



Semi-supervised domain adaptation and collaborative deep learning for dual sentiment analysis

Savitha Mathapati¹ · Ayesha Nafeesa¹ · R. Tanuja¹ · S. H. Manjula¹ · K. R. Venugopal¹

© Springer Nature Switzerland AG 2019

Abstract

Sentiment classification is a much needed topic that has grabbed the interest of many researchers. Especially, classification of data from customer reviews on various commercial products has been an important source of research. A model called supervised dual sentiment analysis is used to handle the polarity shift problem that occurs in sentiment classification. Labeling the reviews is a tedious and time consuming process. Even, a classifier trained on one domain may not perform well on the other domain. To overcome these limitations, in this paper we propose semi-supervised domain adaptive dual sentiment analysis that train a domain independent classifier with few labeled data. Reviews are of varying length and hence, classification is more accurate if long term dependency between the words is considered. We propose a collaborative deep learning approach to the dual sentiment analysis. Long short term memory (*LSTM*) recurrent neural network is used to handle sequence prediction to classify the reviews more accurately. *LSTM* takes more time to extract features from the reviews. Convolution neural network is used before *LSTM* layers to extract features resulting in the reduction of training time compared to *LSTM* alone.

Keywords Dual sentiment classifier · Domain adaptation · Convolution neural network · Long short term memory · Sequence prediction · Semi-supervised learning

1 Introduction

In recent years, there has been a large data-growth in the internet in the form of reviews. As per the traditional method, Bag of Words (*BoW*) technique that uses vector approach is employed to explore the sentiment of the review. Though it is a simple method of classification, it is not efficient in classifying the sentiments present in Text. This is because, *BoW* tampers the order of the words and it also splits the structure of the syntax resulting in loss of semantic information. To remove basic deficiency present in *BoW* technique, almost all the efforts made had only slight changes in improvising the accuracy of the classification. The most commonly known difficulty is the problem of the polarity shift.

Polarity shift is the linguistic phenomena by nature. Negation is the main form of the polarity-shift problem. For example, consider a word “*don't*” as a negation word and by adding this negation word in the review “*I like this movie*” become “*I don't like this movie*”. Hence, sentiment of the review is reversed from positive to negative. Many methods have been discussed to highlight the polarity shift problem. Most of them require a complex linguistic knowledge or external human annotations. High-level dependency on external resources makes *BoW* highly inefficient in the practical applications. Effort have been made to point the polarity-shift problem without the usage of external resources, but results are far from the expected satisfactory results.

As a solution to the polarity shift problem in classifying the sentiment, a method or technique called dual

✉ Savitha Mathapati, hiremathsavitha@gmail.com | ¹University Visvesvaraya College of Engineering, Bangalore University, Bengaluru, India.



sentiment analysis has been indulged. In this method, the first step is to convert reviews into its reverse polarity using a data expansion technique. This method converts the given review to its opposite sentiment review in a one to one correspondence. In second step, dual training (DT) is used to train both original review as well as reversed review for a statistical classifier. Finally, dual prediction (DP) is used to test the reviews to not only to check how positive/negative the review is, but also it check how negative/positive is the reversed review. Both original-reverse reviews are considered while both training testing phase. To perform the training and the testing part, we make use of *Naive Bayes Classifier*.

Motivation: Xia et al. [1] proposed Dual Sentiment Analysis concept for supervised sentiment classification. Supervised classification need more labeled data. Labeling data is expensive and time consuming process. Here, classifier is trained for each domain separately. Reviews are of varying length and classification is more accurate if sequence prediction is considered while classifying.

Contribution: In this paper, we propose semi-supervised domain adaptation for dual sentiment classification that need less labeled data and single classifier is trained to classify many domain reviews. Later, this classifier will adapt to a particular domain. Review is a sequential data and hence, to add sequence prediction, collaborative deep learning is proposed for dual sentiment analysis. LSTM does sequence prediction and this results in better accuracy of classification. To speed up the training of the classifier, CNN is used for feature extraction and LSTM is used for classification.

Rest of the paper is organized as follows. Section 2 gives the summary of the previous works. Problem definition is defined in Sect. 3. Concept of dual sentiment analysis is discussed in Sect. 4. In Sect. 5, Domain Adaptive Semi supervised sentiment classification is explained. collaborative deep learning concept is explained in Sect. 6. Section 7 gives the details of the implementation part. Performance of the implementation and its Results are discussed in Sect. 8. Finally Sect. 9 gives the conclusions of the paper.

2 Related work

In this section, we are going to discuss some of the previous works on Sentiment Analysis, polarity shift, dual sentiment analysis technique and Deep Learning techniques for text classification. In recent times, Sentiment Analysis is widely used to extract sentiments from various sources like Tweets, movie or product reviews [2]. Pang et al. [3] has given the complete reviews on the different classification techniques used in Sentiment Analysis. Wilson et al. [4] proposed that there might be some positive

words that can be used to represent the negative polarity and also that there may be usage of negative words for representation of the positive-polarity. So the usage of lexicons is entirely dependent on the particular context. Positive-words and negative-words might be used for neutral-polarity as well. Research tells that just by analysis of lexicons, polarity of words doesn't determine the polarity of the context. Hence, more complex-linguistic methods are used to analyze the polarity of the context along with the consideration of lexicons. Choi and Cardie et al. [5] suggested that instead of following Bag of Words technique to detect the sentiment of the review, method of finding out the relation between the words that is called as compositional-semantic approach is used. Instead of finding the sentiments of the single word, find out the sentiment of the whole data under consideration in a collective manner. This approach is efficient and outperform the other machine learning strategies.

Pang et al. [3] proposed dependency-tree technique to classify the sentiment of English and Japanese reviews using condition based random-fields. Sentences that are subjective contain words that affect the polarity of related words which might also reverse the polarity of the sentiment. Such cases are difficult to track using the Bag-of-Words technique. In this method syntax dependent structure of subjective-sentences is exploited. The polarity of the sentiment of each dependent sub-tree is a sentence that cannot be detected while training the data. The polarity of the complete sentence is computed with consideration of the interaction of the hidden-variables. Sum-product kind of strategy is used to come to conclusion. Experimental results on English and Japanese dataset fetched results that are more efficient than the baseline methods. Hence, this task is considered as sequence labeling problems [6–9].

In recent years, deep Learning concept has been in use to work on text related applications, e.g. text classification and information extraction. Socher et al. [10] implemented Recursive Neural Network for sentiment classification. Compositional Computationis is used to represent sentences into vector representation. Kim [11] worked on convolution neural network for sentiment classification and proved that it is better than Recursive Neural Network. Le and Mikolov [12] proposed unsupervised learning method with embedding that classifies documents or sentences into positive or negative.

Aspect level sentiment classification is also a very important part of classification. Research is done on aspect level sentiment classification using neural networks. Dong et al. [13] proposed a variation of recurrent neural network for aspect level sentiment classification. In [14], Lakkaraju implemented both sentiment classification and aspect detection using recurrent neural network. Reviews are

long and hence sequence prediction plays a vital role. Many studies are done on sentiment classification using LSTM. Tang et al. [15] proposed LSTM model using Target vector. Target vector is created by taking the average of target word vector. This target vector is given as input at each iteration.

3 Problem definition

The objective of the proposed work is to build a semi-supervised domain adaptive classifier for dual sentiment analysis. Collaborative deep learning techniques are also applied for Dual Sentiment Analysis to improve the accuracy of the classification. Domain Adaptive word expansion technique is applied to adapt the classifier to a particular topic. This reduces the requirement of labeled data and training classifier for each topic. Collaborative CNN and LSTM is used for classification as it considers sequence prediction in the reviews. This enhances the accuracy of classification.

