*I.J. Intelligent Systems and Applications,* **2019, 5, 36-45**
Published Online May 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2019.05.05

# A Novel Text Representation Model to Categorize Text Documents using Convolution Neural Network

**M. B. Revanasiddappa and B. S. Harish**
Department of Information Science and Engineering,
Sri Jayachamarajendra College of Engineering, Mysuru 570006, India
E-mail: revan.cr.is@gmail.com, bsharish@sjce.ac.in

Received: 24 March 2018; Accepted: 23 May 2018; Published: 08 May 2019

***Abstract*—**This paper presents a novel text representation model called Convolution Term Model (CTM) for effective text categorization. In the process of text categorization, representation plays a very primary role. The proposed CTM is based on Convolution Neural Network (CNN). The main advantage of proposed text representation model is that, it preserves semantic relationship and minimizes the feature extraction burden. In proposed model, initially convolution filter is applied on word embedding matrix. Since, the resultant CTM matrix is higher dimension, feature selection methods are applied to reduce the CTM feature space. Further, selected CTM features are fed into classifier to categorize text document. To discover the effectiveness of the proposed model, extensive experimentations are carried out on four standard benchmark datasets viz., 20-NewsGroups, Reuter-21758, Vehicle Wikipedia and 4 University datasets using five different classifiers. Accuracy is used to assess the performance of classifiers. The proposed model shows impressive results with all classifiers.

***Index Terms*—**Text Documents, Convolution Neural Network, Representation, Feature Selection, Categorization.

## I. Introduction

Automatic Text Categorization (TC) is a process of assigning a new text document into one or more pre-defined classes based on its content [1-4]. From last two decades, text categorization has taken more attention by researchers due to huge number of text documents available on the World Wide Web (WWW). TC is successful technique to process and manipulate text documents over the internet. Usually, text documents are unstructured in nature, so it is very difficult to process and understand directly by machine. Hence, it is essential to represent unstructured text document into machine understandable structured form. Thus, representation of text documents is a major step in the process of text categorization [5]. There are various text representation models like Bag of Words [6], Vector Space Model [7], Binary Representation [8], Ontology [9], N-Grams [10], Universal Networking Language [11], Symbolic Representation [12] developed for effective text categorization.

On the other hand many researchers developed Artificial Neural Network (ANN) based text representation models [13-17]. However, existing text representation models fails to preserve the semantic relationships between terms in a text document. Semantic relationship captures the associations that exist between the terms, as well as the structure of terms, and also assist to address the impact of ambiguous terms. Among existing neural network methods, Convolutional Neural Network (CNN) based approaches have received more attention and successfully applied to categorize text documents [18, 19]. CNN is initially proposed by Lecun [20] and used convolutional filters to extract the local features.

In traditional ANN, the relationship between input and output units is determined by matrix multiplication. In CNN, convolution is used instead of general matrix multiplication. In this way, it reduces the number of weights and parameters in the network. In addition, it minimizes complexity of network, which leads to reduction in memory size and enhancement in performance. Moreover, learning algorithms avoid the feature extraction procedure due to directly considering input to the network. Another advantage of convolution is, it helps to learn semantic information of the text documents and also minimize the impact of ambiguous terms [21, 22]. The basic idea of convolution is single-hand sliding window concept, which splits text documents into flexible phrases. Further, convolution also helps to learn representation at multiple levels [23].

The theory of CNN based text categorization is absolutely similar to that of computer vision task [24]. CNN is feed-forward neural network and it consists of convolution layer, pooling layer and activation function [25, 26]. In CNN, convolutional layer is composed of various different convolution filters, which are employed to calculate different feature map. Convolution filter is applied on input text to extract the most significant terms

 

and the extracted terms are represented in hierarchical form. In particularly, each term of feature map is associated to a region of neighboring terms. The primary objective of using pooling layer is to accomplish shift-invariance and to induce the fixed length vector form. Usually, pooling layer is placed in between two convolution layers. After several convolution and pooling layers, softmax function is used to categorize text documents [26]. CNN reduces the feature extraction burden and also it preserves the semantic relationship. Due to these advantages, CNN is widely applied to categorize text documents. Eventhough the CNN performs better but unfortunately it suffers from high computation time.

In CNN, convolutional layer helps to capture the semantic relationship. It is our notion that, instead of using whole CNN model, we can use only convolution layer results and follow the traditional text categorization process on the results. By this process we can preserve semantic relationship and also we can reduce the time complexity. Based on this notion, in this paper we have proposed a new Convolution Term Model (CTM) to represent a text document. In the proposed model, initially embedded matrix (Term Document Matrix) is constructed by applying pre-processing techniques like stemming and stop word elimination [27]. Further to capture the semantic relationship we apply convolution filter to the Term Document Matrix (TDM). Eventhough, resultant convolution feature matrix has less dimension than original TDM but still it is in higher dimensional. To reduce the high dimensionality (feature space), we employed feature selection methods. The feature selection method selects the feature subset from CTM resultant matrix. The feature selection plays a vital role to speed up the process of computation as well as to improve the performance of classifier [28-31]. Finally, classifiers are used to find effectiveness of the proposed model in terms of categorization accuracy.

Overall, the major contributions of this paper are summarized as follows:

- We propose a novel text representation model called Convolution Term Model (CTM). The main objective of this model is to reduce the feature extraction burden and to preserve the semantic relationship.
- To handle higher dimensionality of CTM resultant matrix, we employed feature selection methods: Chi-Square, Information Gain (IG) and Distinguish Feature Selection (DFS).
- We evaluated the effectiveness of proposed model using five different classifiers through conducting experiments on four standard benchmark datasets.

The rest of the paper is organized into five sections. Followed by the introduction, related literature is thoroughly reviewed in Section 2. Section 3 presents the proposed model followed by feature selection method and categorization. Further, the experimental results and discussion are given in Section 4. Finally, we conclude the paper followed by future work in Section 5.

