A novel deep learning architecture for sentiment classification

Conference Paper · March 2016

DOI: 10.1109/RAIT.2016.7507953

3 authors:

Rahul Ghosh
Samsung
2 PUBLICATIONS **36** CITATIONS
SEE PROFILE

Kumar Ravi
Institute for Development & Research in Banking Technology
17 PUBLICATIONS **1,356** CITATIONS
SEE PROFILE

Ravi Vadlamani
Institute for Development & Research in Banking Technology
107 PUBLICATIONS **2,239** CITATIONS
SEE PROFILE

https://doi.org/10.1109/RAIT.2016.7507953
PRE-PRINT VERSION

A novel deep learning architecture for sentiment classification

Rahul Ghosh[3]
[3]Indian Institute of Technology
Guwahati-780139, India
rahul.ghosh@iitg.ac.in

Kumar Ravi[1,2]
[2]School of Computer & Information Sciences,
University of Hyderabad, Hyderabad-500046, India
kumar_ravi66@yahoo.co.in

Vadlamani Ravi[1,*]
[1]Center of Excellence in Analytics,
Institute for Development and Research in Banking Technology,
Castle Hills Road #1, Masab Tank, Hyderabad - 500057, India
rav_padma@yahoo.com

***Abstract*—Evolution of plethora of e-commerce sites resulted in fierce competition among their providers. In order to acquire new and retain existing customers, various producers and market managers effectively employ online feedback analytics tools. Most of the online feedback analysis tools are built using sentiment analysis models. Sentiment analysis evolved in the last one and half decades for review mining process. An important sub-task of sentiment analysis called sentiment classification is used mainly to decide whether a written review is expressing either positive or negative sentiment towards a target entity. In order to have better sentiment classification accuracy, we proposed a hybrid deep learning architecture, which is a hybrid of a two layered Restricted Boltzmann Machine and a Probabilistic Neural Network. The proposed approach yielded better accuracy for five different datasets compared to the state-of-the-art.**

Keywords—Sentiment analysis, Dimensionality reduction, Deep learning, Online learning, Restricted Boltzmann Machine, Probabilistic neural network.

I. Introduction

With the increasing number of online activities and commercial transactions, people are becoming more dependent on online reviews for purchasing decisions. Online reviews play a crucial role to decide the quality and performance of a product or service for potential customers as well as service providers. According to a survey, 79% of customers rely on online customer reviews as much as personal recommendations [1]. Furthermore, Word-of-Mouth (WOM) is more important than any other online media like Facebook, Twitter, etc [1]. Therefore, day-by-day online reviews are becoming more important than personal recommendations. In order to extract better insights from online reviews, research industries as well as academia are working rigorously towards sentiment analysis and opinion mining for the last two decades [2].

Sentiment analysis is used to perform the computational study about emotion, sentiment, opinion, and attitude towards an entity [2]. One of the most important tasks of sentiment analysis is sentiment classification (SC). SC is the process to decide polarity of sentiment expressed in a text. Out of various sentiment analysis tasks, SC is the one of the most challenging tasks due to contextual issues. The contextual issue refers to context based sentiment classification, where the polarity of an opinion changes with respect to different context. If we consider an example, "*The price of the camera is low*", here an adjective *low* has positive polarity. Now we consider another example "*The light of the flash is low*", then *low* has negative polarity. In order to address this kind of issue, we proposed a novel deep learning architecture for sentiment classification. The proposed deep learning architecture is a hybrid of a two layered Restricted Boltzmann Machine (RBM) [3,4] and a Probabilistic Neural Network (PNN) [5]. Restricted Boltzmann machine provides dimensionality reduction and Probabilistic Neural Network were used for sentiment classification respectively. The proposed approach yielded better accuracy for five different datasets compared to the state-of-the-art.

This rest of the paper is organized as follows: The literature survey is presented in Section II. Section III introduced the proposed architecture. The experimental setup is outlined in Section IV. Section V presents results and discussion. Section VI concluded the paper with some future directions.

II. Literature Review

Sentiment classification can be performed at word, aspect, phrase, concept, sentence, statement, and document-level [2]. This study is mainly based on document-level sentiment classification. Therefore, we reviewed related study based on document-level sentiment classification. Sentiment classification can be accomplished using supervised, unsupervised and semi-supervised based approaches [2, 6]. This study mainly focuses mainly on supervised based study; therefore, we reviewed supervised [7-15, 19-30] and semi-supervised [16-18, 35] study based literature.

