Newspaper Article Classification using Machine Learning Techniques

J Sree Devi, M. Rama Bai, Chandrashekar Reddy

***Abstract:** Newspaper articles offer us insights on several news. They can be one of many categories like sports, politics, Science and Technology etc. Text classification is a need of the day as large uncategorized data is the problem everywhere. Through this study, We intend to compare several algorithms along with data preprocessing approaches to classify the newspaper articles into their respective categories.*

Convolutional Neural Networks(CNN) is a deep learning approach which is currently a strong competitor to other classification algorithms like SVM, Naive Bayes and KNN. We hence intend to implement Convolutional Neural Networks - a deep learning approach to classify our newspaper articles, develop an understanding of all the algorithms implemented and compare their results. We also attempt to compare the training time, prediction time and accuracies of all the algorithms.

***Keywords:** Newspaper articles, CNN, SVM, Naïve Bayes, KNN.*

I. INTRODUCTION

Text classification is a very customary topic for Natural Language Processing. It can be said as one of the most sought after topics to explore in order to gain a better understanding on concepts of Natural Language Processing and machine learning. There are many algorithms which segregate text into different categories. All the traditional natural language processing algorithms have been known to majorly operate on words to decide predefined classes for particular text or text-documents. Many researchers have found out that, Convolutional Neural Networks(or ConvNet) that is a deep learning algorithm, basically developed for image processing tasks, also shows competitive results when as compared to the traditional natural language processing techniques[1]. Applying convolutional neural networks to classification of text has been explored in earlier works. This approach is proven to be competitive. The data set used in this study has been taken from AGs News Classification Dataset(.csv), separated into training data and testing data and the Twenty Newsgroup Dataset. After the initial preprocessing of the data, supervised learning algorithms will be applied along with the study of some classification algorithms. After that, a convolutional neural network model will be built and applied to learn its accuracy output on the same data set as comparison to traditional natural language processing algorithms. We would then compare the accuracy of CNN with that of the traditional models. The ability to categorize documents is highly remarkable these days. For example newspaper articles can actually be classified into 'news', 'sports', 'business', etc. To consider another case classifying hotel reviews into 'positive' or 'negative'. Important features for document classification may contain word frequency and structure. Our study looks at the task of classifying articles from the MIT newspaper 'The Tech'. We have a large collection of already categorized documents, so we are able to make use of supervised classification techniques.

In the present study we make an attempt to compare a few popular techniques for text classification and study advantages and limitations of each. Techniques include Data Preprocessing (like bag-of-words, n-grams), Machine Learning Classifiers - Support Vector Machine, Naive Bayes, K Nearest Neighbors and Deep Learning Method (like convolutional neural network). We will also try to understand why we need alternatives to each technique. Eventuallywe look forward to solve the problem of classifying Newspaper articles effectively. Such models can be very easily integrated into an online news portal, which will reduce the human effort of manually categorizing every article into classes.

II. SYSTEM DESIGN

System Architecture

System basically compromises of the following steps as shown in figure 1:

1) Importing the data sets and other python libraries needed.
2) Next we vectorize the articles in the corpus. For vectorization we use Sci-Kit Learn'sCountVectorizer to create a sparse matrix of the count of each word in an article.
3) For better results we then calculate the inverse term frequency for the words using Sci-Kit Learn'sTfidfTransformers. Having got this sparse matrix we would apply classification algorithms on this vectorized word matrix to predict classes for data in test data set.
4) Compare the accuracies, training times, testing times, predictions, etc. to prepare comparativereport.

Revised Manuscript Received on March 30, 2020.

* Correspondence Author

J. Sree Devi*, Department of CSE, MGIT, JNTUH, Hyderabad, India. E-mail: jasthysreedevi@gmail.com

Dr M Rama Bai, Department of CSE, MGIT, JNTUH, Hyderabad, India. E-mail: rama@mgit.ac.in

Chandrashekar Reddy, Department of CSE, MGIT, JNTUH, Hyderabad, India.

