International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V70I1P217
Volume 70 Issue 1, 154-163, January, 2022

Enhanced Sentiment Classification for Dual Sentiment Analysis using BiLSTM and Convolution Neural Network Classifier

Mamatha M[1], Rakshith Shenoy[2], Thriveni J[3], Venugopal K R[4]

Dept. of CSE, UVCE, Bangalore, India

[1]mmtha.s@gmail.com, [2]shenoyrakshith96@gmail.com, [3]drthrivenij@gmail.com

Abstract — *Sentiment Classification is one of the fundamental tasks in sentiment analysis that aims to classify the orientation of a given text (e.g., positive or negative). Analysis of sentiment in the text provides an advantage for customers in services and analysis. The text classification in sentiment analysis is performed using Bag-of-words(BOW) model, which is a machine learning approach. Dual Sentiment analysis(DSA) with BiLSTM and CNN is used to address the polarity shift problem that arises in classification. These classifiers perform sequence prediction and provide better results when compared to other methods. Initially, a data expansion technique is proposed that makes use of opposite labels of positive and negative sentiment for each training and test review sentence. Next, in dual training, the probabilities of original and reverse reviews are trained on the classifier. Predictions in dual prediction are done by considering two sides of one review. As the work is carried over on text reviews, the lexicon-based dictionary is used. The proposed model is evaluated on four multi-domain datasets. As compared to SVM and other classifiers, our methods give better results.*

Keywords - *Bag-of-words, BiLSTM, Dual Sentiment Analysis, Machine Learning, Neural Networks, Sequence Prediction*

I. INTRODUCTION

Sentiment analysis is a way of extracting subjective information on text using natural language processing techniques. This gives knowledge of whether a person is talking positively or negatively about some topic. Different datasets are used for sentiment analysis, such as Twitter data, Product reviews, Restaurant reviews, Stock market. The process of sentiment analysis is a complex task with various steps:

1) Data gathering: Collection of data from various sources, which is either structured or unstructured. NLP is used to perform classification on disorganized data.
2) Text-data construction: Irrelevant data during the analysis process, if identified, is eliminated, and then cleaning is performed.
3) Emotion Recognition: Data is further examined to identify the sentiment in it. It is done by finding the emotional information in the sentences.
4) Sentiment Categorization/Organization: Once the emotions in the sentences are identified, they are classified into positive, negative, and neutral.
5) Result Demonstration: After the classification completes, the output is represented in the form of graphs like a bar graph or pie chart. Data can be analyzed on frequency, time, accuracy, etc.

As the emergence of the web has increased the way people communicate, the importance of word-of-mouth understanding has reduced. This is also a time-efficient way of communication for the customers. The attitudes and reactions can be found easily through opinion analysis.

Bag-of-words(BOW) model, a traditional method, is applied to perform sentiment classification of text. The text in the sentences or document is divided into a set of words/fixed-length vectors. This model is simple to understand and provides flexibility for customization on specific text data. BOW method is used in most machine learning algorithms for feature extraction and information retrieval. But, the problem of the BOW model is it does not provide leverage co-occurrence statistics between words, and it breaks the order of words in the sentence, and some semantic information is removed or is lost. Along with this polarity, the shift is a difficulty in BOW [1] [2] [3].

Polarity shift occurs when the polarity expressed by the words in the sentence is dissimilar/ disparate to the polarity of the sentence. The polarity can be reversed by performing negation on text in the sentence. The word orientation in a sentence can be changed from positive to negative and vice-versa. For example, In the sentence ”The chair is not comfortable”, ’comfortable’ is a positive word but the polarity of the sentence is negative because of the word ’not’, which is a negation word. Thus, BOW fails under such circumstances where 2 sentiments opposite texts are regarded as similar.

Machine learning and Deep learning performs extremely well in data prediction and classification of text for large data length and solving polarity shift problem. CNN is a neural network model used to solve the problems in opinion mining. It is a multilayer model composed of input, hidden layers, and output layers. It is used to extract the features which are combined into larger networks [4]. Semantic and syntactic features are captured by the

convolution filters to carry out sentiment classification tasks.

There are various approaches to address the problem of polarity shifting [5] [6] [7]. Most of them require extra linguistic knowledge and human interpretations. Polarity shift problems are also addressed without extra annotations and semantic knowledge [8] [9]. For document-level sentiment classification, negation alone is not sufficient to deal with the polarity shifting problem.

A. Motivation

Xia et al. [10] proposed dual sentiment analysis to perform sentiment classification on a supervised method. This requires a large amount of labeled data. Labeled data is more expensive than unlabelled information and is time-consuming. Different classifiers are trained separately on each of the domains. As the length of the review sentence varies, the accuracy also fluctuates.

B. Contribution

In this paper, we propose dual sentiment analysis with Bidirectional long short-term memory(BiLSTM) on supervised data to do classification on different domains. Deep learning methods are proposed to obtain accurate results through sequential data. The original data is fed once from the beginning to end and vice-versa. By doing this, the information is preserved in both the beginning and end. Bidirectional LSTM performs sequence prediction to obtain better classification accuracy.

Convolution Neural Networks(CNN) is another deep learning technique used for feature extraction. It is done in the convolution layer. As there is a local correlation between the neurons of close-by layers, the classification performed is better. As each word is assigned a weight in the hidden layer, an exact match is found, and it does it repeatedly. By this task, the loss reduces, and accuracy increases.

The rest of this paper is organized as follows. In Section II, related work is presented. In Section III, background work is explained. In Section IV, the proposed method is explained. Implementation details are given in section V. In Section VI, and performance analysis is presented. Section VII concludes the paper.

II. RELATED WORK

In this segment, we will discuss some earlier works on polarity shift, dual sentiment analysis, sentiment analysis, and deep learning used for classification.

