DOI: 10.21917/ijsc.2021.0335

DEEP CNN WITH SVM - HYBRID MODEL FOR SENTENCE-BASED DOCUMENT LEVEL SENTIMENT ANALYSIS USING SUBJECTIVITY DETECTION

K. Raviya and S. Mary Vennila
PG and Research Department of Computer Science, Presidency College, India

Abstract

With the growth of e-commerce reporting, online customer reviews have evolved rapidly, voicing the sentiment or opinion of customers about goods. The analysis of belief could provide useful data for us. Sentiment analysis on social media like Twitter or Facebook, is now the comprehensive way of understanding about the views of customers and has extensive variety of applications. In the context of NLP, automated text classification can be a fundamental activity and it can help people to pick essential information from vast text resources. Sentiment analysis may be a computational technique that plays a key role in automating the retrieval of subjective knowledge, i.e. customer's sentiment from online text reviews or opinion from social network like Twitter and Facebook. Lexicon-based and machine learning-based methods are two main approaches widely used in sentiment analysis activities. In machine learning based framework, Sentiment analysis is a text recognition task. The outcome depends not only from the soundness of the algorithm for machine learning, but also with the appropriate features. In recent years, the most recent technological advancements, like deep-learning techniques, have resolved a number of standard challenges caused by the lack of lexical tools in the region. It has been exhibited that deep-learning models are auspicious and potential tool to NLP challenges. In this work, the fusion of deep CNN with SVM will automatically detect and extract subjective sentence-level features to perform sentiment analysis of online product review dataset with highest accuracy and less computation time.

Keywords:

Sentiment Analysis, Deep Learning, Convolutional Neural Network, Bigdata

1. INTRODUCTION

The growth of the Internet of Things (IoT) [1] has led to such developments in communication technologies. The major growth of social network and communication devices such as smart phones and laptops, etc. empower people to attach with one another to come up with enormous amounts of massive data. Moreover, because of the influx of latest technologies, the quantity of knowledge is anticipated to rise continuously. Over the years, the widespread use of innovations and extraordinary data flow have also led to the expansion of huge data analytics in business. It refers to 2 parts, because the word suggests: big data and business analytics. Big Data stands for data whose size goes beyond the potential to record, store, handle, and evaluate by traditional database software tools. Big data, however, is often characterised by the most features of volume, variety, and velocity. Analytics, on the opposite hand, refers to the ability to use statistics, simulation, and optimization tools to realize data insights and to use data-driven business decisions. The overall sentiment analysis method for the extraction of social media big data is shown in Fig.1.

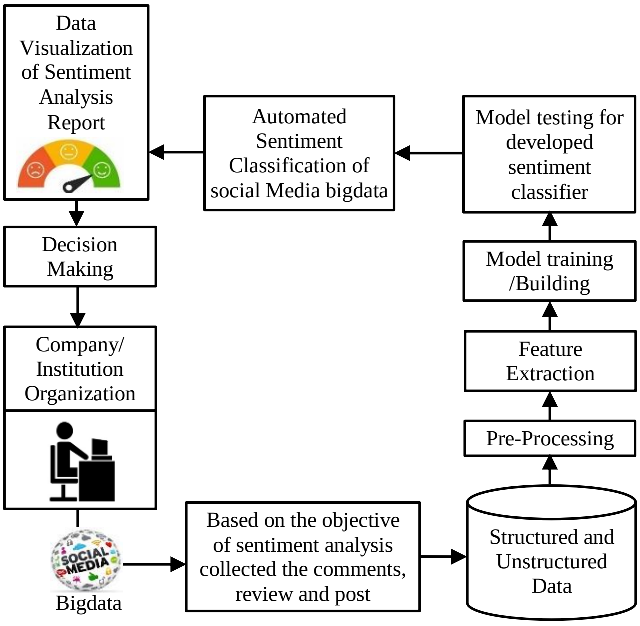

Fig.1. Sentiment Analysis using social media bigdata

First of all, the most method is that the gathering of knowledge from big data sources, like social media forums. It must be held in account that the data gathered should be related to the target of the sentiment analysis system. For instance, if the corporate needs to gather the sentiment about their products and then the information are mined with keywords that signify the products. This also requires setting the target of the sentiment analysis to clarify the goal of this analysis, so as to see the related keywords. The next step is really the pre-processing of the information so as to eliminate rid of noise and irrelevant content then. It is tailed by the development and assessment of the sentiment analysis framework. Especially machine learning methods commonly involve training and testing of algorithms. After constructing and evaluating the model in real data sets (test data), the developed model is operated on newly collected big data to automatically extract and classify the sentiments.

Sentiment classification is a method of automatically analysing customer opinions in text reviews in survey responses and social media communication, allows brands to put attention attentively to their customers, and customize the products and services to full fill their needs. It is the strategy of computationally identifying and categorizing opinions voiced in a piece of text, particularly so as to work out whether the person's attitude about selected topic, product, etc. is positive, negative, or neutral. It is typically conducted in 3 stages.