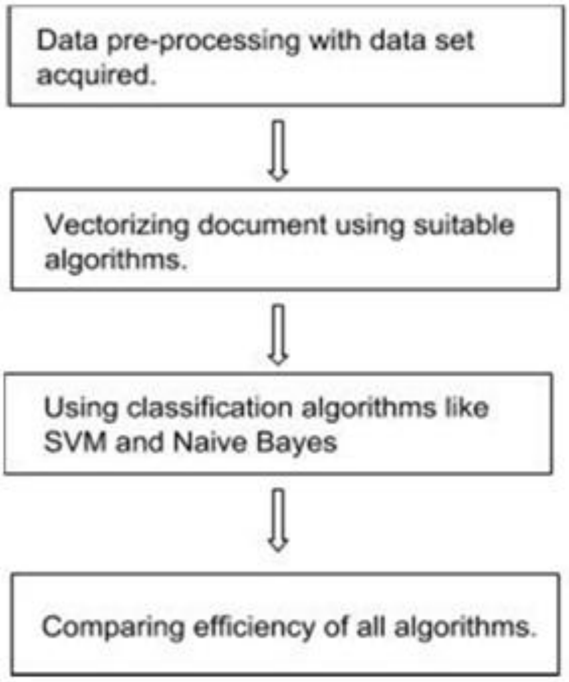
Data pre-processing with data set acquired.
Vectorizing document using suitable algorithms.
Using classification algorithms like SVM and Naive Bayes
Comparing efficiency of all algorithms.

Figure 1:Flow Chart of Newspaper Article Classification.

III. IMPLEMENTATION

A. Bag-of-Words usage in the study

In this model, a text is represented as the vector of its words, ignoringsentence structure and even the order of the words but keeping frequency. Frequency of each word is used as a feature to train the classifier[11].

1. Allocate an integer id to every word in the text of the training set.
2. For the text #n, count the no. of existences of each word W and store it in X[n, m] as the value of feature #m, m is the index of word W in the dictionary.

Here we have used CountVectorizerfunction available in the Sci-kit learn library of Python to convert the group of text documents to a sparse matrix representation [10].

There is an issue with the occurrence count that is longer the documents, higher the count values. We need to downscale the weights for words that occur in many text documents.This downscaling is called TF-IDF which stands for "Term Frequency times Inverse Document Frequency".We use the TfidfTransformer() function of the Sci-kit learn library to produce the term frequencies from the matrix of token counts.

After achieving the features, we train a classifier to predict the category of an article. Here we have implemented two different classifiers, Naive Bayes and Support Vector Machines, for predicting classes of documents in test dataset.

a) Limitations of Bag-of-Words Approach:

Bag-of-words takes into account the existences of each word, neglecting the semantics and grammar of the natural language.

Thus while dealing with Natural languages, we need to take into consideration, the usage of words, semantics and meaning of the sentence the words are a part of N-Grams is one such technique, where we vectorize not one but more than one words together, which convey much more information, than just the number of occurrences.

N-Grams

In N-grams we make the given text into slices, slices of words or slices of characters. First we consider character slicing in it we can append blank character in the starting and ending of each word. Let us consider '_' as a blank character. We have to select the variable N in N- grams i.e N can be one of 1, 2, 3 and so on .when we consider N = 2 it is bi-grams, when N = 3 it is tri-grams, when N=4 it is quad-grams and so on.

Example:In-character slicing for the word "HELLO" will be:-

Bi-grams: _H, HE, EL, LL,O_
Tri-grams: _HE, HEL, ELL, LLO, LO_
Quad-grams: _HEL, HELL, ELL, LLO_

Generally a word or string of length L has L+1 bi-grams, tri-grams, quad-grams etc. when padded with blank characters.

Example : Word slicing for the sentence "We have limited amount of resources to use"willbe:-

Uni-grams: we, have, limited, amount, of, resources, to use
Bi-grams: we have, have limited, limited amount of resources to use.
Tri-grams : we have limited, have limited amount, limited amount of, amount of resources, of resources to, resources touse[12].

N-Grams usage

Normally we use some words more frequently than other. We can combine Zipf'slaw[16] with above statement and can state as :

The occurrence of n most frequent word in text is proportional to 1/n.

