International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

A Novel Deep Learning Based Sentiment Analysis of Movie using Hybrid CNN_SVM Algorithm

Raviya.K, Mary Vennila.S

Abstract: ***Data flow in web is becoming high and vast, extracting useful and meaningful information from the same is especially significant. The extracted information can be utilized for enhanced decision making. The information provided by the end-users is normally in the form of comments with respect to different products and services. Sentiment analysis is effectively carried out in these kinds of compact review to give away the people's opinion of any products. This analyzed data will be efficient to improve the business strategy. In our work the collected online movie reviews are analyzed by using machine learning sentiment classification models like Random Forest, Naive Bayes, KNN and SVM. The work has been extended with CNN and hybrid CNN-SVM deep learning models to achieve higher performance. Comparing the workings of all the above classification models for sentiment analysis based upon various performance metrics is the main objective of the paper.***

Keywords: ***Machine learning, Sentiment analysis, Movie review, Algorithm, Random Forest, Naive Bayes, KNN, SVM, CNN***

I INTRODUCTION

The process of analyzing end-user opinion or view as positive or negative or neutral via in the form of text with respect to any product or topic is sentiment analysis. Sentiment analysis is a highly focusing area in Natural Language Processing. In this field, machine learning technique is considered to be a dominating process.

The basic task of machine learning is to extract complex features from the reviews which are in the form of text and additionally it figures out relevant features and selects a classification algorithm [4].

There are several drawbacks is seen while using traditional lexicon-based approaches: there is always dependency of lexicon which is reliable and consistent whenever there is unpredictability of opinion words, languages and contexts. Because of these dependencies maintaining domain independent lexicons is becoming tough. In contrast to this deep learning has an alternative potential to handle traditional methods. Deep learning has exposed performance of superiority in NLP tasks especially in sentiment analysis.

Manuscript published on November 30, 2019.

* Correspondence Author

Mrs.Raviya.K*, Department of BCA, Gurunanak College, Chennai, India . Email: raviyamca@gmail.com

Dr. Mary Vennila.S,Department of BCA, Gurunanak College, Chennai, India. Email: raviyamca@gmail.com

With minimal external contribution, learning complex features that has been extracted from the data is the core objective of deep learning techniques [8]. Yet another characteristic of deep learning is that they require a huge set of data for better performance.

Conversely, it is not understandable that whether the traditional approach's domain specialization capacity be capable of surpassing deep learning's generalization capacity with all task of NLP. But promising outcomes are thrown out in most of the applications when these two techniques are appropriately combined for sentiment classification [5].

In this paper higher performance rate has been produced when using hybrid CNN with SVM which has been derived from the combination of machine learning and deep learning-based sentiment classification techniques.

II METHODS

A. Random Forest:

This supervised learning algorithm is also called to be random decision forest and highly implemented for the purpose of classification, regression and many more tasks. A forest is meant to be collection of trees. It generally comprises of huge quantity of individual decision trees which usually perform as ensemble. A class prediction is thrown out from each tree in the random forest and model prediction is finalized from the class that has more number of votes [1].

With the concept of bagging and bootstrapping, random decision forest has been regarded with robust and accurate classifier.

B. Naive Bayes:

It is one amongst the simplest classifying algorithms for text document. This probabilistic method is developed upon Bayes' theorem and independence assumption of Naive among given set of inputs. In Naive Bayes the basic assumption is to create an independent and equal contribution to the outcome from each feature.

$$P(I|J) = \frac{(P(J|I)*P(I))}{P(J)}$$

With respect to machine learning, selecting the best hypothesis (I) with the given data (J) is often considered as potential interest. As per the theorem stated above, P(I|J) is the probability of hypothesis I with the given data J, which is called as posterior probability. When the hypothesis I was true, then the probability is P(J|I) with the data J. P(I) is the prior probability of I when hypothesis I is being true

regardless of the data. P(J) is the probability of the data regardless of the hypothesis.