1) *Sentiment Analysis:* Sentiment analysis is a collection of tools and techniques to extract and identify semantic information at the sentence, phrase, or document level. The main goal is to find the attitude, whether positive or negative, using NLP approaches [11] [12]. Sentiment analysis is used as a means to improve the quality of products in industries, analyze public sentiments, tweets, movie and restaurant reviews [13]. [6] Wilson et al. suggested ideas to find the difference between prior and contextual polarity phrases. For this, a combination of machine learning and various other features is applied. Initially, each phrase is classified as polar/neutral. Next, all the phrases polarity from the first step is noted down(positive, negative, or neutral). Along with this new notation, contextual polarity is described. Choi and Cardie et al. [14] proposed a compositional-semantic approach to detect opinions in review instead of the BOW method. This is an efficient approach when compared to the others. This work finds the opinion of the whole sentence instead of finding the sentiment of each word at a time.

2) *Polarity Shift:* To overcome the polarity shift problem that arises from the difficulties of BOW used for classification, dual training, and the dual prediction model is applied. It generates the opposite polarity of the original text. For example, ”I don’t like this movie” reversed polarity is ”I like this movie”. Next, both original and reversed text is processed through dual training and prediction [15] [16] [17].

To perform classification on sentence-level and document level, two different techniques are used: lexicon-based and corpus-based methods. The dictionary-based approach includes computing the orientation for a document from the overall semantic orientation scores of words or phrases in the text collected from the dataset [18] [19]. This approach uses manually created dictionaries. Adjectives are used as indicators for polarity representation of terms in the text.

Positive word: great, awesome, pretty.

Negative words: ugly, worst, disappoint.

Corpus-based/machine learning method uses classifiers built from the structured text in sentences, where the text is a set of words. Then, supervised machine learning algorithms are applied to perform classification [8].

Nowadays, deep learning comes up as a machine learning technique used for text classification on a range of resources. Aspect extraction is a subtask of sentiment analysis. Deep learning approaches and neural networks are used for extracting the aspects in data mining. Adyan et al. [20] suggested applying a deep feed-forward neural network to handle a large amount of unstructured data. In [21], Lakkaraju et al. carried out work on both aspect category identification and opinion classification by applying recurrent neural networks. Sequence prediction is followed as the text reviews are too long. To overcome this difficulty, Tang et al. [22] developed 2 target-dependent LSTM models. In this, the target vector is considered natural, and the text is evaluated. By this, the sentiment classification accuracy raises significantly/remarkably. In [23], Dong et al. proposed an adaptive recursive neural network(AdaRNN) on Twitter data to generate sentiment words and put them as target vectors based on syntactic relation between them.

Wang et al. [24] proposed a hybrid model using convolution neural networks and recurrent neural networks to perform text classification and obtained comparably good results. The extracted feature from CNN is passed as input to RNN. RNN performs sequential prediction to obtain the long-term dependencies.

[25] represented the kernel in convolution layers of various sizes to obtain the feature vector in the data of textual form. The results obtained on sentiment analysis are outstanding by tuning the hyperparameters and keeping the vectors static.

The objective of the proposed work is to set up a BiLSTM and Convolutional neural networks classifier for dual sentiment analysis. These are deep learning techniques to increase performance accuracy at the classification stage. This considers sequence prediction on reviews to obtain accurate classification results.

III. Background Work

In the previous work, the Dual Sentiment Analysis technique was applied to solve the polarity shift problem caused by Bag of Words (BOW). This system was run on different classifiers such as linear SVM, logistic regression, and naive Bayes for multi-domain datasets, and the accuracy was noted. This problem can be addressed using the word embedding technique and by reversing the sentiment in the reviews. The data expansion technique is used as similar to the previous work, and the classifier has been trained with word embedding for different classifiers. Then, a pair of original and reverse reviews is learned, and classification is performed in dual training.

By applying the long short term memory(LSTM), which is a recurrent neural network(RNN) model, the vanishing gradient problem is addressed. It learns the relevant sentence and forgets non-relevant ones based on the training model. This gradient is stored in memory cells using gates. Gates value is computed from the previous value and current value, and at each input, the state gate can erase, read and write into the memory cell.

The methodology contains 2 major phases: Data Expansion technique and Dual sentiment analysis. In the Data expansion technique, each original review creates a reverse review and later is paired to train the classifier. Dual sentiment analysis phase, to predict each test sample say x, we use the reverse sample say $\bar{x}$. This technique is called Dual Prediction.

The process of word embedding in natural language processing is where textual words or phrases in a sentence are mapped to vectors/real numbers. i.e., the text is converted into numbers. The strategy typically includes a mathematic implanting from a high-dimensional scanty vector space to a lower-dimensional thick vector space. It moves from a sparse representation to a dense representation. It is used to learn the sentiment polarities in memory networks. Representing words to understand the aspect word meaning is important. The pipeline of word embedding is shown in Fig. 1.

Mikolov et al. [26] [27] suggested the word2vec model perform word embedding in NLP tasks. It is an efficient prediction model to learn embed words from text data. This neural network model contains a Continuous Bag of Words and Skip-gram model.

Syeda Rida-E-Fatima et al. [28] proposed work on choosing the pertained word vector model that impacts the accuracy of the sentiment. To identify the resultant association between the feature and opinion word, a deep learning-based multilayer dual-attention model is applied. In our work, we have improved on the pertained word embedding to capture the better sentiment.

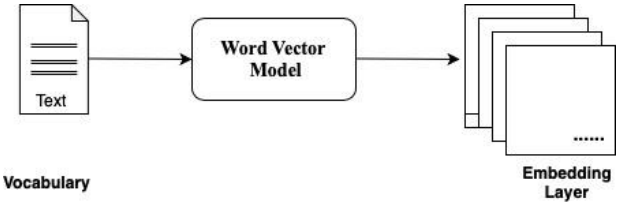

Fig. 1. Word Embedding Model for Sentiment Analysis

A. Data Expansion Technique

In this section, the data expansion technique is introduced by creating sentiment-reversed reviews based on the original reviews. On the basis of the dictionary, with each initial review, a set of reversed reviews shall be drawn up in compliance based on the Text Reversion technique. For a negative review, the scope of negation is identified and converted to its antonym.