4 Semi-supervised domain adaptation for dual sentiment analysis

Semi-supervised domain adaptation is a process of training a classifier with less labeled data and a single classifier that adapt to different domains. In this section, we discuss about this topic.

4.1 Data expansion technique

In this section, we are going to discuss about data expansion technique that generate reversed review from the original review. Two rules are followed to generate reversed reviews with the help of antonym dictionary. They are:

- Text Inversion: Sentiment words present out of scope of negation in the review are replaced with their antonyms. Sentiment words are not reversed if they are in scope of negation, but negation words are removed such as *no*, *not*, *cannot*.
- Label Inversion: Label of the reversed review is updated with the reversed label of the original review.

For example, if the original review is *"I don't love Animals"* is inverted to *"I love Animals"*. Here, as *"love"* is in scope of negation, it is not reversed. *don't* is removed. Label of the review is reversed from *negative* to *positive*.

4.2 Dual sentiment analysis model

The fundamental conceptual structure of dual sentiment analysis (DSA) is presented in this section. Two steps that are followed in DSA are (1) Dual Training and (2) Dual Prediction.

4.2.1 Dual training

In Dual Training stage, the original reviews used for training are inverted to generate opposite-reviews using Data Expansion Technique. The training reviews are labeled as *"original-training-set"* and *"reverse-training-set"*. The label of the reverse-review are changed to opposite of their corresponding original reviews. The training is performed with the combination of both original and reverse reviews, hence the name dual-training. DSA make use of Naïve-Bayes Classifier, while both combination of reviews are together used for training. Let us consider the original review *"I don't like this book. It is boring"*, whose label is negative. Using the data expansion method, the negation word is searched and if found, it is removed and the next immediate sentiment word is not replaced by its antonym. If negation word is not present then the sentiment word is replaced with its antonym. Label is reversed to its opposite and the reversed-review is *"I like this book. It is interesting"*, whose label is positive.

4.2.2 Dual prediction

In dual prediction, each original sample review is taken as input and it is represented as x . In order to predict the label of x , a reverse of sample-original review is generated that is represented as x' . As the name dual prediction indicates, dual side of review is considered to predict the label of original sample-review. It not only considers the level of positivity or negativity of original review x , but also considers positivity or negativity of the reversed review x' . The drawback in the Bag-of-Words model is highly reduced in the dual-prediction. This helps in the accurate classification of the test sample reviews.

Let us indicate posterior probabilities of x and x' as $p(+ | x)$ and $p(+ | x')$ respectively. Here, $+$ represents positive or negative. Two sides of the review are considered while doing dual prediction. Positive sentiment degree of a review is found using two components. They are:

- How much positive is the review x , $p(+ | x)$.
- How much negative is the reversed review x' , $p(- | x')$.

Similarly, Negative sentiment degree of a review is found using two components. They are:

- How much negative is the review x , $p(- | x)$.
- How much positive is the reversed review x' , $p(+ | x')$.

Combination of the positive and negative predictions is used for dual prediction as indicated in Eqs. (1) and (2).

$$p(+ | x, x') = (1 - \eta).p(+|x) + \eta.p(-|x') \tag{1}$$

$$p(- | x, x') = (1 - \eta).p(-|x) + \eta.p(+|x') \tag{2}$$

Here η is a trade off parameter that is used to vary the weight of $p(.|x')$. Ideal performance is when η ranges between 0.5 to 0.7.

- Review (a)—“I don’t like this book. It is boring.”
- Review (b)—“I like this book. It is interesting.”

When Bag-of-Words method is applied to Review (a), based on the average score or term frequency, it decides the given review as positive in spite of presence of the word “don’t”. This is actually a fault and miss prediction as Review (a) is of Negative polarity. This is due to the presence of word “Like” has high score as positive word. But, the same Review (a) when it is decided using the DSA, it predicts the accurate result which gives it as a negative. This is because DSA will decide the polarity of the sentence by considering the polarity score of its reversed review. Review (b) is the reversed review of Review (a) which gets high score as positive sentiment due to the presence of words “Like” and “Interesting”. Hence Review (a) has the negative polarity.

5 Word expansion technique for domain adaptive sentiment classification

In this section, we are going to discuss about how a domain adaptive classifier is trained using semi-supervised learning method. The First step in Domain Adaptive Word Expansion Technique (DAWET) is to build a labeled dataset. Here, labeled dataset is built by combining equally the original reviews and its reversed reviews from all the domains. We are considering only positive and negative classes for review classification. The input reviews used for training are represented as $(x_i, y_i) + (x'_i, y_i)$, where x_i indicates feature vector for the input data and sentiment class of x_i is represented by y_i . x'_i is the reversed review of x_i .

5.1 Computation of feature values

Two components of text features are General Sentiment Words and Domain Adaptive Sentiment Words. General sentiment words are commonly used words to express

sentiment and these words are downloaded from the web. For the implementation of DAWET, General sentiment words are taken from two sources, (1) WordNet Affect and (2) Public sentiment lexicon. WordNet Affect contains a list of words and their corresponding class label. Public sentiment lexicon has two list of words, one containing positive and another with negative words. Both these corpus combine to form Common Sentiment Word set and it is denoted by P .

Domain adaptive sentiment Words: For the domain adaptive classification, along with the labeled reviews from all domains, it requires domain adaptive words from a particular topic to which classifier is getting adapted. Few confident reviews from all domains are selected and only frequently occurring nouns, adjectives are selected to form domain adaptive word set. This set is denoted by Q . These selected confident reviews are reversed using dual sentiment analysis. Frequently occurring nouns, adjectives are selected from these reviews and it is denoted by Q' . Finally, size of the text features is $(P + Q + Q')$.

5.2 Domain adaptive training

Domain Adaptive Training is a process of adapting a classifier built on mixed domain labeled reviews to a particular domain d . Two steps are followed in each iteration through DAWET while adapting to a particular domain and they are (1) Train the classifier, (2) Feature extraction and updation. These two steps help the classifier transform towards a particular topic.

5.2.1 Train the classifier

In DAWET algorithm, we are training the classifier using few labeled reviews and testing using unlabeled reviews. Topic specific words from different domains and general sentiment words combine to form Labeled set L . This classifier is weak and general as it is trained using general words resulting in reduction of classification accuracy. Hence, this classifier is trained using unlabeled review set U from a particular domain d . This will help the classifier to get adapted to domain d . Unlabeled reviews that are selected for training cannot be selected randomly. Spam or irrelevant reviews leads to misleading the classifier. The confidence of the unlabeled reviews t_j is defined as S_j and it is obtained by the Eq. (3)

$$S_j = \frac{\max_y \{w_y^T x_j\}}{\sum_y w_y^T x_j} \tag{3}$$

Here w is the weight of the topic adaptive words. Given a confidence threshold τ , number of reviews t_j that satisfy S_j

$\geq \tau$ are selected for training the classifier in each iteration. These selected reviews are reversed to form t'_j . These reviews $t_j + t'_j$ form Unlabeled dataset U .

