International Journal of Engineering & Technology, 7 (1.1) (2018) 449-455

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Index split decision tree and compositional deep neural network for text categorization

N. Ravikumar [1] *, Dr. P. Tamil Selvan [2]

[1] Research Scholar, Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamilnadu, India
[2] Assistant Professor, Computer Science, Karpagam Academy of Higher Education, Coimbatore, Tamilnadu, India
**Corresponding author E-mail: nravikumar67@yahoo.com*

Abstract

Text categorization with machine learning algorithms generally reckons to possess horizontal set of classes. Several advanced machine learning algorithms have been designed in the past few decades. With the growing research work for text categorization, it has become important to categorize the research outcome and provide the learners with an effective machine learning method, a framework called, Hierarchical Decision Tree and Deep Neural Network (HDT-DNN).It investigates machine learning algorithms to create horizontal set of classes and it is used for classification of text. With this objective, a novel and efficient text categorization framework based on decision tree model is used in order to categorize text according to superior and subordinate level. The text to be categorized is presented in the form of a tree with parent text category being superior to all. The intermediate level represents the text that is both superior and subordinate. Then Deep Neural Network model is presented initiating compositional model, where the text has to be categorized, as a layered integration of primitives from the constructed decision tree model. The extra layers enable composition of features from lower layers, potentially modeling complex text with fewer units than a similarly carried out shallow network producing hierarchical classification. The significance of the impact of HDT-DNN framework is evaluated through empirical study. Extensive experiments are carried out and the performance of HDT-DNN framework is evaluated and compared with existing state-of-art methods using parameters such as precision, classification accuracy, classification time, with respect to varied number of features and document size.

***Keywords**: Text Categorization; Machine Learning; Decision Tree; Deep Neural Network; Compositional Model and Hierarchical Classification.*

1. Introduction

With the availability of text documents in web pages and electronic representation, it is of considerable significance to tag the contents with an assumption set of thematic classes. This is referred to as Text Categorization (TC). Common use of text categorization includes the classification of web pages under hierarchical classes. With the extensive use of TC, it limits the documents belonging to specific classes to avoid the classes from becoming extremely substantial. There have been lots of researches on TC during the last two decades.

On the other hand, classification of high dimensional data with insignificant observations is becoming more customary. During the last two decades, several classification methods and feature selection (FS) algorithms have been constructed to enhance the prediction accuracy. However, the result of an FS algorithm has higher rate of influence typically in high dimensional data.

A new evaluation measure Q-statistic for Feature Selection (Q-FS) was designed in [1] not only to provide stability for the selected feature subset but also ensure prediction accuracy. Followed by this, the Booster of an FS algorithm was designed that increased the value of the Q-statistic of the algorithm applied to extract a different subset. Empirical studies based on the synthetic data when applied to Q-statistic for Feature Selection also resulted in the improvement of prediction accuracy in addition to the stability of the feature being selected at minimum time interval. However, with the horizontal representation of classes, the method could not obtain high performance for certain specific data.

A Maximum Discrimination (MD) [2] method presented a feature selection model that ranked original features with the objective of increasing discerning implementation for TC with naive Bayes classifiers used as learning algorithms. The MD method, when compared to the conventional models, measured the goodness of a feature without training a classifier in an explicit manner. On the other hand the MD method selected these features that provided maximum discrimination in terms of new divergence measure, namely, Jeffreys-Multi- Hypothesis divergence (JMH-divergence). The MD method also designed an efficient approach to rank the order of features with the goal of creating maximum JMH divergence. Compared to the conventional feature selection approaches that ranked features only exploring intrinsic characteristics, the MD method involved the learning model, to analyze the optimality of the selected features. Experiments conducted on benchmarks demonstrated their promising performance improvement in terms of accuracy of text being categorized and F1 measure with different number of features. Though accuracy of text being categorized was ensured, the time taken to arrive at the optimality of features being selected was not analyzed. Against this background, a novel text categorization method with machine learning algorithms, where the input data are represented in their original structural form as Reuters-21578 Text Categorization Collection Data Set called, Hierarchical Decision Tree and Deep Neural Network (HDT-DNN) framework is presented. The framework first identifies optimal feature using decision tree model for each parent text category and intermediate level with respect horizontal set of classes. Next, the machine learning initiates compositional model from lower layers, potentially modeling complex text with lesser units than a similarly carried out shallow network. The combination of decision tree model and machine learning for text categorization is used to create horizontal set of classes for text classification. In summary, the contribution of this paper is threefold.

- The new Index Split Decision Tree algorithm is robust and capable of handling horizontal representation of classes. Since the framework considers both parent text category and intermediate level among the samples, even if text to be categorizes changes with respect to categories, it can still maintain a good performance, as the intermediate level that is utilized considers both the superior and subordinate text into their respective subspaces correctly.
- A Compositional Deep Neural Network model categorizes the text based on the compositional model among samples. Unlike previous work that merely considers ranking of features and reorders, original data with Jeffreys-Multi- Hypothesis divergence (JMH-divergence), HDT-DNN takes compositional model from lower layers and shallow network into account.
- The HDT-DNN framework integrates decision tree with machine learning for horizontal representation of classes to arrive at the optimality of features. This setting fits each individual sample with its horizontal representation with respect to the learned categories, consequently resolving text accuracy being categorized and precision issues for text categorization effectively.

