ISSN 2347 - 3983
Volume 8. No. 10, October 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter628102020.pdf
https://doi.org/10.30534/ijeter/2020/628102020

Aspect-Based Sentiment Analysis Using Hybrid CNN-SVM with Particle Swarm Optimization for Domain Independent Datasets

Raviya K[1], Dr. Mary Vennila S[2],
[1] Research Scholar, PG and Research Department of Computer Science,
Presidency College, Chennai, India, raviya.mca@gmail.com
[2]Associate Professor, Head & Research Supervisor,
PG & Research Department of Computer Science, Presidency College, Chennai, India,
vennilarhymend@yahoo.co.in

ABSTRACT

With the rise of online technology in recent years, social networking sites, discussion forums and blogs, e-commerce websites have gained immense importance. To enhance customer shopping experience, these websites generally provide platform for people to expose their view point about several aspects of the product. Individuals and organizations are posting their opinion about product, service and person on these platforms. Aggregating people sentiments, feelings, and interests articulated on these regions is a key broadcast, this is the attraction of sentiment analysis (SA). It is an application of NLP that emphases on automatic determination and classification of the sentiment in a huge quantity of text or speech. The classical sentiment analysis is a method of classification of sentiments conveyed in a text as positive, negative or neutral. Aspect-level sentiment classification is a well-grained job and provide additional in-depth information than over-all sentiment analysis, because it tries to forecast the sentiment polarities of each aspect or entities in the text review. Recently, deep neural networks have shown exciting and inspiring result in this field. Particularly convolutional neural network (CNN) has fascinated extensive attention since its amazing functioning in several applications including text analytics. However, building a powerful hybrid aspect-based sentiment analysis model utilizing CNN can be highly complex and expensive. In this paper, we suggested novel intelligent framework based on hybrid convolutional neural network and support vector machine (SVM) for aspect-based sentiment detection and classification of online product reviews. The architectural efficiency is augmented by standardising the parameter values of CNN and SVM using PSO algorithm by including the concepts of single-objective optimization (SOO). The suggested hybrid framework combined the SVM 's outstanding classification performance with CNN 's powerful feature-learning ability. The hybrid framework is competent to producing high accuracy for given data sets and more superior to traditional aspect-based sentiment analysis method.

Key words: Aspect based sentiment analysis, Convolutional Neural networks, Particle Swarm Optimization.

1. INTRODUCTION

Research flashes that consumers spend about 8 hours a day on online media, especially social network and mobile net applications. It has entirely transformed the life routine of the people. For purchasing any product or receiving any service, clients now rely heavily on review-based information accessible through many shopping websites, tweets, blogs etc. They need to ensure that the items that they purchase or the amenities that they obtain are of exclusive. Whether ordering an item in an online website or like to go to the restaurant for dine or planning to see a film in the theatre, they always try to find other users' point of view about the items and facilities before they enjoy themselves. The enormous capacity and velocity of user-produced content makes it very hard to manually experience all the data. However, one needs to read and extract all the reviews with the purpose of getting an informed opinion on a product or service.

Unfortunately, it is not a simple task to achieve and the massive amount of time and execution essential for this task. Therefore, tools and methods need to be developed to assist consumers in extracting the required data from the large volume of collected reviews. Sentiment Analysis [1] is the most well-recognised and complex activity in the context of the processing of natural languages. It is the technique to looking at a viewpoint attached in a text by determining whether it is positive, negative, or neutral. The level of polarity that is well established as opinion mining is also noticed (high, moderate, low). It analyses the feeling, thinking, attitude of collecting opinion on different websites as a review. This knowledge not only benefits the person requesting data, but also enables the method of streamlining different corporate decisions in order to increase the eminence of goods or amenities.

Opinion mining can be achieved by two primary methods, the first is the approach to machine learning and the next is a dictionary-centred approach. In addition, the method based on machine learning is divided into two sections, both supervised and un-supervised. There is a large amount of existing techniques for sentiment analysis in different areas [4, 5, 6]. It should be observed that maximum number of these current study is to identify the document 's sentiments [4] or sentence level [5]. However, methods that focus on such rough-level analysis do not always meet the needs of users according to their expectations. sentiment analysis on user-generated reviews can provide useful information for suppliers and customers. Instead of predicting the total perceptual polarity, better-aspect-based sentiment analysis (ABSA) is proposed to well understand feedback than conventional methods. The primary objective of the study of aspect-level sentiment analysis can be considered to be the actions of aspect extraction and then detect the sentiments that are stated in the review text [2, 3]. Two tasks are concerned with most aspect-based sentiment analysis methods, namely aspect identification and classification of sentiment. The flow is shown in Fig.1. In order to acquire the sentiment class of each aspect, the outcome of the aspect detection model and the initial word vectors are moved to the sentiment classification model.

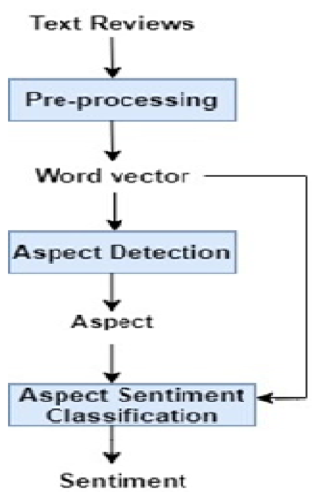

Figure 1: Process Flow of the Aspect level Sentiment Analysis.