5.2.2 Feature extraction and updation

Text features are extracted from unlabeled reviews set U . Those are General Sentiment features and Domain-Adaptive Sentiment features. General Sentiment features remain same till the end of training the classifier. Domain adaptive feature set is updated with domain adaptive words in each iteration of the algorithm. The weight of the domain adaptive sentiment word d is calculated by computing the frequency of occurrence of that word in unlabeled set as belonging to class y and it is defined in Eq. (4)

$$\phi_y(d) = \sum_{y_j=y} f_j(d) \quad (4)$$

Here $f_i(d)$ indicates term frequency of the domain adaptive sentiment word d in the review t_j and $\phi_y(d)$ is the summation of the term frequency of the topic adaptive sentiment word d in the review t_j with the predicted class being y . Similarly, let $f_i(r)$ is the term frequency of the domain adaptive word r in the review t_r . Here, t_r is the reversed review of t_j . $\phi_y(r)$ is the summation of the term frequency of the domain adaptive word r in the review t_r with the expected class label y . Finally, the feature value of the word d is taken as the highest of the addition of weight of any word. So the feature value of domain adaptive word of review t_j is given by Eq. (5).

$$x_d = \max_y \{ \phi_y(d) \} \quad (5)$$

In Eq. (5), x_d is the feature value for the word d . Similarly, feature value of domain adaptive word r of review t_r is given in Eq. (6).

$$x_r = \max_y \{ \phi_y(r) \} \quad (6)$$

Domain adaptive words are added both from reviews and its reversed reviews. For training process, all words are not considered. The words that pass the threshold of significance are considered for addition and it is calculated using the Eq. (7).

$$\varpi_d = \frac{\max_y \{ \phi_y(d) \}}{\sum_y \phi_y(d)} \quad (7)$$

In Eq. (7), ϖ_d is the significance of the word d . For word r from reversed review, significance is computed using the Eq. (21).

$$\varpi_r = \frac{\max_y \{ \phi_y(r) \}}{\sum_y \phi_y(r)} \quad (8)$$

Finally, feature values, x_d are calculated for the words from the unlabeled training set as given in the Eq. (9)

$$x_d = \beta_d \cdot \max_y \{ \phi_y(d) \} \quad (9)$$

where β_d is the selection vector and defined in Eq. (12).

$$\beta_d = I(\varpi_d \geq \theta) \quad (10)$$

In the Eq. (12), $I(\cdot)$ is the pointer function and θ is the significance threshold. When pointer function $I(\cdot)$ returns 1, the word d is selected and d is not selected if $I(\cdot)$ returns 0.

Feature values, x_r are found for the words from the unlabeled training set and it is indicated in Eq. (22)

$$x_r = \beta_r \cdot \max_y \{ \phi_y(r) \} \quad (11)$$

where β_r is the selection vector and defined in Eq. (12).

$$\beta_r = I(\varpi_r \geq \theta) \quad (12)$$

5.3 Word expansion using unlabeled reviews with dual sentiment analysis

Unlabeled reviews U from a domain e are used to convert the weak classifier to a domain adaptive classifier. The selected unlabeled review t_j is predicted to belong to class y'_j according to Eq. (1). Class of review t_j is found using Eq. (13).

$$w_{y'_j}^T x_j = \max \{ w_y^T x_j \} \quad (13)$$

The selected unlabeled review t_j is reversed to form review t_r is predicted to belong to class y'_r according to Eq. (1). We write the Eq. (14) to find the class of review t_r . Here, for which class of the review $\{ w_y^T x_j \}$ is maximum, that class is assigned to the respective review. Weight of the review x_j for each class is calculated and class of the review is decided depending on the maximum value of the w_y^T of review x_j .

$$w_{y'_r}^T x_r = \max \{ w_y^T x_r \} \quad (14)$$

Algorithm *DAWET* gives the procedure to train a classifier that adapts to a particular topic or Domain. This avoids the noise that is added into the system. Unlabeled reviews t_j and its reversed reviews t_r that satisfy $S_c > \tau$ are selected for training the classifier. Here, τ is the confidence threshold. Words that satisfy the condition $\varpi > \theta$ are selected and considered for Domain-adaptive word expansion.

Domain-Adaptive words are extracted from the confident, significant words that satisfies the condition $w > g$ where w weight of the topic adaptive word and g represents threshold to select the topic adaptive words from each iteration. Augmented set L is the combination of unlabeled reviews U and Labeled set L . Domain-Adaptive words are selected and feature values(weight) are updated in every iteration. Words that are not selected in an iteration, carry their weight to the next iteration. Adaptation procedure stops when number of iterations $I > M$. Here, M indicate maximum number of iterations. Finally, trained classifier is a result obtained by the classifier trained using the augmented labeled data set L . Semi-supervised domain adaptation is represented in *DAWET* Algorithm.

layer, information is updated sequentially and output is obtained from output layer.

Convolution neural network (CNN) is a feed-forward artificial neural network. CNN is made of three types of layers: Input, output and hidden layers. Hidden layer is made of convolution of several layers with non linear activation function. Convolution is applied on the input neuron to get the output. In CNN, sentences are represented in the form of matrix.

6.1 Collaboration of LSTM and CNN

In this section, we are going to discuss about the architecture of *LSTM + CNN* model. Word embedding is applied to

Algorithm DAWET:Domain Adaptation with Word Expansion Technique on Domain e

- 1: **Input**
 - 2: L : labeled reviews consisting of Positive and Negative sentiment classes on multiple Domains;
 - 3: U : unlabeled reviews from Domain e ;
 - 4: **Output**
 - 5: C : Classifier trained on features x on Domain e ;
 -
 - 6: Confidence score S_j of Reviews is calculated using equation (3);
 - 7: **while** number of iterations $I > M$ **do**
 - 8: Select the l most confident and unlabeled reviews t_j in each sentiment class that satisfy condition $S_j > \tau$;
 - 9: Find the reverse of the reviews t_j ;
 - 10: Calculate the weight x_d of the domain adaptive sentiment word d using (5);
 - 11: Calculate the significance ϖ of domain adaptive words using equation (7);
 - 12: Select the domain adaptive words whose significance $\varpi > \theta$ from each class;
 - 13: Add the Domain adaptive words to the labeled set that satisfy the condition $w > g$ from each class;
 - 14: Feature values are updated;
 - 15: Reviews are added to estimated class label from U to L ;
 - 16: Increment I ;
 - 17: **end while**
-

6 Collaborative deep learning approach for dual sentiment analysis

Neural network is one of a famous approach to solve complex problems. Nowadays it is successful in NLP also. In *RNN*, each neuron is given previous neuron output along with the layer input. Hence, this model is used to classify sequential data. Long short term memory *LSTM* is a type of a recurrent neural network (RNN). *LSTM* uses sequence of words in reviews to predict the result. At each

get the input and this is given as input to the convolution neural network to extract high level features. *CNN* extracts high level features from the input. Its output is given as the input to the *LSTM* recurrent neural network. This is followed with a classifier layer.

6.1.1 The embedding layer

In this layer, words are converted into feature vector that considers both syntactic and semantic information. Sequence of words $[w_1, \dots, w_{|s|}]$ that are derived from the

vocabulary V is the input to this layer. Word Embedding matrix is by concatenating embedding of all words in V and is represented by $W = R^{l \times |V|}$.

6.1.2 The convolution layer

Let the input sentence is represented by $s = \{w_1, w_2, \dots, w_n\}$. $x(i) = R^k$ is the k dimensional word vector of i th word of a sentence of n words. It is represented by Eq. (15)

$$x_{(1:n)} = x_1 \oplus x_2 \oplus \dots \oplus x_n \quad (15)$$

Here \oplus is a concatenation operator. Let $x_{i:i+j}$ represents the concatenation of words $x_i, x_{i+1}, \dots, x_{i+j}$. Convolution operator also includes a filter that is applied to h words window that results in new feature values. For example, window of words $x_{i:i+h-1}$ generate feature c_i and it is given by Eq. (16)

$$c_i = f(W \cdot x_{i:i+h-1} + b) \quad (16)$$

Here b is the bias and f is the non-linear function. Feature map c is produced by applying the filter to all possible window of words in a sentence $\{x_{1:h}, x_{2:h+1}, \dots, x_{n-h+1:n}\}$. Hence feature map is represented in Eq. (17)

$$c = [c_1, c_2, \dots, c_{n-h+1}] \quad (17)$$

These feature maps are given as input to LSTM network. LSTM captures long term dependencies.