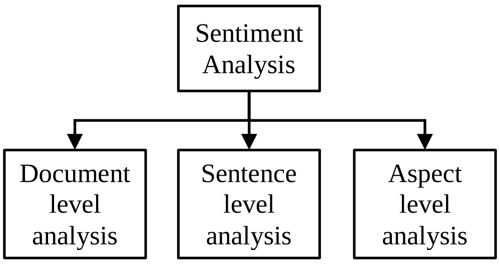

Fig.2. Levels of Sentiment Analysis

First the subject to which the feeling is directed is discovered, then the polarity of the emotion is measured and finally the degree of polarity is determined with the assistance of a sentiment score which signifies the strength of the sentiment. Sentiments are often classified at different levels shown in Fig.2: Aspect level, sentence level and document level. Aspects level sentiment classification categorizes the feelings supported the sentiment's polarity of every aspects or feature of some target entity and sentence level classification on the opposite hand categorizes each sentence supported their opinion polarity regarding some topic. In document level classification, the polarity of entire document is set. It categorizes the complete document as positive or negative or neutral class.

Traditionally, sentiment analysis can be a machine learning technique focused on features. The technological advancement in fields, such as Big Data and Deep Learning technology have influenced and benefited this field's evolution. Deep learning is a vital machine learning branch that utilizes high-order optimization of various nonlinear integration structures. Recently, many authors have successfully been exploring deep learning technology within the field of NLP to unravel the matter of text using sentiment classification. CNNs are commonly employed in computer vision, however they've recently been employed to varied NLP tasks with the necessity of minimal pre-processing and therefore the results were promising.

Kim [2] proposed a text classification model, examining pre-processed vectors of word as input and utilizing convolutional neural networks to realize sentence-based classification process. Xing et al. [3] also examine CNN to resolve the polarity conviction difficulty of twitter data set. However, CNN has given excellent advancements in the categorization of text, it gives close attention for local characteristics and neglects the conceptual significance of terms, thus reducing the accuracy of classification. Therefore, enhancements to the CNN model are in progress. The fusion model introduced during this paper incorporates the benefits of CNN to derive local characteristics and applied SVM for the classification of sentiments. Additionally, the fusion model's classification accuracy is enhanced.

2. RELATED WORK

One of the primary tasks in the natural language sector is the classification of text sentiment. In the area of sentiment analysis, several researchers were so far carried out in-depth studies. During the early phase of the study, the machine learning-based algorithm proved its dominance in the text classification issues, considering the fairly tiny size of the dataset. Among them, The Text classification based on SVM was suggested by Sun et al. [4]. This problem is solved by the classifier and has obtained strong research results. For binary sentiment classification, Kennedy and Inkpen [5] implemented the model with characteristics of unigram and bigram and attained 84.4% of result for movie reviews by applying SVM. In the research of Gautam and Yadav [6] used SVM for sentiment classification of Twitter texts along with the semantic analysis and obtained 89.9% overall accuracy. The SVM with n-gram characteristics was also used by Tripathi et al. [7] and 88.94% of accuracy for sentiment classification was obtained. For binary classification with Twitter messages, Bahraini et al. [8] operated on the number of positive and negative words as characteristics and achieved 86.7% of output. Neethu et al. [9] merged unigram features with Tweet features and gained 90% of accuracy using SVM with binary classification. In general, all of these studies using the SVM classifier achieved 70-90% accuracy, and the SVM was the most efficient model. While the use of machine learning models has been successful with the general limitation that their output varied and depending on how for the features have been defined; for various data, it would take a lot of effort by domain experts to achieve better results by integrating other tools such as ontology, lexicon and it will cost a lot of domain experts effort and time.

One of the alternatives to such a constraint is the deep learning model, since it is known to automatically capture random patterns. In addition, as described in [10], the use of the deep learning for sentiment analysis can offer a representation of contextual features that generalize well in different domains. The advantages of deep learning for the classification of sentiments are discussed in the next subsection. Social media is an important data sources for SA. Social networks are constantly expanding, producing knowledge that is much more dynamic and interrelated. As the size of datasets continuously expanding, the booming technology includes deep learning methods offers the chance to accommodate large quantities of data for text classification. Several studies focused on developing efficient models to resolve the ever-growing challenge of big data and applying sentiment analysis for broad variety of applications from economic forecasting and advertising strategies to medical analysis and some other fields. However, in order to provide realistic proof of their results, very few of them pay full concentration to examining various deep learning techniques. Liao and Junbo [11] Wang suggested CNN to understand circumstances based on Twitter data sentiment analysis and obtained an overall accuracy of 95.39%.

Several scholars have analysed text-based sentiment analysis. The analysis of the detection of subjectivity, however, is sometimes ignored or defined in the above reviews as a subsection. In comparison, we have concentrated only on methods of identification of subjectivity in this study. Another downside of earlier research on this subject is that recent word-vector models such as convolutional neural networks are not covered. For text sentiment classification, Kim [2] uses a simple CNN model. Experimental findings specify that the CNN model performs better in the semantic classification of phrases than conventional models. In order to extract text feature information, Rozental [12] projected a fusion model using BiGRU with CNN. The hybrid CNN-SVM model is used in most of the current research related to image classification and has achieved high accuracy. In this paper, to handle the classification of text sentiment, we take this fusion of deep learning with machine learning CNN-SVM technique. The experimental result also showed that CNN-SVM outshines other models, providing a reasonable stability between the runtime of the CPU and accuracy.