## II. Related Works

In literature, various intuitive models have been proposed for text representation [5, 32]. Bag of Words (BoW) is widely used representation model in text categorization [6]. Unfortunately, BoW suffers from loss of information, high dimensionality and fails to identify the semantic relationship between terms in text documents. On the other hand, many researchers developed neural network based text representations methods [13-16]. In an attempt to utilize the power of neural network for text representation, Le and Mikolov [13] proposed a simple approach that learns sequence distributed vector representation for text. This approach extracts the ordering of words and also semantic information of the words in an efficient way. Gupta and Varma [14] developed a Doc2Sent2Vec text representation model to learn document representation. This model consists of two steps, in the first step, the model learns sentence embedding with the help of standard word-level language model. In the second step, the model learns document representation with help of sentence level language model. Keller and Bengio [15] proposed a novel non-probabilistic representation model. This model provides the rich internal representation of terms (words) and documents using neural network. Li et al., [16] proposed Text Concept Vector model which represents the concept level of text. In this model, initially input text is mapped to conceptualized text. Further, taxonomy knowledge base is used to extract the concept of text. Finally it generates the concept level representation of text by making use of neural network.

Recently, Convolution Neural Network (CNN) has provided new solutions and taken more attention in text categorization task. Convolution Neural Network (CNN) is one of the popular Neural Network technique [26]. It considers each term fairly through convolutional layer, and leverages sliding windows with varying width and filters to generate feature map. Further, pooling task is utilized to obtain an output. CNN also makes contribution to text representation. Kim [33] utilized convolutional neural network and proposed a new approach for sentence classification. This approach used single convolutional layer which make use of multiple width and filters. Later, max pooling layer extracts the informative features. Finally, extracted features are fed into output layer. Zhang et al., [18] presents the empirical study on character-level CNN for text categorization. The various traditional and deep learning models were compared and applied on large datasets. However, analysis result shows that character-level CNN achieved better results on large datasets.

Johnson and Zhang [34] presents bag-of-word conversion in the convolution layer and CNN is applied directly to high-dimensional features, without using one-dimensional (pre-trained) word vectors like word2vec. The same work is further enhanced by integrating with unsupervised region embedding of words [35]. Zhang et

 

al., [36] explored the use of character level CNN, without using any pre-trained embeddings. The proposed model uses the deep networks for text categorization and sentiment analysis. Huang et al., [21] proposed a character-aware Convolution Neural Network model, which has three stages. In the first stage, the model generates sentence semantic representation considering sentence-level as an input and it depends on only character. In the second stage, abnormal characters are considered and these are combination of misspelling, ungrammatical expression and emotion icons. Lastly, the model is computed on Microsoft Research Paraphrase (MSRP) and Paraphrases on Twitter (PIT) data. Mass et al., [22] proposed the word representation model to capture the semantic and sentiment relationship of words. This model generates the vectors based on unsupervised probabilistic approach.

Zhang et al., [37] proposed a new CNN model named as Rationale Augmented-CNN (RA-CNN) for text categorization. In this model, the concept of rationale is integrated into neural network model. The model start by computing the probability of rationale and contribution score of each sentence. Further, document is represented by aggregating all the sentences. Li et al., [38] presents the document representation model based on neural network for deceptive spam review. This model estimates the important weight of each sentence to capture the semantic information and then integrate them for document representation. Most of the existing CNN based models are applied to supervised learning for Natural Language Processing (NLP) applications. While, Xu et al., [39] used the power of CNN on unsupervised learning NLP application like Short Text Clustering.

From the literature review, it is observed that lot of works are reported on CNN for text categorization. In CNN, convolution layer discovers the composite features through convolution filter from padded text and these features can describe the hidden semantic relationship of terms in the text document. In addition to that it also captures multi-scale contextual information. Considering these advantages of convolution layer, in this paper, we propose a novel text representation model called Convolution Term Model (CTM). The next section presents the proposed model in detail.

## III. Methodology

In this section, we describe the details of text categorization process, which includes Pre-processing followed by CNN based text representation model, Feature Selection and Categorization.

### A. Pre-processing

In text documents each term is considered as a feature. But some terms are irrelevant and unwanted. Thus, it is essential to apply pre-processing to remove unwanted and irrelevant terms. The pre-processing techniques like stemming and stop word elimination is applied. After pre-processing, word embedding matrix (term document matrix) is constructed. Let us consider that there are $N$ number of documents which belongs to k number of pre-defined classes i.e., $C = C_1, C_2, C_3 ..., C_k$ . Each class contains $n$ number of documents i.e., $D = D_1, D_2, D_3, ..., D_n$ and $m$ number of features (terms) $T = t_1, t_2, t_3, ..., t_m$ . The word embedding matrix $Q$ of size $N \times m$ is constructed as follows:

$$Q(i, j) = tf(T_i, D_j). \quad 1 \le j \le N, 1 \le i \le m \tag{1}$$

Where, $tf(T_i, D_j)$ is the frequency of $i^{th}$ term in the $j^{th}$ document. Each entry in the matrix represents the appearance count of the term in the document. This word embedding matrix representation fails to capture the semantic relationship. Thus, to capture semantic relationship of terms in the text document, in the next step we are proposing a new representation model called as Convolution Term Model (CTM).

### B. Representation

The CTM is based on Convolution Neural Network (CNN), which captures the semantic relation between terms in the text documents. The basic idea of capturing the semantic relationship is to define an operation like convolution to perform semantic composition over input matrix, where window is used with varying width. The convolution operation computes inner product of filter matrix and input matrix. This inner product helps to preserve the semantic relationship of terms and also it exploited the multi-scale contextual information, which minimizes the impact of ambiguous terms in the text documents.