6.1.3 Long short term memory neural network

The convolution filters capture information from text for a limited window size, for example, 3 or 4 words. Hence, CNN cannot capture long term dependencies in sequential data. We, humans do not start thinking from scratch each time, but we use previous thoughts and reacts to the present. Traditional neural networks do not address these issue. Recurrent neural network has a solution to use previous knowledge to predict the output. RNN has loops that help information to remain in the network. Neural Network S takes input (i_t) and output (h_t). Loop helps in passing information from one step to the next step of the network. Some times, remembering short term memory is enough to predict. For example, in the sentence, "Sun is shining bright in the sky". Here, it is easy to predict word "sky". But in some cases, this does not work. For example "I grew up in France. .I speak fluent French". Here, there is a long gap between language French and the France. It is difficult to predict language French as there is gap between French and France. RNN is not very accurate in predicting such long term dependencies. LSTM is a special kind of RNN that can handle long term dependencies as it can remember information for a longer time.

LSTM Walk through: LSTM contains memory cells and it decides what information needs to be stored and what needs to be disposed from the cell state. Steps that are followed are:

- **Step 1 LSTM** check the values of h_{t-1} and i_t . Outputs 0 or 1 for each information in the cell state C_{t-1} . 1 indicates keep the information and 0 indicate discard the information. Sigmoidal layer that is also called as input gate layer find the values that need to be updated and it is represented in Eq. (18).

$$q_t = \sigma(W_f \cdot [h_{t-1}, i_t] + b_f) \quad (18)$$

- **Step 2** A vector of new candidates C_t is created using \tanh layer that could be added to the cell state.

$$n_t = \tanh(W_n \cdot [h_{t-1}, i_t] + b_n) \quad (19)$$

$$C_t = \tanh(W_c \cdot [h_{t-1}, i_t] + b_c) \quad (20)$$

- **Step 3** Now we have to update the old cell C_{t-1} state to a new cell state C_t . After forgetting the information that are selected to forget, we have to multiply the old state C_{t-1} by q_t . New state C_{nt} is found using Eq. (21)

$$C_{nt} = q_t * C_{t-1} + n_t * C_t \quad (21)$$

- **Step 4** In this final step, we are going to find the output. First, sigmoidal layer decides which all states of the cell are considered for output and it is represented in Eq. (22). We need to output the information that we decide on. hence, cell states are passed through \tanh and then it is multiplied with the output of sigmoidal function as indicated in Eq. (23). This allow only information to pass that are selected.

$$o_t = \sigma(W_o \cdot [h_{t-1}, i_t] + b_o) \quad (22)$$

$$h_t = o_t * \tanh(C_{nt}) \quad (23)$$

7 Implementation

In this section, we are going to discuss about the implementation of dual sentiment analysis concept using Semi-supervised domain adaptation and deep learning concepts. We have considered positive and negative sentiment classification on four domains of Multi-Domain Sentiment Dataset [16]. They are Book, DVD, Electronics and Kitchen Domain.

7.1 Dataset

We use multi-Domain dataset that contains English reviews that are taken from Amazon.com. Reviews are from four domains: *Book*, *DVD*, *Electronics* and *Kitchen*.

Table 1 Multi-domain sentiment dataset

Dataset	Number of positive reviews	Number of negative reviews
Book	1000	1000
DVD	1000	1000
Electronics	1000	1000
Kitchen	1000	1000
Total	4000	4000

Each domain contains 1000 positive and 1000 negative reviews and total of 4000 reviews are present. Details of the dataset is indicated in Table 1. For supervised dual sentiment analysis, each domain reviews are divided into 5 parts. Four parts are used for training and other for testing the classifier. Naive bayes classifier is used to classify the reviews.

7.2 Data expansion

The implementation of the *DSA* algorithm needs labeled data for training and unlabeled data for testing. First, the original labeled data is converted to its reverse polarity. Both original and reversed data is used to train the classifier. This makes the classifier model more stable and robust. The module of “Train dataset” consists of original review and its reversed review that are used for training the classifier. The module “Test dataset” consists of test review that are to be examined using the trained classifier.

SDA-DSA uses Antonym Dictionary that is formed by extracting the words from WordNet. Wordnet is a huge repository of the data present in the English language and also many more language support has been provided by the Wordnet. Nouns, verbs, adjectives and adverbs are combined into groups of synonyms (synsets), each expressing a unique concept. Synsets are linked internally by methods of conceptual semantic and lexical relations.

7.3 Semisupervised domain adaptation

The implementation of Semi-supervised Domain Adaptation for Dual Sentiment Analysis (*SDA-DSA*) requires both labeled data as well as Unlabeled data. Reviews are taken from the Multi-Domain labeled data set as mentioned in Table 1. (*SDA-DSA*) is a semi-supervised learning and hence we take few labeled data for training. (*SDA-DSA*) is also Domain adaptive classification. First, the classifier is trained using labeled reviews and its reversed reviews. Labeled set *L* is a mix of labeled data from all four domains of both positive and negative polarity. Training is done considering 1% and 5% of the dataset as labeled set. Out

Table 2 Comparison of classification accuracy values on different sample ratios

Sample Ratio	-	1%		5%		
		80 Reviews		400 Reviews		
L	-	<i>DSA-WN</i>	<i>TASC</i>	<i>SDA-DSA</i>	<i>TASC</i>	<i>SDA-DSA</i>
<i>Book</i>	0.82	0.85	0.90	0.83	0.91	
<i>DVD</i>	0.83	0.84	0.92	0.84	0.93	
<i>Electronics</i>	0.85	0.86	0.92	0.86	0.92	
<i>Kitchen</i>	0.82	0.84	0.91	0.85	0.93	

of 8000 reviews, 80 are considered as labeled set *L* for 1%. These 80 reviews are reversed to its opposite polarity. Hence, now we have 160 reviews for training. Similarly, for 5% of dataset, 400 labeled mixed reviews from all four domains are considered. These 400 reviews are polarity reversed and are used for training.

Classifier trained on the labeled set from mixed reviews from all domain is made to adapt to a particular topic *e*. This is done by adapting the classifier to a particular topic *e* by training the classifier with the domain adaptive words extracted from the unlabeled reviews.

7.4 Dual training

In the stage of training, all the original sample reviews are converted into their reverse and made opposite. It is represented as “Original training set” and “Reverse training set” respectively. In data expansion technique there exists one-one correspondence among original and reverse reviews. The training of the classifier is done by maximization along with the combination of together original and reverse reviews. Once the training data is ready with both “original dataset” and “reverse dataset”, then it is used to train into the Bayesian-classifier. This is the classifier of the statistical nature. It calculates the probabilities of the tuples or the given class. The classification is based on the method of Bayes theorem.

8 Results and analysis

In this section, we are going to discuss the performance of the proposed domain adaptation with semi-supervised learning and collaborative LSTM-CNN with dual sentiment analysis.