Pang and Lee [7] employed Naïve Bayes, Maximum Entropy, and Support Vector Machine (SVM) for SC of movie reviews. McDonald et al. [8] developed a clique based sentiment classification model. They applied MIRA algorithm [9, 10] for sentiment classification. In another study, Dang et

[*] Corresponding Author: Phone: +91-4023294042; FAX: +91-40-23535157.

al. [13] employed information gain for feature subset selection and SVM for sentiment classification. They experimented with (a) digital camera and (b) dataset of Blitzer et al. [12]. Saleh et al. [14] employed SVM for classifying sentiment on (a) Pang and Lee [11] dataset, b) Taboada and Grieve [15] dataset, and c) SINAI corpus. Huynh et al. [16] trained stacked Convolutional Restricted Boltzmann Machine (CRBM) using word embeddings on the first one billion characters from Wikipedia[1] for subjectivity classification and movie review dataset[2] for sentiment classification. They reported subjectivity and sentiment classification accuracy of 78.3% and 78.7% on MPQA subjectivity lexicon and movie review (MR) [11] dataset respectively. Zhou et al. [17] proposed five layered (including input and output) Active Deep Networks (ADN) for sentiment classification. They experimented with Pang and Lee [11] and multi-domain [12] datasets. Severyn and Moschitti [18] employed single layer convolutional neural networks, which was extended with max pooling and a soft-max classification layer. They performed phrase level and message level sentiment classification of Semeval-2015 dataset and obtained an accuracy of 84.79%. Moraes et al. [19] employed balanced and imbalanced learning for four datasets viz. Pang and Lee [11], GPS, Books, and Cameras review dataset. The performances of both classifiers ANN and SVM deteriorated for the unbalanced dataset. Basari et al. [20] performed sentiment classification of movie reviews using a hybrid method of Particle Swarm Optimization (PSO) and SVM. Zhang et al. [21] obtained synonyms of some of the selected aspects using word2vec [22-24] and performed sentiment classification on 10,000 reviews on clothes using SVMperf [25]. They reported an accuracy of 89.95% and 90.30% using opinion words and Parts-of-Speech (POS) based feature selection. Ghiassi et al. [26] developed unsupervised feature selection using statistical analysis for sentiment classification of tweets. They employed a special architecture Artificial Neural Network (DAN2) for multi-class classification, which outperformed SVM. Fattah [27] developed several term weighting schemes for Bag-of-Word (BOW) representation of text documents for sentiment classification. Ravi et al. [28] applied χ^2-feature selection on mobile phone reviews. They proposed principal component analysis-extreme learning machine and evolving clustering method-extreme learning machine for sentiment classification. They also applied probabilistic neural network and recurrent neural network for online sentiment classification. Then, Vinodhini & Chandrasekaran [29] reduced dimensionality using principal component analysis for 500 (250 +ve & 250 -ve) reviews on the digital camera. Out of four classifiers viz. SVM, Naive Bayes, bagged SVM and Bayesian boosting, Bayesian boosting outperformed all other methods by yielding the highest precision of 83.3%. Ravi et al. [30] applied sentiment analysis in the educational sector in order to determine a correlation between qualitative and quantitative feedback obtained by online customers. Hajmohammadi et al. [35] performed cross-lingual sentiment classification by applying active learning and self-training in tandem. In order to reduce time and effort from manual labeling, they performed training on unlabelled samples from target language and labeled samples from source language.

III. Proposed Architecture

The proposed deep learning architecture is a hybrid of RBM and PNN as presented in Fig. 1. At the first step, RBM performs dimensionality reduction. At the next step, PNN performs sentiment classification. Both steps are presented in this section.