The first job is to define all aspect terms appeared in the review. It can be defined as a problem of sequence labelling because it relies heavily on structural and contextual data., while the second task estimate sentiment polarity for the given aspects. In Fig.2 the review contains two aspect terms ‘Location’ and ‘Food’. Sentiment of these two aspect terms are opposite in the review. it has a positive sentiment for the one aspect term (Location) and negative sentiment for the next aspect term (Food).

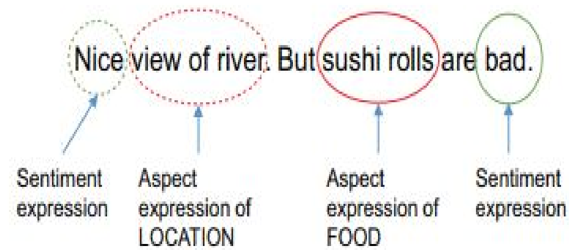

Figure 2: Text review with aspect-based sentiments

Analysis of sentiments at the aspect level will help customers to gain more understanding of others opinion about different aspects of the resulting object. The decision taken afterwards is therefore, more insightful and achievable. In this paper, we suggested a novel hybrid method for aspect-based sentiment analysis based on convolution neural network with SVM optimized by PSO for improved classification accuracy. The system setup of the suggested technique is shown in Fig. 3.

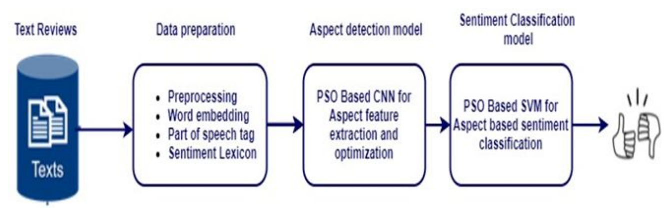

Figure 3: System setup.

To sum up, the suggested approach contains three key contributions. Firstly, the proposed techniques straightaway extract the aspect features from the embedded words by applying CNN (aspect detection model). Secondly, the outstanding small sample learning facilities of the SVM is joint with the formidable deep feature adaptive learning ability of CNN to enhance the classification accuracy (sentiment classification model). Finally, the PSO is offered for the parameter tuning of CNN and SVM, which can highly increase the F-measure on aspect feature extraction and classification accuracy and generalization functioning of SVM. The Hybrid approach with ten-fold cross-valuation reveal the efficiency and advantage of the proposed method.

2. RELATED WORK

A substantial amount of research work in sentiment analysis has been conducted in the last few years. A wide variety of classification algorithms have been used to classify aspect-based sentiments in many applications. The extraction of aspects from views was primarily examined by Hu and Liu [2]. They also incorporated the gap among direct and indirect aspects. The authors, however, addressed precise aspects only and applied a set of rules created on statistical explanations. We will review the latest works relevant to the study of aspect-based sentiment in this section. The authors of [7] improvise the aspect-based sentiment analysis by integrating common domain knowledge into an ontology. For both aspect identification and aspect sentiment classification, it is possible to increase effectiveness. They suggested two separate algorithms, a review-level and an accumulated sentence-level algorithm, and improved with the application of an ontology. The precision is substantially advanced for these two algorithms with the use of ontology. But manual creation of ontology is very time-consuming and labour-exhaustive process. For sentiment analysis, recurrent neural networks (RNN) and convolution neural networks (CNN) have recently shown advanced results. Wang and Liu [8] are the key reference we applied to construct deep learning based on CNN to implement aspect-level sentiment analysis. Aspect detection not only generates predicted aspects but also generates likelihood in every aspect of every word that occurred. It will be manipulated as an input for the sentiment classification model. In order to do the sentiment classification, Xue and Li [9] applied Gated-CNN. The aspect function regulates the transmission of sentiment with embedded aspect of the specified aspect class. Kim [10] recorded a series of convolution neural network (CNN) experiments for sentence-level classification tasks on the pre-trained word vector. Simple CNN with little hyper parameter tuning gives outstanding performance on various datasets. Ekawati and Khodra [11] construct a framework consisting of three steps: aspect identification, categorization of aspects, and classification of sentiments. Cahyadi and Khodra[12] developed a new approach with similar framework and use deep learning on aspect categorization and classification of sentiments. Abdelghani et al. [13] offered DE-CNN frame work to define optimal parameters of CNN, and augment the implementation of Arabic sentiment analysis. The effectiveness of the projected DE-CNN method is measured on the basis of five Arabic datasets. The outcomes of the experiments indicate that DE-CNN is more accurate and it take less amount of time than other algorithms. Ravindra Kuma et al. [14] projected a model for aspect-level sentiment analysis by deep convolutional neural network with PSO for multi objective optimization effectively joining three actions (a)the development of ontologies for the mining of semantic characteristics (b) Word to vector conversion for the handled corpus (c) CNN for aspect- based sentiment classification and it achieved 88.52% better accuracy. For aspect-based sentiment analysis Recently, several methods were proposed which worked on selection of features before classifying reviews to improve classification accuracy. Shad Akhtar et al. [15] presented a cascaded framework of feature selection method for trio distinct classifiers, namely CRF, SVM and ME, and successful PSO-based ensemble construction techniques for aspect term extraction and sentiment classification that achieves higher efficiency in two distinct domains. In many published works, the PSO is used with SVM classifier to advance the output of the classifier [16,17]. The implementation of PSO in a CNN-SVM joint model is the observing factor in the proposed work. As per our perception, this is the first instance of the implementation of a PSO-optimized CNN-SVM architecture for aspect-based sentiment analysis. The PSO-based CNN-SVM architecture attained highest result then other methods. Results confirms that the handling of PSO enhances the accuracy of classification and also reduces computational time.