8.1 Semi-supervised domain adaptation for dual sentiment analysis

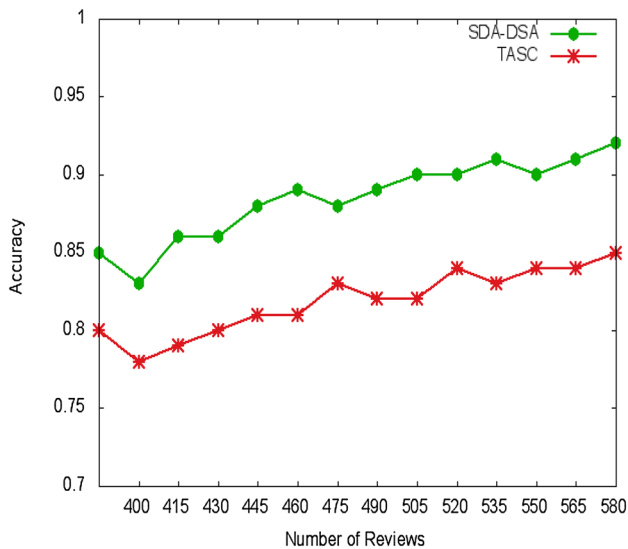
Semi-supervised domain adaptation using dual sentiment analysis (*SDA-DSA*) classify each domain reviews

Table 3 Comparison of accuracy values for book reviews of 5% sample ratio with step length = 15

Iterations	1	2	3	4	5	6	7	8	9	10	11	12	13	14
<i>SDA-DSA</i>	0.85	0.83	0.86	0.86	0.88	0.89	0.88	0.89	0.90	0.90	0.91	0.90	0.91	0.92
<i>TASC</i>	0.80	0.78	0.79	0.80	0.81	0.81	0.83	0.82	0.83	0.84	0.83	0.84	0.84	0.85

Table 4 Comparison of accuracy values for DVD reviews of 5% sample ratio with step length = 15

Iterations	1	2	3	4	5	6	7	8	9	10	11	12	13	14
<i>SDA-DSA</i>	0.86	0.83	0.84	0.85	0.86	0.88	0.89	0.90	0.90	0.92	0.91	0.93	0.92	0.93
<i>TASC</i>	0.78	0.76	0.77	0.76	0.78	0.79	0.79	0.80	0.81	0.81	0.82	0.83	0.83	0.84

**Fig. 1** Comparison of accuracy for book reviews of 5% sample ratio with step length = 15

using Naive Bayes Classifier. In Table 2 *DSA-WN* classification accuracy values are listed for each domain. These results are compared with two Semi-supervised domain adaptive methods. They are *TASC* [17] and our proposed method *SDA-DSA*. *TASC* performed marginally better than *DSA-WN*. *DSA-WN* is supervised learning and Dual Sentiment Analysis are the positive points of *DSA-WN*. Still, *TASC* being semisupervised learning performed better than *DSA-WN* as *TASC* extracted topic adaptive words from the reviews itself. *DSA-WN* uses dual sentiment analysis to address polarity shift problem. Whereas, *TASC* does not address polarity shift problem. Hence our method, *SDA-DSA* address polarity shift problem by applying dual sentiment analysis using semi supervised learning. As shown in Table 3, we observe that *SDA-DSA* performs better than *TASC* and *DSA-WN* with respect to Book, DVD, Electronics and Kitchen domains considering 14 iterations for both 1% and 5% sample ratio.

Sample ratio is the percentage of mixed labeled data from all domains selected for training the classifier. 80

reviews are selected from 1% sample ratio and 400 reviews are selected from 5% sample ratio. When 1% sample ratio is taken for training the classifier, *SDA-DSA* performed better than *TASC* and *DSA-WN* with accuracy of 0.90 for Book domain. Similarly *SDA-DSA* performed better on all other domains. We compare our results when Sample ration of 5% is considered. *SDA-DSA* perform better than *TASC* and *DSA-WN* with accuracy of 0.91. *SDA-DSA* with sample ratio 5% performed better than *SDA-DSA* with 1% sample ratio. This is because the number of labeled data used for training the classifier is more and hence more accurate results. There is 2% increase in the accuracy of classification when sample ration is changed from 1 to 5% for Kitchen domain.

We have compared the performance of all four domain on *TASC* and *SDA-DSA* algorithm for 5% sample ratio. We have considered upto 14 iterations for training and step length of 15 unlabeled reviews are selected in each iteration to train the classifier. Hence, after each iteration, 15 unlabeled reviews from the particular domain for which the classifier is getting adapted to are selected for training the classifier. In the process of classifier adapting to a particular Topic, in each iteration 15 reviews are considered for training. Table 4 tabulates accuracy values after each iteration of *Book* domain. In every iteration, 15 reviews are selected for training. In the first iteration, performance is better compared to second iteration. In the first iteration all reviews used for training are labeled. In the second iteration, accuracy reduces as selected reviews are unlabeled. From third iteration performance improves. Compared to *TASC*, *SDA-DSA* perform better in every iteration. Variation of accuracy values for *Book* domain for step length 15 are plotted in Fig. 1. Similar performance is observed in *DVD*, *Electronics* and *Kitchen* domain and variations are listed in Tables 5, 6 and 7 respectively. Variations are plotted in Figs. 2, 3 and 4 of *DVD*, *Electronics* and *Kitchen* domain respectively.

The observed of the variation of accuracy values when step length is changed from 15 to 30 is tabulated in Table 7 of *Book* domain. in the first iteration, classifier with 30 step length perform better than classifier trained with step length 15. This is because, more labeled data is available.

Table 5 Comparison of accuracy values for electronics reviews of 5% sample ratio with step length = 15

Iterations	1	2	3	4	5	6	7	8	9	10	11	12	13	14
SDA-DSA	0.87	0.86	0.86	0.88	0.89	0.90	0.91	0.90	0.91	0.92	0.92	0.93	0.92	0.93
TASC	0.77	0.78	0.77	0.79	0.80	0.81	0.81	0.83	0.85	0.84	0.85	0.87	0.89	0.86

Table 6 Comparison of accuracy values for kitchen reviews of 5% sample ratio with step length = 15

Iterations	1	2	3	4	5	6	7	8	9	10	11	12	13	14
SDA-DSA	0.87	0.84	0.84	0.85	0.86	0.87	0.89	0.88	0.90	0.91	0.92	0.93	0.94	0.94
TASC	0.79	0.76	0.76	0.77	0.78	0.78	0.79	0.79	0.80	0.81	0.81	0.82	0.83	0.85

Table 7 Performance of SDA-DSA algorithm on 15 and 30 step lengths on book reviews of 5% Sample ratio

Iterations	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15 Steps	0.85	0.83	0.86	0.86	0.88	0.89	0.88	0.89	0.90	0.90	0.91	0.90	0.91	0.92
30 Steps	0.87	0.85	0.87	0.85	0.83	0.80	0.79	0.75	0.76	0.73	0.72	0.72	0.68	0.64

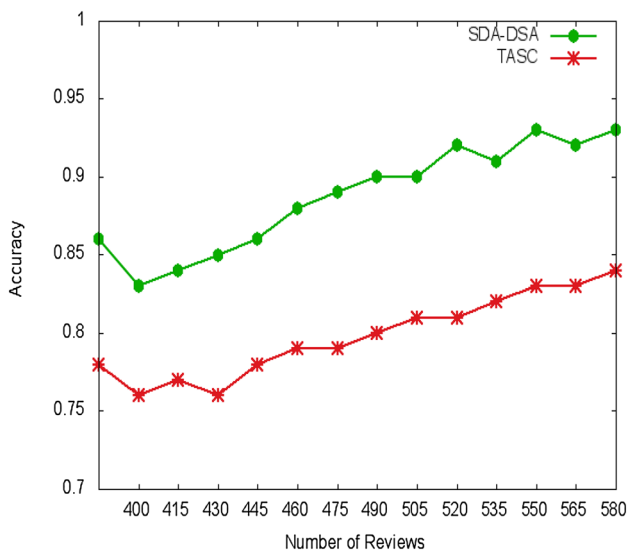


Fig. 2 Comparison of accuracy for DVD reviews of 5% sample ratio with step length = 15