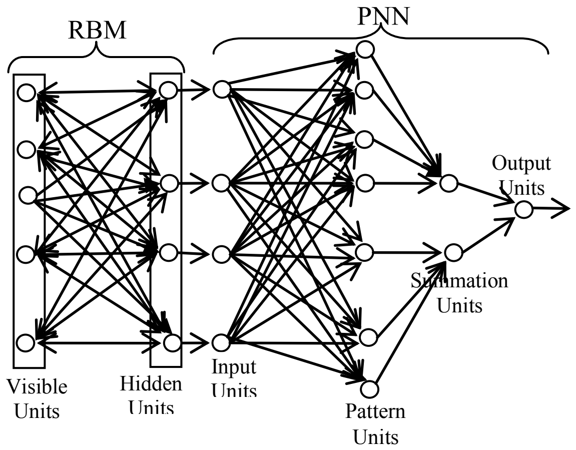

Fig. 1. The proposed hybrid architecture.

1) Dimensionality Reduction using Restricted Boltzmann Machine (RBM): Restricted Boltzmann Machine is a two layered generative model that represents a probability distribution [3]. Training an RBM essentially means learning the parameter values such that the RBM represents a probability distribution that fits the training data in the best possible way. After successful completion of the learning phase, the RBM provides a probabilistic distribution of the given data through the visible layer and the hidden layer. There are connections between the visible layer and the hidden layer but lacks the connection between the units that belong to the same layer. So, it forms a complete bipartite graph, where the visible units constitute of the components of an observation and the hidden layer represents the dependencies between various units of the visible layer (various features). Due to this functionality of RBM, it can be used to extract the non-linear features.

In an RBM with n binary visible and m hidden layer units, the joint probability distribution of P(v,h) is calculated using equation (1)

$$P(v,h) = \frac{1}{Z} exp\,(-E(v,h)) \tag{1}$$

where visible units, $v \in \{0,1\}^n$, hidden units, $h \in \{0,1\}^m$, and $E(v,h)$ is given by equation (2)

$$E(v,h) = -\sum_{i=1}^{n} a_i\, v_i - \sum_{j=1}^{m} b_j\, h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i\, w_{ij} h_j \tag{2}$$

and Z is a normalization factor given by

$$Z = \sum_{v,h} \exp\,(-E(v,h)). \tag{3}$$

[1] http://mattmahoney.net/dc/enwik9.zip.
[2] http://ai.stanford.edu/~amaas/data/sentiment/.

In equation (2), ai, and bi are biases for visible and hidden units respectively. The wij are weights between visible and hidden units.

a) Learning method: The parameters of RBM can be optimized with the use of stochastic gradient ascent on the log-likelihood of the training vector. Given the joint distribution, the probability assigned to the visible layer by the RBM can be calculated by summing over all possible hidden vectors using equation (4)

$$P(v) = \frac{1}{Z}\sum_h \exp\left(-E(v,h)\right). \qquad (4)$$

The derivative of the log of the probability function of the training vector is represented as

$$\frac{\partial \log P(v)}{\partial w_{ij}} = \langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{model} \qquad (5)$$

where $\langle v_i h_j\rangle_{data}$ and $\langle v_i h_j\rangle_{model}$ denote expectations under the distribution specified by $P\langle h|v\rangle$ and $P\langle v,h\rangle$, respectively. So, the stochastic gradient ascent is performed using equation (6),

$$\Delta w_{ij} = \epsilon(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{model}) \qquad (6)$$

where ϵ is the learning rate pre-specified by the user. Given a visible vector, the activation probabilities of the hidden units can be calculated using equation (7)

$$P\left(h_j = 1|v\right) = \sigma(b_j + \sum_{i=1}^{n} v_i w_{ij}) \qquad (7)$$

where j = 1, 2...., m and σ(.) is the sigmoid function. Similarly, given a hidden vector, the activation probabilities of the visible units can be calculated using equation (8)

$$P(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^{m} h_j w_{ij}) \qquad (8)$$

In order to speed up the learning process, momentum, θ, can be set using equation (9)

$$\Delta\theta_i(t) = x_i(t) = \alpha x_i(t-1) - \epsilon\frac{dE}{d\theta_i}(t). \qquad (9)$$

Here, x(t) is the velocity at the time t and for the constant gradient, velocity will exceed $\epsilon\frac{dE}{d\theta_i}$ by a factor of 1/(1- α).