The input to CTM is a word embedding matrix $Q$ . In CTM, convolution transformation is applied over the word embedding matrix $Q$ , where filter '$w$' is applied to a window of '$p$' terms to produce a new feature. This new feature presents the semantic relation and composite features of individual terms in a given documents. Let $T_i$ is the new term feature generated from a window of terms $t_{i:i+p-1}$ and it is computed as follows:

$$T_i = f(w.t_{i:i+p-1} + \alpha). \tag{2}$$

Where, $\alpha$ is a bias term and $f$ is a non-linear function. The convolution filter is applied to each possible window of terms in a document $\{t_{1:p+1}, t_{2:p+2}, ..., t_{m-p+1:m}\}$ to produce a convolution feature map $F$ , which consists convoluted features i.e.,:

$$T = \{T_1, T_2, T_3, ..., T_{m-p+1}\}. \tag{3}$$

The equations 2 and 3 help to capture the semantic relationship of the terms in the text document. Fig. 1. illustrates the representation of text documents using proposed model by considering four text documents:

 

$\{D_1, D_2, D_3, D_4\}$ and these documents presented in Table 1.

Table 1. Illustration

| |
|---|
| $D_1$ : Sachin Tendulkar is a god of cricket game |
| $D_2$ : Kohinoor diamond is very famous in the world |
| $D_3$ : Sachin is a very famous cricketer in the world |
| $D_4$ : Sachin Tendulkar is the Kohinoor diamond of the game of cricket |

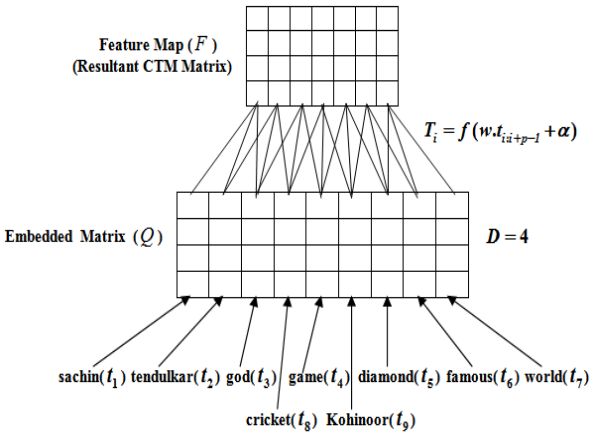


Fig.1. Text representation using proposed model (CTM).

The proposed model (CTM) begins with stemming and stop words are eliminated in the given documents. Further, dictionary will be formed and the newly formed dictionary contains 'sachin' ( $t_1$ ), 'tendulkar' ( $t_2$ ), 'god'( $t_3$ ), 'cricket'( $t_4$ ), 'game' ( $t_5$ ), 'Kohinoor' ( $t_6$ ), 'diamond' ( $t_7$ ), 'famous' ( $t_8$ ), 'world' ( $t_9$ ), and these are represented into word embedding form $\square^{D\times m}$ i.e., $4\times 9$, where, $D$ is the number of documents in word embedding matrix and in the above example, we use $D$ =4. These terms are given as input to the proposed model. The main idea of proposed model is to discover the semantic relationship of terms. After applying convolution, we get convolution feature map $F$ of size $\square^{D\times j}$ , where $j$ is number of convoluted features (semantic).

The size of convolution feature map $F$ is less than original word embedding matrix $[N\times m] > [N\times j]$ , where $j = m - p + 1$. However, still the dimension of the convolution feature map is high. The higher dimensionality of convolution feature map not only degrades the categorization performance, but also increases the computational time complexity. To address this problem, in the next step we applied feature selection methods on convolution feature map $F$ .

### C. Feature Selection

In this paper, we employed well known feature selection methods like: Chi-Square ($\chi^2$) [31], Information Gain (IG) [40] and Distinguish Feature Selection (DFS) [41] methods.

*Chi-Square* ($\chi^2$) : Chi-Square ($\chi^2$) [31] selects the discriminative features according to its correlation with respective class. The Chi-Square can be expressed with the following formula:

$$\chi^2(T_j) = \sum_{T_j\in\{0,1\}} \sum_{C_k\in\{0,1\}} \frac{\left(O_{T_j,C_k} - E_{T_j,C_k}\right)^2}{E_{T_j,C_k}} . \quad (4)$$

Where, $O_{T_j,C_k}$ is the number of observed frequency for each term $T_j$ and class $C_k$ and $E_{T_j,C_k}$ is the expected frequency for each term $T_j$ and class $C_k$ . Chi-Square ( $\chi^2$ ) computes the expected frequency $E$ and observed frequency $O$ which varies from each other.

*Information Gain (IG)*: IG [40] computes the quality of bits of information acquired by knowing the presence or absence of term (feature) in the document for categorization decision. IG of terms ( $T_j$ ) is computed as:

$$IG(T_j) = \sum_{C\in\{C_k,\bar{C}_k\}} \sum_{T\in\{T_j,\bar{T}_j\}} P(T_j, C_k) * \log \frac{P(T_j, C_k)}{P(T_j)P(C_k)} . \quad (5)$$

Where, $P(T_j, C_k)$ is the joint probability of class $C_k$ and occurrence of term $T_j$ , $P(T_j)$ is the probability of term, $P(C_k)$ is the probability of class, $\bar{T}_j$ indicates term is not present and $\bar{C}_k$ indicates class is not present.

*Distinguish Feature Selection (DFS):* DFS [41] determines the feature, which is discriminating between classes and also semantically similar to the document. The DFS of term ( $T_j$ ) is computed as:

$$DFS(T_j) = \sum_{i=1}^{k} \frac{P(C_i \mid T_j)}{P(\bar{T}_j \mid C_i) + P(T_j \mid \bar{C}_i) + 1} . \quad (6)$$

Where, $P(C_i \mid T_j)$ is the conditional probability of class $C_i$ given presence of term $T_j$ , $P(\bar{T}_j \mid C_i)$ is the absence of term in conditional probability and $P(T_j \mid \bar{C}_i)$ is the absence of class in conditional probability.