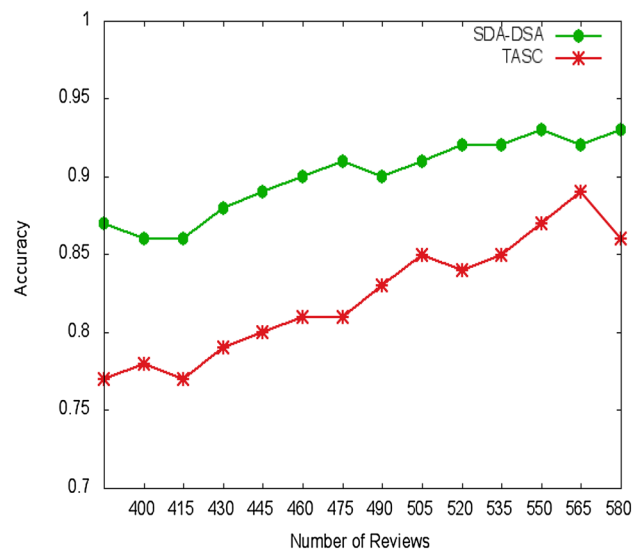


Fig. 3 Comparison of accuracy for electronics reviews of 5% sample ratio with step length = 15

Performance remains better till iteration 3. After iteration 3, accuracy of classification starts decreasing when step length considered is 30. This results is due to more number of unlabeled data used for training and hence misguiding the classifier. As the iteration proceeds, accuracy of classifier with step length 30 goes on decreasing compared to accuracy of classifier with step length 15. Performance of accuracy of classifier with step length 15 goes on increasing after each iteration. Figure plots the variation of accuracy values of *Book* domain with step length 15 and 30. Similar observations are present when *DVD* reviews are classified with step length of 15 and 30. Values are tabulated in Table 8 and plotted graph in Figure.

8.2 Collaborative deep learning for dual sentiment analysis

In this section, we are going to discuss about the performance of the *LSTM*, *CNN* and *LSTM + CNN* for Dual Sentiment Analysis. We have compared the performance of SDA-DSA for 15 and 30 step length on book and DVD reviews and it is represented in (Figs. 5, 6) respectively. We compare the accuracy of classification considering positive reviews and its reversed reviews using *LSTM*, *CNN* and *LSTM + CNN* for various epoch values and it is represented in Fig. 7. As we increase the number of epochs and re-train the model, the accuracy of the resulting model

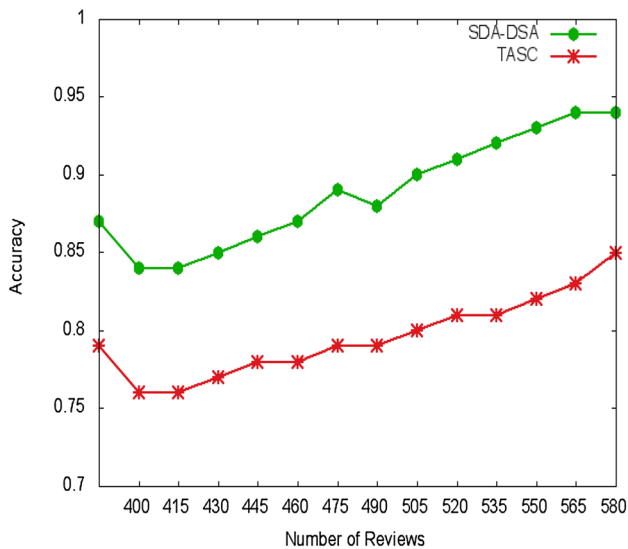


Fig. 4 Comparison of accuracy for kitchen reviews of 5% sample ratio with step length = 15

starts to increase. The graph outlines the training process for 3 different models—*LSTM*, *CNN* and a collaborative model that includes both *LSTM + CNN*. In epoch 1, the accuracy of *CNN* is the least followed by *LSTM* and then *LSTM + CNN*. As the number of epochs increase, the

accuracy also increases. All the three models perform better with increase in the epochs. This is due to the fact that model gets more amount of training data. The Fig. 7 was plotted using the results of a model trained with only positive and reversed positive (that are obtained by reversing the positive reviews) reviews. *LSTM + CNN* perform better with accuracy 0.987 compared to *CNN* with accuracy 0.975 and *LSTM* alone with accuracy 0.980 at the end of tenth epoch. Figure 8 is a plot of model trained on negative and reversed negative reviews. Performance of the combined *CNN* and *LSTM* model is fairly good with an accuracy of 0.98.

Figure 9 represents variation of the accuracy for various percentage of the training data used for training the classification model. The input data for the model is prepared using (1) positive and reversed positive reviews, (2) negative and reversed negative reviews. Out of the total available reviews, we start with taking 10% of the total data for training, i.e. 200 reviews. In the graph, we compare four different models—*Naive Bayes*, *CNN*, *LSTM* and a combination of *LSTM + CNN*. We observe from the plot that the *Naive Bayes*'s classification without Dual Sentiment Analysis model has the least accuracy compared to *CNN*, *LSTM* and *LSTM + CNN*. There is a improvement of the accuracy value at 50% training data and after that again it reduces. For the other models, the accuracy is much better

Table 8 Performance of SDA-DSA algorithm on 15 and 30 step lengths on DVD reviews of 5% Sample ratio

Iterations	1	2	3	4	5	6	7	8	9	10	11	12	13	14
15 Steps	0.86	0.83	0.84	0.85	0.86	0.88	0.89	0.90	0.90	0.92	0.91	0.93	0.92	0.93
30 Steps	0.88	0.86	0.83	0.81	0.81	0.79	0.77	0.75	0.76	0.74	0.72	0.71	0.68	0.61