The initial review mark is flipped to get the reversed review, and part of the speech has also been tagged. Thus the reversed analysis may not be grammatically right or as strong as a human-generated one, but caution should be taken to ensure that it maintains the strength of the opposite opinion of the original review. Assigning the smaller weight to the model would protect the model being incorporating the low-quality analysis.

A two-step rule for data expansion based on the antonym dictionary is performed. They are:

- Text Reversion: Sentiment words that are not in the scope of negation are reversed with their antonyms. But, negation terms such as don't, no, not are removed.
- Label Reversion: The name of the original review sentence is reversed and updated to its opposite label.

It is shown with an example,

"I don't like oranges. It tastes worst."

Class Label: **Negative**

Initially, text reversion is performed. In the first part of the sentence negation word "don't" is eliminated, and the sentence becomes "I like oranges". As the word "like" is in the scope of negation, it is retained back. In the second bit, the sentiment word "worst" is reversed with the opposite word "good". Next, in label reversion, the original class name of the sentence was negative. It is now reversed to positive polarity.

Sentiment degree metric is proposed for selecting unique training reviews to perform data expansion. The measure of sentiment polarity is estimated using

$$m(z) = |p(+|\vec{z}) - p(-|\vec{z})| \quad (1)$$

Where $p(+|z)$ and $p(+|\vec{z})$ are posterior probabilities and z is the training data review. A good result is attained by choosing a set of training reviews instead of the complete data.

B. Dual Sentiment Analysis

In this section, the DSA framework is presented in detail.

- Dual Training Phase: In this stage, the original training set is reversed to its opposite set using the data expansion technique. Data has a one on one correspondence between

original and reverse reviews. The original data consists of " the original training set," and reversed data consists of "reverse training set". First, the scope of negation is checked in the sentence, and sentiment words are retained back. Negation words such as not, don't are reversed. Then, the complete class label of the original sentence is reversed. This method adapts easily to other NLP classifiers.

- Dual Prediction Phase: It is the second step in dual sentiment analysis executed after dual training. Two-component predictions as the weighted combination are utilized as the output in this stage. It solves the problem that arises from the BOW as it misclassifies the original text polarity irrespective of the negation word. The review sentence," I don't like chocolates," should have a negative class polarity. But, the traditional BOW model miscategorizes the polarity as positive even in the presence of the negation word "don't". This is because the word "like" has a high positive score in BOW. This problem is solved by DSA in the current stage, where it decides the sentence polarity of a review by calculating the reversed review score. Hence, classification accuracy increases, and prediction errors are reduced.

z And $\bar{z}$ are the original and reverse test review samples. The posterior probability of z and $\bar{z}$ are $p(*|z)$ and $p(*|\bar{z})$ '*' denotes positive or negative. For each test sample, a reversed test sample is created in the dual prediction phase.

Dual prediction of a review is obtained using two components:

1) To measure how positive the test review z is, we consider $p(+|z)$ the original test review and also consider $p(\text{-}|\bar{z})$. i.e., how negative is the reversed test review?
2) To measure how negative the test review z is, we consider $p(\text{-}|z)$ and also $p(+|\bar{z})$ is considered, i.e., probability of how positive is the reversed test review.

For dual prediction score function, both positive and negative predictions are used as

$$p(+|z,\bar{z}) = (1-\alpha).p(+|z) + \alpha.p(\text{-}|\bar{z}) . \quad (2)$$

$$p(\text{-}|z,\bar{z}) = (1-\alpha).p(\text{-}|z) + \alpha.p(+|\bar{z}) \quad (3)$$

Where α is a trade-off parameter that lies between ($0 \leq \alpha \leq 1$)? The highest performance result is obtained when α it is in the range [0.5-0.7].

Let us consider an example to see how dual prediction addresses the polarity shift problem.

- Original review: I don't like to watch movies. It is boring.
- Reversed review: I like to watch movies. It is fascinating.

By applying the BOW method to the original sentence, "like" as it is positive contributes to a high positive score. The overall polarity of the sentence is misclassified as positive in spite of the negation word "don't". In DSA, the dual prediction classifies the polarity of the sentence based on the score of the reversed review. The output of DP is a weighted combination of two-component predictions. The wrong prediction of the original sample is recompensated by predicting the reversed review. Thus, the prediction errors caused by the polarity shift are possibly reduced.

IV. Proposed Method

In Fig. 2, the system model is explained. The original and reverse train data is passed to the system. Before word embedding, each word is mapped to a real-valued vector. Words that are similar or nearest in the vector space to the target word are identified. Next, feature extraction is done. Few features will be reduced so that noisy data is eliminated. Then, dual training is carried over on the extracted data. Using the data expansion technique, the original data is reversed to its opposite, and then the BiLSTM or CNN classifier does the classification using the feature vectors. By combining the original and reverse test review, a weighted score is obtained with high probability, and the prediction is done. By doing this, prediction errors are reduced in polarity shift. Finally, the sentiment polarity is obtained.