3. PRELIMINARIES

The proposed frame work is achieved in the phases of data collection, pre-processing, Word2vec conversion and execution of aspect detection and sentiment classification models. we described the phases one by one.

3.1 Dataset

Data is gathered through Python web scrapping and pre-processing to extract valuable insights from the raw dataset. Word2vec is obtained by using an unsupervised neural network for the processed dataset. Finally, with tenfold cross-validation, a vector shape of the used corpus is trained via the CNN for aspect feature extraction and it is followed by SVM for aspect- based sentiment classification. We browse different online sources and collected 17,154 reviews created by users, which belong to three distinct domains. The list of aspect categories for three domains are mentioned in Table1 is described and compiled.

Table 1: Aspect categories of different domains

Domains	Aspect Categories
Smart Phone	Display, Camera, Storage, Battery, Price, Misc.
Bluetooth Head phone	Sound Quality, Range, Reliability, Price, Misc.
Car	Design, Safety, Fuel consumption, Comfort, Price, Misc.

We adopt the same Sem-Eval mutual task scheme for dataset annotation. We identify and save different aspect categories of each review analysis along with their related sentiment in an XML format.

3.2 Pre-processing

In this phase, reviews collected on the internet provide syntactic characteristics and it is not helpful in aspect-based sentiment analysis. There are also lots of stop words, slang words, URLs, etc. that are appeared in the reviews that slow down the classification process of sentiment. These HTML tags and special terms are also the greatest challenge for extracting aspects in product reviews. To clean up the unprocessed data, the pre-processing steps are followed as shown in Fig. 4.

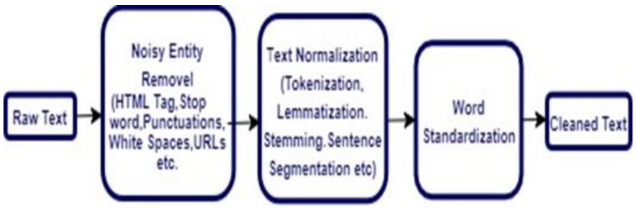

Figure 4: Pre-processing technique

These data are cleaned by using the Panda library and only the useful instances of data are kept. Pre-process step includes sentences splitting, normalization, and removal of stop word then tokenization. We can isolate expressive words and letters in the text and eliminate white spaces by tokenizing the data. After tokenization, the words of the sentences are added to Part-of-Speech tags such as 'noun' and 'adjective'. We lemmatize terms to identify various forms of the word as the same. It means we are finding a word's dictionary type. The final phase is to parsing the data that find the grammatical shape of the sentence. Later on, this information can be used to classify the interrelated words. Finally, to perform word level embedding, Word2vec is used.

4. PROPOSED TECHNIQUE

This paper proposes an efficient two-stage process of aspect detection and sentiment classification method for fine-grained aspect-based sentiment analysis. In addition, we want to examine the advantages of a hybrid classifier that combines deep learning and supervised machine learning. with PSO algorithm. The process flow of this approach is as follows.

1. A systematic Word2vec is then generated using an unsupervised neural network for the given pre-processed data and it is used to trained through the CNN classifier.
2. The convolutional neural network (CNN) has been employed for the extraction of aspect-based sentiment features.
3. Tuning of CNN parameters is done with particle swarm optimization (PSO) for aspect detection and optimization.
4. CNN's soft-max output layer is replaced by SVM in our proposed process.
5. PSO is often used to optimize SVM parameters for aspect-based sentiment classification.

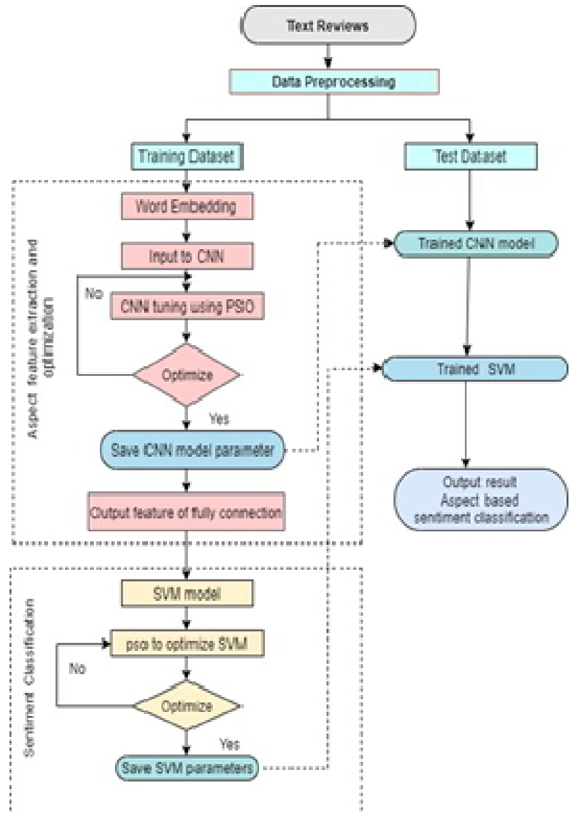

Figure 5: Architecture of proposed models

The primary attention of this work is on the role of CNN for the extraction of aspect-based features. The CNN-PSO extracts optimal features from the review data based on aspects, but does not constantly give the desired outcomes for classification. High classification efficiency can be attained by introducing an efficient classifier like SVM [18] in the CNN architecture by changing the output layer with SVM classifier. The complete architecture of the suggested models is exposed in Fig. 5. There are various machine learning methods applied for data classification. The SVM is robust machine learning classifiers. In this work, the CNN is accompanied by the SVM classifier for extracting and classifying the aspect sentiment features from the review data set. We have developed a hybrid PSO-based CNN-SVM frame work, which is a fusion of a powerful feature extraction and classification technique. The PSO is adapted for improving the parameters of both CNN and SVM to obtain better aspect feature extraction and sentiment classification performance.