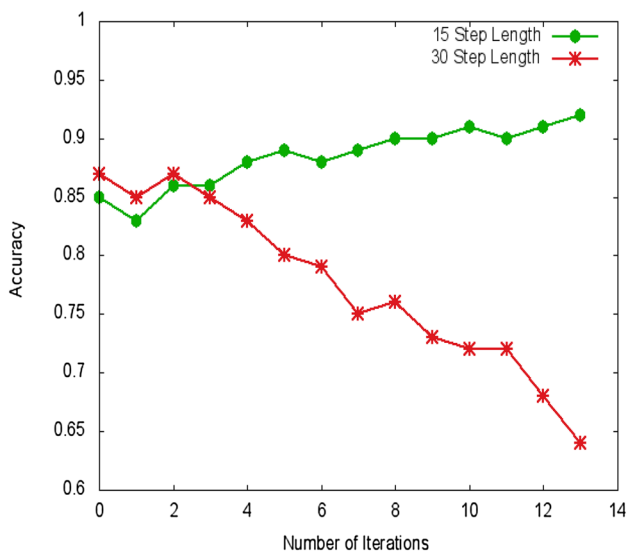


Fig. 5 Comparison of SDA-DSA for 15 and 30 step length on book reviews

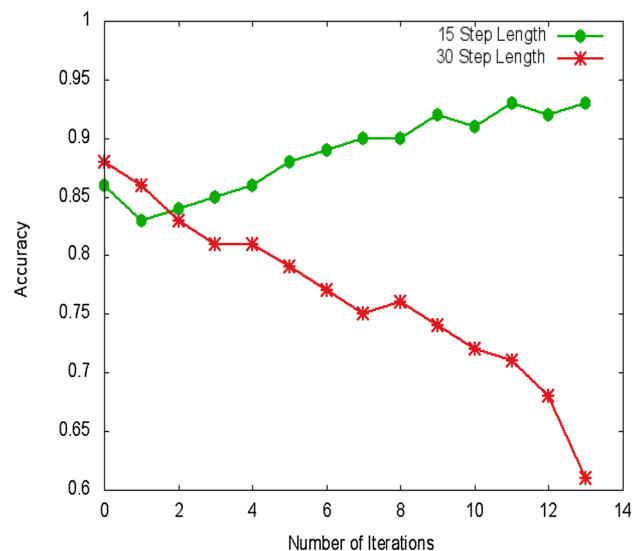


Fig. 6 Comparison of SDA-DSA for 15 and 30 step length on DVD reviews

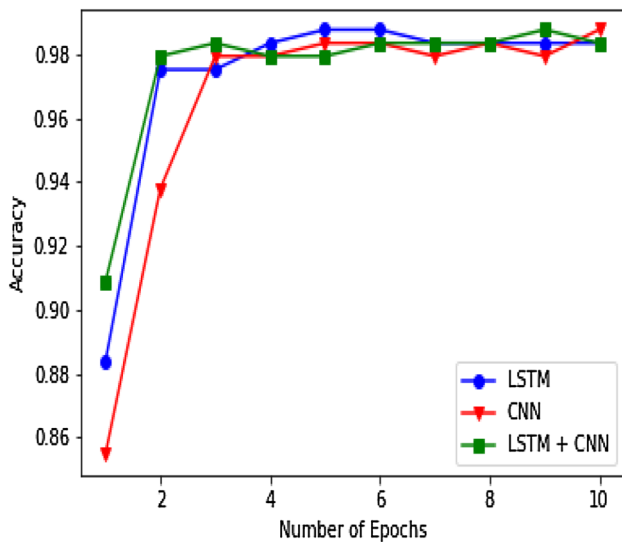


Fig. 7 Accuracy variation for different epochs of *Book* domain

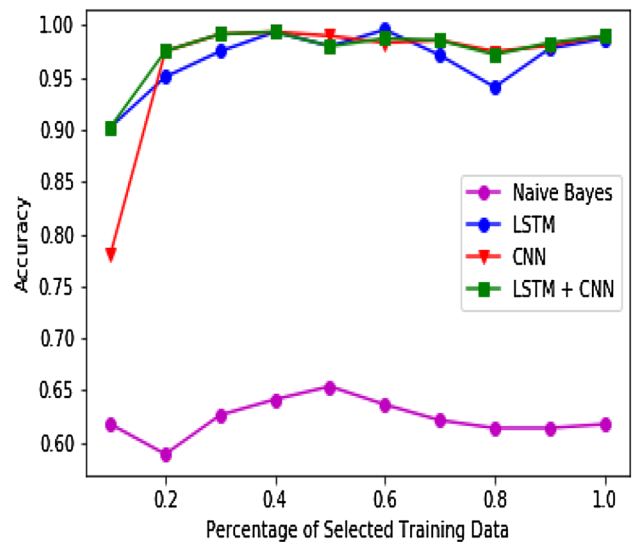


Fig. 9 Accuracy for different percentage of selected training data

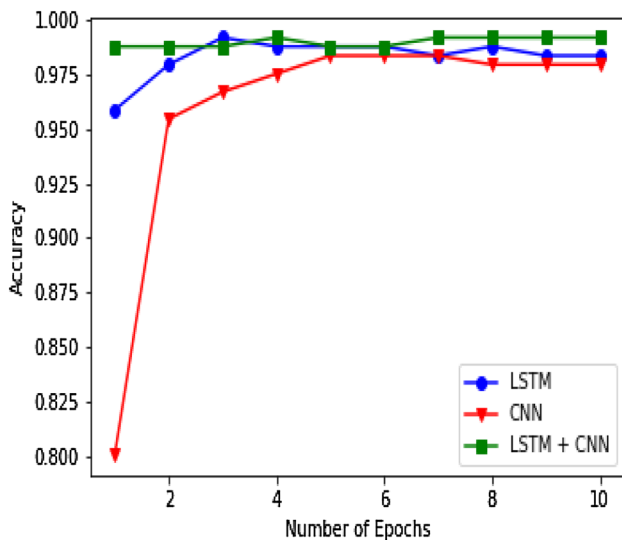


Fig. 8 Accuracy variation for different epochs

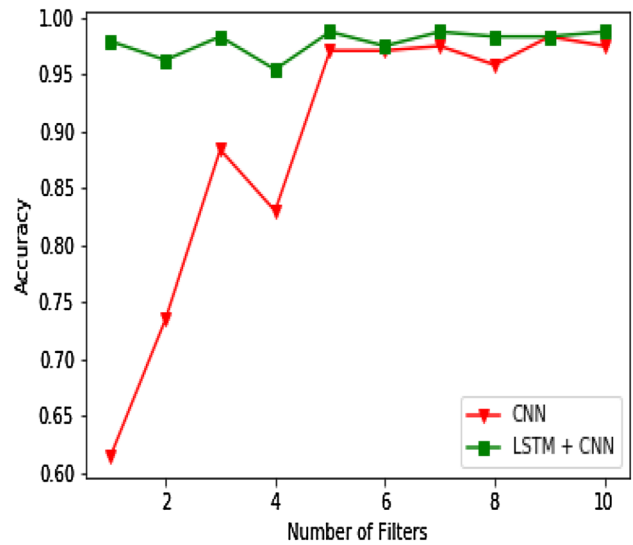


Fig. 10 Accuracy for different filter size

than *Naive Bayes*, ranging from 75 to 98%. The *LSTM + CNN* model gives the best and the most stable results as it combines the two robust models, *LSTM* and *CNN*. *LSTM + CNN* model gives 98% accuracy when training data is 100%. *LSTM + CNN* model perform better than *CNN* and *LSTM* when the training data is 30% and 40%. This is the main advantage of *LSTM + CNN* model.

Number of filters used for training Neural Network based models such as *CNN* is also an important measure that influence the accuracy of the model. Filter size specifies how many neighbor's information can the Neural Network see when processing the current layer. Selecting the right

number of filters is important for building a right model that can give good results in its prediction phase. Performance is plotted in Fig. 10. We start with a filter size of 1 which indicates a 1*1 filter and go up to a filter size of 10 which indicates a 10*10 filter. For the *CNN* based model, the accuracy is low for smaller number of filters. As the filter size increases, the accuracy improves and becomes stable after filter size 5. For *CNN+LSTM* model, the accuracy is 98% when filter size is 1 and maintains a steady accuracy for variation of filter size.