The selected feature subsets from feature selection methods (Chi-Square, IG and DFS) are represented in $T = T_1, T_2, ..., T_z$ , where $z$ is the number of selected features $(z < j)$ . In the next step selected feature subset are fed into classifier.

### D. Categorization

In this paper, to evaluate the efficacy of proposed model, we employed five most widely used classifiers viz., Naïve Bayes (NB) [42, 43], k-Nearest Neighbor (kNN) [44], Support Vector Machine (SVM) [45], RBF Neural Network (RBF NN) [46] and Convolutional Neural Network (CNN) [18, 21, 35]. The performance of

 

the classifier is evaluated in terms of categorization accuracy.

*Naïve Bayes (NB) classifier*: Naïve Bayes [42, 43] is a simple formal probabilistic classifier, which is based on Bayes theorem. NB models the distribution of documents in each class by make use of probabilistic model, which assumes that distribution of features are independent to each other in a document. The NB classifier can be described as follows:

$$c_{NB} = \arg\max_{C_k \in C} P(C_k) \prod_{l=1}^{z} P(T_l \mid C_k) . \quad (7)$$

Where, $P(T_l \mid C_k)$ indicates the priori probability of class $C_k$ and $P(T_l \mid C_k)$ indicates the conditional probability of term $T_l$ given class $C_k$.

*k-Nearest Neighbor (kNN):* kNN [44] is one of the simplest classifier used to categorize the text documents. It is similarity based classifier, which uses similarity (distance) measures to perform categorization. kNN evaluates the closeness of training documents by similarity measures (Euclidean, cosine etc.,) and assign a label to test document k neighbors. In kNN, k value represents the number of neighbor documents being compared. Let us consider a test document $X$ with $T_1, T_2, ..., T_z$ features. To predict the class of test document, kNN uses the class label of k closest neighbors. Finally, test document assigns to a class, which has highest score. The kNN classifier decision rule can be written as:

$$f(X) = \arg\max Score(X, C_k)$$
$$= \sum_{D_n \in kNN(X)} Sim(X, D_n)\, y(D_n, C_k) \ . \quad (8)$$
$$Where,\ y(D_n, C_k) = \begin{cases} 1 & D_n \in C_k \\ 0 & D_n \notin C_k \end{cases}$$

Where, $f(X)$ is the class label assigned to the test document $X$, $Score(X, C_k)$ presents the score of the class $C_k$ with respect to $X$, $kNN(X)$ presents the set of k-nearest neighbors of test documents, $Sim(X, D_n)$ presents the similarity between $X$ and training documents $D_n$ and $y(D_n, C_k)$ indicates the categorization for documents $D_n$ with respect to class $C_k$.

*Support Vector Machine (SVM)*: SVM [45] is a supervised learning categorization technique, which is extensively used in categorization problems. SVM is a form of linear classifier. A document $D$ is represented by set of frequency count of terms $T$. A single SVM can only separate two categories a positive category (represented by $y = +ve$) and negative (represented by $y = -ve$). In the space of input vectors, an optimal separating hyper-plane may defined by setting $y = 0$ in the following linear equation:

$$y = \vec{w} \cdot \vec{D} + b . \quad (9)$$

Where, $\vec{D}$ is the vector of document term frequency, $\vec{w}$ is the vector of coefficients and $b$ is the bias. The SVM attempt to determine the optimal separating hyper-plane with the maximum distance $\xi$ (named as margin), which is appeared between positive ($+ve$) and negative ($-ve$) example of the training set. The text documents with distance $\xi$ from optimal separating hyper-plane are called support vectors and they find out the actual location of the optimal separating hyper-plane. An unknown document is categorized to $+ve$ category, if it's computed function value is y>0, otherwise in to $-ve$ category.

*Radial Basis Function-Neural Network (RBF-NN):* RBF-NN [46] is a feed-forward neural network. It has three fixed layers input, hidden and output layer. Approximation ability is the major strength of RBF and Gaussian function as an activation function in the hidden layer.

$$y_v^* = \sum_{u=1}^{Y} \beta_{vu} \exp\left( -\frac{\|D - \mu_u\|}{2\sigma_u^2} \right). \quad (10)$$

Where, $y_v^*$ is the $v^{th}$ output, $\beta_{vu}$ is the interconnecting weight between $v^{th}$ output neuron and $u^{th}$ Gaussian neuron, $D$ is the number of input documents, $\mu_u$ and $\sigma_u$ are the center and width of the gaussian function of the $u^{th}$ neuron.

*Convolution Neural Network (CNN):* CNN [18, 21, 36] is one of the recent successful technique in neural network. It consists of multiple convolution layers, pooling layers and output layer. The convolution layer uses different convolution filters to generate new feature map. Further, max-pooling reduces the dimensionality of feature matrix size by extracting sub-sequence maximum values. In our experiments, max-pooling is applied over each row of the feature map $T$ that extracts sub-sequence of maximum values. Finally, activation function (Sigmoid function) is employed in output layer. In convolution layer, convolution operation computes inner product of filter matrix and input matrix, and it is presented in equation (2). The max-pooling layer extracts the maximum value, which is computed by following equation:

$$p_h^* = \max_i [T_{hi}], \ \forall i = 1, ..., (m - p + 1) . \quad (11)$$

Where, $p_h^*$ presented as the most informative and discriminate feature that extracted from the feature map $T$. In output layer, sigmoid function is applied on $p_h^*$ to compute class label, which is described as follows:

$$s_h = \frac{\exp(p_h^*)}{\sum_{q=1}^{T} \exp(p_q^*)}. \quad (12)$$

Where, $s_h$ defines the class label.