b) Contrastive Divergence Algorithm: Contrastive Divergence (CD) algorithm, developed by Hinton in 2002, is a fast learning algorithm which initiates by setting the visible unit as the training data [4]. Then, the hidden units are computed using eq (7). Once the binary states of the hidden layer have been determined, the visible units are reconstructed from these hidden units using eq (8). The updation of the weight is performed using equation (10)

$$\Delta w_{ij} = \in(\langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{recon}) \qquad (10)$$

where $\langle v_i h_j\rangle_{recon}$ is the distribution that is obtained by running Gibbs sampling for one step. Pseudo code for RBM is presented in Fig. 2. The given algorithm can be extended for Gaussian visible units as proposed by Hinton [3]. We only need to change the way we compute the activation probabilities. For a given hidden vector, the activation probabilities of the visible units can be calculated using equation (11)

$$P(v_i|h) = \mathcal{N}(a_i + \sum_{j=1}^{m} h_j w_{ij}, 1) \qquad (11)$$

where $\mathcal{N}(x,1)$ is a Gaussian distribution with mean x and variance 1.

```
Procedure
   Initialize the weight matrix W, bias vectors
   a and b, momentum v.
   Set the states of visible unit $v_1$ as the
   training vector
   While i < Max_Iter
     For j = 1, 2 …, m (all hidden units)
        Compute $P(h_{1j} = 1|v_1)$ using equation (7)
        Gibbs Sampling $h_{1j} \in \{0,1\}$ from $P(h_{1j}|v_1)$
     End For
     For i = 1, 2 …, n (all visible units)
        Compute $P(v_{2i} = 1|h_1)$ using equation (8)
        Gibbs Sampling $v_{2i} \in \{0,1\}$ from $P(v_{2i}|h_1)$
     End For
     For j = 1, 2 …, m (all hidden units)
        Compute $P(h_{2j} = 1|v_2)$ using equation (7)
     End For
        //Update rule:
        W := W + $\in (P(h_1 = 1|v_1)\, v_1^T - P(h_2 = 1|v_2) v_2^T)$
        a := a + $\in (v_1 - v_2)$
        b := b + $\in (P(h_1 = 1|v_1) - P(h_2 = 1|v_2))$
        x := updation of momentum;
   End While
End Procedure
```

Fig. 2. Pseudo-code for RBM's fast learning algorithm based on CD.

B. Sentiment Classification

The online learning is the neural network learning process, which trains the network in only one epoch. Therefore, learning process is used to be very much faster than other type of learning like semi-online and offline [28].

1) Probabilistic neural network: The probabilistic neural network is a parallel neural network architecture, which provides online learning [5, 28]. PNN is mainly based on non-parametric estimations of probability density functions and bayesian decision strategy. It has four layers involving input, pattern, summation, and output layer. For an input dataset of dimension *n*, and number of samples *m*, the PNN will have *n* input inputs and *m* patterns units, where dataset should be normalized in unit length. For an input pattern vector *X* and weight vector W_i, each pattern unit calculates $Z_i = X.W_i$ at the first stage. At the next stage, pattern unit applies activation function on Z_i like $z_i' = \exp[(Z_i - 1)/\sigma^2]$. z_i' will be passed to summation layer, which contains only two units, called summation units. Each of summation units performs summation on samples received from previous layer on the basis of training pattern category. The output unit decides the category of each sample on the basis of a single variable weight, C_k, using equation (12).

$$C_k = -\frac{h_{B_k} I_{B_k}}{h_{A_k} I_{A_k}} . \frac{n_{A_k}}{n_{B_k}} \qquad (12)$$

where h_A and h_B represents prior probabilities of class A and B; I_A and I_B are the misclassification function; n_{A_k} and n_{B_k}

represents number of training patterns from category A_k and B_k respectively.

C. The time complexity of the proposed architecture

For an input dataset of dimension n and number of samples m, the time complexity of the restricted Boltzmann machine will be $O(I*B*\max(n, h)) \approx O(I*B*n)$. Here, I is the number of iterations, B is the batch size, and h is the number of hidden units. Furthermore, the dimension sizebeing reduced will be equal to the number of hidden units. The time complexity of PNN will be $O(m)$. Therefore, the total time complexity of the proposed architecture will be be $O(I*B*n) + O(m) \approx O(I*B*n)$.

IV. Experimental Setup

The whole experiments were performed into four steps namely data collection, text preprocessing and dimensionality reduction, online sentiment classification, and reporting results. The whole approach is depicted using a workflow diagram in Fig. 2. Results and discussion are presented in Section V.