SVM-Support Vector Machine:

In machine learning, SVM algorithm is a supervised learning model used for both classification and regression analysis. Even though it is used in both analyses the main focus is for classification purpose. SVM is a discriminative classifier by which the output is thrown as optimal hyper-plane that separates the two classes [10]. In general SVM training algorithm make a model which allot new examples for one category or to other, that make a non-probabilistic binary linear classifier.

I Linear Kernel SVM

Kernel is meant as dot-product and rephrased to:

$$K(x, xi) = sum(x * x_i)$$

In which distance measure or similarity between support vectors and new data is defined by the kernel K. As distance is a linear combination of inputs, this dot-product which is a measure of similarity is utilized for linear kernel.

II Radial Kernel SVM

When compared with linear kernel, radial kernel is more complex. As an example, let us consider the below equation:

$$K(x, xi) = \exp(-gamma * sum((x - xi\hat{}2))$$

Here SVM algorithm has to be intimated about the parameter gamma. If gamma is 0.1, then value is meant to be good in default with its range from 0 and 1. Within feature space, complex regions can be created by this kernel, which in turn have the transformation capacity of space dimension from low to high.

III Polynomial Kernel SVM

This kernel can be utilized as an alternate to dot-product. Let us take the below equation:

$$K(x, xi) = 1 + sum(x * xi)\hat{}d$$

Here the SVM learning algorithm should be specified about the degree of polynomial with hand written note. It will be same as linear kernel when d becomes 1. In the input space this kernel allows curved line.

C. KNN

K-nearest neighbours is a supervised machine learning algorithm which is simple and easy for solving problems in both classification and regression. But only in classification analysis it is used in a broader way. All available cases are stored in the KNN and a new case is classified with the major votes of the k neighbours. With the assistance of distance function the case is allocated with the classes which are usually common within their k-nearest neighbour [7].

The distance function may be Hamming, Manhattan, Minkowski and Euclidean. Hamming is used for categorical variables and the other three are utilized for continuous function. If 1 is the value of k, then the case will be assigned for the class with the nearest neighbour. Sometimes while performing KNN model, selecting K becomes a challenging task. For our real life scenarios KNN can be easily mapped.

D. CNN

CNN is mainly implemented in image processing and it is successfully applied in NLP task nowadays. It is a multilayer perceptrons, which means fully connected networks. Words in the CNN architecture are broken as features and then these features are sent to convolutional layer. The convolution result is aggregated or pooled as representative number. The represented number is sent to a fully connected neural structure. This in turn decides the classification on the basis of weights allotted for every feature in the text. Extracting the input feature is the main objective of convolution and sampling the convolution matrix is covered by pooling [6].

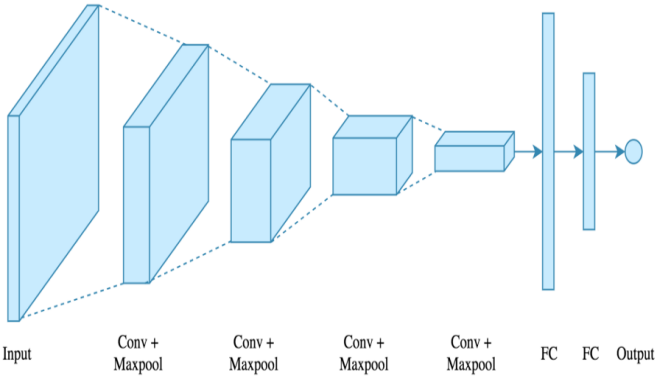

Fig.1 CNN Architecture

E. Hybrid CNN_SVM

Learning invariant features with web page sources are done effectively in CNN, where as they do not often generate optimal classification outcomes. On the other hand with fixed kernel operation SVMs are unable to learn complicated invariances, instead with the use of soft-margin approaches they maximize the margins to produce good decision surfaces. To examine and suit the hybrid system is the ultimate scope of the proposed system. In this system the CNN has been trained in learning the features which are comparatively invariant with respect to inputs of irrelevant differences. Subsequently SVM in combination with non-linear kernel produces optimal solution for the purpose of segregating classes within the learned feature.