4.1 Aspect category detection model

Since CNN can't explicitly understand the sentence. We need to turn the text data into numerical form in order to follow the deep learning method for the NLP function [19]. CNN's first layer is the embedding layer of terms. This layer is used to obtain the word embedding vectors computed by using word to vector method. For this layer, the output is the list of words with their resultant 300-dimensional vector form w_i. and then it passes through a number of layers that extract its features with an augmented degree of complexity, A logistical distribution of each aspect is the fully-connected layer output. The best performance is attained by designing the network architecture and CNN tuning for the task of optimized aspect category detection as follows.

4.1.1 Convolutional neural network (CNN)

The characteristics of the aspect term relies on the phrases around it. We thus exercised a five-word window in a sentence all around each word, i.e., ± 2 words. We generated the local features window and calculated them to be middle word features. Then, CNN was then supplied with the function vector. One input layer, two convolution layers, two max-pool layers, and a fully integrated soft-max output layer were included in the network. Network feature extraction can be done at the level of sentences and phrases. The innovation of the network construction is to have two convolutional layers that allow sentences and words of any size to be handled. The first layer contained 100 feature maps with filter range of two and 50 feature maps in second convolution layer with filter range of three. The phase in each layer of convolution is one because we required to tag each word. convolution layers are followed by max-pool layers. In the max-pool layers the pool size we used of 2. We exercised regularisation on penultimate layer with dropout with a restriction on weight vector of L2-norms, of 30 epochs. Using a non-linear function, the outcome of each convolution layer was calculated; we have used hyperbolic tangent in our case. The CNN creates local features in a sentence around each word and then integrate these features in to global feature vector. The dimensionality Lx × Ly of these layers is 3 × 300 and 2 × 300, respectively, As the kernel size was different for the two convolution layers. The embedding layer was 65× 300, where the highest count of words in a phrase is 65, and the word embedding dimensionality is 300 per term. The procedure was implemented for each word and each character in the sentence. In contrast to the conventional max-likelihood leaning method, after converting all tokens in the sentence, we trained the network using back propagation. After convolution, we saved the weights, biases of all token and only back propagated the oversight to precise them until all tokens were handled by means of training scheme. For that case, if a training example s had n terms then we denoted the input vector as s_1: n = s_1 $s_i \in R$, For the word si, k is a k-dimensional vector of features. We noticed that the architecture of this network yielded high performance on domain independent text review dataset.

4.1.2 Feature extraction

Aspect-based sentiment analysis uses the following features:

1. **Word embeddings**: the CNN use word embeddings feature as an input for the system. Each and every word was thus encoded as a 300-dimensional word vector that was supplied to the network. Word2vec is for word level embedding.
2. **Part of speech tags:** We used the word's POS tag as its additional characteristic. In order to find the aspect from reviews, Pre-processed reviews are given to POS tagger. Most of the words of aspect are moreover nouns or clumps of nouns. We used 6 basic sections fixed as a six-dimensional binary vector of speech classes (noun, verb, adjective, adverb, preposition, conjunction). Stanford Tagger as a POS tagger for transferring words to part of speech classes.
3. **Sentiment lexicon:** It is the extensively used features for sentiment analysis It is based on sentiWordNet, which assigns positivity and negativity sentiment score to each synset. All the terms of the desired aspect that occur in the contextual sense of the preceding five and subsequent five words are gathered and summed. As a function, the sentiment score value obtained as a result of this function is used.

All the vectors of features have been concatenated and fed to CNN. Therefore, the final feature vector is 300 dimensional for each word. The word vocabulary is a v^{word},of fixed size and words consist of fixed-size v^{char} characters from character vocabulary. The sentence consists of $[W_1, W_2, \ldots, W_n]$ of n words, and conversion of W_n into $Vn = (r^{word}; r^{wchar})$ comprise of two fold sub-vectors: the embedding word level $r^{word} \in R^{d^{word}}$ and the embedding character level $r^{wchar} \in R^{d^{ll}_{n}}$ .Word level embedding is expected to obtain syntactic and semantic details, and character level embedding is projected to grab type and morphologic details.

4.1.3 Word embedding

Word embeddings are distributed text representations that encode the words' semantic and syntactic properties. Column vectors convert the level of word embedding in an embedded matrix $w^{word} \in R^{d^{word}} \times v^{word}$ The embedding of word level belongs to each column $w_i^{word} \in R^{d^{word}}$ from that the i^{th} word vocabulary is produced. A word W converts into word level embedding by means of the matrix-vector product: $r^{word} = v^w w^{word}$ where $v^w \leftarrow$ vector dimension $|v^{word}|$ has value of 1 at index w,0 on other places. The w^{word} matrix is a learning parameter. The user who selects the hyperparameter is the embedding word level size of d^{word}. In this article, Word2vec is utilized for word level embedding. Word2vec is tool offered by Google in 2013 in the Appache License 2.0.0 as an open source. Without any human involvement, it extracts the features from a specified text corpus. Mainly, it will work fairly fine, even if the size of a text is small or merely a single word. Word2vec provides an acceptable word sense by having a suitable corpus meaning and uses, and it also works quickly with large amounts of data. The context of words is one of the highest significant aspects of deep learning that is entirely content by the use of Word2vec for the characterization of larger volume. Data source on Google News (approximately words of 100 billion) is used to trained the vectors in the proposed technique.