This paper is ordered as follows: Section 2 provides some existing text categorization methods and mechanisms provided by different researchers, Section 3 describes the implementation of the improved framework, Hierarchical Decision Tree and Deep Neural Network (HDT-DNN), Section 4 provides experimental details, Section 5presents results and analysis and Section 6 offers Conclusion.

2. Related works

The extensive availability of web documents in electronic media necessitates an inevitable mechanism to label documents with default topics set, known as Text Categorization (TC). By constructing the TC task as a classification problem, many existing learning approaches can be applied.

A text classification algorithm using multi-pass sieve framework was investigated in [4]. This was performed with the objective of categorizing the text from Portable Document Format (PDF). Feature selection plays a major role in text categorization. With this objective, in [5], particle swarm optimization was performed to select optimal features. Using the generated optimal features, text categorization was carried out resulting in the improvement of computational complexity. A comparative study on machine learning techniques was performed in [6]. However, using conventional machine learning methods, solution for feature engineering can not be found. To address this issue, convolutional neural network was applied in [7] reducing the computational cost and memory use.

A comparative study of k-nearest neighbor and decision tree was analyzed in [8]. Yet another machine learning model involving topic-independent features for text categorization was presented in [9], resulting in better classification accuracy. However, structural features were not considered during classification of text documents due to the unary relations used. To address this issue, binary relations using certain lexemes and relation names as features in [10] were used resulting in the classification accuracy.

With the era of big data and enormous growth in textual data, conventional mode of text categorization proved to be less efficient. A novel text categorization algorithm was investigated in [11] using genetic selection feature optimization. This in turn resulted in the improvement of classification accuracy with smaller feature sets. However, the frequency of the term appearing in a text was not considered. To address this issue, in [12], Multinomial Naïve Bayes Probabilistic model was used providing better scores for text classification. A pilot study was conducted in [13] using deep learning model to classify retinal images.

One of the biggest challenges in TC is the learning from high dimensional data. On one hand several thousand terms in document result in computational burden. On the other hand, certain irrelevant and redundant features have directly a negative impact on predictive performance of classifiers for text categorization. To minimize the impact of curse of dimensionality and to speed up the learning process, it becomes necessary to perform feature reduction to reduce feature size.

A human perceptive model was investigated in [14] by introducing two new concepts, subjective perception and subliminal stimulation for color categorization on the basis of two dimension. A Zipf’s law-based feature selection and use of linear SVM weight for feature ranking was investigated in [15] to address issues related to dimensionality. This hybrid feature selection method improved the classification performance. A comparative study of five text categorization algorithms was provided in [16].

An improved global feature selection scheme was presented in [17] according to the discriminative power on classes and these labels were used while producing the feature sets. This in turn resulted in the improvement of performance of classification in terms of micro-f1 and macro-f1. In [18], a survey of feature selection and classification techniques for text categorization was presented. However, as the model was based on dense structure, with the increase in the size, time for text categorization also increased. To address this issue, a sparse model for text categorization using regression was presented in [19]. With this, a good trade off between performance and sparsity were said to be achieved. Regularization Extreme Machine Learning was applied in [20] for text categorization that was proved to be faster than the conventional learning models.

In the analyzed papers, the authors employs different mechanisms for text categorization with no changes on-the-fly to create horizontal set of classes and optimality of features and uses them for the classification of text. In this way, horizontal set of classes and optimality of features have to be explored. The work proposed in this paper takes into account horizontal set of classes and feature optimality properties using Hierarchical Decision Tree and Deep Neural Network (HDT-DNN) framework, which is discussed in the forthcoming sections.

3. Methodology

In this section, the application of machine learning algorithms for horizontal set of classes and therefore the categorization of text, namely, Hierarchical Decision Tree and Deep Neural Network (HDT-DNN) framework are investigated. Initially, a Decision Tree model is constructed with the given Reuters-21578 Text Categorization Collection Data Set as input. The text to be categorized is represented in the form of a tree with the objective of selecting optimal features. The parent text category is considered to be superior to all. The intermediate level represents the text that is both superior and subordinate. With this feature selection is made in an efficient manner through decision tree construction.

With the constructed decision tree, Deep Neural Network model is applied to produce hierarchical classification. The layers enable composition of features from lower layers. As a result, complex text is modeled in a potential manner with fewer units than similarly performing shallow network. Figure 1 shows the block diagram of Hierarchical Decision Tree and Deep Neural Network (HDT-DNN) framework.

