A Brief Survey on Text Document Classification

Vishnu Panickar#1, Priyanka Kashyap#2, Ashish Kawale#3, Sujit Pradhan#4, Nihar M. Ranjan#5

Department of Computer Engineering, JSPM's Narhe Technical Campus, Pune

[1]vishnu.panickar234@gmail.com
[2]pkashyap0409@gmail.com
[3]ashishvk99@gmail.com
[4]sujitpradhan78@gmail.com
[5]nihar.pune@gmail.com

Abstract

Data is considered as the backbone of IT sector. Every day a lot of data is being generated across the internet, and most of this data is in unstructured form. Such as data collected from various social media sites like Facebook, twitter etc. The data collected from these sources can be vital for different organizations for their business. These data are stored in the form of large document files. Maintaining, retrieving and organizing these data is very difficult as what actually the document contains is not known. It is not practically possible to manually read each and every file and assign a label to the document. To make this process easier and efficient document classifier can be used. A text classifier basically analyses the data, and tags it with a suitable label that matches its contents. Document classification is one of the branches of text classification, where the classifier is able to tag a suitable class to the document from a list of predefined classes, which makes the process of organizing and maintaining the files in a better way. Document classification can be used in the field of library science where a lot of data from various fields are stored for analysis and decision making. Traditional machine learning techniques for text classification are relying on the bag of words representation of documents to generate features in which they are simply ignoring context. Moreover, these models require lexical features like unigram, bi-gram or n-grams to mark their presence /absence in the labelled documents. There are some serious issues arise by using these types of feature representation such as data sparsity problem and curse of dimensionality. Therefor to overcome these limitations We will use Neural Networks for building the classification model, Neural Networks has an edge over other traditional classification models due to its effective feature extraction methods and its ability to maintain a co-relation between the words.

Keywords: *Classification, Neural Networks, Data Mining, Machine Learning, Convolutional Neural Network.*

1. INTRODUCTION

Today the advancement in technology and social platform leads to generate lots and lots of data, mostly they are in form of documents. There are huge amount of text document that are being shared in social media platform and that too sometimes without the knowledge of the content. These documents could be related to terrorist, racial discrimination which is definitely going to harm our society. However, this raises the new concern of how we are going stop some undesired bad documents to spread. Therefore, we should know what is the category of the text document that is being broadcasted whether it is good or bad. Therefore, document classification plays an important role in today's tech world. One of the other area where text document plays an important

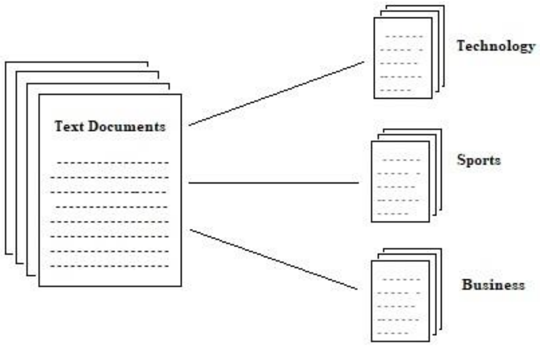

Fig 1.1 Document Classification

role is library science. Document are nothing but the long blocks of structured texts with words, sentence and paragraphs, etc. So to practically classify these

long text document is not an easy task. Keyword matching is one of the technique used in commercial document classification to classify, but the problem in this technique is to build a suitable keyword list for each category is a very challenging task that always requires domain knowledge. And moreover as keyword matching only consider the different independent keywords without taking the context information into consideration These texts are represented as features of a particular document, and the feature representation is one of the key issues in text classification. Feature representation of traditional machine learning approaches use bag of words or n-gram techniques to generate feature vector as text representation to train classifiers. Traditional machine learning techniques use such vector representations as input to train the models. However, such vectors miss to capture the semantics of the words. We will use Neural Networks for building a classification model which has better performance that these traditional classifiers.

2. EXISTING WORK

Several work on classification has been done over the years. Various methods were used for classification of text which were efficient to some extent. Sang-Bum Kim in his work used naïve bayes classifier for classification of text. Sang Bum identified the problem of rough parameter estimation which occurred in multinomial naïve bayes classification. Rough parameter estimation was done by calculating the chances of a class in the entire document. This method failed for long documents, as the length of the documents increased the number of terms increased in parameter estimation. The other problem was handling categories that very few training data. Naïve bayes classifiers does not perform well with little amount of training data. Sang-bum Kim introduced a multivariate Poisson model for naïve bayes classification and weight - enhancing method to improve the performance of rare categories. Sang-bum Kim's model was able to reduce the problems of rough parameter estimation and handling rare categories that occurred in traditional classifiers, but it was little costly in terms of time and space.