The Fig. 11 indicates the relationship between Pool Size and Accuracy. Pool size indicates the amount of reduction in spatial size of the quantity of parameters. We run the model with the most common pooling sizes of 2*2 and 4*4. Both

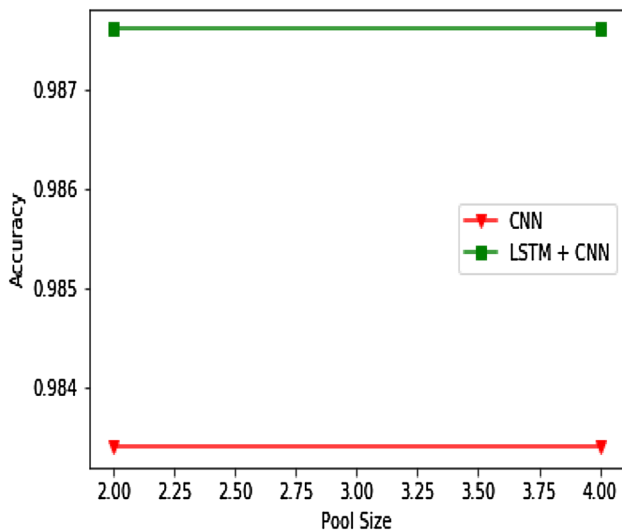


Fig. 11 Accuracy for pool size 2*2 and 4*4

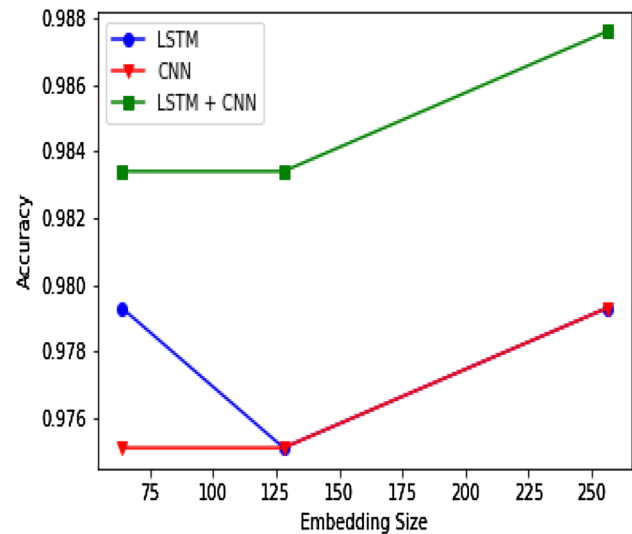


Fig. 13 Variation of accuracy for different embedding size

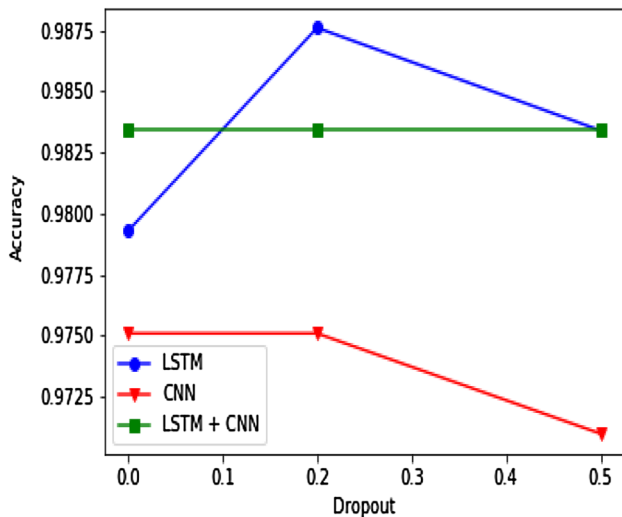


Fig. 12 Variation of accuracy for different dropout values

the models give same accuracy irrespective of the pool size considered. The Accuracy is higher for the *LSTM + CNN* compared to that of the *CNN* model. Filter size considered is 5*5. Dropout is a technique of making the model forget a portion of the data that it has learned. This results in reducing the over fitting of the model. We run all the three models *CNN*, *LSTM* and *LSTM + CNN* with the dropout sizes of 0, 0.2 and 0.5. Experimental results are plotted in Fig. 12. The accuracy of *CNN* is the least among the three. The *CNN* model has an accuracy of 97.5% when the dropout is 0. As we increase the dropout, the accuracy reduces by a small amount which removes subtle over fitting in the model. The same is the case with the *LSTM* model. The *CNN* model seems to have constant accuracy value throughout the iterations.

The Embedding layer is initialized with random weights and learns an embedding for all the words in the training dataset. It uses the learned embedding for predicting words later in the prediction phase. We use embedding sizes of 64, 128 and 256. In Fig. 13 observe that the models generally perform better with a higher embedding size. The accuracy of the *LSTM + CNN* model is better than *CNN* and *LSTM* for all the embedding sizes.

9 Conclusions

A novel idea called dual sentiment analysis is used to solve the problem of polarity shift for the classification of sentiment. Dual Sentiment Analysis create reverse review for every single original training review using Data Expansion Technique. labeled data is costly and time consuming process. Semi-supervised domain adaptive dual sentiment analysis (*SDA-DSA*) use less labeled data and single classifier can adapt to different Domains. Proposed (*SDA-DSA*) perform well compared to supervised Dual Sentiment Classification with good accuracy. Reviews are long and hence long term dependency need to be addressed. Collaborative *LSTM + CNN* for dual sentiment analysis classify the reviews considering long term dependency with higher accuracy compared to *naive bayes*, *CNN* or *LSTM*.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Xia R, Xu F, Zong C, Li Q, Qi Y, Li T et al (2015) Dual sentiment analysis: considering two sides of one review. *IEEE Trans Knowl Eng* 27(8):2120–2133
2. Zhuang L, Jing F, Zhu X-Y (2006) Movie review mining and summarization. In: Proceedings of the fifteenth ACM international conference on information and knowledge management, pp 43–50
3. Pang B, Lee L et al (2008) Opinion mining and sentiment analysis. *Found Trends Inf Retr* 2:1–135
4. Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing, pp 347–354
5. Choi Y, Cardie C (2008) Learning with compositional semantics as structural inference for subsentential sentiment analysis. In: Proceedings of the conference on empirical methods in natural language processing, pp 793–801
6. Li F, Han C, Huang M, Zhu X, Xia Y-J, Zhang S, Yu H (2010) Structure-aware review mining and summarization. In: Proceedings of the 23rd international conference on computational linguistics, pp 653–661
7. Wu Y, Zhang Q, Huang X, Wu L (2009) Phrase dependency parsing for opinion mining. In: Proceedings of the 2009 conference on empirical methods in natural language processing, vol 3, pp 1533–1541
8. Ma T, Wan X (2010) Opinion target extraction in Chinese news comments. In: Proceedings of the 23rd international conference on computational linguistics, Posters, pp 782–790
9. Zhang Q, Wu Y, Li T, Ogihara M, Johnson J, Huang X (2009) Mining product reviews based on shallow dependency parsing. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pp 726–727
10. Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng A, Potts C (2013) Recursive deep models for semantic compositionality over a sentiment Treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing, pp 1631–1642
11. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv preprint [arXiv:1408.5882](https://arxiv.org/abs/1408.5882)
12. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning, pp 1188–1196
13. Dong L, Wei F, Tan C, Tang D, Zhou M, Xu K (2014) Adaptive recursive neural network for target-dependent Twitter sentiment classification. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, vol 2, pp 49–54
14. Lakkaraju H, Socher R, Manning C (2014) Aspect specific sentiment analysis using hierarchical deep learning. NIPS workshop on deep learning and representation learning
15. Tang D, Qin B, Feng X, Liu T (2015) Target-dependent sentiment classification with long short term memory. *CoRR*, [arXiv:abs/1512.01100](https://arxiv.org/abs/1512.01100)
16. <http://www.cs.jhu.edu/~mdredze/datasets/sentiment/>
17. Liu S, Cheng X, Li F, Li F (2015) TASC: topic-adaptive sentiment classification on dynamic tweets. *IEEE Trans Knowl Data Eng* 27(6):1696–1709

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.