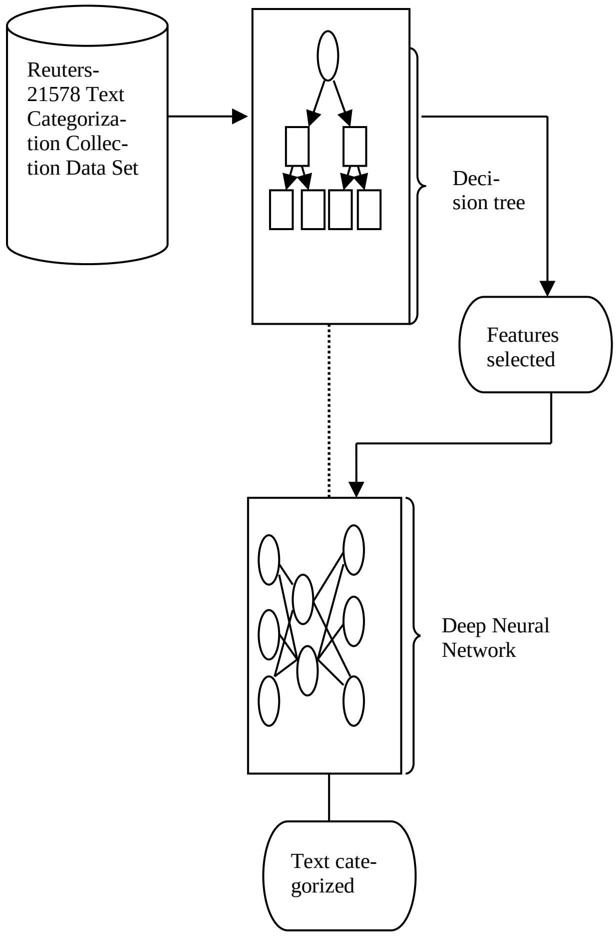

Fig. 1: Block Diagram of HDT-DNN Framework.

As shown in the figure, the block diagram of Hierarchical Decision Tree and Deep Neural Network (HDT-DNN) framework is split into two parts.

The two parts are construction of Decision Tree model and Deep Neural Network. Decision Tree construction is performed using Index Split Value.

With this Index Split Value, the features (i.e. text) to be selected for text categorization are obtained. The second part involves the Deep Neural Network construction.

The Deep Neural Network construction using the Conditional Probable Compositional Text Categorization algorithm reduces the number of features used repeatedly for text categorization and therefore resulting in the improvement of precision.

Index Split Value-based Decision Tree

Let us consider a decision tree association with a document ‘D’, where each root node ‘RN’ comprises all documents, each internal node ‘IN’ being a subset of documents separated on the basis of an attribute, and leaf node ‘LN’ labeled with a class respectively. Let us further initiate a class of representations for the purpose of feature (i.e. text) selection on the basis of decision trees. Figure 2 given below shows the block diagram of Index Split Value-based Decision Tree model.

As illustrated in figure 2, the block diagram of Index Split Value-based Decision Tree model consists of the input obtained from the Reuters-21578 Text Categorization Collection Data Set. With the obtained dataset as input, preprocessing is performed with which the features are selected. For this purpose, simple attributes on strings are used and is represented as given below.

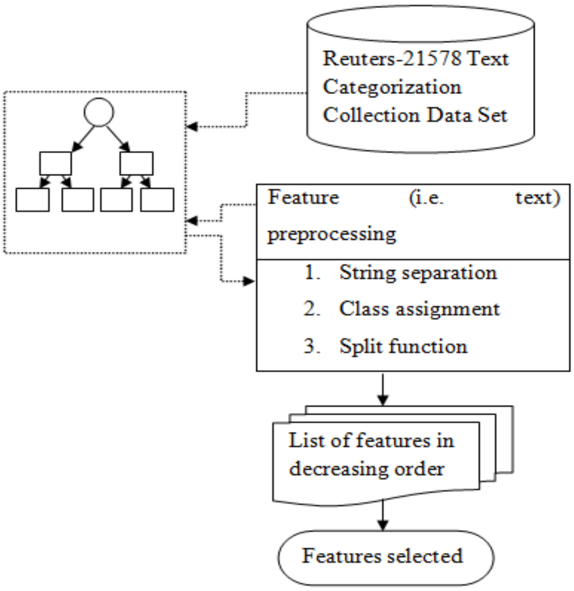
Reuters-21578 Text Categorization Collection Data Set
Feature (i.e. text) preprocessing
1. String separation
2. Class assignment
3. Split function
List of features in decreasing order
Features selected

Fig. 2: Block Diagram of Index Split Value-Based Decision Tree Model.

$$\text{keyword }(w)(p) = \begin{cases} 1, \text{if a string p contains w} \\ 0, \text{otherwise} \end{cases} \quad (1)$$

Let ‘$DT = \{\text{keyword }(w)\}$’. Thus, from the above given equation (1), a feature selection tree in HDT-DNN framework consists of a binary tree. Given a set of classes ‘C’ and collection of text documents ‘D’ obtained from Reuters-21578 Text Categorization Collection Data Set, the objective is to find the correct topic or topics for each document. Then, the fixed set of classes is mathematically written as given below.

$$C = \{c_1, c_2, \dots, c_k\} \quad (2)$$

From the above given equation (2), with the set of predefined classes ‘$c_1, c_2, \dots, c_k$’, the objective of HTD-DNN framework is the approximation of unknown class assignment function with the predefined classes and set of documents. With this, the function is mathematically formulated as given below.

$$\text{fun}: D * C \rightarrow \{0, 1\} \quad (3)$$

From the above given equation (3), ‘$D = \{x_1, x_2, \dots, x_n\}$’ represents the set of all documents and ‘C’ represents the set of all classes respectively. Also, from the above equation, the resultant value of ‘fun (D, C)’ is ‘1’ if the document ‘D’ belongs to a class ‘C’. On the other hand, if the resultant value of ‘fun (D, C)’ is ‘0’, the document ‘D’ does not belongs to a class ‘C’.