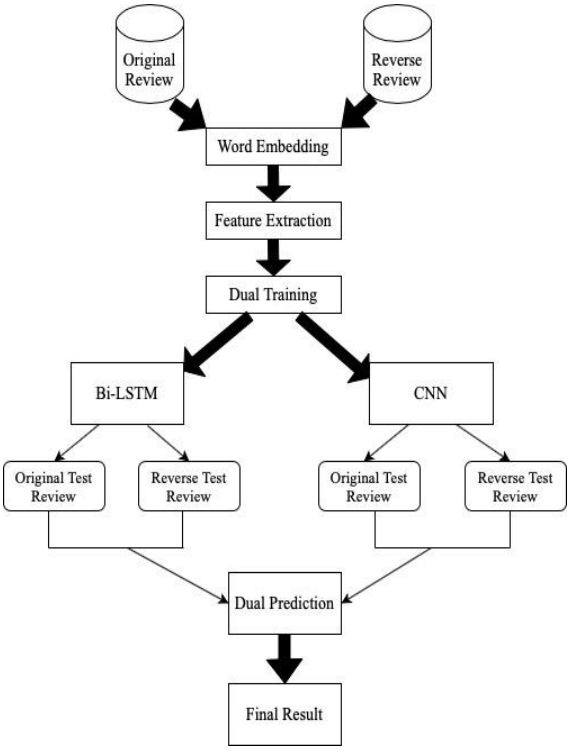

Fig. 2. System Design

In RNN architectures, all the layers are independent. Vanishing gradients was a major problem in this method. The network includes unfolding the network layers for each input time stamp, basically making a deep network that requires weight updates. To overcome this problem, BiLSTM and CNN are applied, which are highly efficient methods as it uses both forward and backward data.

C. Bidirectional Long Short Term Memory(BiLSTM)

The recurrent neural network is unable to capture long-distance semantic dependencies. This is called the vanishing gradient problem, where RNN loses the data when it travels across many layers. Because of this, the dependencies are not identified properly. To overcome this problem, BiLSTM, which is an extension of RNN, is used. In the recurrent neural network model, the knowledge can only be distributed in the forward direction at every time step. BiLSTM combines two recurrent neural network models to capture the context information in both directions. This maintains the sequential order between the data in a sentence. It stores the past and the future data, which is in the form of input and output at any point. Bidirectional long short term memory architecture is presented in Fig. 3.

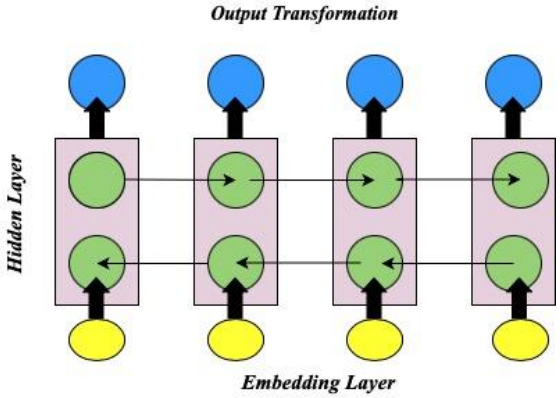

Fig. 2. BiLSTM Architecture

In the proposed method, BiLSTM extracts hidden lexical semantics of words in the vocabulary. It provides long term dependent sequence information of the target sentence. The memory cells are able to remember this historical information with a back and forth mechanism. The one-layer process from left to right and the other from right to left.

At time t, the hidden layer unit function is $\vec{h}$ for the sentence $x = \{x_1, x_2, \ldots\ldots, x_n\}$ is determined on the previous hidden state at time t-1. The input at the current state is computed based on the hidden unit function $\overleftarrow{h}$ and hidden unit $\overleftarrow{h_{t-1}}$ state. The final output of hidden states is generated by concatenating the $\overleftarrow{h}$ *and* $\vec{h}$.

The hidden semantics vector for the aspect is calculated as below for hidden state:

$$\vec{h} = LSTM(x_1, x_2, x_3, \ldots\ldots\ldots, x_n) \quad (4)$$

$$\overleftarrow{h} = LSTM(x_1, x_2, x_3, \ldots\ldots\ldots, x) \quad (5)$$

$$h = \vec{h} * \overleftarrow{h}. \quad (6)$$

$$h_1^t = \sigma[(W_{ij})_t, h_1^{t-1}] \quad (7)$$

Each time cell memory is calculated as in Eq. 7.

The Gated Recurrent Unit combines forget and input gates into a single unit called Update gate. This merges the cell state and hidden state.

- The forget gate holds information on what to forget in the cell state when new information enters.
- The input gate encodes the new information that enters the cell state and controls them.

$$z_t = \sigma\left[W^z\left(w_{ij}, h_1^{t-1}\right)\right] \quad (8)$$

$$r_t = \sigma\left[W^r\left(w_{ij}, h_1^{t-1}\right)\right] \quad (9)$$

$$\widetilde{h_t} = \tanh\left[W\left(r_t, w_{ij}\right)\right] \quad (10)$$

$$h_t = (1 - z_t) * \widetilde{h_{t-1}} + z_t * h_t \quad (11)$$

where z_t and r_t are the forgot gate and input gate and h_t is a candidate hidden layer. 'W^z' and 'W^r' is the weight of the state and h_t is the final state.

a) Pooling: Neural pooling functions take the BiLSTM output as a subsampling layer. Max and average functions are used for pooling to capture the highest value and average value of each dimension. This technique minimizes the computational complexity arising from upper layers.

$$h^2 = \left[\begin{bmatrix}\max(h_{i1}^t)\\ \ldots\ldots\\ \max(h_{1n}^t)\end{bmatrix}, \begin{bmatrix}\text{avg}(h_{i1}^t)\\ \ldots\ldots\\ \text{avg}(h_{1n}^t)\end{bmatrix}\right] \quad (12)$$

where h^2 is the output of the layer?

b) Softmax: The softmax layer predicts the resultant sentiment of the review. The output from the pooling layer is fed as input to this layer.

$$y' = softmax[w(h^2) + bias] \quad (13)$$

Where w is the weight of the layer and y' is the predicted label.

D. Convolutional Neural Networks

CNN is a deep learning feed-forward artificial neural network technique composed of artificial neurons. It takes the text/image as input and performs pre-processing and feature extraction to classify the polarity of text. To do this, it passes through a number of filters of different sizes and on a number of layers. In-text classification, CNN is applied to word embedding.