A. Data Collection

We experimented with Pang and Lee [11] dataset (MOV) and Blitzer et al. [12] multi-domain dataset. The former dataset contains 1000 positive and 1000 negative movie reviews. The latter dataset contains 1000 positive and 1000 negative reviews on each of Books (BOO), DVDs, Electronics (ELE), and Kitchen appliances (KIT). In all these datasets, a review with more than 3 stars as user ratings was considered as a positive review, user ratings with less than 3 stars were considered as a negative review, and user ratings with 3 stars were considered as neutral. Here, a positive review implies that a review is expressing positive sentiment for a product or a movie. Similarly, a negative review implies that a review is expressing negative sentiment for a product or a movie. The neutral reviews were not considered for the experiment, hence discarded. For classification purpose, all positive reviews were considered under positive class. And, all negative reviews were considered under negative class. Hence, binary sentiment classification is performed on all datasets.

B. Text Preprocessing

In order to apply any supervised learning on unstructured data, we need to convert them into structured data e.g. Document-Term Matrix (DTM). DTM is a kind of BOW model, which represents a text corpus as a combination of rows and columns. Here, each row represents a text document and a column represents each distinct word as a feature. Each document was represented using the Term Frequency-Inverse Document Frequency (TF-IDF) representation. TF-IDF provides relative importance of each feature with respect to all documents, therefore it yields better performance. In order to prepare the DTM, we performed tokenization, stop word removal, and stemming. Tokenization is the process to break a text into relevant chunks of text like unigrams, bigrams, etc. The `StringToWordVector` filter is available in Weka 3.7.12, which was used to perform tokenization and prepare DTM [31]. Tokenization was performed on the basis of the appearance of punctuation marks and white spaces, where we specified punctuation marks and whitespaces as an input to `StringToWordVector` filter. Furthermore, the whitespaces included a blank space, a tab space, and a carriage return. We removed stop words, which appears very often in whole text and doesn't contribute to the analysis. A list of 571 stop words were supplied as an input to Weka, which were collected using "`SMART`" option of `stopwords()` command of R language [32]. Stemming was performed using Snowball stemmer [33], which brought words into its root form. As for example, *happy, happier,* and *happiest* were brought into its root form *happi*. The TFTransform and IDFTransform option of WEKA were enabled to obtain TF-IDF values.

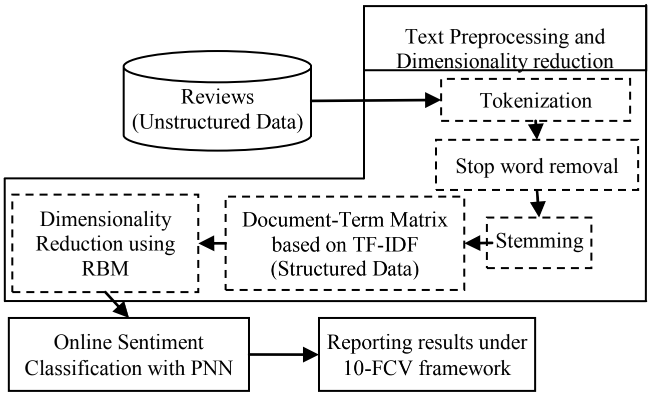
Text Preprocessing and Dimensionality reduction
Reviews (Unstructured Data)
Tokenization
Stop word removal
Stemming
Document-Term Matrix based on TF-IDF (Structured Data)
Dimensionality Reduction using RBM
Online Sentiment Classification with PNN
Reporting results under 10-FCV framework

Fig. 3. The experimental methodology.

This process yielded a number of features of 33435, 5942, 5242, 7083, and 6290 for MOV, BOO, DVD, ELE, and KIT dataset respectively. The *max-min normalization* was applied to normalize DTM in the unit length according to prior requirement of PNN.

1) Dimensionality Reduction: We performed feature space dimensionality reduction for each of the dataset and reduced to 50, 60, 70, 80, and 100 dimensions. Let us consider the movies dataset. Initially, its feature space dimension was 33435, which was reduced each of them to 50, 60, 70, 80, and 100 using RBM to conduct five different experiments. We set step size, a total number of iterations, batch size, and momentum rate as 0.09, 4000, 50, and 0.9 respectively. Here, step size and batch size refers to learning rate and a number of samples per batch respectively. So, the number of batches will be 2000/50 = 40 and the number of epochs will be 4000/40 = 100.