Dino Isa and Lam Hong Lee in their work introduced support vector machine along with naive byes classifier. Support Vector Machine has been analytically proven better as compared to other classifiers. But it was not easy to vectorize the text data into numeric form by using SVM, and methods like TF-IDF had problems of dimensionality. So Dino Isa and Lam Hong Lee introduced a hybrid method consisting of naive bayes and SVM. Their system was divided as follows: naive bayes was used to pre-process the data, i.e. vectorize the text and the SVM was used as a classifier. This hybrid system utilized the simplicity of bayesian formula as a vectorizer and the capability of SVM to generalize efficiently as a classifier. Naive bayes formula vectorized the documents by using the probability distribution where the dimension of features is based on the number of categories available. The hybrid model improved the performance of the classifier when its performance was compared with the performance of traditional naive bayes classifier. This model does not perform well where there is high percentage of similarity between the keywords. In such cases the naive bayes classifier shown greater accuracy as it uses the highest probability category to identify the correct class to the document. In comparison to other hybrid models such as TF-IDF and SVM, the naïve bayes – SVM had better performance in terms of training time and testing time and the hybrid model has only one second addition in terms of training and testing time when compared to the traditional naïve bayes classification approach.

Jung-Yi Jiang, Liou and Lee in their work proposed a fuzzy similarity based self-constructing algorithm which was used for feature clustering. Feature clustering was able to decrease the dimensionality of feature vectors. The two major approaches, feature selection and feature extraction are used for reducing features and forming clusters of words based on similarity test. By feature selection, a new feature set is obtained, which is the subset of the original word set. This obtained subset is then used as the input set for classification tasks. Feature clustering, which is an efficient approach is then used to group words that are similar to each other, into the same cluster and is characterized by a membership function with statistical mean and deviation. The user need not specify the features in advance. When all the words are fed, the clusters are formed automatically. The clusters formed in this approach are of incremental and self-constructing nature, which is an important factor in the calculation of similarity. No clusters exist at the beginning, and clusters are formed according to necessity. Every time a new cluster is formed, the corresponding membership function gets initialized. If the new word pattern is combined into an existing cluster, the membership function related to that cluster gets updated accordingly. The training data in this model uses SVM (support vector machines) classifier as it is better than other classifying techniques. In order to make the method more flexible and robust, SVM finds the maximum hyperplane in feature space. The fuzzy model gives better results in terms of speed and it can obtain better extracted features when compared with other classification approaches.

Ali Arshad, Saman Riaz and Licheng JIAO in their work propose a novel approach using DFCM-MC by utilizing multiple intra clusters to extract the information about the new features which can control the redundancy for the multiple class which are imbalance for classification, here the classification is associated with maximum similarity of the features between the different multiple intra clusters. Further they have improved the classification performance of their model for classification. In order to enhance the prediction ability, they design a feature extraction technique with help of random sampling to handle the problem of imbalanced data. Basically, in their approach they focusing to eliminating the redundant features and then handle the problem of multiple class imbalance data for classification the data. Here they have design new approach for multiclass imbalanced data classification which is DFCM-MC. Which is the extension of their previous work. They have extended it for the binary imbalanced dataset to the multiclass imbalanced data dataset by utilizing the decomposition technique on two layers. Where the first layer decomposes the semi-supervised data into both supervised and unsupervised, which is continuously and simultaneously operate during the training process where the information is extracted from unlabelled data to support the development of good classifier. In the second layer of the model, the supervised data is further decomposed into subsets depending upon the number of classes for(one-vs-one) a deep relation between the supervised and unsupervised data. However according to their knowledge, very few works are done on multi-cluster to overcome the issue of the class imbalanced problem. As features learning is a crucial process for the realization of the embedded information in field of data analysis. By transforming the data into a low dimension for efficient learning. The whole classification performance highly depends on the features extraction which is used as input to design the classifier. As its believed that more features are redundant, irrelevant causing more risk by making the system complex and furthermore growths the time and cost. Hence the features can be reduced into 2 ways of features extraction and then features selection. However very few works are done for combine features reduction technique to enhance the performance of classification on the multi-class imbalanced problem and easy to implement with efficient and better results. Their main motivation to utilize the combine features reduction techniques was to handle all the problems of imbalanced data and also eliminate the all irrelevant and redundant features and noisy data for the classification of the data by using the proposed DFMC-MC based features extraction technique and features selection technique (Random under sampling (RUS) and Random over sampling(ROS)).

3. PROPOSED SYSTEM

In our proposed model we are going to use convolutional neural network to classify a document. Convolution neural network is usually used in image classification; it has a great feature of detecting patterns among the text. This feature can help in classification of long text documents without missing the important data in the document. In our work we will use the following layers and classify the document on the basis of five classes.