Text classification using CNN is similar to image classification. The difference is that text data is converted

into word vectors. Each neuron in convolutional neural networks takes input. Performs weighted sum and passes it on using activation function and finally the output. Loss is calculated in the softmax layer, which is the output layer of CNN.

Initially, word embedding is performed that converts words into embedding vectors. It is a lookup table obtained from the data. For this, word2vec is used that achieve the vector representation of words in the dataset. It uses a neural network model. Next, the vectors are passed on to the convolution layer to perform several operations to reduce the sentence into the low-dimensional matrix. Further, the filters start reducing the matrix to obtain a group of features. This is then reduced and passed as output to the pooling layer for downsampling. It is a data reduction process to learn the higher-order representation of data. 0The activation function is used to obtain a non-linear relationship on the output. For this, Rectified Linear Unit(ReLU), Sigmoid or tanh function is used.

The steps performed are:

- **Data Exploration:**

The dataset is first to read, and if unstructured, it is converted into labeled data. This data is read using pandas, a data analysis and manipulation tool in python. It is a major step carried over in data mining to refine the dataset. As the data in our work is clean, we just load it into numpy, which works efficiently with arrays without applying any cleaning technique. The text and label fields are built into vectors which makes the execution easier.

- **Tokenization:**

It is performed to prepare the dataset before it is passed as input to the network. As neural networks understand numerical, the data/ words are converted into unique integer values. These are called tokens. Based on the word occurrence, these tokens are assigned. The word that has the highest occurrence in the complete dataset is assigned the token ’0’. The association of words and tokens is learned from the nltk import word tokenize function. Here, the token mapping to the word is shown. The value of the token always starts from (0-100,00). Stopwords are also added up to the dataset. After it is done with all the words, tokenization is performed on the complete dataset. Example : {”the” - 0, ”is” - 1, ”then” - 2, ”of” - 3, ”which” - 4}.

- **Data Annotation**:

It is a process of modeling the dataset for the training of machine learning models. Once the text is tagged and then trained, the system will easily learn to recognize the patterns of the annotated data. Annotations can be in the form of keywords or phrases, tagging of text, highlighting the words in a document, translation from one language to another, parts-of-speech, and identification of relation. There are different types of text annotations, sentiment, linguistic, semantic, and entity annotation.

All the words and phrases are annotated to make them understandable. In our work, Natural language processing is used to perform the data annotation task. It is done for the machine learning models to understand the sentiment within the text in a sentence. We are not creating a separate hardcode annotate library.

- **Vocabulary Size:**

We need to know the size of the vocabulary for the embedding layer. Vocabulary is built to train the sentence length and the word training. This is done after the review sentence length is fixed. It is done using

SentenceLength = (len(tokens)for Tokens in data[”tokens’])
Sort(listset(training words))

The input to this step is got from the training data after performing tokenization.

- **Word Embedding:**

It is a process of representing the text in the data in the form of a vector. The words with similar semantic meanings will have the same real-valued vector representation. As the system cannot understand the text to process it, the data/text in the vocabulary is converted into vector or numerical form, which makes the task easier. Embedding is created using input, output, and hidden layers.

Word embedding in our work is carried over on the word2vec model, which is a statistical method of modeling. It builds a vocabulary from the training data and then gets the vector form of words. The Skip-gram model, which is a variant of word2vec, is used as it gives accurate results. The complete context is predicted using a target word. The input layer takes the vector value, and the output is a softmax layer that gives the prediction probability of each word. Hidden layer holds a set of nodes equal to the embedding size.

- **Padding:**

After preprocessing, the length of the sentences in the dataset changes. The input provided to the classifier should be of the same size and shape in the neural network. Padding is done to reshape the size of these sentences in both train and test data which gives accurate outcomes. We set it to the maximum number of words in the sentence using the max sequence length() function. If a sentence is longer than the specified length, the sentence is truncated by dropping the words from the last or end of the sentence rather than the beginning.

- **Embedding:**

Usually, sentiments with identical labels will have the same meaning and vector. Embedding is a process of retrieving the sequence number for each word with their sentiment in the form of a vector. Once the embedding layer provides the feature vectors, the convolution layer is used. The embedding matrix dimension is fixed to 300. The information contained is large if the embedding size is higher. It works as a sliding window.

5 different filter sizes are applied in the model in the hidden layer. They are [2, 3, 4, 5, 6]. The layers in this model are:

1. MaxPooling 1D for each layer
2. Dropout layer
3. Dense layer
4. Dropout layer
5. Dense layer

Max pooling is used to perform downsampling; it selects the largest value in the feature map in every step to form an output vector. This output is a feature map with distinguished elements from the previous map. This is done to reduce the number of parameters. The dropout layer is used to reduce the complexity by preventing overfitting problems. It is set to 0.1. The outputs from all the layers are finally concatenated. In the dense layer, each neuron or input node is connected to each output node.

As our dataset is multi-domain text reviews, to perform classifications, the feature matrix is 1-dimensional. Hence, Conv1D is used. The filter in conv1D is 200, and the activation function used is 'ReLU'. The dropout rate is set to 0.5. The validation split is 0.1.

a) Conv1D: Conv1D means the kernel moves in 1 direction only. It is one cross relational operation where the window moves from the leftmost input array and slides down to the rightmost array. It is called the sliding technique.

Example: As shown in Fig. 4, the width of the one-dimensional input array is 6, and the kernel array is 2. The output array width is calculated using:

$$6\text{-}2\text{+}1 = 5 \text{ (Input)}$$

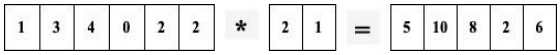

Fig. 4. Conv1D cross-relational operation

The first element in the output array is obtained by multiplying the leftmost element in the input array with the kernel element and then adding them up together. It is shown below:

$$1*2 + 3*1 = 5$$

$$3*2 + 4*1 = 10$$

The same way it is performed for multi-input cross-correlation operations.