From the traditional CNN structure, our proposed model is optimized in five different layers. The layers are Input Layer and Embedded Layer, Convolutional Layer, Pooling Layer and fully connected layer with SVM classifier. According to this structure the initial step is to train word embedding for every word in the given dataset. Then this trained dataset is used as CNN model's input feature. In turn this is given iterative training with in the midst of other network factors. Finally the extracted features are given as input to the SVM classifier that has been trained and produced the final output. As per this model CNN acts as an excellent feature extractor and SVM does the classification task.

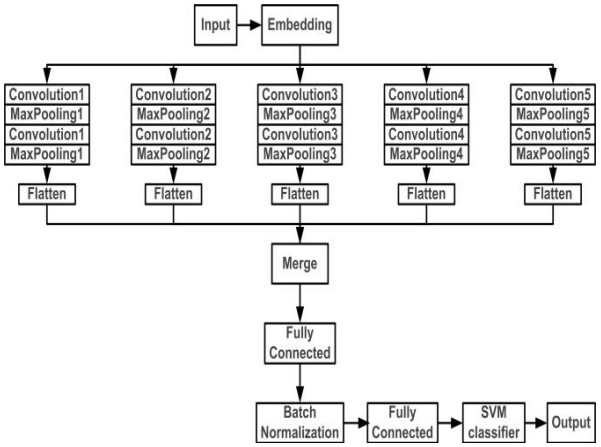

Fig.2 Hybrid CNN_SVM Architecture

Input and Embedded layer:

Input layer is the first layer and it receives the input, followed by embedded layer in which data pre-processing is carried out. Embedded layer is a matrix where the word vectors analogous with the words within the sentence are arranged in order from top to bottom. This is performed with an assumption of words from the sentence as 'm', word vector dimension as 'n' and then forming m x n matrix.

Convolutional layer:

By convolutional operation numerous feature maps are obtained from the input layer, wherein k x n represents convolutional window. Here longitudinal words are represented by k and word vector dimension as n. Number of columned feature maps will be obtained from convolutional window.

Pooling layer:

Sub-sampling is another name of pooling layer which is used for reducing the input data size. CNN pooling can be done in many ways, but max-pooling is the most common one that is used. With respect to our model, we have five convolutional and pooling layers in varied sizes like 4 x 100, 5 x 100, 6 x 100, 7 x 100 and 8 x 100 for the feature extraction of text.

Fully connected layer:

This is the last layer that connects one or more fully connected layers. The ultimate aim of this layer is to make multi-dimensional inputs to one-dimensional. The resulted output from this layer is normalized and given to SVM classifier for final classification. The pseudecode for Hybrid CNN_SVM is as follows:

Pseudocode

$\boldsymbol{input}$: *a data set of opinion sentences that is already represented by word vectors*
$\boldsymbol{output}$: *class labe of each data*
$window\_size = \{h1, h2, h3, \dots, hn\}$,
$z = \emptyset$
$data = \{X1, X2, \dots, Xm\}$
for each h in window_size:
$w \leftarrow initializeFilter(h, m)$
for X in data:
$s \leftarrow size(X)$
$c = \emptyset$
for i in 0: $s - n + 1$
$x \leftarrow concantenate(Xi{:}i{+}h{-}1)$
$temp \leftarrow nonLinear(wTx + b)$
$c \leftarrow c \cup temp$
end
$\hat{c} \leftarrow \max(c)$
$z \leftarrow z \cup \hat{c}$
end
end
$w \leftarrow initializeWeight(size(z))$
$skor \leftarrow svm(wT\ z + b)$
if skor > 0
return "*positive*"
else
return "*negative*"
end

III RESULT AND DISCUSSION

Online movie review dataset are collected and analyzed with the available classification models. In total the dataset consists of 2000 text documents with positive and negative reviews.