4.1.4 Character embedding

In all characters of words that choose the main aspects for the classification process, rigorous process for obtaining type and morphological details from words are taken into account. The local features are created over each character of the word by a convolutional method and further max operation is employed to integrate them to create fixed-size word embedding at the character level. Since words consist of 'm' character [k1, k2, ..., km], each of km character is converted into the r_m^{char} embedding character. Column vectors interpret the modularizing character in the $w^{char} \in |v^{char}|$ embedding matrix. The matrix-vector product gets its embedding r^{char} due to k characters: $r^{char} = v^k w^{char}$ where v^k vector size of $|v^{char}|$ is with value of 1 at index K and 0 on other places. The input of the convolution layer is the set of embedded characters $\{r_1^{char}, r_2^{char}, r_3^{char}, \ldots r_n^{char}\}$. $W^0 \in R^{c_u^0 \times d^{char} r^{char}}$ ← [Weight matrix of the Convolution layer] Then, in a given term, local characteristics are extracted using the similar matrix on each character of a window element.

4.1.5 Sentence level embedding

A sentence x with m words is provided $\{w^1;w^2;...w^m\}$ which is then translated in to joints of word level and $\{u^1,u^2;\ldots,u^n\}$ are embedding level character. The next step is the representation of extraction at the sentence level r_x^{st}.
Two main problems faced by the sentence-level extraction of the feature set method is

1. Various sentence sizes
2. The essential data can be in any position in the sentence.

These problems are undertaken by the usage of convolution layer to determining the feature vector of large sentence r^{st}. In the frame work of CNN, the next convolution layer functions similarly to the layer used to derive character-level characteristics. Specific features are generated in this layer about each word, and they are connected after using max process to construct fixed size feature maps of the sentence. For each window size of k^{word} a matrix vector process is implemented by a second convolution layer in a consecutive window series {{u1,u2,... ,un}. The vector $x_m \in R^{(d^{word} + c_u^0) k^{word}}$ does the chain of embedding k^{word} in n^{th} word. where $w^1 \in R^{(d^{word} + c_u^0) k^{word}}$ ← [weight matrix of Convolutional layer]. Ultimately, r_x^{st} vector with the feature vector universal qualified to x sentence is evaluated in two adjacent layers of neural network.

4.1.6 CNN tuning using PSO

Particle Swarm Optimization (PSO) is an inhabitants-dependent stochastic optimization approach [21] focused on the genetic algorithm-like effect of flocking birds. PSO also begins with a collection of random solutions and by updating the generations, it searches for the global optima. In this section, we described PSO based optimization for aspect feature selection and optimization in this model. To optimise more than one function simultaneously, multi-objective method of optimization is used. Initially, a single-objective method of optimization function is developed to evaluate the best suitable feature sets for aspect term extraction and sentiment classification. The PSO is single objective method of optimization technique (SOO) in which single function is optimized at a time [20]. The features we are utilizing for our activities contain lexical, syntactic and semantic level information. We select the most important features in a way that increase the objective functions If the classifiers are trained in these unique combinations, we optimise the F-measure for aspect feature extraction, while we optimise the classification accuracy for sentiment classification. A number of vectors, some of which refers to a specific grouping of features is the output of this operation. Aspect optimization is done by tuning CNN using PSO as follows. During neural network training, choice of the number of iterations and learning rate has a major influence on network performance. Hence this paper chooses these two parameters to be optimised. The binary PSO approach will be used in this step to iteratively refine the combination of features. Finally, in the case of binary vectors, T quantized implementation process can be produced. The PSO input location vector composed of CNN weight and bias variables to generate the metrics of analysis as optimal solutions. CNN contains two most important layers such as convolution and max-pooling to bring the final fully connected layer. These layers midway relations consisting of weights. CNN parameters are now condensed in the way of X = {W, B} into the PSO particles, where W = $\{W_i^j\}$ is the weight and B = $\{b_i^j\}$ is the bias vector. Three operations, namely assessment, comparison and imitation, rule the entire PSO process. The assessment process tests how well to measure each particle, i.e. The applicant's resolution fixes the issue at hand. Through comparing distinct solutions, the evaluation process aims to find the finest particle. The process of imitation creates innovative particles centred on some of the top particles that have been realized up to this point. Until to reach the specific stopping condition, these three processes are repeated. The aim is to obtain the particle that is finest to resolve the problem of destination. The two significant notions in PSO are velocity and neighbourhood. In every PSO iteration, by changing the velocity variable, A solution moves to a unique value. For each solution particle, the vector of velocity is revised comparative to global best and local best positions (Gbest and Lbest) when resolving for the cost function C. The v_i is the velocity and x_i is the position at current state j of the particle i:

$$v_i^{k+1} = w \times v_i^k + C_1 \times r_1 (lbest_i^k - x_i^j) + C_2 \times r_2 (Gbest^k - x_i^j) \quad - (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad - (2)$$

C_1, C_2 are learning factors for social and cognitive terms, w is inertia weight, random numbers r1, r2 $\in$(0,1). The pseudo code for feature evaluation using PSO is outlined in Fig6.