Based on the resultant values, each internal node is assigned with a string whereas each leaf node is assigned with a class name. Each feature selection tree selects an input string as follows: With the basic assumptions that the decision trees make several vertical and horizontal cuts on the data domain, the data domain is mapped to several classes (responses). In HDT-DNN framework, two sub domains are generated. These two generated sub-domains form a left and a right tree. Hence, these decision trees select the features by developing multiple sub-domains.

An input string ascertains a distinctive path from the root node to a leaf node. Now a horizontal cut is chosen to cut the left sub domain (left tree). At each internal node the right side ‘RS’ to a child is considered if the input string contains the string labeled at the right side node as a substring. Similarly, the right sub domain is divided with a horizontal cut. On the other hand, at each internal node the left side ‘LS’ to a child is considered if the input string contains the string labeled at the left side node as a substring.

Finally, the class that the input string belongs tois the class the leaf reaches, either the right side node or the left side node. In order to select the features for horizontally structured set of classes, parallel indexing or Index Split Value-based Decision Tree model is applied in the HDT-DNN framework. In Index Split Value-based Decision Tree model, first, the features are split according to indexing or based on the descending order. Therefore, parallel indexing is said to take place, where the split function is evaluated on the one hand and indexing takes place on the other. The left split and right split is mathematically formulated as given below.

$$\mathrm{LS}_{\mathrm{sv,f}}(\mathrm{D}) = \ \mathrm{p} \ \in \mathrm{D} \tag{4}$$

$$\mathrm{RS}_{\mathrm{sv,f}}(\mathrm{D}) = \mathrm{D} - \mathrm{LS}_{\mathrm{sv,t}}(\mathrm{D}) \tag{5}$$

From the above equation (4) and (5), the split value 'sv' for arriving at the features 'f' to be selected is mathematically formulated as given below.

$$\mathrm{RV}\ (\mathrm{sv,f,D}) = \ \mathrm{LS}_{\mathrm{sv,f}}(\mathrm{D}) * \ \mathrm{RS}_{\mathrm{sv,f}}(\mathrm{D}) \tag{6}$$

From the above obtained resultant value 'RV', ranking is performed on the basis of the maximum of the 'RV' and finally, the list of features in decreasing order is returned. The pseudo code representation of Index Split Decision Tree algorithm is given in the algorithm below 1.

```
Input: Document 'D = {$x_1, x_2, \ldots, x_n$}', Root Node 'RN', Internal Node 'IN', Leaf Node 'LN'
Output: optimal features selected (F = $f_1, f_2, \ldots, f_n$)
Begin
  For each document '$D$' with root node '$RN$'
    Perform separation of strings using equation (1)
    Obtain set of categories using equation (2)
    Obtain the resultant value of '$fun\ (D, C)$' using equation (3)
    If '$fun\ (D, C) = 1$ '
      Document '$D$' belongs to a class '$C$'
    End if
    If '$fun\ (D, C) = 0$ '
      Document '$D$' does not belongs to a class '$C$'
    End if
    For each right side node or the left side node
      Measure split value using equation (4) and (5)
    End for
    Measure the resultant value '$RV$' using equation (6)
    Return the list of features in the decreasing order of $RV$
  End for
End
```

Algorithm 1: Index Split Decision Tree algorithm.

From the Index Split Decision Tree algorithm given above, one of the great features of decision tree algorithms is that it intrinsically evaluatesthe appropriateness of features for the efficient separation of objects or text corresponding to several classes. This opportunity is directly used in Index Split Decision Tree algorithm for the purpose of feature selection. As a result, the horizontal representation of classes, i.e., feature splitting and indexing is performed in a parallel manner, thereby boosting the performance. Index Split Decision Tree algorithm automates analytical model building that continuously evaluates and learns from data and also accesses hidden insights. In this way, good feature representation is said to be learnt for a given task and therefore minimizes the time taken to arrive at the optimality of features being selected.

Compositional deep neural network

Uponsuccessful construction of decision tree and optimal features being selected with it, the Deep Neural Network is applied to produce hierarchical classification. In the HDT-DNN framework, Deep Neural Networks initiates compositional model by grouping multiple instances, where the text hasto be categorized, from the constructed decision tree model.

The foremost contribution of the HDT-DNN framework is hierarchical classification of documents. Q-statistic for Feature Selection (Q-FS) works well for a limited number of classes, but performance falls with growing number of classes. In the Compositional Deep Neural Network model, the problem is addressed by generating architectures that train deep learning on the basis of the level of the document hierarchy. Figure 3 shows the block diagram of Compositional Deep Neural Network for Text Categorization.

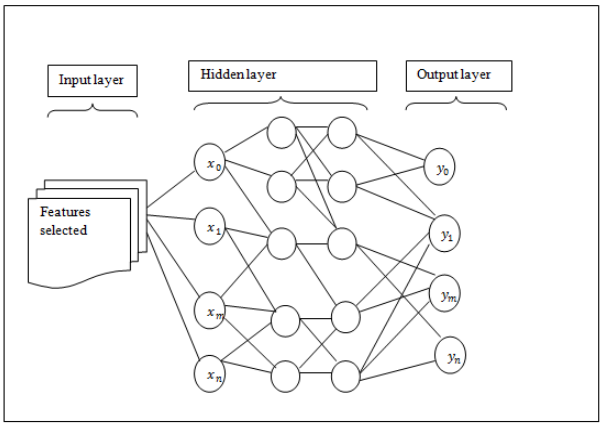

Fig. 3: Block Diagram of Compositional Deep Neural Network for Text Categorization.