As the classifier needs to be trained by the extracted features, the collected dataset is exploited. Implementation is carried out with the consideration of features in combination of n-grams. This process of n-grams extraction from the review dataset consists of the following steps:

1. Filtering is the process of eliminating usernames, characters, URL's and special words.
2. Tokenization process uses spaces as delimiters and removes symbols like punctuation marks to divide the text and to tokenize them. In further to this text processing two more processes like Lemmatization and Stemming are also followed.
3. Exclusion of stop words ("an", "the" and "a") from the processed text with the use of created word set is known to be removing stop words.
4. Next step is to construct n-grams. In this process n-grams are constructed from information of continuous vocabulary. Usually negation words such as "no" and "not" is associated with words either in front or middle or at the last. Here let us take an example: "I will not play ball", this sequence of word will be resulted in two bigrams namely, "I will+not", "will+not play", "not+play ball".

The dataset that are processed has been divided in two groups, one for training and the other for testing in different sizes. These sets are implemented using Random Forest, Naive Bayes algorithm, Support Vector Machine and K-Nearest Neighbour. By implementing the collected corpus in the above algorithms, the performance factors like precision, recall, F-score and accuracy were obtained.

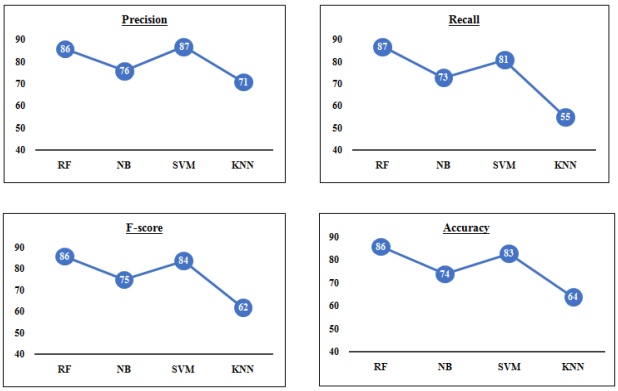

Fig.3 Performance measures of four algorithms

Performance measures of four machine learning algorithms are shown in fig-1. As per the observation, SVM produces peak precision whilst Naive Bayes produced least precision. Random forest and SVM produces peak recall whilst KNN produced least recall. Again random forest and SVM produces peak F-score and accuracy whilst KNN produced least F-score and accuracy.

$$\text{Precision} = \frac{\text{The no. of correctly classified samples of this type of polarity}}{\text{The no. of marked samples of this type of polarity}}$$

$$\text{Recall} = \frac{\text{The no. of correctly classified samples of this type of polarity}}{\text{The no. of this samples of this polarity}}$$

$$\text{F - Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Table – 1 Performance measures of all six algorithms

Machine learning Algorithm	Precision	Recall	F-score	Accuracy
RF	86	87	86	86
NB	76	73	75	74
SVM	87	81	84	83
KNN	71	55	62	64
CNN	91	90	89	90
CNN_SVM	94	92	90	92

Finally we observe that SVM model is suitable for the given online movie review dataset. For further improvisation in sentiment classification models, we introduced deep learning algorithms like CNN and hybrid CNN_SVM and compared the performance metrics with all the models. Based upon the performance factors, table-1 and fig-2 depict high performance of hybrid algorithm when compared with other algorithms.