The general usage language consists on lot of words which are common. In some cases, when we are classifying same type of data then some words present in all groups. Those are not much useful while classifying the data.

Generating frequency Profile:

- Data is modified by discarding numbers and punctuations, the necessary blank spaces are added to thedata.
- Then generate all possible n-grams (let's say 1 to 5) including the blanks aswell.
- Then fit the data into a hash table along with frequency such that each n-gram has its own count.
- Now sort these n-grams in descending order of occurrences then remove the count and store onlyn-grams.

Usually the top n-grams are the common words we use frequently in the human language.Now comparing two texts using n-grams[13].After generating two frequency profiles one of each text,We measure the place of each n- gram in the profile with respect to another as shown in figure 2.

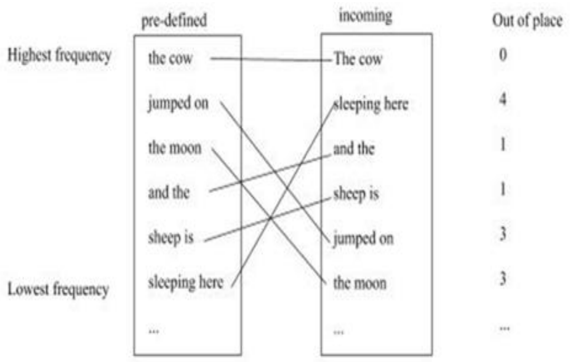

Figure 1:N-gram approach

While classifying data into multiple groups we will take one group calculate sum of relative position of each n-grams of the text to that of the group. We will perform this on all group, then we will classify the text to a category that has the minimum sum. This is how the n-gram classification works.

K-Nearest Neighbors

K Nearest neighbors is a method that can be used for both classification and regression. Here, we will discuss the classification aspects of the same**.**Through the following example, we will try to illustrate the process that K nearest neighbors algorithm follows as shown in figure 3:

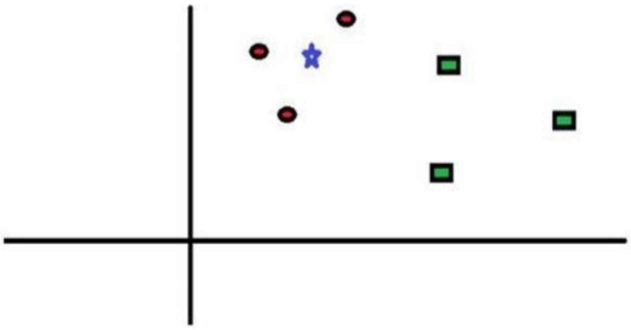

Figure 2:plot of data tuples

This is a spread of red circles and green squares. We need to find the class among the circles and squares that the blue star will belong to. When we apply k nearest neighbors to the above problem, let’s take an arbitrary value for k, k=3 [14]. We find the three nearest neighbors to the star in the Cartesian plane by finding the distance between all data points(circles and squares) and the star, and taking the minimum three.

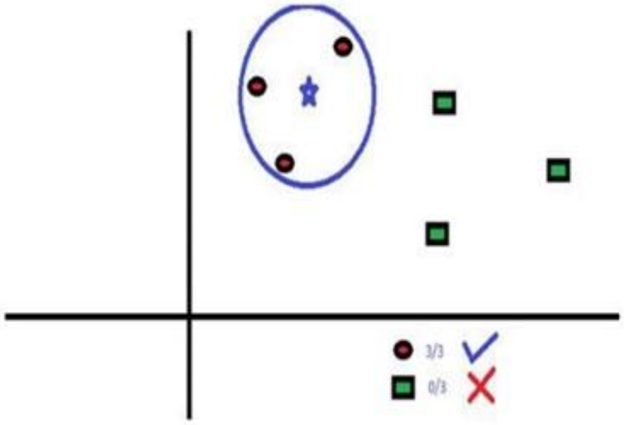

Figure 4: K-nearest neighbors

So we can see quite clearly to which class of points the star belongs to. With a given K value we can make boundaries of each class. These boundaries will segregate red circles from green squares. The value of K can be selected experimentally and may lie between 1 and 10.
KNN is a simple classification algorithm but it can still give competitive results to other algorithms.