As shown in the above block diagram, the Compositional Deep Neural Network for Text Categorization is trained with the standard back propagation algorithm using both Error Function (Equation 7) and Binary Step (Equation 8) as activation functions. The output layer uses Normalized Exponential Function (Equation 9).

$$f(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{7}$$

$$f(x) = \begin{cases} 0, for\ x < 0 \\ 1, for\ x \geq 0 \end{cases} \tag{8}$$

$$\sigma\ (x_i) = \frac{e^{x_i}}{\sum_{n=1}^{N} e^{x_n}}, for\ i = 1,2, \ldots, n \tag{9}$$

As given in above equation, (7), (8), (9) '$x\ \in X, X\ \in F$ ', at time 't'. In the proposed framework, the Compositional Deep Neural Network model uses both Error Function and Binary Step as activation functions. The purpose of using both Error Function and Binary Step is to reduce the classification error and, finally, to improve the classification accuracy of text being categorized. The Compositional Deep Neural Network model then applies the probabilistic classifier based on the Bayes theorem for efficient text categorization. Suppose the number of documents is 'n' and each document has the label 'c', '$c\ \in c_1, c_2, \ldots., c_k$', where '$k$' corresponds to the number of labels, then conditional probability is carried out mathematically for the given number of documents and classes as given below.

$$Prob\ (C \mid D) = \frac{Prob\ (D \mid C)\ Prob\ (C)}{Prob\ (D)} \tag{10}$$

From the above conditional probability value, mapping of compositional values are performed for text categorization and is mathematically represented as given below.

$$T = \ C_{MAP} = \ Prob\ (C \mid D) * \sigma\ (x_i) \tag{11}$$

$$T = \ C_{MAP} = Prob\ (x_1, x_2, \ldots, x_n \mid D) \ * \ \sigma\ (x_i) \tag{12}$$

With the above said compositional model followed in addition to the conditional probability model, the Compositional Deep Neural Networks enables composition of features or text from lower layers. This provisioning therefore possess the advantage of designing complex text with fewer units, therefore resulting in the overall improvement in the classification accuracy of the text being categorized. The pseudo code representation of Conditional Probable Compositional Text Categorization algorithm is given below.

```
Input: Document ‘$D = \{x_1, x_2, \ldots, x_n\}$’, Classes ‘$C = c_1, c_2, \ldots., c_k$’
Output: text to be categorized ‘$T = \{t_1, t_2, \ldots, t_n\}$’
1: Begin
2: For each features selected ‘$F$’
3: Measure Error Function using equation (7)
4: Measure Binary Step using equation (8)
5: Measure Normalized Exponential Function using equation (9)
6: Measure conditional probability using equation (10)
7: Perform mapping of compositional values with resultant conditional probability using equation (11)
8: End for
9: End
```

Algorithm 2: Conditional Probable Compositional Text Categorization Algorithm.

As given in the above Conditional Probable Compositional Text Categorization algorithm 2, the compositional text categorization refers to the features or text that is modeled as an explicit combination of features in the lower layer. The advantage of this compositional model for text categorization in the proposed framework was that it provided rapid reasoning through indexing inherently producing easy restoration from partial observations and visualization of features. As a result, a number of features used repeatedly for text categorization are reduced resulting in the improvement of precision and overall accuracy.

4. Experimental setup

In this section, experimental results are presented to show the effectiveness of the Hierarchical Decision Tree and Deep Neural Network (HDT-DNN) framework. A well known dataset Reuters-21578 Text Categorization Collection for text categorization is used in the following experiments. The comparison of the proposed Hierarchical Decision Tree and Deep Neural Network (HDT-DNN) framework is made with two other text categorization methods, namely Q-statistic for Feature Selection (Q-FS) [1] and Maximum Discrimination (MD) [2].

As described in Section 1, Q-FS involves high dimensional data classification and MD is one of the state of the art feature selection methods for text categorization approaches. For convenience, the proposed framework is abbreviated as HDT-DNN, standing for Hierarchical Decision Tree and Deep Neural Network. A computer with Intel(R) Core(TM) 2 Quad CPU Q6600 2.40 GHz, 4 GB of RAM is used to conduct the experiments. The programming language used is JAVA.

Dataset details

Experiments are conducted using the Reuters-21578[3], test collection of Distribution 1.0. The collection of distribution 1.20 appeared in Reuter’s newswire in the year 1987. The collection comprises 22 data files, a Standard Generalized Markup Language (SGML), Document Type Definition (DTD) file corresponding to the format of the available data, and six files providing the categories used for indexing. The Reuters-21578 collection is available at http://www.daviddlewis.com/resources/testcollections/reuters21578/. In the Reuters-21578, we Mod Apte split (9603 training and 3299 testing documents) is used, and two category sets---the 10 largest categories and 90 categories---with at least one training example and one testing example.