3. SENTENCE LEVEL SENTIMENT ANALYSIS

Reviews and blogs are consisting of both subjective and objective terms. Subjective sentences are important and express the mood, opinion or belief of the consumer, whereas objective sentences are insignificant and contain factual data [13]. On the basis of this concept, we filter the sentences that are factual and retain sentences that are highly subjective. First of all, it decides if the sentence is subjective or objective in the sentence level sentiment analysis, and if the sentence is subjective, then the attitude of all subjective sentences is evaluated. A deep learning technique is proposed to perform this task to extract the opinion or attitude of subjective sentence in the product review to find the final polarity of review. The work flow of our system in shown in Fig.3.

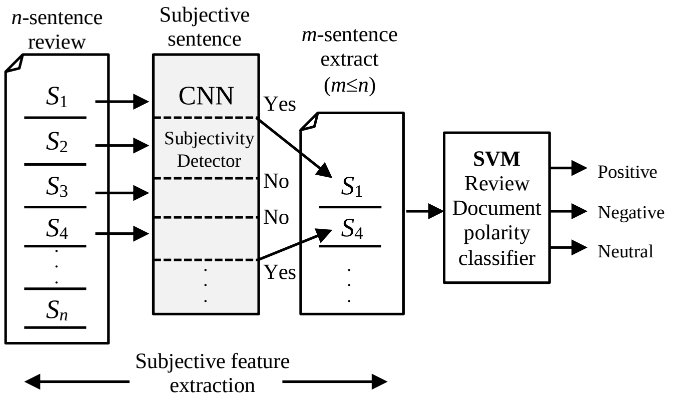

Fig.3. System setup for Document sentiment classification

Document-level sentiment classification may be considered as a special case of categorization of text sentiment instead of topic-based classification. Normal classification methods for machine learning, such as support vector machines (SVMs), can therefore be extended to the complete document itself, as Pang et al. [14] have done in 2002. As default polarity classifiers, we refer to such classification methods. As mentioned above, however, we might be able to boost the classification of polarity by eliminating objective sentences. Therefore, as shown in Fig.3, we propose to use a subjectivity detector that decides whether or not of sentence is subjective: rejecting the objective terms produces an extracted feature that should better reflect the subjectivity of the review document to the sentiment classifier. To our knowledge, previous work with document-level sentiment polarity has not incorporated sentence-level subjectivity detection. Yu and Hatzivassiloglou [15] used a methods of sentence level analysis and to resolving whether the document is subjective or not, but he is not pipeline two algorithms for polarity classification of documents.

The motive of Beineke et al. [16] as single sentence selection process is to identify the polarity of sentiments of a document, but they are not assessing the accuracy of the polarity classification results. There are several characteristics that are less important for sentiment analysis. To train the classifiers, extract and use the most relevant feature set. Now, conventional methods and methods based on deep learning are the widely used methods of text classification. The traditional text classification approaches are mainly based on machine learning and are classified using statistical principles. The deep learning methods primarily use neural network to extract text characteristics that can combine low-level features to form more abstract high-level features. The goal of this paper is to integrate these two methods to propose a CNN-SVM a hybrid model for document level sentiment analysis.

3.1 CNN

Deep neural network models in the domain of computer vision [17] and speech identification [18] have obtained impressive results. Many researchers employed deep neural networks in natural language processing to build word embedding [19] and to do text analytics over the learned word embeddings. The CNN model has achieved very good results in many ways, such as grammar analysis, search for words, sentence modelling, and other traditional NLP tasks. We adopt convolutional neural network for feature extraction. For computer vision, it is possible to obtain a portion of the pixel related data, not just to retrieve the pixels bit by bit, the information of the features can be retrieved piece by piece, each piece contains information of several pixels; when we convert the text into a matrix, it can also be treated as same as the matrix of an image pixel, so we can extract the subjective characteristics from the text data by performing the same operation. In order to create a more stable hybrid deep learning network to make use of extracted features, the CNN soft-max layer is replaced by the non-liner SVM-based classification at the final level.

3.2 SVM

SVM is a type of supervised algorithm for regression and classification. In terms of certain generalization requirements, it is a classifier that seeks optimal separating hyperplane. There is an efficient process, called the kernel process, for non-linear difficulties that allows the machines to generate complex nonlinear definitions within the unique space. There are several functionalities that can be used as an SVM kernel. There are four common kernel term of functions are in Table.1.

Table.1. Common Kernel Functions

Kernal	Formula
Linear	$K(x,y)=x.y$
Polynomial	$K(x,y)=(x.y+c)^d$
Gaussian RBF	$K(x,y)=\exp(-\gamma\|x-y\|^2)$
Sigmoid	$K(x,y)=\tanh(x.y+c)$

The choice of different kernels is used to construct different SVM classifier. In order to ensure a better result, the non-linear composition of kernels is suggested for SVM on sentiment classification. These kernels are sufficient for certain problems. In non-linear combination kernels, the adjustable parameters are the weight of every other sub-kernel.