B. Naive Bayes

With Naive Bayes classifier,

- The class of data set can be identified easily and quickly. It also works well for Multi-ClassClassification.
- When the assumption that our data is independent of the features of each other holds, then even with lesser training data Naive Bayes classifier gives excellent results.

Naïve Bayes implementation in the study

We create an instance of the model available in Sci-kit learn library and then fit the training data using fit() function. Later we predict classes for test data using predict() function. If we wish to, we can calculate metrics like accuracy, rms error, etc.

C. Support Vector Machines(SVM)

SVM is a supervised machine learning algorithm that is used to classify data as well as for the regression problems. It finds its use in most cases in the classification problems. In SVM, we represent each feature of our dataset as a point in our coordinate system. The algorithm tries to find out a hyper-plane that splits the two classes with as much accuracy as possible.
For example consider the following figure as shown in figure 5:-

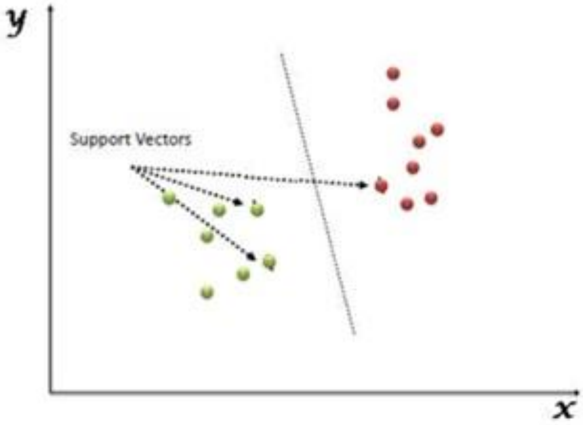

Figure5:Hyper-plane separating two classes

1) The distance between the nearest points in both the categories is called the margin. Such a hyper-plane is selected that finds out the maximum margin for both the categories. If the data is nonlinear, the a special technique called kernelization is used. In this process, the data is projected into a three- dimensional plane and then a hyper-plane dividing the two categories is found. Then the points are projected back into the 2 dimension. So, the hyper-plane might be somewhat circular

SVM usage in the study:

We also used Support Vector Machines(SVM), which is also a widely used classifier(although a bit slower than Naive Bayes). We create an instance of the SGD Classifier available in Sci-kit learn library and then repeat to process of training the model on training data and predicting classes for test data.

To implement SVM in newspaper article classification, the first step is to remove stop words, then the punctuation marks are removed. The next step is to remove the digits since they too do not contribute to the categorization of articles. After all the data preprocessing, next step is to use bag of words or n- grams to create sparse matrix which contains the words as vectors, count as the feature. On this the support vector machine classifier is applied. The results for the same have been analyzed and discussed in the results section along with the inferences.

D. Convolutional NeuralNetworks(CNN)

CNNs are deep artificial neural networks used for Image classification. They take the local features and perform convolutions in every layer. Convolutional Neural Networks works based on the fact that the data in images and paragraphs consists of a sensible architecture. In convolutional neural networks the layers contains neurons that are set in 3 dimensions that are width, height and depth.

It basically consists of three types of layers that are - Pooling layers
- Activation layers
- Fully connected layer.

After every network layer, an activation function is applied to find out the node that should be fired. In simple terms, we apply different non linearities by applying the activation functions on each layer to match the output nodes. ReLU activation function is mostly used for Convolutional Neural Networks that maps negative values to 0 and the positive values stay as it is.

In our model, we took an input layer of 15,000 neurons. Our next layer was a ReLU activation function which gave us a positive output for each input, i.e. mapping negative to zero. Then, we drop out some of the nodes in a drop-out layer. This is the first layer. We repeat the above steps for the next layer, which is the hidden layer[16]. These are then in turn are connected to output layer. All these three layers have 512 neurons each. In the output layer, SoftMax is applied.

The output layer is a 4X1 array which contains some values. The maximum of the four values is given as output category, i.e. 0 if the first index has the maximum value, 1 for second index and so on.