V. Implementation

In this section, we evaluate our approach of dual sentiment analysis performance on polarity classification using the BiLSTM and CNN classifier on four English datasets that are from different domains.

A. Dataset

The dataset for this work is reviews taken from Amazon.com of multi-domains: Kitchen, Book, DVD, and Electronics. Each domain in the dataset has 2000 review sentences:1000 positive and 1000 negative. These reviews are the customers rating from ⋆1 to ⋆5. ⋆1 and ⋆2 reviews are labeled negative, and ⋆4 and ⋆5 are positive polarity sentences. Table I shows the details of all the datasets used to perform DSA.

TABLE I
DATASET SPECIFICATION

Amazon Review	Positive	Negative	Train Data	Test Data
Books	500	500	1000	200
Kitchen	500	500	1000	200
Electronics	500	500	1000	200
DVD	500	500	1000	200

The aim of this work is to evaluate the DSA model under various settings. The reviews in the dataset are divided into 5-parts. One part is used for testing, and four parts are used for training the classifier. The dimension of the embedding is 300. Adam optimization method is used to train the network through time and learning rate of 0.01.

B. Data Expansion

Implementation of dual sentiment analysis using BiLSTM and CNN is carried over using labeled data for both training and testing purposes. The reviews from Table I are divided on 80:20 ratio for train and test data. Initially, the original labeled data is converted to reverse labeled data. Next, the classifier is trained using both these data. Thus, the system becomes stable as it has an original and reversed review.

The lexicon-based antonym dictionary is got from the WordNet database. It has a group of English word synonyms called synsets. A specific synset of words is identified based on POS tags present in WordNet. The words are grouped based on their semantic relations, and their opposites are obtained easily. This is a simple dictionary that is readily available in readable form in English language and many more. The problem of lexicon dictionary is domain consistency that is solved by developing corpus-based dictionary.

Based on the dataset available if we consider total 4000 original training reviews, they are all reversed to their opposite semantic relation. Thus, there is 8000 total training data including both actual and reverse review sentences. These are then used for training.

In the training process the reviews are grouped into reversed review based on their opposites. After this the total data is passed on to each of the classifiers separately which executes the data in a sequential manner by storing it in both input and output layer.

C. Construction of Word Embedding

Word embeddings exhibit the meaning of the word based on the text occurrence in the corpus. The word vectors will represent the meaning of the word in case the text corpus is very huge. This will be helpful in various applications. Word2vec and GloVe are the word embedding models for learning word vectors from the corpora. The word embedding for BiLSTM is demonstrated using GloVe model as it takes a count of the word to word co-occurance globally. For CNN, word2vec

is applied. Fig. 5 illustrates the mapping of sentiment to word embeddings with examples.

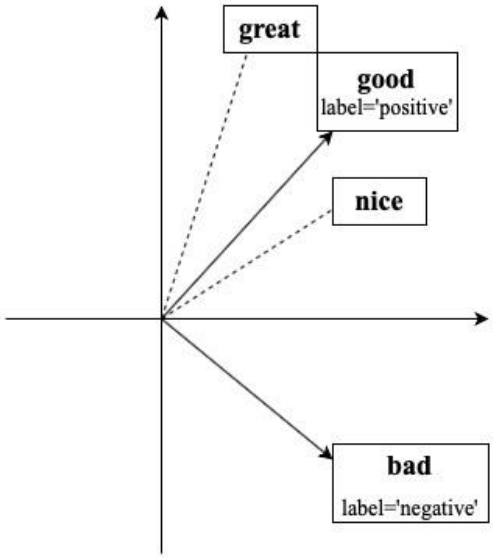

Fig. 5. Sample sentiment text to word embedding mapping

Let T_j is a combination of tokens, that is $T_j = \{t_{i1}, t_{i2}, t_{i3}, \dots\dots, t_{in}\}$ be a training goal. Key goal is to maintain correlation and extract aspect $A_j = \{a_{i1}, a_{i2}, a_{i3}, \dots\dots, a_{in}\}$ and opinion $O_j = \{o_{i1}, o_{i2}, o_{i3}, \dots\dots, o_{in}\}$.

For a given vocabulary $S = \{s_1, s_2, s_3, \dots\dots, s_n\}$ and aspect $A_j = \{a_{i1}, a_{i2}, a_{i3}, \dots\dots, a_{in}\}$, map each word into vector form ad convert into 300-dimensional word vector of size N. The embedding layer now contains two-dimensional matrix of word vector $W = \{w_1, w_2, w_3, \dots\dots, w_n\}$.

1) CNN Model for Text: It defines the convolution dimension and kernel for inputs. The max-overtime pooling layer is used that corresponds to max-pooling layer which takes various time steps as input on each channel. The text on CNN is divided into:

- Multiple 1-dimensional convolution kernels are used to perform input calculations.
- Max-overtime pooling is done on all output channels. Then, the pooled output values are concatenated into a vector.
- The vectors those are concatenated is modified into output for each of the category. It is done through the fully-connected layer.
- In the above step, dropout layer is used to prevent overfitting.

In Fig. 6 the input sentence is of 11 words and hence the sequence is of width 11 with channel input 6. Each word is represented as a 6-dimensional word vector form. The kernel width of the two 1-dimensional vector is 2 and 4 and channel output is 4 and 5. The width of one of the output channel is $11 - 2 + 1 = 10$. The width of the channel with channel output 5 is $11 - 4 + 1 = 8$. Next, max-overtime pooling is performed on each channel and the pooling outputs are concatenated into 9-dimensional vector. This is modified into 2-dimensional final output using fully connected layer. This shows the prediction whether positive or negative sentiment.

For a learning algorithm number of epochs and batch-size must be specified. These are integer values. Epoch is a hyper parameter that specifies the number of times the algorithm runs on the dataset and learns. Batch size is the amount of data processed at a single time before the model is updated to calculate the gradient and update the weights. Its size should be more than or equal to one and it is lesser than or equal to number of samples in the dataset.