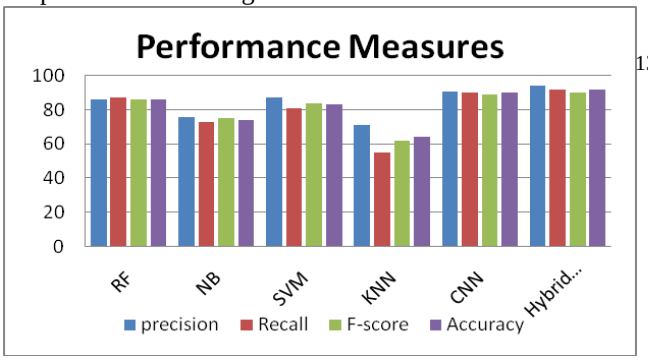

Fig.4 Performance measures of all algorithms

IV CONCLUSION

One of the important tasks that have been always focused in sentiment analysis is text classification. Sentiment classifiers are used widely in companies for many applications like brand monitoring, market research, product analytics, work force analytics and customer support. Machine learning and deep learning technologies are becoming one of the most focused areas of research for implementation of sentiment analysis. In this study we have tried in solving the classification of text in sentiment analysis with merging of deep learning and traditional machine learning algorithms. At the outset word vector that are pre trained where embedded into CNN model of improved category. Next, to improvise classification task we utilized our proposed Hybrid CNN with SVM algorithm. Experiments have proved in this paper that sentiment classification accuracy is high compared to other classification models. In future, our focus is to build a custom classifier for sentiment analysis, topic labelling, language detection and intent detection.

REFERENCES

1. Y. Kim, "*Convolutional Neural Networks for Sentence Classification* Proceedings of the 2014Conference on Empirical Methods in Natural Language Processing (EMNLP)", 2014, pp.1746-1751.
2. Akhila Ravi,Akash Raj Khettry, "Amazon Reviews as Corpus for Sentiment Analysis Using Machine Learning- Springer ICACDS 2019", 2019, CCIS 1045, pp. 403-411.
3. Gitanjali, Kamlesh Lakhwani, "A Novel Approach of Sensitive Data Classification using Convolution Neural Network and Logistic Regression",IJITEE, June 2019, ISSN:2278-3075,Volume-8 Issue-8.
4. X. Zhang, J. Zhao and Y. LeCun, "Character-Level Convolutional network for text classification. In Advances in neural information processing System", pp. 649-657.
5. K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber and L. E. Barnes, "Hierarchical deep learning for text classification", 16th IEEE International Conference on Machine Learning and Applications(ICMLA), December 2017, pp. 364-371.
6. Brian Keith,Exequiel Fuentes,and Claudio Meneses, "A Hybrid Approach for sentiment analysis Applied to Paper Review", In Proceedings of ACM SIGKDD(KDD'17), 2017, 10 Pages
7. Brian Keith,Exequiel Fuentes,and Claudio Meneses, "A Hybrid Approach for sentiment analysis Applied to Paper Review", In Proceedings of ACM SIGKDD(KDD'17), 2017, 10 Pages
8. Huaiguang Wu, Daiyi Li and Ming Cheng, "Chines Text Classification Based on Character-Level CNN and SVM", Springer Nature, ISICA 2018, 2019, CCIS 986, pp. 227-238.
9. Oscar Araque, Ignacio corcuera, J. Fernando sanchez-Rada, Carlos A. Iglesias, "Enhancing Deep Learning Sentiment Analysis With ensemble techniques in social applications", Elsevier Expert System with Application, 77 (2017) 236-246.
10. F. Pedregosa, A. Gramfort, Michal,V Scikit, "Learn: Machine Learning in Python. J.Mach. Learn. Res", 2012, 12(10), pp. 2825-2830.
11. Nirmala Devi and K. Jayanthi, "Sentiment Classification Using SVM and PSO", Int. J. Adv. Eng. Technol, 2016, VII(II), 411 – 413.
12. Sonagi and D. Gore, "Efficient Sentiment analysis using Hybrid PSO-GA Approach. Int. J.Innov. Res. Comput. Commun. Eng", 2017, 5(6), 11910-11916.
13.

AUTHORS PROFILE

K.Raviya Associate Professor,Department of BCA, Gurunanak college,Velachery,Chennai-42 Tamilnadu, India