IV. TESTING AND RESULTS ANALYSIS

Testing

Here we will aim to compare different text classification techniques, namely bags-of-words, bag-of-n-grams and convolutional neural networks. We will compare them according to the time taken for each algorithm to train with classifiers including SVM, Naive Bayes and KNN. Then we shall compare each algorithm with their accuracy in classifying the test data when different sizes and different training data in terms of content isprovided to each model.

Thus we shall be comparing the traditional models and deep learning models and provide statistical analysis of each models with respect to other.

Here we are expecting that convolutional neural networks model will provide good results in terms of accuracy in every case[17]. In the rest of the models we are expecting n-grams to be better in more cases than convolutional neural networks based on the size of data set provided. It is expected that, bag-of-words and convolutional neural networks will perform almost with nearly same accuracy in most of the cases depending of data set provided.[18]

To confirm our results, we will use different available data sets to match our results for each case to reach to a conclusion.

Result:After successfully applying the techniques and classification algorithms we attempted to compare them on various metrics.

The various metrics are:

- Accuracy on test data
- Training time
- Prediction time

We can graphically represent the comparison of accuracies of all the algorithms the following way:

The first metric on which we compared the algorithms was accuracy of different machine learning algorithms and deep learning namely CNN and we got the following results as shown in figure 6.

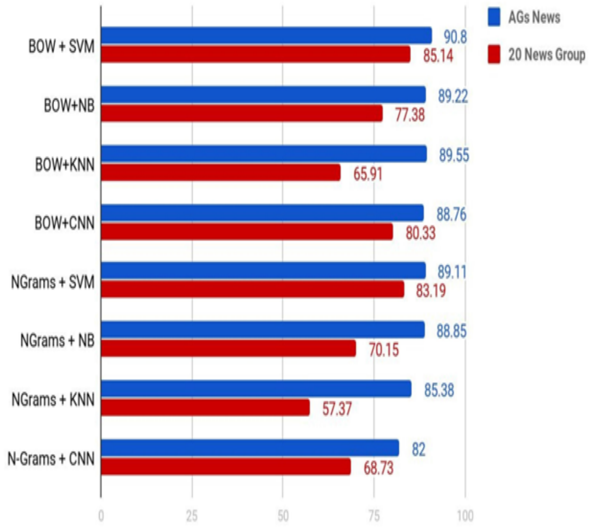

Figure 6:Accuracy of models

The very first observation we can make is that accuracies on the 20 NewsGroup dataset are always lesser than those for AG's News dataset irrespective of the algorithm used. We owe this to the following three reasons :

- The first and more obvious reason is the fact that articles in 20 NewsGroup dataset are classified into 20 categories whereas those in AG's news dataset are categorized into just The more the number of categories, the higher the chances of mistakes in categorizing them among those categories.
- The second being the ratio of testing to training data for both the datasets. The AG's News Dataset has 120,000 training news articles to 7,200 testing articles[19]. Whereas,the20NewsGroup dataset has 11,314 training articles to 7,532 testing articles.

Thus the models were very well trained with more training data and had to do relatively less work predicting classes for a smaller testing data in case of AG's News dataset.

- Thirdly, while preprocessing the raw data, we used an approach for removing stop words(punctuations, undesired characters and very frequently appearing words) on the AG's News Dataset only, improving the quality of the model and the accuracy eventually.

The second observation we can make is that we see that SVM outperforms K-NN and Naive Bayes when applied to either of the data preprocessing techniques. This is because SVM tries to separate the classes with hyperplanes as far apart as possible, creating clear and distinct boundaries among data points. Whereas K-NN predicts classes based on nearness to data points. Some of the data points might infiltrate into chunks of data points of some other class, resulting into lesser accuracy. Moreover, unlike SVM, Naive Bayes consider all features to be mutually independent, thus neglecting co-relations among them, which otherwise might be very important to the dataset [18].

Another inference from the above graph is that when data is pre-processed with N-Grams, the model performs poorly as compared to Bag-of-Words. This is because sequence of words are less likely to appear in an article than individual words. Thus, accuracy decreases as value of N increases.