4. PROPOSED HYBRID CNN-SVM MODEL

Kim [2] suggested CNN for sentence level sentiment classification with fully connected soft-max layer is used as a classifier. This layer of classification is too simple for the role of sentiment classification. Fortunately, the CNN's pooling layer output values can be considered as feature vectors of the subjective sentence. CNNs are successful in learning the text's invariant characteristics, but will not always achieve optimal results for classification. Contrarily, SVMs do not learn complicated invariances through their fixed activation functions, but yield good decision interfaces by using soft-margin approaches to optimize margins. In this context, the proposed method is intended to investigate a hybrid framework where only CNN is equipped to learn subjective characteristics and is relatively invariant to insignificant input variations. In this case, a non-linear kernel SVM can thus just provide optimal solution in the learned feature space for class labels. The CNN's output layer is exchanged by SVM, i.e., the CNN's fully connected layer serves as an input to the SVM. For document-level sentiment classification, we should conduct subjectivity detection on each and every individual sentence by applying convolutional neural network on each sentence in isolation. Here CNN act as subjectivity detector and SVM for document level sentiment classifier. The document level classification model is trained and tested on the output generated CNN, the sentence level subjectivity detectors to discover the sentiment polarity of the text reviews. In this work, we introduce CNN based SVM that handles CNN as the implicit feature learner and it is robustness to handle negation as well as its ability to identify the subjective sentence and extract subjective features that is important for sentiment prediction then SVM act in the role of sentiment classifier. The outputs of CNN, the distributed feature descriptions of the input sentences remain considered as features of SVM and it is trained by applying these document sentiments labelled vectors. The integrated architecture is supposed to incorporate the benefits of CNN and SVM for the prediction of document sentiment polarity.

5. CNN ARCHITECTURE

The network takes the words array in the sentence as input, and routes it through a set of layers where characteristics are extracted with varying degrees of complication. The network obtains information from the level of the character and sentence to the level of the document. In our network, the process flow consists of three major parts. Fig.4 describes the end-to-end layers of the network. The network is therefore made up of eight layers:

- ***Word Vectorization***: We use fixed-length word2vec word embeddings as input data. The first layer of CNN is the embedding layer and it is used to do word vectorization.
- ***Sentence Vectorization***: It is denoted as word representation sequences to fixed-length sentence vectors for each sentence. In our network architecture, the key innovation is the use of two convolutional layers, allowing it to operate characters and phrases of any scale. For each convolution, one max pooling layer is there.
- ***Document Vectorization***: It involves concatenation of the sequence of sentence vector representation to the document vector. This is entirely done by one concatenation and k-max pooling layer. Finally, the fully connected layer connects all the feature for sentiment polarity classification and pipe line to SVM for final prediction.

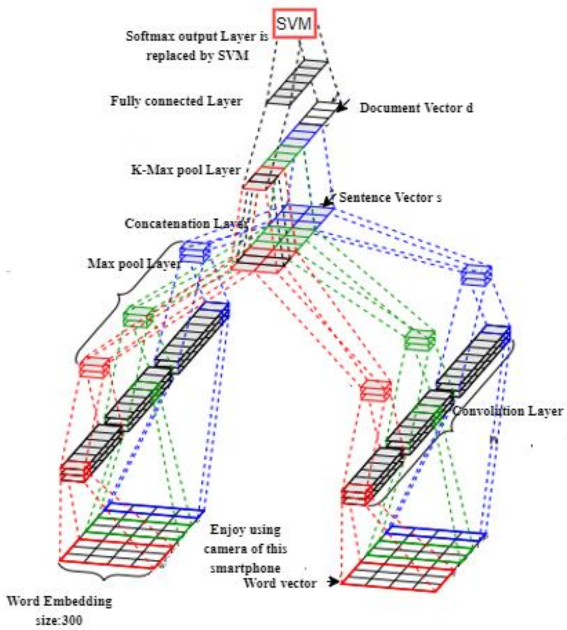

Fig.4. CNN-SVM Architecture

5.1 EMBEDDING LAYER

CNN's initial layer is the embedding layer converts words into real-value feature vectors that take phenotypic, syntactic, semantic details from the words. Word2vec is for word level embedding. We are handling fixed-sized word diction of V_{word}, and words are comprised of characters from a fixed-sized character V_{char}. Given sentence containing of N terms $\{w_1, w_2,..., w_N\}$, every word w_n is changed into a vector $u_n = [r^{wrd}; r^{wch}]$ is comprised of 2 sub-vectors: the embedding of word-level $r^{wrd} \in R^{d^{wrd}}$ and the character-level embedding $R^{cl_u^o}$ of w_n. Although embedding at word-level intended to acquire syntactic and semantic data, embedding at character-level collect morphologic and shape details. The sentence x with N terms $\{w_1, w_2,..., w_N\}$, has been transformed to combined word and character-level embedding $\{u_1, u_2,...,u_N\}$. Data source on Google News (approximately words of 100 billion) is used to trained the vectors in the proposed technique.

5.2 CONVOLUTIONAL LAYER

CNN uses the Convolutional layer to learn the text's local characteristics. The convolution method creates local characteristics around each character of the word and then mixes them to generate a fixed-sized embedding of character-level terms of the word using a max operation.

The first convolution operates a matrix-vector process to the window of size k^{chr} of consecutive windows of the system $\{r_1^{chr}, r_2^{chr},..., r_M^{chr}\}$. Let us express the vector $z_n \in R^{(d^{chr}+cl_u^o)k^{chr}}$ is the sequence of character embedding m, its $0.5(k^{chr}-1)$ of left neighbours, and $0.5(k^{chr}-1)$ of right neighbours.

$$z_m = \left(r^{chr}_{m-0.5(k^{chr}-1)}, ..., r^{chr}_{m+0.5(k^{chr}-1)}\right)^T \tag{1}$$