Evaluation metrics

The following metrics are used to compare the categorization effectiveness of each method. The performance measures precision, Recall, classification time and classification accuracy are defined below:

Precision ‘P’ refers to the ratio of the predicted documents for a given class that are categorized correctly‘$C_{cat\ correct}$’ to the total class found ‘$Total_{C\ found}$’. Precision is mathematically formulated as given below and is measured in terms of percentage (%).

$$P = \frac{C_{cat\ correct}}{Total_{C\ found}} \tag{13}$$

Recall ‘R’ refers to the ratio of the documents for a given class that are classified or categorized correctly ‘$C_{cat\ correct}$’ to the total correct class ‘$Total_{C\ correct}$’. Recall is mathematically formulated as given below and is measured in terms of percentage (%).

$$R = \frac{C_{cat\ correct}}{Total_{C\ correct}} \tag{14}$$

Classification accuracy ‘CA’ refers to the overall classification performance of the text being categorized. In other words, it is the ratio of text categorized correctly ‘$T_{cat\ correct}$’ to the overall text ‘$overall\ text$’ in the document. It is measured in terms of percentage (%) and mathematically expressed as given below.

$$\mathrm{CA} = \frac{\mathrm{T_{cat\,correct}}}{\mathrm{overall\ text}} \tag{15}$$

Classification time refers to the time taken for the text to classify. In other words, classification time involves the product of time taken for the text to categorize correctly and the document size. It is measured in terms of milliseconds (ms) and is mathematically expressed as given below.

$$\mathrm{CT} = \mathrm{Time}\ (\mathrm{T_{cat\,correct}}) * \mathrm{D_{size}} \tag{16}$$

5. Results and analysis

In the experiments, comparison of machine learning based text categorization method with two other text categorizations with machine learning approach using the global combination functions of the activation and Normalized Exponential Function is investigated. To verify the performance of HDT-DNN, comparison is made with the standard Q-statistic for Feature Selection (Q-FS) [1] and Maximum Discrimination (MD) [2]. All experiments are carried out in JAVA environment running in a .40 GHz CPU and 4 GB memory. For each experiment, the test is run 10 times their average values are taken as the results.

Performance analysis of precision

In this section, the precision efficiency for text categorization is presented, where the experimental setup includes observations in the range of 100 – 1000. The results of ten experimental runs conducted to measure the precision efficiency is shown in figure 4. The precision efficiency obtained using the framework HDT-DNN offers comparable values unlike the state-of-the-art methods. Figure 4 shows the results of precision on Reuters-21578 Text Categorization Collection Data Set with different number of features in the range of 100 to 1000. In this figure, the x-axis indicates the number of features used and the y-axis indicates the precision. Two-thirds of the documents were used for training and the rest for testing.

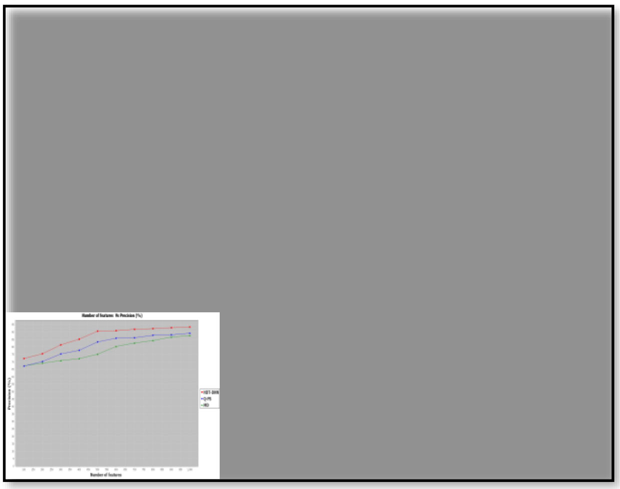

Fig. 4: Precision Efficiency of HDT-DNN, Q-FS and MD Varying Number of Features.

Figure 4 depicts the precision performance of three different text categorization methods HDT-DNN, Q-FS and MD. All methods obtained their best value, when the number of features was in the range of 100 to 1000. Though precision was increasing the number of features, it was found to remain an average. However, the proposed HDT-DNN framework outperformed all the contrast methods with the best precision value of 93.33%, whereas Q-FS and MD obtained 89.25% and 87.67% respectively, when the entire training set was used. All of the methods obtained better results when the size of number of features increased. In all training cases, though downtrends were not observed with the increase in the number of features, improvement was observed with very small changes in the precision value.

To obtain different numbers of selected features, different split values ‘sv’ are used in HDT-DNN. The number of features selected is identical to the number of classes. For the other methods, the number of selected features should be specified in advance by the user either based on ‘Q’ statistic or a hypothesis. Based on the split value, ranking is performed in HDT-DNN framework. Therefore the separation of objects is done in an efficient manner for any number of features. Obviously, the HDT-DNN framework has a higher precision rate than the Q-FS and MD. For example, the HDT-DNN framework acquires 92.82% of precision for 800 features, while Q-FS acquires 88.14% of precision and MD 86.56%. For 900 features HDT-DNN framework acquires 97.82% of precision, but Q-FS and MD acquires 96.14% and 95.56% respectively.