The second metric on which we compared the algorithms was on Training time and we got the following results as shown in figure 7:

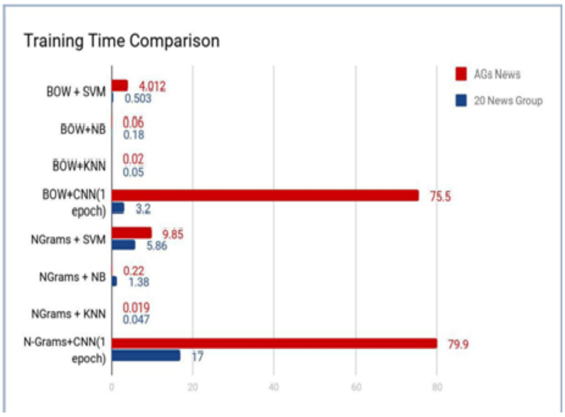

Figure 7: Training time comparison

Here we observe that AG's news dataset takes much more time to train than the 20 NewsGroupdataset, primarily because of its much larger number of articles in the training dataset, 1,20,000 to 11,314.

CNN is mathematically the most complex algorithm involving lots of calculations, hence taking the most time to train. Also training time is much dependent on the structure of the network, hidden layers, the number of neurons on input, hidden and output layers, dropout values, etc.

Also we observe that data preprocessed with N-grams causes more time to train the model than with Bag-of words.

The third metric we compared the algorithms was on Prediction Time on the test dataset. We got the following results as shown in figure 8:

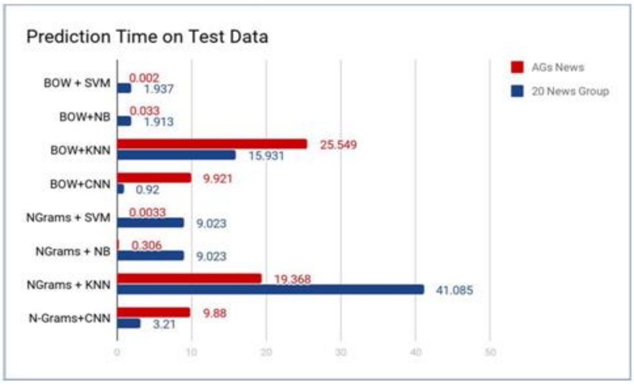

Figure 8:Prediction time on Test Data

a) The first very evident observation we make is the huge prediction time for K-NN algorithm. This is because, instead of simply calculation a cost using pre-calculated hyperparameters on the test data, KNN calculates distances to every datapoint in the training dataset to predict classes.

Hence, larger the training data, the more time it takes to predict categories.Also, more number of categories in the 20 NewsGroup dataset increases the prediction time as Naive Bayes has to calculate 20 probabilities and SVM has to hypertune 20 one-versus-all hyper-planes compared to only 4 each in case of AG's News Dataset[19]. The prediction time of CNN is again dependent on its structure.

We mainly used two data pre-processing techniques, namely Bag-of-Words and N-Grams. Hence we compared the time both these techniques take for vectorizing the datasets as shown in figure 9.

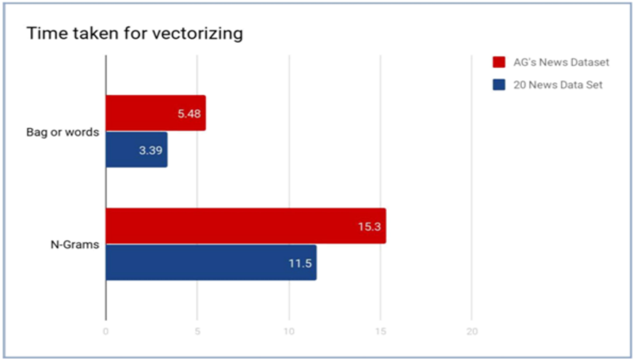

Figure 9: time taken for vectorizing

b) The process for tokenizing the input strings is more complex when using N-grams as compared to Bag-of-Words. Also the above times include time to calculate TFIDF values and the sparse matrix which again is more complex when using a sequence of words than individual words. Also it takes more time to vectorize the AG's News dataset due to its much larger size.