## IV. Experiments

### A. Dataset Description

To assess the effectiveness of the proposed model, we conducted experimentation on four different standard datasets viz., 20-NewsGroups, Reuter-21758, Vehicle Wikipedia and 4 University Datasets. The 20-NewsGroups is one of the popular standard dataset for text categorization. It contains 18846 documents which are distributed evenly into 20 classes [47]. Reuters-21578 dataset is collected from Carnegie Group Inc. and Reuters Ltd, and it contains 21578 documents which are distributed across 135 classes non-uniformly [48]. Vehicle Wikipedia dataset is extracted from Wikipedia pages and it consists 440 documents of vehicle characteristics, which spread across four categories of vehicle i.e., Aircraft, Trains, Cars and Boats with low degree of similarity [49]. The 4 University dataset contains 8282 WWW-pages collected from computer science department of various universities in January 1997 by the World Wide Knowledge Base (WebKb) project of the CMU text learning group [50]. The 8,282 pages were manually classified into 7 different classes such as student, faculty, staff, department, course, project and others.

### B. Experimental Setup

During the experimentation, it is necessary to split the dataset into training and testing set to validate the proposed method. The large training data results in overfitting of the model. On the other hand, small training data results in underfitting the model. The whole reason for split comes from the fact that, we often have limited and finite data. So we want to make the best use of it and train on as much data as we can. Thus, in our experiments, to validate the proposed model, we split the dataset into training and testing phase in 60:40 ratios respectively. The training set is 60% documents of each class of dataset, used to build our proposed model. On the other hand, testing set is applied on proposed model to assess the performance. In the proposed model we empirically fixed convolution filter size as 3. Since the size of CTM resultant matrix is high, further we employed three well known feature selection methods to reduce matrix size. By each feature selection method, initially, we conducted experiments by fixing 100 numbers of features from CTM matrix by empirically.

Further, we varied the number of features from 100 to 500 with an increment of 100. However, decreasing below 100 numbers of features and increasing above 500 features does not yield good results. Hence, we restricted number of features to vary between 100 to 500. To demonstrate the efficacy of proposed model, we used five different classifiers viz., NB, kNN, SVM, RBF-NN and CNN. We considered accuracy as evaluation metric to assess the effectiveness of the proposed model.

### C. Experimental Results

Table 2 shows the performance comparison of five different classifiers on 20-NewsGroups dataset. From Table 2, we can observe that CNN classifier performed better compared to other classifier with all 3 feature selection methods. CNN classifier with DFS method outperformed all the other classifiers and feature selection methods.

Table 2. Categorization Results on 20-NewsGroups

| Feature Selection Method | Number of features selected | Accuracy (%) | | | | |
|---|---|---|---|---|---|---|
| | | Naïve Bayes | kNN | SVM | RBF-NN | CNN |
| Chi–Square Method | 100 | 64.23 | 68.74 | 72.89 | 71.63 | 81.23 |
| | 200 | 65.72 | 69.85 | 74.56 | 73.57 | 82.98 |
| | 300 | 67.15 | 70.52 | 76.90 | 74.25 | 84.77 |
| | 400 | 68.90 | 72.35 | 77.32 | 75.11 | 86.52 |
| | 500 | 69.09 | 74.04 | 79.81 | 77.86 | 87.33 |
| Information Gain (IG) | 100 | 67.43 | 69.56 | 78.98 | 71.85 | 84.43 |
| | 200 | 68.77 | 70.65 | 79.12 | 73.45 | 85.96 |
| | 300 | 69.05 | 71.23 | 80.45 | 74.97 | 87.77 |
| | 400 | 70.86 | 72.89 | 81.56 | 76.05 | 88.90 |
| | 500 | 71.32 | 74.57 | 83.66 | 78.23 | 90.10 |
| Distinguishing Feature Selection (DFS) | 100 | 71.23 | 72.34 | 84.58 | 82.56 | **86.88** |
| | 200 | 73.89 | 73.58 | 85.66 | 83.73 | **87.90** |
| | 300 | 75.65 | 75.42 | 87.11 | 84.32 | **89.12** |
| | 400 | 76.67 | 78.90 | 88.42 | 85.54 | **90.88** |
| | 500 | 77.13 | 80.10 | 89.92 | 86.98 | **91.10** |

 

Table 3. Categorization Results on Reuters-21578

| Feature Selection Method | Number of features selected | Accuracy (%) | | | | |
|---|---|---|---|---|---|---|
| | | Naïve Bayes | kNN | SVM | RBF-NN | CNN |
| Chi–Square Method | 100 | 60.67 | 61.74 | 65.98 | 63.57 | 66.83 |
| | 200 | 61.28 | 63.54 | 66.39 | 64.96 | 67.98 |
| | 300 | 62.61 | 64.79 | 68.28 | 65.89 | 69.05 |
| | 400 | 64.33 | 65.32 | 69.61 | 66.20 | 70.35 |
| | 500 | 66.85 | 66.90 | 70.84 | 67.90 | 71.54 |
| Information Gain (IG) | 100 | 63.98 | 64.55 | 68.72 | 64.38 | 68.53 |
| | 200 | 65.19 | 65.27 | 69.51 | 65.21 | 69.29 |
| | 300 | 66.53 | 66.07 | 70.58 | 66.86 | 70.55 |
| | 400 | 67.85 | 67.32 | 71.83 | 67.98 | 71.28 |
| | 500 | 68.90 | 68.55 | 72.77 | 69.02 | 72.56 |
| Distinguishing Feature Selection (DFS) | 100 | 67.52 | 68.74 | 73.89 | 72.89 | **81.74** |
| | 200 | 68.90 | 69.08 | 74.90 | 74.56 | **82.95** |
| | 300 | 70.28 | 70.25 | 75.60 | 75.08 | **83.55** |
| | 400 | 71.77 | 72.95 | 77.02 | 77.26 | **86.90** |
| | 500 | 73.56 | 74.44 | 79.82 | 79.85 | **88.10** |