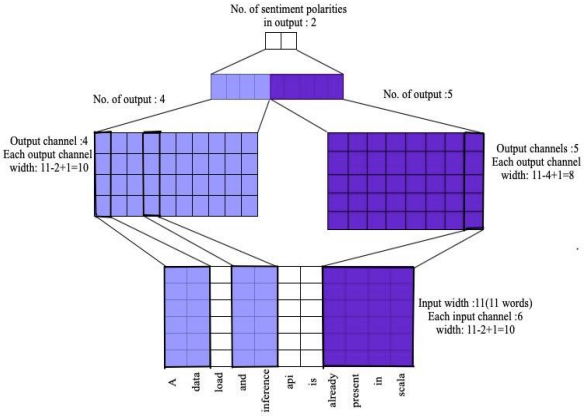

Fig. 6. Text Processing Design using CNN

VI. Performance Analysis

In this section, the result analysis of BiLSTM and CNN are discussed. The results of deep learning model is compared with machine learning methods, namely SVM, naive bayes and logistic regression. The main task of analysis is to classify the sentiment in the sentences. BiLSTM is a better model to identify long-term dependencies and performs sequence predictions. In CNN every layer other than output layer uses ReLU as activation function.

In Table II the accuracy obtained from BiLSTM and CNN on the kitchen data is shown. It is observed that CNN performs better than BiLSTM at epoch 5 and drops down between 0.82 to 0.85 in the next epochs. The best accuracy recorded is 0.916 at epoch 50. The classification is accurate as pooling operation in the pooling layer reduces the dimensionality of the extracted feature vectors. Downsampling is performed on the feature maps by reducing the number of hidden units in hidden layer.

TABLE II
ACCURACY COMPARISON ON KITCHEN DOMAIN FOR STEP LENGTH=15

Epoches	5	10	15	20	25	50
DSA-BiLSTM	0.887	0.86	0.875	0.89	0.89	0.895
CNN	0.91	0.824	0.89	0.85	0.857	0.916

It is observed in Table III that the accuracy for book domain on BiLSTM model lies between 0.825 to 0.84 and the highest accuracy of 0.84 is achieved at epoch 50. But, the result of CNN is not stable and it reaches an accuracy of 0.87 at epoch 15 and 50. This is achieved because CNN performs feature extraction.

TABLE III
ACCURACY COMPARISON ON BOOK DOMAIN FOR STEP LENGTH=15

Epoches	5	10	15	20	25	50
DSA-BiLSTM	0.835	0.84	0.83	0.825	0.832	0.84
CNN	0.78	0.824	0.87	0.84	0.86	0.87

The accuracy on DVD and electronics domain is shown in Table IV and Table V. The result on BiLSTM is increasing consistently from 0.86 to 0.88 but in CNN the results are not stable. It fluctuates from 0.77 at epoch 5 to 0.83 at epoch 15 and increases to 0.87 at epoch 20 and again at epoch 50. This result is achieved because CNN performs feature extraction technique which increases the accuracy.

TABLE IV
ACCURACY COMPARISON ON DVD DOMAIN FOR STEP LENGTH=15

Epoches	5	10	15	20	25	50
DSA-BiLSTM	0.87	0.86	0.873	0.873	0.87	0.88
CNN	0.775	0.824	0.835	0.87	0.85	0.87

TABLE V
ACCURACY COMPARISON ON ELECTRONICS DOMAIN FOR STEP LENGTH=15

Epoches	5	10	15	20	25	50
DSA-BiLSTM	0.87	0.864	0.863	0.87	0.87	0.875
CNN	0.72	0.81	0.835	0.857	0.86	0.883

Table VI shows the classification results on different domains. The deep learning techniques are compared with traditional methods. As compared to the NLP techniques CNN performs better. The polarity classification is higher than BiLSTM in the proposed technique.

TABLE VI
ACCURACY COMPARISON OF POLARITY CLASSIFICATION USING BILSTM AND CNN

Dataset	SVM	NB	Log Reg	BiLSTM	CNN
Books	0.809	0.837	0.823	0.84	0.87
Kitchen	0.879	0.895	0.886	0.91	0.92
Electronics	0.849	0.859	0.857	0.857	0.887
DVD	0.816	0.84	0.836	0.873	0.87
avg.	0.838	0.857	0.85	0.87	0.89

The loss on BiLSTM and CNN are plotted in Fig. 5. It is calculated in the dropout layer and the dropout rate is set to 0.5. These models use binary cross entropy and 'Adam' optimizer to optimize the loss function. The loss is minimal when the model works better with small values rather than bigger numbers and it is 0.0 if the model predicts accurate probabilities. Loss in CNN is less than BiLSTM as it regularly checks for overfitting in the dropout layer.

The cross entropy loss is calculated using

$$BCE = -r_1 \log\big(f(t_1)\big) - (1 - t_1)\log\big(1 - f(t_1)\big)$$

$$= \sum_{i=1}^{C'=2} r_i \log\big(f(t_i)\big) \quad (14)$$

where C_1 and C_2 are the two classes. It is set to $C' = 2$, as it is binary classification problem. For C_1, $r_1[0,1]$ and t_1 are ground truth and score. For C_2, $r_2 = (1 - t_1)$ and $t_2 = (1 - f(t_1))$ are ground truth and score.