1. Input Layer: We are going to give the whole document as an input to our model. But in a document there will be huge amount of words and these words are ultimately transformed into features and feed to our neural network. But because there are lot of features it will take a lot of time to train our model and it may take a couple of days. So to reduce the training time and complexity we will extract only 1000 words from the entire document which will be randomly selected and extract the features from these limited set of words. The extracted words will be then pre-processed by removing the stop words, removing whitespaces, converting the document into lowercase, removing punctuations and finally tokenization of the words. After the pre-processing step is completed the data will be passed to the next layer i.e. Embedding Layer.

2. Embedding Layer: The embedding layer helps to map the input token sequences to word vectors. Word embedding's are nothing but the representation technique where the words have same meaning have similar representation. Word embedding is one of the best and most popular representation technique for representing the document vocabulary. Word embedding helps in capturing the context of a word in a sentence, also it identifies semantic and syntactic similarity, and relation with other words. To show the importance of word embedding in our model, consider the following similar sentences:

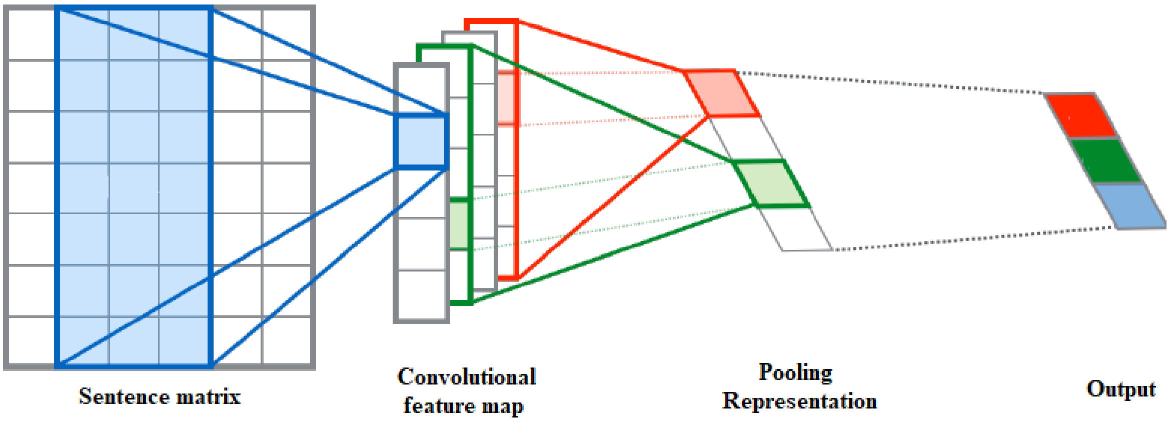

Fig 3.1 Proposed Model

A man is playing a guitar and a man is playing keyboard. They hardly have man is different meaning. It's like man is playing an instrument. If we construct a set of all words, then S={man, is, playing, guitar, keyboard}.
Suppose we create a representation using bag of words (one hot encoded) for each of these words in S.

We can clearly see the length of our one-hot encoded vector would is 5. The values in the vectors will have all zeroes except for the element which is the index that represent the specific word. That particular element would be one. The encodings below would explain this better. Man=[1,0,0,0,0],is=[0,1,0,0,0],playing= [0,0,1,0,0], guitar = [0,0,0,1,0], keyboard = [0,0,0,0,1]. If we closely observe these encodings, we can say that it is 5 dimensional space, where each word takes one dimension and it has no link to other words, this means 'guitar' and 'keyboard' are considered as different which is not true. Our objective is to possess words with similar context occupy close spatial positions. Mathematically, the cosine of the angle between such vectors should be close to 1, i.e. angle close to 0. The word embedding matrices are provided as input to the convolution layer, which extracts distinctive word patterns hidden within the training data. In simple terms, an embedding layer learns and try to find the optimal mapping of each of the unique words to a vector of real numbers. The most common application of an Embedding layer is for text processing.

3)Convolutional Layer: The convolution layer applies filters of different size to these word embedding matrix to extract features as vector corresponding to each filter. Convolving the same filter through embedding of every word of a word extract features that are independent of word position. So basically the convolution layer is networks apply a filter to an input to create a feature map that summarizes the presence of detected features in the input. We will use convolutional filter of size 3, 4 and 5 in our model. Filters can be handcrafted, such as line detectors, but the main function of the of convolutional neural networks is to learn the filters during training in the context of a specific prediction problem

4)Pooling: The pooling method helps to combine the vectors from different convolution filters into a single-dimensional vector. This is done again by taking the max or the average value observed in resulting vector from the convolutions. In our model we are going to max-pooling approach. A pooling layer is another building block of a CNN. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently. The most common approach used in pooling is max function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently. The most common approach used in pooling is max pooling. Max pooling is a sample-based discretization process. Here the objective is to down-sample an input representation reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned. This is done into part to help over-fitting by providing an abstracted form of the representation. As well, it reduces the computational cost by reducing the number of parameters to learn and provides basic translation invariance to the internal representation. Max pooling is done by applying a max filter to (usually) non-overlapping sub regions of the initial representation. After pooling the result is then concatenated and the activation function is applied to get the result.