Initial feature set: G_d= {g_1,g2,.....,g3}

Number of MFCV subsets: T

Population size in single PSO : M

Maximum number of iterations: I

Output:

Optimal d' features bestF{g_1,g2,......,g3}

Begin

1) Divide review set R into T different training sets and evaluation sets
E_t (S_tU E_t = R, t=1, 2 , ...T)
2) For t=1 to T do
3) Initialize feature evaluation result
$f_t(G_d)$ = (0,0,....0)
4) For I = 1 to I do
5) Randomly generate M binary particles $\{X_p^i\}_{p-1}^{M}$ on search space S_t
6) Compute the individual fitness (X_p^i)
(p = 1, 2, ...,M) of these particles on E_i
7) Update global optimal solution $gbest_q$ (1≤q≤d) and local optimal solution lbest with fitness (X_p^t)
8) Adjust the position X^t= [$X_1^t, X_2^t, X_3^t, \ldots X_m^t$]^T of M particles to optimize the feature combination
9) End For
10) Obtain and save t-th feature evaluation result by sum rule as $f_t(G_d) = \sum_{q-1}^{M} X_q^t$
11) End For
12) Select the best features
bestF(g_1,g2,.....,g3)=decending$\left(\sum_{t=1}^{T} f_t\,(G_d)\right)$

Figure 6: Feature evaluation using PSO

This model 's output is the distribution of probabilities over the quantity of labels. The label vector is a size of n, The aspects specified in advance, O = {O_1 , O_2 , …, O_n}. For k ∈ [1 , n] O_k = 1/y where y is the maximum number of related aspects present in the review, else O_k = 0. The model will use soft-max cross-entropy to measure losses. We construct similar models with no hidden layer to generate the matrix of aspect that denotes the likelihood of each word occurring in each aspect, so weights on the output layer have the factor of W x A where W is the set of different words and A is the set of aspects. As an aspect matrix, the weights are extracted and used.

4.2. Sentiment Classification Model

The second model on this approach adopts SVM-PSO for handling aspect-based sentiment classification. But, SVM gets a linear hyperplane by exploring input data in to higher dimension and generalizes well. In our proposed analysis, we integrate this principle by substituting the soft-max regression with SVM on CNN's output layer. Once after the aspect categories are defined, Sentiments related to these aspects are classified in to one of the 4 classes of sentiment polarity, namely positive, negative, neutral and conflict. For all pre-defined aspects, every sentence has a label. If a sentence is not expected to have a particular feature, there is no label on that aspect. Otherwise, the sentence with aspect labels are given to sentiment classification model. The input to sentiment classification model is output of CNN, the predicted aspect word vectors w_i which is convolved with their respective embedded sentiment(si) and POS tag (pi) for each word. The embedded sentiment is collected from sentiment lexicon, it contains of words with sentiments of positive and negative, neutral etc.

4.2.1 Parameter Tuning of SVM for sentiment classification

A support vector machine (SVM) is an effective machine learning method established by Vapnik [23]. SVM is frequently used as a robust learning algorithm for classification and regression and it is highly viable for tasks related to higher range of text classification. but the SVM classifier's penalty factor c and kernel parameter depend primarily on manual assortment, which has a significant effect on the SVM classifier's accuracy. In general, nonlinear fitting capacity is expressed by the penalty factor c. If the parameter c is enormously high, the generalisation capability of the SVM will be lost [22]. The size of the kernel parameter disturbs the distribution of the kernel mapping of the sample data. Parameter tuning is therefore necessary for the SVM to achieve high accomplishment in the classification. Hence, we suggested PSO based SVM for sentiment classification model in this paper. This model is mainly designed for enhance the accuracy of the SVM classifier by defining the best useful characteristics and the best principles for regularisation of SVM based kernel parameters c, σ and ε are determined. The regression function of SVM is represented by the Equation (3).

$$K(x_i, x) = exp(-\|x_i - x\|^2/2\sigma^2) + \frac{C\varepsilon}{2\sigma^2} \quad (3)$$

$[10^{-3}, 10^{+3}]$ are the range of C and σ then [0,1] is the range of ε

To experience this improved classification model, this technique comprises two machine-learning algorithms by refining the SVM parameters by means of PSO. It begins with n-randomly selected particles and iteratively search for the optimal particle. Each particle is an m-dimensional vector and signifies the solution of the applicant. For each selection strategy, the SVM classifier is designed to assess its effectiveness via the cross-validation process. The PSO governs the collection of possible subdivisions that provide the greatest accuracy for forecasting. The fitness of each particle is then measured. To quantify the quality of each particle, the classification capacity of the function subset related to the location of each particle vector must be determined. The support vector machine (SVM) is used to calculate the discriminability of features for this purpose. The PSO operation is regulated by the velocity vector and position vector using equations 1 and 2.

Geometrically, in the hypothetically very high-dimensional space, the goal of SVM is to discover a linear discriminating function with the highest probability (i.e. to optimize the range among the positive and negative data points). Once after the fitness process, the comparison of global finest and local finest parameters has been evaluated. The weight and position of each particle will be adjusted till to converges the value of the fitness process. The global finest particle in the swarm is supplied to the SVM training classifier after converging. Finally, it will train the SVM classifier.

5. RESULTS AND DISCUSSION

In order to see the efficiency of the overall framework, we conducted an experiment on all three domains i.e. 'smart phone', 'head phone', 'car'. The data is separated into two parts, with 70% for training and the rest for models testing.

