu/~jrennie/writing/loocv.ps.gz, December 2003.

[25] Lan, Man, Chew Lim Tan, Jian Su, and Yue Lu. "Supervised and traditional term weighting methods for automatic text categorization." IEEE Transactions on Pattern Analysis and Machine Intelligence 31, no. 4 (2009): 721-735.

[26] Artificial Intelligence - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Artificial_intelligence

[27] Machine Learning - Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Machine_learning

[28] http://datacafeblog.com/wp-content/uploads/2015/12/slide_6-960x670.jpg

[29] https://codesachin.files.wordpress.com/2015/11/kohonen1.gif

[30] https://deeplearning4j.org/img/tfidf.png

[31] https://i.ytimg.com/vi/ajTc5y3OqSQ/hqdefault.jpg

[32] Ramos, Juan. "Using tf-idf to determine word relevance in document queries." In Proceedings of the First Instructional Conference on Machine Learning. 2003.

[33] Harish, Bhat S., Devanur S. Guru, and Shantharamu Manjunath. "Representation and classification of text documents: A brief review." IJCA International Journal of Computer Applications, Special Issue on Recent Trends in Image Processing and Pattern Recognition (2) (2010): 110-119.

[34] ChandraShekar, B. H., and G. Shoba [sic]. "Classification of Documents Using Kohonen's Self-Organizing Map." International Journal of Computer Theory and Engineering 1, no. 5 (2009): 610.

[35] Hu, Xiaohua, Xiaodan Zhang, Caimei Lu, Eun K. Park, and Xiaohua Zhou. "Exploiting Wikipedia as external knowledge for document clustering." In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 389-396. ACM, 2009.

[36] Saarikoski, Jyri, Jorma Laurikkala, Kalervo Järvelin, and Martti Juhola. "Self-Organising maps in document classification: A comparison with six machine learning methods." In International Conference on Adaptive and Natural Computing Algorithms, pp. 260-269. Springer Berlin Heidelberg, 2011.

[37] Wei Ye, Samuel Maurus, Nina Hubig and Claudia Plant."Generalized Independent Subspace Clustering." In 16th International Conference on Data Mining, IEEE, 2016.

[38] Suh, Sangho, Jaegul Choo, Joonseok Lee, and Chandan K. Reddy. "L-EnsNMF: Boosted Local Topic Discovery via Ensemble of Nonnegative Matrix Factorization." http://www.joonseok.net/papers/lensnmf.pdf, 2016.

[39] Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[40] Wang, Yuanqing, Wenjun Wang, Weidi Dai, Pengfei Jiao, and Wei Yu. "A Fused Multi-feature Based Co-training Approach for Document Clustering." In Information Science and Control Engineering (ICISCE), 2016 3rd International Conference on, pp. 38-43. IEEE, 2016.

[41] Naive Bayes - Wikipedia, the free encyclopedia.
https://en.wikipedia.org/wiki/ Naive_Bayes_classifier

[42] http://stanfordnlp.github.io/CoreNLP/

[43] Apache Spark - Wikipedia, the free encyclopedia.
https://en.wikipedia.org/wiki/ Apache_Spark

[44] Sarazin, Tugdual, Hanane Azzag, and Mustapha Lebbah. "SOM Clustering using Spark-MapReduce." In Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International, pp. 1727-1734. IEEE, 2014.

[45] K-Means clustering- Wikipedia, the free encyclopedia.
https://en.wikipedia.org/wiki/ K-means_clustering

[46] Semi-supervised learning- Wikipedia, the free encyclopedia.
https://en.wikipedia.org/wiki/ Semi-supervised_learning

VITA

Rakesh Reddy Bandi was born on November 30, 1993 in Andhra Pradesh, India. He completed his Bachelor's degree in Computer Science and Engineering from Jawaharlal Nehru Technological University in Hyderabad. During his under graduation, he worked as an Intern in Blackbucks Engineers and company for last 3 semesters. Mr. Rakesh Reddy Bandi started his masters in computer Science at the University of Missouri-Kansas City (UMKC) in August 2015, specializing in Data Sciences and Software Engineering. While he was studying in UMKC, he has worked as Student Developer at UMKC Information Services-Internal Applications. Upon completion of her requirements for the Master's Program, Mr. Rakesh Reddy Bandi plans to work as a Big Data Engineer.