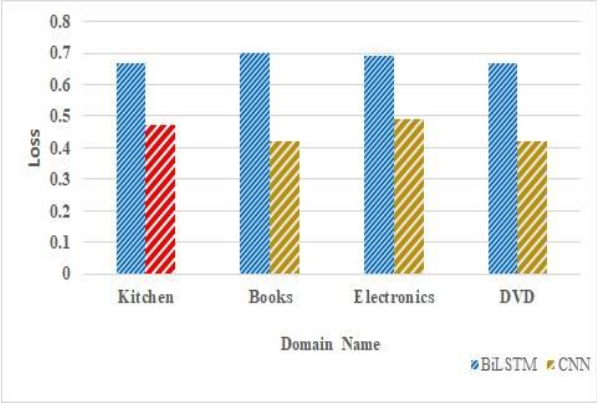

Fig. 7. Loss Rate Comparison

VII. Conclusions

In this paper, polarity shift problem caused by sentiment classification is overcome by using deep learning methods on dual sentiment analysis. Data expansion technique creates reverse reviews, i.e., reviews opposite to original review labels. BiLSTM identifies the long term dependencies in text. CNN has the ability to extract the features. Embedding is performed using GloVe.

The BiLSTM and CNN models are compared on four multi-domain datasets with different variations. Better classification accuracy is obtained by applying neural methods compared to machine learning techniques. CNN achieves higher accuracy compared to naive bayes, SVM, logistic regression and BiLSTM. This is because CNN acts as a feature extraction model. As it first performs feature extraction and then classifies the data, there is improvement in the accuracy obtained.

REFERENCES

[1] S. Zirpe and B. Joglekar, Polarity Shift Detection approaches in Sentiment Analysis: A Survey, International Conference on Inventive Systems and Control(ICISC), (2017) 1–5.

[2] R. Xia, F. Xu, J. Yu, Y. Qi, and E. Cambria, Polarity Shift Detection, Elimination and Ensemble: A Three-stage Model for Document-Level Sentiment Analysis, International Journal on

Information Processing and Management, 52 (2016) 36–45.
[3] A. Jangde and P. Malviya, Opinion Analysis Using Polarity Shift Model, International Journal of Computing and Technology, 10 (2017) 1–309.
[4] Y. Goldberg, Neural Networks method for Natural Language Processing, Synthesis Lectures on Human Language Technologies, 4 (2017) 34–38.
[5] L. Shoushan and C.-R. Huang, Sentiment Classification Considering Negation and Contrast Transition, 23rd Pacific Asia Conference on Language, Information and Computation, (2010) 297–306.
[6] T. Wilson, J. Wiebe, and P. Hoffmann, Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis, Association for Computation Linguistics, 35 (2009) 399–433.
[7] L. Shoushan, S. Y. Mei Lee, Y. Chen, C.-R. Huang, and G. Zhou, Sentiment Classification and Polarity Shifting, Proceedings of the 23rd International Conference on Computational Linguistics, (2010) 635–643.
[8] B. Pang, L. Lee, and S. Vaithyanathan, Thumbs Up? Sentiment Classification using Machine Learning Techniques, Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, (2002) 79–86.
[9] S. Das and M. Chen, Yahoo! for Amazon: Extracting Market Sentiment from Stock Message Boards, Asia Pacific Finance Association Annual Conference(APFA), (2001) 79–86.
[10] R. Xia, F. Xu, C. Zong, Q. Li, Y. Qi, and T. Li, Dual Sentiment Analysis: Considering Two Sides of One Review, IEEE Transactions on Knowledge and Data Engineering, 27 (2015) 2120-2133.
[11] M. V. Mantyla, D. Graziotin, and M. Kuutila, The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers, Computer Science Review, 27 (2018) 16–32.
[12] B. Pang and L. Lee, Opinion Mining and Sentiment Analysis, Foundations and Trends in Information Retrie 2 (2008) 1–135.
[13] R. Prabow and M. Thelwall, Sentiment Analysis: A Combined Approach, International Journal of Informetrics, 3 (2009) 143–157.
[14] Y. Choi and C. Cardie, "Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis," Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP), (2008) 793–801.
[15] R. Xia, T. Wang, X. Hu, S. Li, and C. Zong, Dual Training and Dual Prediction for Polarity Classification, Association for Computation Linguistics, (2013).
[16] H. Deshmukh and P. L. Ramteke, An Overview of Sentiment Analysis Model for Polarity Classification by User Perspective Review, Inter- national Journal of Advanced Research in Computer Engineering and Technology (IJARCET), 6 (2017) 77–79.
[17] P. K. Manna and S. Bodkhe, Three-Stage Sentiment Analysis by Polarity Shift Detection, Elimination and Ensemble, International Journal of Advanced Computational Engineering and Networking, 3 (2015) 58–61.
[18] P. D. Turney, Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, (2002) 417–424.
[19] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, Lexicon- Based Methods for Sentiment Analysis, Association for Computational Linguistics, 37 (2011) 267–307.
[20] A. M. Ramadhani and H. S. Goo, Twitter Sentiment Analysis using Deep Learning Methods, International Annual Engineering Seminar, (2017) 1–4.
[21] H. Lakkaraju, R. Socher, and M. Chris, Aspect Specific Sentiment Analysis using Hierarchical Deep Learning, (2014) 1–9.
[22] D. Tang, B. Qin, X. Feng, and T. Liu, Effective LSTMs for Target-Dependent Sentiment Classification, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, (2016) 3298–3307.
[23] D. Li, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, (2014) 49–54.
[24] X. Wang, W. Jiang, and Z. Luo, Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, (2016) 242–243.
[25] Y. Kim, Convolutional Neural Networks for Sentence Classification, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014).
[26] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, Efficient Estimation of Word Representations in Vector Space, International Conference on Learning Representations, (2013) 1–12.
[27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, Distributed Representations of Words and Phrases and their Compositionality, Proceedings of the 26th International Conference on Neural Information Processing Systems, 2 (2013) 1–12.
[28] S. Rida-E-Fatima, A. Javed, A. Banjar, A. Irtaza, H. Dawood, H. Da- wood, and A. Alamri, A Multi-Layer Dual Attention Deep Learning Model with Refined Word Embeddings for Aspect-Based Sentiment Analysis, IEEE Access, 7 (2019) 114 795–114 807.