V. CONCLUSION

We applied all the algorithms and graphically compared the accuracies, time taken for training, and testing and time taken for vectorizing using preprocessing approaches like Bag of Words and N-Grams. We observed the following from our implementation.

Retrieval Number: E2753039520/2020©BEIESP
DOI: 10.35940/ijitee.E2753.039520
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

- SVM gives more accuracy with a moderate training time and predicts very quickly.
- If we wish to devote less time towards training and shift the load towards testing, KNN is preferred as it takes less time for training.
- If we have a lot of articles to test, and prefer to choose an algorithm which takes less time for testing, either CNN [20] or SVM can be considered with data preprocessing through Bag of Words.

ACKNOWLEDGEMENT

The authors wish to thank Mr M Chandrashekar Reddy for carrying out some simulation as a part of his project work.

REFERENCES

1. XIANG ZHANG, JUNBO ZHAO, YANN LeCUN, “Character-level Convolutional Networks for Text Classification”, Courant Institute of Mathematical Sciences, New YorkUniversity.
2. C. dos Santos and M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 69–78, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.
3. Y. Kim. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar, October 2014. Association for ComputationalLinguistics.
4. R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. CoRR, abs/1412.1058,2014.
5. M. IKONOMAKIS, S. KOTSIANTIS, V. TAMPAKAS, “Text Classification Using Machine Learning Techniques “, University of Patras, GREECE, WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, August 2005, pp.966-974.
6. Zhenya Zhang, Shuguang Zhang, Enhong Chen, Xufa Wang, Hongmei Cheng, TextCC:New Feed Forward Neural Network for Classifying Documents Instantly, Lecture Notes in Computer Science, Volume 3497, Jan 2005, Pages 232 –237.
7. Forman, G., An Experimental Study of Feature Selection Metrics for Text Categorization. Journal of Machine Learning Research, 3 2003, pp. 1289- 1305
8. Kim S. B., Rim H. C., Yook D. S. and Lim H. S., “Effective Methods for Improving Naive Bayes Text Classifiers”, LNAI 2417, 2002, pp.414-423
9. BharathSriram, Dave Fuhry, EnginDemir, HakanFerhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pages 841–842.ACM.
10. William B. Cavnar and John M. Trenkle, N-Gram-Based Text Categorization, Environmental Research Institute of Michigan P.O. Box 134001 Ann Arbor MI 48113-4001 pp1
11. Zhong, S. (2005, August). Efficient online spherical k means clustering. InNeural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on (Vol. 5, pp. 3180-3185).IEEE.
12. Mei, J. P., & Chen, L. (2014). Proximity-based partitions clustering with ranking for document categorization and analysis. Expert Systems with Applications, 41(16),7095-7105.
13. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119.2013.
14. G. Lev, B. Klein, and L. Wolf. In defense of word embedding for generic text representation. InC.Bie-mann, S. Handschuh, A. Freitas, F. Meziane, and E. Mtais, editors, ”Natural Language Processing and Information Systems”, vol 9103 Lecture Notes in Computer Science, pp3550.
15. Yiming Yang, Jan O. Pedersen, A Comparative Study on Feature Selection in Text Categorization, Carnegie Mellon University, USA, VerityInc.
16. Zipf, George K., Human Behavior and the Principle of Least Effort, an Introduction to Human Ecology, Addison-Wesley, Reading, Mass.,1949
17. https://www.analyticsvidhya.com/wp-content/uploads/2015/10/SVM_1.png
18. Ding-Xuan Zhou, “Universality of Deep Convolutional Neural Networks” , Department of Mathematics, arXiv[cs:LG],
19. 20 July2018.
20. David Stutz, “Understanding Convolutional Neural Networks”, FakultätfürMathematik, Informatik und Naturwissenschaften, August 30, 2014.
21. Xiang Zhang, JunboZhao ,YannLeCun, “Character-level Convolutional Networks for Text Classification”, Courant Institute of Mathematical Sciences, New York University,arXiv:1509.01626[cs.LG].