The primary convolutional layer calculates j^{th} component of the vector $r^{wch} \in R^{cl_u^1}$ as follows, this is the character-based embedding of w, as come first. Here $W_0 \in R^{cl_u^o \times d^{chr} k^{chr}}$ is the weight-matrix of the convolutional layer. To abstract local features on each character vector of a given phrases, the same matrix has been used. We just use max over to all character ranges of the keyword to derive a global fixed-sized function vector for the term. Methods for extracting a sentence-wide feature set often deal with two key issues: phrases of various sizes; and at any place in the expression, sensitive details will exist. We fix these issues by using a convolutional layer to measure the sentence-wide function vector r^{sent}. In our neural network, the next convolutional layer operates in a quite parallel to those used to retrieve character-level characteristics of words. This layer generates local characteristics on each word of the sentence and join them to create a fixed-sized feature vector for the sentence by using a max operation. The 2nd convolutional layer operates matrix-vector process to window of different size k^{wrd} of consecutive windows for the order $\{u_1, u_2, ..., u_N\}$. Let us describe the vector $z_n \in R^{\left(d^{wrd}+cl_u^o\right)k^{wrd}}$ the chain of a sequence of k^{wrd} embeddings, centralized in the nth word:

$$z_m = \left(u_{n-0.5\left(k^{wrd}-1\right)}, ..., u_{n+0.5\left(k^{wrd}-1\right)}\right)^T \tag{2}$$

This layer calculates j^{th} term of the vector $r^{sent} \in R^{cl_u^1}$ as comes next

$$\left[r^{sent}\right]_j = \max_{1<n<N}\left[W^1 z_n + b^1\right]_j \tag{3}$$

Here $W_0 \in R^{cl_u^o \times d^{chr} k^{chr}}$ is the weight-matrix of the convolutional layer. To obtain local features on all word vector of a given sentence, the same matrix has been used. For the sentence, we retrieve "global" fixed-sized function vector using the max over all word ranges of the sentences.

5.3 MAX POOLING LAYER

CNN's third and fifth layer belongs to a max pooling layer. The feature vectors of the convolutional layer are pooled after the convolution process, and all the feature maps are consolidated and measured. The CNN model utilizes max pooling in this article. This operation establishes the pooling layer to obtain an *m*-dimensional function vector, here m is the number of filters. Multiple filters of different window sizes are used for the CNN model. Finally, the vector r_x^{sent}, the global function vector of sentence *x* is transformed by this layer, which extract one more level of representation. We take its highest value as the pooling layer's function since the most prominent features will be extracted. Here two max-poling layers are used for two convolutional layers.

5.4 CONCATENATION LAYER

Convolution and max pooling are employed on each sentence in the document. Finally, we combine the vectors retrieved for each sentence. Finally, the vector r_x^{sent}, the "global" vector of sentence x, is evaluated by two more network layers to extracts a further level of support for entire document sentiment label aggregation and classification.

5.5 K-MAX POOLING LAYER

After processing individual sentences, this layer is intended to aggregate sentence vectors into document vectors. The representation of a document is a variable-sized convolution of all its subjective sentence vectors. For all of these 600 features, we select the maximum of the subjective sentences of the document to obtain the document representation. This gives the whole document of 600-dimensional real-valued representation.

$$d^{network} \in R^{r_x^{sent}} \tag{4}$$

5.6 FULLY CONNECTED LAYER

CNN's eighth layer is the fully linked layer that links all features and calculates a score for each sentiment labels. In this job, the parameters are reduced by k-max-pooling layer after the function map and the optimum features are obtained. Then all the optimal local features obtained are linked through a completely connected layer then this output is an SVM input that is the subjective feature vector of the entire document.

$$d(x) = W^3 h\left(W^2 r_x^{sent} + b^2\right) + b^3 \tag{5}$$

5.7 SVM FOR CLASSIFICATION

Nineth Soft-max output layer is replaced by SVM to determine the sentiment polarity of the entire document. Finally, the subjective features are classified by SVM classifier.

Let $x_i = \left(x_i^{(1)}, x_i^{(2)}, ..., x_i^{(m)}\right)^T$ are *k*-dimensional vector of features of i^{th} text, $y_i \in \{-1,1\}$ is a class of i^{th} text. By training to identify a hyperplane, the SVM can distinguish the vector of features of the document.

$$\vec{\omega} + x + b = 0 \tag{6}$$

Here $\vec{\omega}$ is denoted as normal vector defining the hyperplane direction, b is the form of displacement which defines the range between the origin and hyperplane. The distance between x_i and hyperplane:

$$r_i = \frac{\vec{\omega} x_i + b}{\|\vec{\omega}\|} \tag{7}$$

Obtaining the optimal hyperplane is to retrieve the closest two distinct vectors and their closeness of the hyperplane is equivalent and the sum of the differences on them to that hyperplane is the farthest. Then it is equivalent to,

$$\min \frac{1}{2}\|\vec{\omega}\|^2 = s.t. y_i\left(\omega^T x_i + b\right) \geq 1,\ i = 1, 2, ..., n \tag{8}$$

Let $\vec{\omega}$ and b are the optimal hyperplane, and utilizing this optimal threshold for the classification the feature vectors of the document.