C. Sentiment Classification

We employed PNN for binary sentiment classification, which is available in NeuroShell 2.0 [34]. Out of available various scaling function, PNN yielded the best accuracy using *logistic*. The *smoothing factor* for each dataset was learned automatically using *genetic calibration,* where a number of generations were set as 100. Out of two available distance metrics, the *city block distance metric* yielded better performance than *vanilla distance*. PNN has been trained for approximately 25 minutes for each fold to have better convergence and smoothing factor.

V. Results and Discussion

In order to report the performance of the classifier, we used performance measures viz. accuracy (ACC), sensitivity (SEN), and specificity (SPE). Accuracy, sensitivity, and specificity are calculated using equation (13), (14), and (15) respectively.

$$Accuracy = \frac{(tp + tn)}{(\mathrm{tp + fn + tn + fp})} \quad (13)$$

$$Sensitivity = \frac{tp}{(tp+fn)} \quad (14)$$

$$Specificity = \frac{tn}{(tn+fp)} \quad (15)$$

where *tp, tn, fn,* and *fp* represent true positive, true negative, false negative, and false positive respectively. The sentiment classification result for an average of 10-FCV accuracy is presented in Table I.

TABLE I. The Average Results of 10-FCV

Dimension size and momentum		MOV[a]	BOO[b]	DVD[c]	ELE[d]	KIT[e]
50; 0.9	ACC	0.801	0.767	0.79	0.7945	0.793
	SEN	**0.933**	0.631	0.645	0.67	0.67
	SPE	0.669	**0.903**	**0.935**	**0.919**	**0.919**
60; 0.9	ACC	**0.808**	0.802	**0.811**	**0.801**	**0.802**
	SEN	0.929	**0.927**	**0.931**	0.94	0.927
	SPE	**0.687**	0.677	0.691	0.662	0.677
60; 0.05	ACC	0.7905	**0.81**	0.780	0.7525	0.7826
	SEN	0.908	0.915	0.901	**0.949**	**0.932**
	SPE	0.673	0.705	0.66	0.556	0.633

[a]Movies, [b]Books, [c]DVD, [d]Electronics, [e]Kitchen.

As we mentioned earlier, we experimented with five reduced dimensions of each dataset. Out of five reduced dimensions, we obtained the best accuracy for 60 feature dimensions for all datasets. We experimented with different step size with a variation of 0.003 from 0.001 to 0.009, and achieved the best result for 0.009. Similarly, we varied momentum with a variation of 0.04 from 0.01 to 1.2, and achieved the best result for 0.9 for all dataset except books dataset. We obtained better sensitivity than specificity using 60 feature dimensions for all datasets. And, we obtained the better specificity than sensitivity using 50 feature dimensions for all datasets except movies dataset.

Even though our results are not strictly comparable to any other study, we compared our results with that of Dang et al. [13]. Since, they experimented with the dataset of Blitzer et al. [12] using content free (F1), content-specific (F2), and sentiment (F3) features. Among them, F1 contains lexical, syntactic, and structured features. Further, F2 contains unigrams and bigrams. And, F3 is generated using sentiment and semantic-oriented approaches. The statistics of features generated by Dang et al. [13] is presented in Table II. The average of 10-FCV accuracy results of Dang et al. [13] is presented in Table III. Compared to Dang et al. [13], our proposed approach outperformed all six experiments performed on Books and DVD datasets. For electronics dataset, our proposed approach outperformed the first and third experiments. Similarly, for kitchen appliances, our proposed approach outperformed the first and third experiments. If we consider the time complexity aspects, the time complexity for feature selection method of Dang et al. [13] depends on the process of generation of unigrams, bigrams, POS tagging, and the length of a sentiment lexicon. While in our case, the time complexity depends on the process of generation of unigrams and the time complexity of RBM. Here, the generation of unigrams is the process to obtain a number of features/terms of DTM. So, our proposed approach is much faster than its counterpart and doesn't depend on POS tagging and other resource like a sentiment lexicon. Furthermore, due to online learning and less number of features, classification will be very much faster than that of Dang et al. [13]. Since, online learning performs classification in a single iteration. Moreover, it is obvious that a classifier will consume more time with a higher number of features. From Table II, we can see that Dang et al. [13] needed ten times or more number of features compared to ours. And, our proposed approach outperformed for specified number of experiments using only 60 features. Therefore, computational time for classification will be quite minimal compared to that of Dang et al. [13]. Finally, the benefits of our proposed approach over existing study can be summarized as follows:

- The proposed approach doesn't depend on outer resource like a sentiment dictionary. The size of sentiment dictionary will affect the time complexity of the existing approach.
- In our proposed approach, we didn't have to perform POS tagging unlike Dang et al. [13], which is very time consuming process.
- The numbers of dimensions, which yielded the best performances, are 50 and 60. So, the size of the dataset will be much smaller than that of Dang et al. [13], hence reduces time complexity of classification.
- In our case, the online learning also helps in reducing the time complexity of classification.

TABLE II. The Features Statistics of Dang et al. [13]

Feature Set	BOO	DVD	ELE	KIT
F1	250	250	250	250
F1 + F2	8,138	7,992	5,282	4,670
F1 + F3	1,141	1,090	641	582
F1 + F2 + F3	9,029	8,832	5,673	5,002
Selected F1 + F2	524–555	597–668	612–670	545–592
Selected F1 + F2 + F3	567–606	647–724	658–715	583–627

TABLE III. Results Comparison using Accuracy

Sl. No.		BOO	DVD	ELE	KIT
1.	**F1**	70.55	65.45	69.85	69.95
2.	**F1 + F2**	75.3	78	80.65	82.4
3.	**F1 + F3**	75.55	72	76.35	77.85
4.	**F1 + F2 + F3**	76.95	78.4	80.9	83.3
5.	**Selected F1 + F2**	77.45	80.45	82.8	83.4
6.	**Selected F1 + F2 + F3**	78.85	80.75	**83.75**	**84.15**
7.	**Our Results (RBM+PNN)**	**81.0**	**81.1**	80.1	80.2

In terms of applicability of the sentiment classification, it can be applied for customer churn prediction. In order to prevent customers from churning, a service or manufacturing industry should consider reviews containing negative sentiment content. In the most of the cases, a review bearing negative sentiment score contains customer complaints regarding a service/ product and its aspects. In this case, identification of negative reviews is quite more important than identification of positive reviews. Hence, the sensitivity should be reported for sentiment classification results in this case. The proposed method yielded the best sensitivity of 93.3%, 92.7%, 93.1%, 94.9%, and 93.2% for MOV, BOO, DVD, ELE, and KIT respectively.

VI. Conclusions and Future Directions

We proposed a hybrid deep learning and online learning based architecture involving RBM and PNN in succession to perform fast sentiment classification, which is not dependent on outer resource like a sentiment dictionary, a POS tagger, etc. We experimented with five different datasets and compared our results with the state-of-the-art. For two datasets, the proposed approach outperformed existing study for all six experiments. For other two datasets, our approach outperformed for two experiments. In terms of feature selection, existing study followed very complex approach to select relevant features. Whereas, we employed dimensionality reduction in order to obtain less number of dimensions compared to existing study. We foresee future research directions for sentiment analysis in three folds. First, an advanced architecture should be developed to perform sentiment classification with less number of labeled data. Second, due to a lack of labeled data, cross-domain sentiment classification should be developed at the finer level. Third, more advanced hybrid techniques need to be developed to achieve better accuracy for sentiment classification.

References

[1] https://www.brightlocal.com/wp-content/uploads/2013/06/Local-Consuner-Review-Survey-20131.pdf. Accessed on 3rd September, 2015.

[2] K. Ravi, V. Ravi, A survey on opinion mining and sentiment analysis: Tasks, approaches and applications, Knowledge-Based Systems, 89 (2015) 14–46.

[3] G.E. Hinton, A practical guide to training restricted Boltzmann machines, Momentum 9, no. 1 (2010): 926.

[4] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural computation 14, no. 8 (2002): 1771-1800.

[5] D.F. Specht, Probabilistic neural networs, Neural networks, Vol. 3, pp. 109-118, 1990.

[6] B. Liu, Sentiment analysis and opinion mining, Morgan and Claypool publishers, May 2012.

[7] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, Proceedings of the ACL-02 conference on empirical methods in natural language processing (Vol. 10, pp. 79–86). Association for Computational Linguistics, 2002.