Table 4. Categorization Results on Vehicle Wikipedia dataset

| Feature Selection Method | Number of features selected | Accuracy (%) | | | | |
|---|---|---|---|---|---|---|
| | | Naïve Bayes | kNN | SVM | RBF-NN | CNN |
| Chi–Square Method | 100 | 72.98 | 71.56 | 78.98 | 76.05 | 84.59 |
| | 200 | 73.10 | 73.80 | 80.56 | 77.60 | 87.40 |
| | 300 | 73.99 | 75.66 | 82.44 | 78.35 | 88.66 |
| | 400 | 75.41 | 77.32 | 85.39 | 79.62 | 89.01 |
| | 500 | 78.04 | 79.06 | 88.90 | 80.45 | 89.88 |
| Information Gain (IG) | 100 | 74.07 | 76.90 | 80.63 | 72.80 | 83.45 |
| | 200 | 75.77 | 78.20 | 82.32 | 74.67 | 85.54 |
| | 300 | 77.89 | 79.43 | 84.06 | 75.35 | 87.16 |
| | 400 | 79.62 | 80.16 | 86.72 | 76.94 | 88.90 |
| | 500 | 80.05 | 82.22 | 88.56 | 77.90 | 89.91 |
| Distinguishing Feature Selection (DFS) | 100 | 78.88 | 80.44 | 84.06 | 83.21 | **88.48** |
| | 200 | 80.25 | 82.69 | 86.90 | 84.44 | **89.03** |
| | 300 | 82.36 | 83.90 | 88.22 | 85.73 | **90.21** |
| | 400 | 84.55 | 85.55 | 89.10 | 86.04 | **91.46** |
| | 500 | 86.30 | 87.88 | 90.26 | 87.50 | **92.58** |

Table 5. Categorization Results on 4 University dataset

| Feature Selection Method | Number of features selected | Accuracy (%) | | | | |
|---|---|---|---|---|---|---|
| | | Naïve Bayes | kNN | SVM | RBF-NN | CNN |
| Chi–Square Method | 100 | 54.98 | 61.03 | 65.93 | 62.56 | 64.58 |
| | 200 | 56.77 | 61.89 | 66.64 | 64.08 | 65.70 |
| | 300 | 57.77 | 63.22 | 67.40 | 65.83 | 66.48 |
| | 400 | 58.90 | 64.59 | 68.73 | 66.02 | 67.91 |
| | 500 | 60.11 | 66.20 | 69.04 | 67.90 | 68.77 |
| Information Gain (IG) | 100 | 55.28 | 57.62 | 67.51 | 65.35 | 68.44 |
| | 200 | 56.71 | 58.39 | 68.70 | 66.61 | 69.40 |
| | 300 | 57.83 | 59.32 | 69.03 | 67.88 | 70.26 |
| | 400 | 58.66 | 60.92 | 69.88 | 68.92 | 71.45 |
| | 500 | 59.85 | 61.66 | 70.52 | 69.73 | 72.39 |
| Distinguishing Feature Selection (DFS) | 100 | 64.88 | 65.22 | 70.47 | 64.55 | **70.56** |
| | 200 | 66.21 | 65.90 | **72.32** | 66.12 | 71.44 |
| | 300 | 67.01 | 66.48 | **73.80** | 67.83 | 72.58 |
| | 400 | 68.56 | 67.21 | **74.55** | 68.90 | 73.93 |
| | 500 | 69.33 | 68.37 | **75.89** | 69.98 | 74.60 |

 

Further, the same set of experimentation were carried out on Reuters-21578, Vehicle Wikipedia and 4 University dataset. For Reuters-21578, the comparison results of proposed model using five different classifiers with three feature selection methods are presented in Table 3. It can be seen from Table 3 the performance of proposed model using CNN classifier with DFS method shows better result of 88.10% for 500 features, compared with other classifiers.

Similarly, Table 4 presents, results on Vehicle Wikipedia dataset. The proposed model achieved better result of 92.58 using CNN classifier for 500 feature of DFS method. Table 5 shows the performance comparison results of proposed model using five different classifiers on 4 university dataset. From Table 5, we can note that the proposed model using SVM classifier obtained good result of 72.32%, 73.80%, 74.55% and 75.89% compared to other classifiers, when 200, 300, 400 and 500 features of DFS method respectively.

### *D. Discussion*

In this paper, we developed a novel text representation model called Convolution Term Model (CTM). The proposed CTM is based on CNN, which reduces the feature extraction burden and also it preserves the semantic relationship between terms in the text document by using convolution operation. The convolution operation reveals that semantic term contains highest score, when the term is semantically related to beside context (left and right). The resultant CTM matrix is of very high dimension. Hence, we applied three feature selection methods to reduce the high dimensionality. It is evident from Tables 1-3, the proposed model (CTM) with Distinguish Feature Selection (DFS) method using Convolution Neural Network (CNN) classifier performed better on 20-NewsGroups, Reuter-21758 and Vehicle Wikipedia datasets. On the other hand, 4 University dataset exhibits the characteristics, which is suitable to categorize text document using SVM classifier. Thus, on 4 University dataset, the proposed model achieved better results with DFS using SVM classifier. The set of experiments reveal that DFS performed better compared to Chi-square and Information Gain (IG) on all the four standard datasets

In feature selection method, the features are selected based on the score. The Chi-square measures the lack of independence between term and class. When a term appears in multiple classes then chi-square assigns high score to that term. Whereas, DFS assigns a low score when a term appeared periodically in multiple classes. Information Gain (IG) assesses the quality of bits of information acquired by knowing the absence or presence of a term in the document for categorization decision. IG selects the feature, which is highly related to respected class. Whereas, DFS assigns relatively high score when term frequently occurs in one class and does not occur in the other classes.