Performance analysis of classification accuracy

This section assesses the classification accuracy of the HDT-DNN framework when using different number of features. Classification accuracy is used as a statistical measure of how well a binary classification test correctly rejects or includes a condition. In other words, the classification accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases (features or text) examined.

The accuracy of the service functionality reflects the degree of proximity to the user’s actual values when using a service, compared to the expected values in terms of the features being selected. The more the correlation of the features being selected, the higher is the classification accuracy and, therefore, the higher the amount of text being categorized. For text categorization, accuracy’s first indicator is the number of features being selected deviating from a promised feature. It is defined as the frequency of failure in fulfilling the promised features to be selected from a document for a specified class. The classification accuracy for the features being selected is separately evaluated for 100 features. All the results are then averaged over 100 features. Figure 5 given below depicts the classification accuracy efficiency obtained using HDT-DNN framework and two different state-of-the-art methods Q-FS and MD.

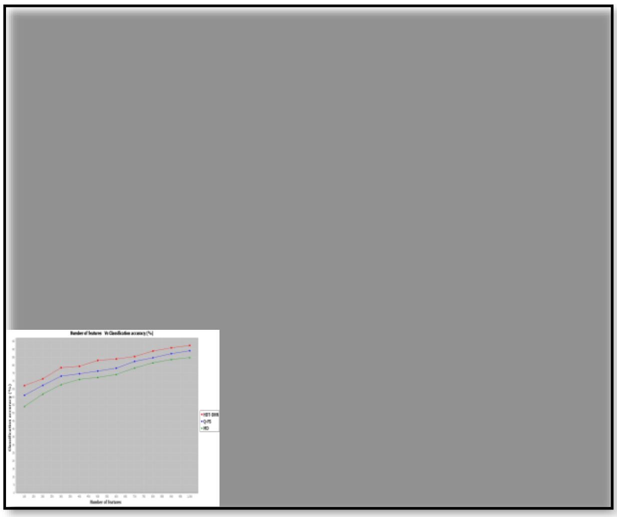

Fig. 5: Classification Accuracy Efficiency of HDT-DNN, Q-FS and MD Varying Number of Features.

To further verify the practicability of HDT-DNN framework, experiments were conducted among 100 to 1000 features. Firstly, in this experiment, two sub-domains are generated to form a left and right tree. Then, this experiment is repeated for 10 different documents, with the size of each ranging from 5MB to 50MB. Figure 5 shows the average classification accuracy of different documents and for different number of features respectively.

From the figure, it is noted that the classification accuracy differs among different text categorization methods like HDT-DNN, Q-FS and MD. This is due to different features acquired from different documents of different sizes. However, the classification accuracy obtained using HDT-DNN framework is found to be comparatively better than when using Q-FS and MD. This is due to the incorporation of error and binary function while applying Compositional Deep Neural Network model to the selected features. This has a positive impact on the features being selected and, therefore, improves the classification accuracy of HDT-DNN framework by 6% compared to Q-FS and 13% compared to MD respectively.

Performance analysis of classification time

To evaluate the efficiency of the HDT-DNN framework, in this section, the classification time for text categorization over a dataset Reuters-21578 collection is discussed. The classification time for text categorization is the time taken to classify the text present in the document in order to categorize the text and identify its presence in the document. The processing time needed to generate the split function and indexing is included in the total classification time. On the contrary, the processing time needed for the construction of both the left side and the right side of the node (i.e. for decision tree construction) is not included in the total classification time as the construction of decision tree can be regarded as an offline process in the proposed HDT-DNN framework. Figure 6 shows the classification time needed by HDT-DNN framework and the classification time needed by the two existing feature selection methods for categorizing the text.

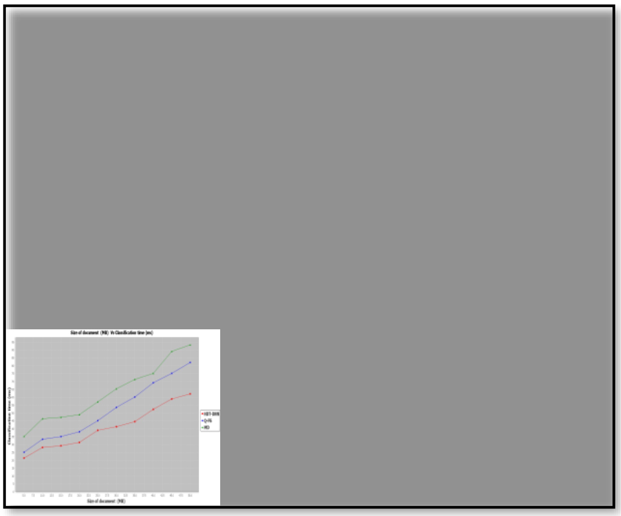

Fig. 6: Classification Time Efficiency of HDT-DNN, Q-FS and MD Varying Number of Features.

As shown in the figure, the HDT-DNN framework runs much faster than Q-FS and MD. For example, for 100 selected features with a document size of 5MB, Q-FS requires 25.16ms and MD requires 35.13ms, but the HDT-DNN framework needs only 21.35ms. As the number of selected features increases, Q-FS and MD run significantly slow.