6. PARAMETER OPTIMIZATION OF CNN-SVM

Our model is trained to reducing the negative probability of training set *D* to a minimum. Taking a sentence *x*, the set of network parameter θ calculates a score $s_\theta(x)_\tau$ of each sentiment class $\tau \in T$. To translate these scores in to the conditional distribution of labels of probability provided the sentence and the parameters θ of the network, here we applied soft-max method over the scores of all tags. To reduce the negative log-likelihood with regard to θ, we employ stochastic gradient descent.

$$\theta \rightarrow \sum_{(x,y\in D)} -\log p(y|x,\theta) \tag{9}$$

where in training corpus D, (x,y) refers to a sentence, and y represents its relative label. The backpropagation algorithm is a perfect choice to measure network architecture patterns efficiently, including the one proposed in this study. The pre-trained word integrations are well-tuned by back propagation of the CNN model training process. Fine-tuning makes it possible to learn very accurate word representations. When the terms may not exist in the embedding of pre-trained words, then they are randomly initialized. The output vectors of the completely connected CNN layer are considered to be representations of the distributed sentence function, and then these representations of sentences are considered to be feature vectors of the document in the SVM classifier.

Table.2. CNN-SVM Hyper Parameter Settings

Parameter	Parameter Name	Values
d^{wrd}	Dimension of word-level embedding	300
K^{wrd}	Word Context Window	5
cl_u^1	No. of units for Word Convolution	300
d^{chr}	Dimension of Char. Embedding	5
K^{chr}	Character context Window	3
cl_u^0	No. of units for Char. Convolution	10
γ	SVM Kernel Parameter	0.1
C	Penalty Parameter	0.01
Convolutional Layer		2
Hidden Layer		4
Activation Function		ReLU
Filter Size		5,6,7
Fully Connected Layer		1
Learning Rate		0.01
Epochs		10
Dropout		0.5
Mini-Batch Size		32
Regularizer		L2

We invest large amount of time, tuning the hyper-parameters of CNN-SVM for good performance. This gives some measure of how robust our approach is and the amount of training epochs ranges from five to ten. Then, we present the selected hyper-parameter values for the proposed method in Table.2.

7. DATASET AND EXPERIMENTAL SETUP

In this study, we conducted the experiments on Amazon online Smart Phone and its accessories review datasets (Training and Testing Datasets) of 2000 reviews collected from Kaggle. Each review is considered as a document of two or three sentences. To eliminate the noise in the data, the extracted datasets are pre-processed. The role of pre-processing is important to make the analysis of sentiment more accurate. We have used Jupyter Notebook [20] as open-source development environment for implementing all baseline and proposed models. This is a web-based application for generating and distributing of documents comprising of test, code, formulas and visualizations and it is used to perform data cleaning, transformation and statistical and machine learning models etc. In order to produce rich set of features, the input data are pre-processed by eliminating unrelated data such as stop word, punctuation etc., then it is followed by stemming, term frequency identification using python coding. The TFLearn is a python package in TensorFlow [21] for developing and speedup all deep learning methods. First of all, the sentences are converted to list of sequences. This is performed by TFLearn vocabulary processor we need to translate these sequences to numerical values, because a neural network model can only perform operations over numbers.

7.1 EXPERIMENTAL RESULTS AND DISCUSSION

Training Phase: The optimization of the parameters for CNN and SVM is performed in the training time by tight experimenting on sample data with 5-fold cross validation. In the training process, accuracy is adopted as main metric since the single metric helps to speed up the optimization of parameters. As the time complexity of the CNN model is determined by the size of the filters and the hidden units, these two parameters are chosen within the acceptable range. We fine tune the filter and hidden unit parameter.

The size of pooling layer feature vector output is the product of these two parameters. In Table.3, the optimization results of CNN-SVM show that the accuracy of the sentiment classification for the given dataset hits a maximum value of 97.09% once the filter value of [5]-[7] and the hidden units are of 150.

Table.3 Experimental Result on Different filters

Filters	Hidden unit			
	50	**100**	**150**	**200**
[1,2,3]	94.65	95.10	95.43	95.67
[2,3,4]	95.01	95.09	95.70	95.65
[3,4,5]	96.11	96.24	96.45	96.32
[4,5,6]	96.52	96.60	96.63	96.60
[5,6,7]	96.80	96.98	**97.09**	96.96
[6,7,8]	96.60	96.82	96.80	96.76

The Fig.5 illustrate the impact of various learning rates in the product review dataset on the efficiency of CNN-SVM. The number of iterations will raise or drop into local minima by observing at the gradient descent method. If the primary value of learning rate set was also low. The cost function will be inconsistent and, causing the algorithm slow, could not hit the actual minimum.

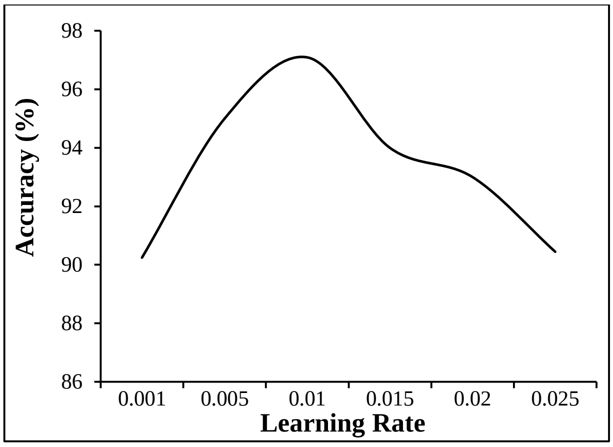

Fig.5. Different learning rate with accuracy

The Fig.5 indicates that the CNN-SVM model's accuracy reached its maximum value once the learning rate was 0.01, the accuracy rate decreased when the learning rate raised again, so the resulting experimental learning rate was 0.01. For SVM parameter optimization, the grid search array of each parameter is: $\sigma = [10^{-3}, 10^{-2},...,10^{2}, 10^{3}]$ and $C = [10^{-3}, 10^{-2},..., 10^{2}, 10^{3}]$. We tried various combination of both parameters on training dataset. The maximum accuracy is attained with 0.01 of C and 0.1 for γ.