[8] R. McDonald et al., Structured models for fine-to-coarse sentiment analysis, Annual Meeting-Association For Computational Linguistics, Vol. 45. No. 1. 2007.

[9] R. McDonald, K. Crammer, F. Pereira, Online large-margin training of dependency parsers. In Proc. ACL, 2005.

[10] K. Crammer, Y. Singer, Ultraconservative online algorithms for multiclass problems, JMLR, 2003.

[11] B. Pang, L. Lee, A sentiment education: Sentiment analysis using subjectivity summarization based on minimum cuts, in: Proceedings of the 42nd annual meeting on Association for Computational Linguistics (p. 271), 2004, July.

[12] J. Blitzer, M. Dredze, F. Pereira, Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, In ACL, vol. 7, pp. 440-447. 2007.

[13] Y. Dang, Y. Zhang, H. Chen, A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews, Sentiment Classification, IEEE Intelligent Systems, July/August 2010.

[14] M.R. Saleh, M.T. Martín-Valdivia, A. Montejo-Ráez, L.A. Ureña-López, Experiments with SVM to classify opinions in different domains, Expert Systems with Applications 38 (2011) 14799–14804.

[15] M. Taboada, J. Grieve, Analyzing appraisal automatically, In Proceedings of the AAAI spring symposium on exploring attitude and affect in text: Theories and applications, 2004, (pp. 158–161).

[16] T. Huynh, Y. He, and S. Rüger, Learning Higher-Level Features with Convolutional Restricted Boltzmann Machines for Sentiment Analysis, In: Advances in Information Retrieval, pp. 447-452. Springer International Publishing, 2015.

[17] S. Zhou, Q. Chen, X. Wang, Active deep networks for semi-supervised sentiment classification, In: International Conference on Computational Linguistics, Coling 2010 Organizing Committee, Beijing, China, 2010, pp. 1515– 1523.

[18] A. Severyn and A. Moschitti, Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pages 959–962.

[19] R. Moraes, J.F. Valiati, W.P. Gaviao Neto, Document-level sentiment classification: An empirical comparison between SVM and ANN, Expert Systems with Applications 40 (2013) 621–633.

[20] Abd. S.H. Basari, B. Hussin, I.G.P. Ananta, J. Zeniarja, Opinion Mining of Movie Review using Hybrid Method of Support Vector Machine and Particle Swarm Optimization, Procedia Engineering 53 (2013) 453 – 462.

[21] D. Zhang, H. Xu, Z. Su, Y. Xu, Chinese comments sentiment classification based on word2vec and SVM perf, Expert Systems with Applications 42, no. 4 (2015): 1857-1863.

[22] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[23] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, In Advances in neural information processing systems, pp. 3111-3119. 2013.

[24] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic Regularities in Continuous Space Word Representations, In HLT-NAACL, pp. 746-751, 2013.

[25] T. Joachims, C.-N.J. Yu, Sparse kernel SVMs via cutting-plane training, Machine Learning 76, no. 2-3 (2009): 179-193.

[26] M. Ghiassi, J. Skinner, D. Zimbra, Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network, Expert Systems with Applications 40 (2013) 6266–6282.

[27] M. Abdel Fattah, New term weighting schemes with combination of multiple classifiers for sentiment analysis, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.04.051.

[28] K. Ravi, V. Ravi, C. Gautam, Online and semi-online sentiment classification, International Conference on Computing, Communication & Automation (ICCCA), pp. 938 – 943, (2015) http://dx.doi.org/10.1109/CCAA.2015.7148531.

[29] G. Vinodhini, R. M. Chandrasekaran, Opinion mining using principal component analysis based ensemble model for e-commerce application, CSI Transactions on ICT (2014): 1-11.

[30] K. Ravi, V. Ravi, V. Siddeshwar, L. Mohan, Sentiment analysis applied to Educational Sector, 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC2015), pp. 117-122, (2015).

[31] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1, (2009).

[32] I. Feinerer, An introduction to text mining in R, R News, 8(2):19{22, Oct. 2008. URL http://CRAN.R-project.org/doc/Rnews/.

[33] http://snowball.tartarus.org/index.php.

[34] Neuroshell: http://www.neuroshell.com/.

[35] M.S. Hajmohammadi et al., Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples, Inform. Sci. (2015), http://dx.doi.org/10.1016/j.ins.2015.04.003.