We constructed and assess two baselines for the aspect model and sentiment classification model prior to conducting the entire experiment. Testing on aspect detection model is done by measuring precision and recall and average f1-score value and accuracy for sentiment classification. The optimization of parameters for CNN and SVM using PSO is carried out in the training process through closed testing on 10-fold cross validation. The different settings were used for different models. We optimise the F-measure for the aspect detection model, then we optimise the classification accuracy for the sentiment classification model. The dataset includes 17,154 user reviews associated with three products. A total of 8,528 positive, 1,837 negative, 6,518 neutral and 271 conflict occurrences of aspect categories. The summary of statistics from the dataset is provided in Table 2. We use optimised CNN and SVM parameters in the testing stage. The achieved performances on given dataset by various models are shown in Table 3, Fig.7. It was noted that the assessment metrics implemented in NLPCC-SCDL are used for our experiment. It is detected that our proposed PSO based CNN_SVM combined model outshines the baseline CNN-based methods. We obtain the average F-measures of 87.17%, 86.49%, 86.68% for aspect category detection model in smart phone, head phone, car domain, respectively. our sentiment classification model reports the accuracy of 88.45%, 87.51%, 90.37 % for the three domains respectively. Related to the CNN-based model, the outcome clearly demonstrates that the contribution of our PSO based CNN_SVM works well for the domain independent dataset.

Table 2: Data statistics

Domains	Aspect Category	Polarity				
		Pos	**Neg**	**Neu**	**Conf**	**Total**
Smart Phones	Display	700	261	763	73	1797
	Camera	160	55	149	6	370
	Storage	305	69	137	13	524
	Battery	110	31	83	4	228
	Price	70	19	30	3	122
	Miscel	290	89	173	21	573
	Total	1635	524	1335	120	3614
Bluetooth Head phones	Sound Quality	800	121	687	40	1648
	Range	643	92	548	14	1297
	Reliability	589	67	431	19	1106
	Price	340	43	99	8	490
	Miscel	250	96	211	12	569
	Total	2622	419	1976	93	5110
Car	Design	943	256	896	11	2106
	Safety	659	35	490	6	1190
	Fuel consume	883	87	567	8	1545
	Comfort	543	72	366	9	990
	Price	987	321	655	16	1979
	Miscel	256	123	233	8	620
	Total	4271	894	3207	58	8430
Over All		8528	1837	6518	271	17154

Table 3: Performance result of two models

Domain	Method	Aspect Category Detection			Sentiment classification
		Precision	Recall	F-score	Accuracy
Smart phone	CNN	85.48	79.25	82.24	81.31
	CNN-SVM	87.42	85.01	86.20	86.76
	CNN-SVM with PSO	88.27	86.10	87.17	88.45
Bluth tooth Head phone	CNN	86.01	81.32	84.17	82.42
	CNN-SVM	87.35	84.11	85.70	86.13
	CNN-SVM with PSO	89.40	83.77	86.49	87.51
Car	CNN	86.21	85.05	85.11	82.67
	CNN-SVM	87.24	85.09	86.40	87.21
	CNN-SVM with PSO	90.15	83.47	86.68	90.37

The current approach used a PSO optimizer with a level of 0.01 learning rate for the aspect detection model. weight restriction was included in the dense layer and dropout is applied in the convolution layer. When embedding words were used along with the POS functionality, the highest precision on this dataset was achieved. This illustrates that whereas the word embedding features are greatest beneficial, the POS function play a key part in extracting aspects. Table 4 illustrate the performance output of various frameworks for aspect extraction.

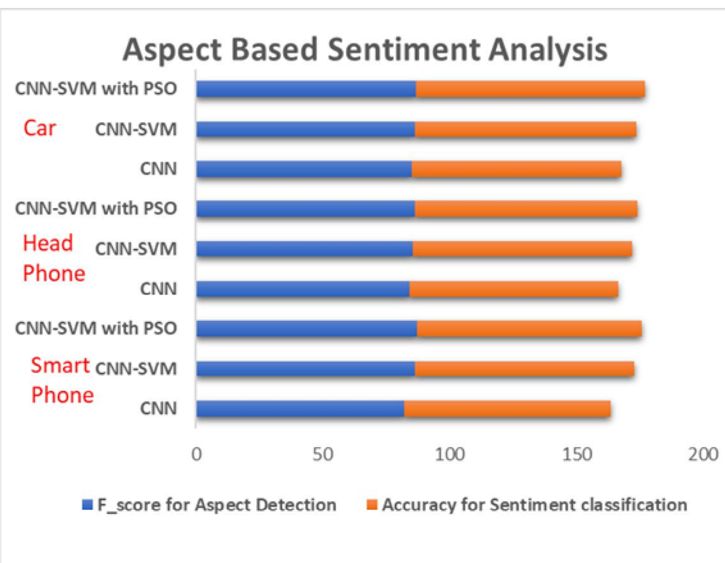

Figure 7. Performance measures in percentage

Table 4: Effect of POS in aspect detection

Domain	Features	Precision	Recall	F-Score
Smart Phone	WE	85.23	84.11	83.91
Smart Phone	WE+POS	88.27	86.10	87.17
Blue Tooth Head Phone	WE	84.60	80.18	83.33
Blue Tooth Head Phone	WE+POS	89.40	83.77	86.49
Car	WE	86.54	81.32	83.32
Car	WE+POS	90.15	83.47	86.68

Some experiments have been done on sentiment classification model by using group of features such as word embeddings (WE), sentiment lexicons (SL), and POS tags (POS) show in Table 5. It can perform well for forecasting polarity labels and produce better result also. Our experimental outcomes exhibited that this hybrid algorithm PSO based CNN_SVM would better understand the text semantics than the pure machine learning algorithm, and extracts more prominent aspect terms. We assume that there are two key approaches in our framework to outshine the advanced methods. First, a deep CNN, which in essence is non-linear, highly extract the aspect features than linear type. Second, the feature of embedding terms assists our system to exactly identify both aspect and related sentiments. The main benefit of our system is to improve the classifiers' performance by using PSO based optimization technique. All this configuration minimises the expense and time.