The selected features are considered as input to classifier to categorize text documents. The proposed model performed better using Convolution Neural Network (CNN) compared to other four classifiers on 20-NewsGroups, Reuter-21758 and Vehicle Wikipedia datasets. Naive Bayes (NB) classifier obtained lowest result among five classifiers. NB is a simplest classifier and easy to implement. It is computationally cheaper compared to other classifiers. Unfortunately, NB fails to learn the interaction between features and conditional independence assumption. It performs very poorly when features are highly correlated. On contrary, k-Nearest Neighbor (kNN) is a non-parametric method. The performance of kNN depends on selecting the k-values and distance measure. However, determining k-value is very difficult when the documents are not uniformly distributed. Although, Support Vector Machine (SVM) is popular supervised learning technique for text categorization, it has capability to learn independently about the dimensionality of feature matrix. However, performance of SVM is dependent on selecting kernel function and soft margin parameter C for non-linear data. The Radial Basis Function-Neural Network (RBF-NN) gives very good result for complex problems and it has ability to handle both discrete and continuous data. RBF-NN has fixed three layers like: input, hidden and output layer. In RBF-NN, determining number of hidden layer is a major challenge and also training rate is relatively slow. Compare to other classifiers, the Convolution Neural Network (CNN) is made-up of one or more convolutional layers followed by one or more pooling layers and output layer. In CNN, convolution layer use convolution filters with varying size. Thus it encourages the performance of CNN. In addition, unlike other neural network methods which use general matrix multiplication, CNN uses convolution operation which reduces the computation time. Pooling layer reduces the dimensionality of feature space by extracting sub-sequence maximum values, which is advantageous to enhance the performance of classifier.

## V. CONCLUSION

In this paper, a novel text representation model called Convolution Term Model (CTM), which uses convolution filter, is presented. The CTM is focused on preserving semantic relationship of terms and also reduce the burden of feature extraction. To reduce the feature space of resultant CTM matrix, we employed three different feature selection methods: Chi-Square ($\chi^2$), Information Gain (IG) and Distinguish Feature Selection (DFS). Further, to assess the performance of proposed model, we used five different classifiers like Naïve Bayes (NB), k-Nearest Neighbor (kNN), Support Vector Machine (SVM), RBF Neural Network (RBF NN) and Convolution Neural Network (CNN). The CTM model is evaluated on four standard datasets such as 20-NewsGroups, Reuter-21758, Vehicle Wikipedia and 4 University Dataset. From the experiments, it is concluded that the CTM preserve the semantic relationship by convolution operation and enhance the performance of classifier. The experimental result reveals that CTM

 

performes superior with DFS method using CNN classifier on 20-NewsGroups, Reuter-21758 and Vehicle Wikipedia. On the other hand, the proposed model using SVM classifier with DFS gives better results on 4 University Dataset.

In future, it is intend to embed multiple convolution layers in the proposed model. Additionally, it can also be planned to develop an optimization technique which automatically decides the number of optimal features.

## REFERENCES


[1] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1-47, 2002.

[2] F. S. Al-Anzi, and D. AbuZeina, "Beyond vector space model for hierarchical arabic text classification: A markov chain approach," *Information Processing & Management*, vol. 54, no. 1, pp. 105-115, 2018.

[3] M. M. Mirończuk, and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification," *Expert Systems with Applications*, 2018.

[4] I. S. Abuhaiba, and H. M. Dawoud, "Combining Different Approaches to Improve Arabic Text Documents Classification," *International Journal of Intelligent Systems and Applications*, vol. 9, no. 4, p.39, 2017.

[5] W. Wei, C. Guo, J. Chen, L. Tang, and L. Sun, "Ccodm: conditional cooccurrence degree matrix document representation method," *Soft Computing*, pp. 1-17, 2017.

[6] Z. S. Harris, "Distributional structure," *Word*, vol.10, no. 2-3, pp. 146-162, 1954.

[7] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613-620, 1975.

[8] Y. H. Li, and A. K. Jain, "Classification of text documents," *The Computer Journal*, vol. 41, no. 8, pp. 537-546, 1998.

[9] A. Hotho, A. Maedche, and S. Staab, "Ontology-based text document clustering," *KI*, vol. 16, no. 4, pp. 48-54, 2002.

[10] W. Cavnar, "Using an n-gram-based document representation with a vector processing retrieval model," *NIST SPECIAL PUBLICATION SP*, pp. 269-269, 1995.

[11] B. Choudhary, and P. Bhattacharyya, "Text clustering using universal networking language representation," *in: Proceedings of Eleventh International World Wide Web Conference*, 2002.

[12] B. S. Harish, M. B. Revanasiddappa, and S. V. Arun Kumar, "Symbolic representation of text documents using multiple kernel fcm," *in: International Conference on Mining Intelligence and Knowledge Exploration*, Springer, pp. 93-102, 2015.

[13] Q. Le, and T. Mikolov, "Distributed representations of sentences and documents," *in: International Conference on Machine Learning*, pp. 1188-1196, 2014.

[14] M. Gupta, V. Varma, "Doc2sent2vec: A novel two-phase approach for learning document representation," *in: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, ACM, pp. 809-812, 2016.

[15] M. Keller, and S. Bengio, "A neural network for text representation," *in: International Conference on Artificial Neural Networks*, Springer, pp. 667-672, 2005.

[16] Y. Li, B. Wei, Y. Liu, L. Yao, H. Chen, J. Yu, W. Zhu, "Incorporating knowledge into neural network for text representation," *Expert Systems with Applications*, vol. 96 pp. 103-114, 2018.

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, pp.1137-1155, 2003.

[18] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *in: Advances in neural information processing systems*, pp. 649-657, 2015.

[19] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 1, pp. 1107-1116, 2017.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, vol. 11, pp. 2278-2324, 1998.

[21] J. Huang, D. Ji, S. Yao, and W. Huang, "Character-aware convolutional neural networks for paraphrase identification," *in: International Conference on Neural Information Processing*, Springer, pp. 177-184, 2016.

[22] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," *in: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1, Association for Computational Linguistics*, pp. 142-150, 2011.

[23] R. Collobert, and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," *in: Proceedings of the 25th international conference on Machine learning*, ACM, pp. 160-167, 2008.

[24] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," *in: Workshop at the European conference on computer vision*, Springer, pp. 474-490, 2014.