Table.4. Experimental Result on Different Iterations

Performance	Number of Iterations						
	1	10	20	30	40	50	100
Training time (s)	42.6	435.3	873.1	1200.5	1564	1934	3441
Test time (s)	3.6	3.8	4.3	4.0	3.6	3.8	3.5
Accuracy (%)	90.25	95.66	96.11	96.16	97.09	96.84	95.65
Loss (%)	34.63	23.93	18.09	17.35	18.28	18.55	18.11

The precision of feature extraction and classification increases with the increasing number of iterations. The CNN and SVM parameters almost remain the same when the amount of training exceeds a certain degree. Then both converge, and the efficiency is optimum. The growing pattern of accuracy in classification is gradually decreased after this period. The training time and the iteration times are highly associated, but the number of iterations does not influence the test time.

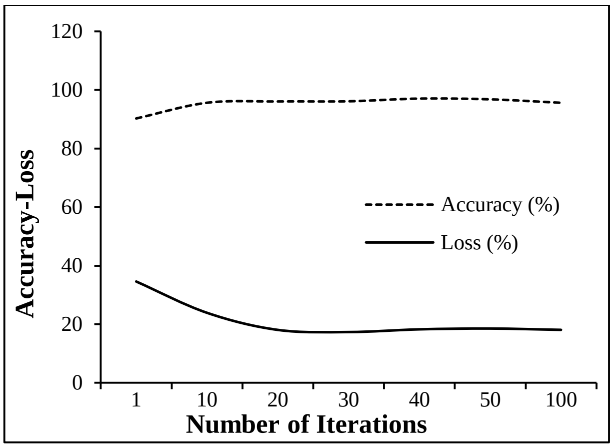

Fig.6. Accuracy Loss vs. Number of Iterations

It was found that the CNN-SVM model offered the best trade-off among no. of iteration with training time and the accuracy of results. Here, we choose the mini-batch form of CNN training, i.e., one group contains multiple samples named batch-size. Mini-batch training can result in a network jitter when there is a lack of training. To confirm that the latest weights will minimise errors in training phase, the previous batch-size samples will update the network weights. But losses can increase for the latter batch-size. However, with more training courses being carried out, the overall number of losses will eventually decrease. The total of hundred iterations were executed in this work, and the impact of these executions is shown in Table.4 and Fig.6.

The CNN-SVM performance is evaluated on various aspects such as different iteration with training and testing time and accuracy and loss Percentage based on various parameters. Also, the validity of the sentiment analysis approach based on convolutional neural network for subjective feature detection and SVM for document polarity classification suggested in this paper is also compared with other classification methods. Usually, the performance factors of machine learning techniques are precision, recall, F-score and accuracy. Here NB, SVM, Random Forest, CNN algorithms are found as baseline models and the performance of these algorithms are compared with Hybrid CNN-SVM algorithm. It is shown in Table.5 that our system is superior to other methods of classification, which can obtain better results. The Recall rate and F1-score also define our model's high performance.

Table.5. Experimental Evaluation

Method	Precision	Recall	F-score	Accuracy
NB	74.6	76.2	69.9	71.5
Random-Forest	82.3	82.0	82.1	81.5
SVM	87.7	86.4	85.2	87.3
CNN	93.7	92.8	90.8	91.7
CNN-SVM	97.07	96.06	96.04	97.09

The Fig.7 shows the performance results of the proposed system for sentiment analysis compared with the existing algorithms. From this result, it is observed that in terms of accuracy, the proposed method accomplished better than existing methods.

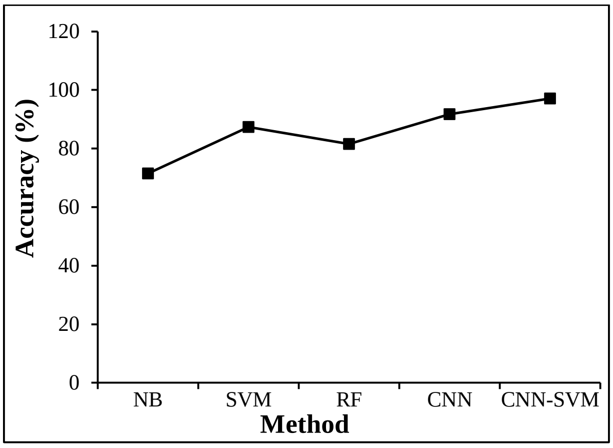

Fig.7. Comparison of our model with existing algorithms

CNN-SVM will produce specific outcomes because the benefits of CNN and SVM are completely used as follows.

- Using Embedding layer Word2vec to produce higher-quality subjective features of the sentences.
- Adding mini batch size reduce the overall losses in CNN model
- Additional performance gain was achieved by the addition of dropout technique in the hidden layers that are completely connected.

We have found empirically that the non-linear SVM method has a broad potential to enhance the efficiency of the model for sentiment classification. Without relying on all details, it can deal with interactions among nonlinear features. Apart from this, CNN-SVM has high generalization ability, it can obtain good effect even if the review contexts in the datasets are quite different. The CNN-SVM model can be generalized to different domains by just fine-tuning.