As can be seen in Figure 6, for text categorization using HDT-DNN framework, achieving classification times of about 62.14ms is possible, when using a document with the size of 50MB with a total of 1000 features (i.e. text). For Q-FS and MD, the resulting classification time is about 82.13ms and 93.24ms respectively with a total of the same 1000 features. Compared to Q-FS and MD, the classification time of HDT-DNN framework highly depends upon the number of selected texts that is obtained using

Index Split Decision Tree algorithm. The Index Split Decision Tree algorithm by automating an analytical modeling performs feature splitting and indexing in a parallel manner. This in turn returns optimal features, thereby reducing the classification time using HDT-DNN framework by 20% compared to Q-FS and 36% compared to MD.

6. Conclusion

This paper proposed a measure, Index Split Value for feature selection approaches based on the index measures for efficient separation of objects, aiming to select the features that offered the maximum discriminative capacity for text categorization. The horizontal structured set of classes was also derived, leading to the other version of the Split Value Criterion for feature selection. Grouping of multiple instances using Compositional Deep Neural Network was then performed to train the features using deep learning based on the document hierarchy. Compared with the existing feature selection approaches that provided stability for selected feature subset and ensure prediction accuracy without considering the time taken to arrive at the optimality of features being selected, the HDT-DNN framework provided learning model in the horizontal representation of classes, ensuring a theoretical way to analyze optimality of selected features and therefore text to be categorized. Experiments conducted revealed the efficiency of the proposed framework in terms of precision, classification accuracy and classification time compared to the state-of-the-art works.

References

[1] HyunJi K, Byong SC & Moon YH, “Booster in high dimensional data classification”, *IEEE Transactions on Knowledge and Data Engineering*, Vol.28, No.1, (2016).

[2] Bo T, Steven K & Haibo H, “Toward Optimal Feature Selection in Naive Bayes for Text Categorization”, *IEEE Transactions on Knowledge and Data Engineering*, Vol.28, No.9, (2016).

[3] Reuters-21578 text categorization test collection, Distribution 1.0. Reuters, (1997).

[4] Duy DAB, Guilherme DFA, Siddhartha J, “PDF text classification to leverage information extraction from publication reports”, *Journal of Biomedical Informatics, Elsevier*, (2016).

[5] Mehdi HA & Setareh H, “feature selection using particle swarm optimization in text categorization”, *JAISCR*, Vol.5, No.4, (2015).

[6] Chanawee C, Kitsuchart P & David RH, “A Comparative Study of Machine Learning Techniques for Automatic Product Categorisation”, *Springer*, (2017).

[7] Joseph DP & Taghi MK, “Improving deep neural network design with new text data representations”, *Journal of Big Data, Springer*, (2017).

[8] Adel HM, Omar AM & Tariq A, “Arabic Text Categorization using k-nearest neighbour, Decision Trees (C4.5) and Rocchio Classifier: A Comparative Study”, *International Journal of Current Engineering and Technology*, Vol.6, No.2, (2016).

[9] Aleksandr S, Tatiana L, Dmitry G, Roman R & Ivan M, “Machine Learning Models of Text Categorization by Author Gender Using Topic-Independent Features”, *Elsevier*, (2016).

[10] Gulin VV & Frolov AB, “On the Classification of Text Documents Taking into Account Their Structural Features”, *Pattern Recognition And Image Processing*, (2015).

[11] Hao C, Wen J, Canbing L & Rui L, “A Heuristic Feature Selection Approach for Text Categorization by Using Chaos Optimization and Genetic Algorithm”, *Hindawi Publishing Corporation Mathematical Problems in Engineering*, (2013),

[12] Guozhong F, Baiguo A, Fengqin Y, Han W & Libiao Z, “Relevance popularity: A term event model based feature selection scheme for text classification”, *Plos One*, (2017).

[13] Joon YC, Tae KY, Jeong GS & Jiyong K, Terry Taewoong Um, Tyler Hyungtaek Rim, “Multi-categorical deep learning neuralnetwork to classify retinal images: A pilot study employing small database”, *Plos One*, (2017).

[14] Doujie L, Zhongyan F & Wallace KST, “Domain learning naming game for color Categorization”, *Plos One*, (2017).

[15] Hari S, “Effective feature selection technique for text Classification”, *Int. J. Data Mining, Modelling and Management*, Vol.7, No.3, (2015).

[16] Ahmed HA & Esraa HAA, “Comparative Study of Five Text Classification Algorithms with their Improvements”, *International Journal of Applied Engineering Research*, Vol.12, No.14, (2017), pp.4309-4319.

[17] Alper KU, “An improved global feature selection scheme for text classification”, *Expert Systems with Applications, Elsevier*, (2015).

[18] Pradnya K & Manisha M, “A Survey on Feature Selection Techniques and Classification Algorithms for Efficient Text Classification”, *International Journal of Science and Research* (IJSR), Vol.5, No.5, (2016).

[19] Wenbin Z, Yuntao Q, Minchao Y & Hangzhou, H, “A Grouped Structure-based Regularized Regression Model for Text Categorization”, *Journal of Software*, Vol. 7, No. 9, (2012).

[20] Wenbin Z, Yuntao Q & Huijuan L, “Text categorization based on regularization extreme learning Machine”, *Neural Comput & Applic*., Vol.22, (2013), pp.447–456. https://doi.org/10.1007/s00521-011-0808-y.