[25] W. Huang, and J. Wang, "Character-level convolutional network for text classification applied to chinese corpus," *arXiv preprint* arXiv:1611.04358.

[26] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, 2017.

[27] A. K. Uysal, and S. Gunal, "The impact of preprocessing on text classification," *Information Processing & Management*, vol. 50, no. 1, pp. 104-112, 2014.

[28] A. Rehman, K. Javed, and H. A. Babri, "Feature selection based on a normalized difference measure for text classification," *Information Processing & Management*, vol. 53, no. 2, pp. 473-489, 2017.

[29] D. B. Patil, and Y. V. Dongre, "A Fuzzy Approach for Text Mining," *International journal of Mathematical Sciences and Computing*, vol.4, pp.34-43, 2015.

[30] B. S. Harish, and M. B. Revanasiddappa, "A New Feature Selection Method based on Intuitionistic Fuzzy Entropy to Categorize Text Documents," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.5, no. 3, pp. 106-117, 2018.

[31] B. S. Harish, and M. B. Revanasiddappa, "A comprehensive survey on various feature selection methods to categorize text documents," *International Journal of Computer Applications*, vol. 164, no. 8, 2017.

[32] B. S. Harish, D. S. Guru, and S. Manjunath, "Representation and classification of text documents: A brief review," *IJCA*, Special Issue on RTIPPR, no. 2, pp. 110-119, 2010.

[33] Y. Kim, "Convolutional neural networks for sentence

classification," *arXiv* preprint arXiv:1408.5882.
[34] R. Johnson, and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *arXiv* preprint arXiv:1412.1058.
[35] R. Johnson, and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," *in: Advances in neural information processing systems*, pp. 919-927, 2015.
[36] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *in: Advances in neural information processing systems*, pp. 649-657, 2015.
[37] Y. Zhang, I. Marshall, and B. C. Wallace, "Rationale-augmented convolutional neural networks for text classification," *in: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, NIH Public Access, pp. 795, 2016.
[38] L. Li, B. Qin, W. Ren, and T. Liu, "Document representation and feature combination for deceptive spam review detection," *Neurocomputing*, vol. 254, pp. 33-41, 2017.
[39] J. Xu, B. Xu, P. Wang, S. Zheng, G. Tian, and J. Zhao, "Self-taught convolutional neural networks for short text clustering," *Neural Networks*, vol. 88, pp. 22-31, 2017.
[40] C. Lee, and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization," *Information processing & management*, vol. 42, no. 1, pp. 155-165, 2006.
[41] A. K. Uysal, and S. Gunal, "A novel probabilistic feature selection method for text classification," *Knowledge-Based Systems*, vol. 36, pp. 226-235, 2012.
[42] D. M. Diab, K. M. and El Hindi, "Using differential evolution for fine tuning naive bayesian classifiers and its application for text classification," *Applied Soft Computing*, vol. 54, pp. 183-199, 2017.
[43] Y. Ko, "How to use negative class information for naive bayes classification," *Information Processing & Management*, vol. 53, no. 6, pp. 1255-1268, 2017.
[44] S. Jiang, G. Pang, M. Wu, and L. Kuang, "An improved k-nearest-neighbor algorithm for text categorization," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1503-1509, 2012.
[45] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," *in: European conference on machine learning*, Springer, pp. 137-142, 1998.
[46] E. P. Jiang, "Semi-supervised text classification using rbf networks," *in: International Symposium on Intelligent Data Analysis*, Springer, pp. 95-106, 2009.
[47] 20newsgroups, http://people.csail.mit.edu/jrennie/20Newsgroups/.
[48] Reuters-21578, http://www.daviddlewis.com/resources/testcollections/reuters21578/.
[49] D. Isa, L. H. Lee, V. Kallimani, and R. Rajkumar, "Text document preprocessing with the bayes formula for classification using the support vector machine," *IEEE Transactions on Knowledge and Data engineering*, vol. 20, no. 9, pp. 1264-1272, 2008.
[50] 4-university, http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

## Authors' Profiles

**M. B. Revanasiddappa**
He received his B.E degree in Information Science and Engineering and M.Tech degree in Software Engineering from Visvesvaraya Technological University, Belagavi, Karnataka, India. He is currently pursuing Ph.D degree in Computer Science from Visvesvaraya Technological University, Belagavi, Karnataka, India. His area of research includes Machine Learning, Text Mining and Soft Computing.

**B. S. Harish**
He obtained his B.E in Electronics and Communication (2002), M.Tech in Networking and Internet Engineering (2004) from Visvesvaraya Technological University, Belagavi, Karnataka, India. He completed his Ph.D. in Computer Science (2011); thesis entitled "Classification of Large Text Data" from University of Mysore. He is presently working as an Associate Professor in the Department of Information Science & Engineering, JSS Science & Technology University, Mysuru. He was invited as a Visiting Researcher to DIBRIS - Department of Informatics, Bio Engineering, Robotics and System Engineering, University of Genova, Italy from May-July 2016. He delivered various technical talks in National and International Conferences. He has invited as a resource person to deliver various technical talks on Data Mining, Image Processing, Pattern Recognition, Soft Computing. He is also serving and served as a reviewer for National, International Conferences and Journals. He has published more than 50 International reputed peer reviewed journals and conferences proceedings. He successfully executed AICTE-RPS project which was sanctioned by AICTE, Government of India. He also served as a secretary, CSI Mysore chapter. He is a Member of IEEE (93068688), Life Member of CSI (09872), Life Member of Institute of Engineers and Life Member of ISTE. His area of interest includes Machine Learning, Text Mining and Computational Intelligence.

**How to cite this paper:** M. B. Revanasiddappa, B. S. Harish, "A Novel Text Representation Model to Categorize Text Documents using Convolution Neural Network", International Journal of Intelligent Systems and Applications(IJISA), Vol.11, No.5, pp.36-45, 2019. DOI: 10.5815/ijisa.2019.05.05

 