4. CONCLUSION

Document Classification using neural networks has a better analytical results as compared with other traditional classification methods. Use of Convolutional Neural Network will help in the finding the important features from the text document. The use of word embedding for finding the features of the documents helps the classifier to get the context of the word in text document. Neural Network also helps in finding the relations between the words due to its ability to find the correlation between the words, so it can classify data effectively using these correlations. By implementing our proposed method large text documents can be classified efficiently which was one of the major drawback of traditional classification methods.

REFERENCES

1. Nihar Ranjan, Rajesh Prasad, "Automatic Text Classification using BP Lion- Neural Network and Semantic Word Processing", Imaging Science Journal Print ISSN: 1368-2199, Online ISSN: 1743-131X

2. Nihar Ranjan, Rajesh Prasad," LFNN: Lion fuzzy neural network-based evolutionary model for text classification using context and sense based features", Applied Soft Computing ISSN: 1568-4946

3. Nihar Ranjan, Rajesh Prasad," Author Identification in Text Mining for Used in Forensic", International Journal of Research in Advent Technology E-ISSN: 2321-9637

4. Sang-Bum Kim, "Some Effective Techniques for Naive Bayes Text Classification" PhD, Department of Computer Science, Korea University, 2006.

5. D.D. Lewis, "Representation and Learning in Information Retrieval," PhD dissertation, Dept. of Computer Science, Univ. Of Massachusetts, Amherst, 1992.

6. A. Rios and R. Kavuluru, ``Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles," in Proc. ACM-BCB, Atlanta, GA, USA, Sep. 2015, pp. 258_267.

7. E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, ``Improving word representations via global context and multiple word prototypes," in Proc. ACL, Jeju-do, South Korea, Jul. 2012, pp. 873_882.

8. P. Cunningham, N. Nowlan, S.J. Delany, and M. Haahr, "A Case- Based Approach in Spam Filtering that Can Track Concept Drift," Proc. ICBR Workshop Long-Lived CBR Systems, 2003.

9. A. McCallum and K. Nigam, "A Comparison of Event Models for Naıve Bayes Text Classification," J. Machine Learning Research 3, pp. 1265-1287, 2003.

10. S.J. Delany, P. Cunningham, A. Tsymbal, and L. Coyle, "A Case- Based Technique for Tracking Concept Drift in Spam Filtering," J. Knowledge Based Systems, vol. 18, nos. 4-5, pp. 187-195, 2004.

12. S. Block, D. Medin, and D. Osherson, P.A. Flach, E. Gyftodimos, and N. Lachiche, "Probabilistic Reasoning with Terms," technical report, Univ. of Bristol, Louis Pasteur Univ., 2002.

13. K. Nigam, J. Lafferty, and A. McCallum, "Using Maximum Entropy for Text Classification," Proc. IJCAI Workshop Machine Learning for Information Filtering, pp. 61-67, 1999.

14. Y. Xia, W. Liu, and L. Guthrie, "Email Categorization with Tournament Methods," Proc. Int'l Conf. Application of Natural Language (NLDB), 2005.

15. X. Su, "A Text Categorization Perspective for Ontology Mapping," technical report, Dept. of Computer and Information Science, Norwegian Univ. of Science and Technology, 2002.

16. J.R. Quinlan,H. Kim, P. Howland, and H. Park, "Dimension Reduction in Text Classification with Support Vector Machines," J. Machine Learning Research, vol. 6, pp. 37-53, 2005.

17. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ``Distributed representations of words and phrases and their compositionality," in Proc. NIPS, Carson City, NV, USA, Dec. 2013

18. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and

R. Salakhutdinov, ``Dropout: A simple way to prevent neural networks from over_tting,'' J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929_1958, 2014.

19. A. Mohamed, G. E. Dahl, and G. Hinton, ``Acoustic modelling using deep belief networks,'' IEEE Trans. Audio, Speech, Language Process., vol. 20, no. 1, pp. 14_22, Jan. 2012.

20. N. Kalchbrenner, E. Grefenstette, and P. Blunsom, ``A convolutional neural network for modelling sentences,'' arXiv:1404.2188, 2014.

21. H. Xu, M. Dong, D. Zhu, A. Kotov, A. I. Carcone, and S. Naar-King, ``Text classification with topic-based word embedding and convolutional neural networks,'' in Proc. BCB, Seattle, WA, USA, Oct. 2016, pp. 88_97.

22. Y. Kim, ``Convolutional neural networks for sentence classification,'' 2014, arXiv:1408.5882. [Online]. Available: https://arxiv.org/abs/1408.5882.