Table 5: Sentiment Classification Accuracy with different features

Features	Smart Phone	Blue Tooth Head Phone	Car
WE	88.21	86.87	89.38
WE+SL	88.33	87.10	89.83
WE+POS	88.40	87.30	90.15
WE+POS+SL	88.45	87.51	90.37

6.CONCLUSION

Aspect level sentiment analysis is an essential activity for many social media analytics such as medial, political, market research, etc. In this study, we work out the experiment to enriching the quality of aspect-level sentiment analysis and its sub-tasks. The main aim of this research is to extract an aspect of the target entity and its associated sentiment on different domain using convolutional neural networks and SVM on the joined approach for the better performance. For the attainment of deep learning, a greater number of researchers make an effort to solve the complex sentiment analysis task by using deep learning algorithms. Here The PSO-based optimization effectively improves the efficiency of the classifiers by training them with the most compatible set of features. The current work consists of a single objective technique of optimization, where we are just concerned with only one objective attribute at one time. We like to discuss in the future how multi-objective optimization that handles the optimization of various target objective function is successful in solving issues and we would like to further refine the process by extending our framework to some other deep learning algorithm with automatic optimization techniques.

REFERENCES

1. B. Pang, L. Lee, **Opinion mining and sentiment analysis, Foundations and trends in information retrieval** 2 (1-2) (2008) 1–135.
2. M. Hu, B. Liu, **Mining and summarizing customer reviews**, in: Proceedings of the 10th KDD, Seattle, WAs, 2004, pp. 168–177
3. B. Liu, **Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies**, Morgan & Claypool Publishers, 2012.
4. P. D. Turney, **thumbs up or thumbs down: Semantic orientation applied to unsupervised classification of reviews,** in: Proceedings of the 40th ACL, 2002, pp. 417–424.
5. S.M. Kim, E. Hovy, **Determining the sentiment of opinions**, in: Proceedings of the 20th international conference on Computational Linguistics, 2004.
6. V. Jagtap, K. Pawar, **Analysis of Di□erent Approaches to Sentence-level Sentiment Classification**, International Journal of Scientific Engineering and Technology (ISSN: 2277-1581) Volume 2 (2013) 164–170.
7. Sophie de Kok, Linda Punt, Rosita van den Puttelaar, Karoliina Ranta, Kim Schouten1, Flavius Frasincar1, **Review-aggregated aspect-based sentiment analysis with ontology features**, Progress in Artificial Intelligence-7 pp.295-306 (2018),
8. Wang B. and Liu M., "**Deep Learning for Aspect-Based Sentiment Analysis**", Stanford University report, 2015
9. Xue W. and Li T. "**Aspect Based Sentiment Analysis with Gated Convolutional Networks**", May 18, 2018
10. Kim Y (2014) **Convolutional neural networks for sentence classification**. arXiv preprint arXiv:1408.5882.
11. Ekawati D. and Khodra M. L., "**Aspect-Based Sentiment Analysis for Indonesian Restaurant Reviews**," in Advanced Informatics: Concepts, Theory and Application (ICAICTA), 2017.
12. Cahyadi A. and Khodra M. L., "**Aspect-Based Sentiment Analysis Using Convolutional Neural Network and Bidirectional Long Short-Term Memory**" in Advanced Informatics: Concepts, Theory and Application (ICAICTA), 2018.
13. Abdelghani Dahou, Mohamed Abd Elaziz, Junwei Zhou and Shengwu Xiong , "Arabic "**Sentiment Classification Using Convolutional Neural Network and Differential Evolution Algorithm**", Hindawi Computational Intelligence and Neuroscience Volume 2019, Article ID 2537689, 16 pages.
14. Ravindra Kumar, Husanbir Singh Pannu, Avleen Kaur Malhi," **Aspect-based sentiment analysis using deep networks and stochastic optimization**", Neural Computing and Applications 32, pages3221–3235(2020), part of Springer Nature,
15. Md Shad Akhtar, Deepak Gupta, Asif Ekbal, Pushpak Bhattacharyya," **Feature Selection and Ensemble Construction: A Two-step Method for Aspect Based Sentiment Analysis**", Knowledge-Based Systems (2017).
16. Mochamad wahyudi, Dinar Ajeng kristiyanti**, sentiment analysis of smartphone product review using support vector machine algorithm-based particle swarm optimization** journal of theoretical and applied information technology September 2016. vol.91. no.1
17. Nirmala Devi, K., Jayanthi, P.: **Sentiment Classification Using SVM and PSO**. Int. J. Adv. Eng. Technol. VII(II), 411–413 (2016).
18. Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: **A Library for Support Vector Machines.** ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27.
19. Ouyang X, Zhou P, Li CH, Liu L (2015) **Sentiment analysis using convolutional neural network**. In: 2015 IEEE, pp 2359–2364
20. D. K. Gupta, K. S. Reddy, A. Ekbal, et al. 2015. **PSO-ASent: Feature Selection Using Particle Swarm Optimization for Aspect Based Sentiment Analysis**. In Natural Language Processing and Information Systems, pages 220–233. Springer.
21. J. Kennedy, R. C. Eberhart, **Swarm Intelligence**, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001
22. S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee, **Particle swarm optimization for parameter determination and feature selection of support vector machines**, Expert systems with applications 35 (4) (2008) 1817– 1824.
23. V. Vapnik, **'The Nature of Statistical Learning Theory'**, Springer- Verlag, 2000, Pages 863-884