CNN is viewed by this model as a subjectivity detector. Actually, it is a feature learner that learns the local features of the input sentences automatically. It is shown to be helpful in improving the representation of functions and learning of features. In addition, the integrated model takes full advantage of the SVM classification effectiveness in terms of improving the classification ability of CNN.

The proposed algorithm is the use of sequential models such as CNN and SVM to boost the effectiveness of the detection of sentence sentiment polarity, as it was shown that improving the accuracy of sentence classification usually led to higher accuracy of document sentiment classification.

8. CONCLUSION

Sentiment Analysis is an evolving data mining field. Since more people like to buy online product as well as provide feedback, reviews and comments, it will become more essential to predict sentiment about the product. This paper explores the sentiment analysis definition and its different levels on which this analysis can be conducted in form of sentence to, document level. We established a new deep learning model in this research that collectively uses the representations of character-level, word-level to sentence-level implementation of sentiment analysis. The key contribution of the paper is the principle of using convolutional neural networks to extract features of the subjective term from character-to sentence level and document level, then SVM for sentiment classification of the polarity of entire text review. In future, we are planning to further evaluate our fusion algorithm to in-corporate this technique in any one of the big data analytical tool like spark ML into consideration.

REFERENCES

[1] Shahid Shayaa and Noor Ismawati," Sentiment Analysis of Big Data: Methods, Applications and Open Challenges", *IEEE Access*, Vol. 6, pp. 1-13, 2018.

[2] Y. Kim, "Convolutional Neural Networks for Sentence Classification", *Proceedings of International Conference on Computation and Language*, pp. 1-13, 2014.

[3] S. Xing, Q. Wang and T. Li, "A Hierarchical Attention Model for Rating Prediction by Leveraging User and Product Reviews", *Neurocomputing*, Vol. 332, pp. 417-427, 2019.]

[4] A. Sun and Y. Liu, "On Strategies for Imbalanced Text Classification using SVM: A Comparative Study", *Decision Support Systems*, Vol. 481, pp. 191-201, 2010.

[5] A. Kennedy and D. Inkpen, "Sentiment Classification of Movie Reviews using Contextual Valence Shifters", *Computational Intelligence*, Vol. 22, No. 2, pp. 110-125, 2006.

[6] G. Gautam and D. Yadav, "Sentiment Analysis of Twitter Data using Machine Learning Approaches Semantic Analysis", *Proceedings of 7th International Conference on Contemporary Computing*, pp. 437-442, 2014.

[7] A. Tripathy, A. Agrawal and S.K. Rath, "Classification of Sentiment Reviews using N-Gram Machine Learning Approach", *Expert Systems and Applications*, Vol. 57, pp. 117-126, 2016.

[8] S.A. Bahrainian and A. Dengel, "Sentiment Analysis using Sentiment Features", *Proceedings of IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies*, pp. 26-27, 2013.

[9] M. Neethu and R. Rajasree, "Sentiment Analysis in Twitter using Machine Learning Techniques", *Proceedings of International Conference on Computing, Communications and Networking Technologies*, pp. 1-5, 2013.

[10] Nadia Nedjah, Igor Santos and Luiza De Macedo Mourelle, "Sentiment Analysis using Convolutional Neural Network Via Word Embeddings", *Evolutionary Intelligence*, Vol. 6, No. 2, pp. 1-13, 2019.

[11] Shiyang Liao, Junbo Wang and Ruiyun Yu, "CNN for Situations Understanding based on Sentiment Analysis of Twitter Data", *Proceedings of International Conference on Advances in Information Technology*, pp. 19-22, 2016.

[12] A. Rozental and D. Fleischer, "Task 1: GRU Neural Network with a CNN Attention Mechanism for Sentiment Classification", *Proceedings of International Workshop on Semantic Evaluation*, pp. 1-6, 2018.

[13] A. Kamal, "Subjectivity Classification using Machine Learning Techniques for Mining Feature-Opinion Pairs from Web Opinion Sources", *Proceedings of International Conference on Computing, Communications and Networking Technologies*, pp. 332-335, 2013.

[14] B. Pang, Lillian Lee and Shivakumar Vaithyanathan, "Thumbs Up? Sentiment Classification using Machine Learning Techniques", *Proceedings of International Conference on Empirical Methods in Natural Language*, pp. 79-86, 2002.

[15] Hong Yu and Vasileios Hatzivassiloglou, "Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences", *Proceedings of International Conference on Empirical Methods in Natural Language*, pp. 132-137, 2002.

[16] Trevor Hastie, Christopher Manning and Shivakumar Vaithyanathan, "Exploring Sentiment Summarization", *Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text: and Applications Theories*, pp. 1-7, 2004.

[17] A. Krizhevsky, I. Sutskever and G.E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks", *Proceedings of International Conference on Neural Information Processing Systems*, pp. 1106-1114, 2012.

[18] A. Graves, A.R. Mohamed and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645-6649, 2013.

[19] T. Mikolov, I. Sutskever and K. Chen, "Distributed Representations of Words and Phrases and their Compositionality", *Proceedings of the IEEE International Conference on Neural Information Processing Systems*, pp. 3111-3119, 2013.

[20] Jupyter, Available at http://jupyter.org/, Accessed at 2020.

[21] TensorFlow, Available at https://www.tensorflow.org/, Accessed at 2020.