Sentiment Analysis of Customer Reviews Using Robust Hierarchical Bidirectional Recurrent Neural Network

Chapter · January 2016

DOI: 10.1007/978-3-319-33625-1_23

2 authors:

Arindam Chaudhuri
Google Tokyo

Soumya K. Ghosh
Indian Institute of Technology Kharagpur

Sentiment Analysis of Customer Reviews Using Robust Hierarchical Bidirectional Recurrent Neural Network

Arindam Chaudhuri and Soumya K. Ghosh

Abstract With tremendous growth of online content, sentiment analysis of customer reviews has become an active research topic for machine learning community. However, due to variety of products being reviewed online traditional methods do not give desirable results. As number of reviews expand, it is essential to develop robust sentiment analysis model capable of extracting product aspects and determine sentiments adhering to various accuracy measures. Here, hierarchical bidirectional recurrent neural network (HBRNN) is developed in order to characterize sentiment specific aspects in review data available at DBS Text Mining Challenge. HBRNN predicts aspect sentiments vector at review level. HBRNN is optimized by fine tuning different network parameters and compared with methods like long short term memory (LSTM) and bidirectional LSTM (BLSTM). The methods are evaluated with highly skewed data. All models are evaluated using precision, recall and $F1$ scores. The results on experimental dataset indicate superiority of HBRNN over other techniques.

Keywords Semantic analysis · Customer reviews · RNN · BRNN · HBRNN

1 Introduction

In the present competitive business scenario vast amount of consumer reviews are written on Web about any product or service [1]. WWW contains an overwhelming volume of customer reviews [2] about different categories of commodities avail-

A. Chaudhuri (✉)
Samsung R & D Institute Delhi, Noida 201304, India
e-mail: arindam_chau@yahoo.co.in

S.K. Ghosh
Department of Computer Science Engineering, Indian Institute of Technology, Kharagpur 721302, India
e-mail: skg@iitkgp.ac.in

R. Silhavy et al. (eds.), *Artificial Intelligence Perspectives in Intelligent Systems*, Advances in Intelligent Systems and Computing 464,
DOI 10.1007/978-3-319-33625-1_23

able. An appreciable number of websites, blogs and forums allow customers to post opinions about products or services. They describe general sentiment of customer towards the product in detail [3]. The aggregated aspect level sentiment analysis is valuable information source when a company is introducing new product and wants to create hype. Carefully managing sentiment of potential customers is paramount to succeeding in creating buzz for new product. For products that already exist, this detailed information extracted from customer reviews is useful to improve quality of service or product. The customer reviews are thus essential to potential customers, retailers and manufacturers in their efforts to understand general opinions of customers and making better decisions. However, as number of reviews expand it becomes difficult for users to obtain comprehensive view of opinions of customers about various aspects manually. Consequently proper analysis and summarization of reviews are required to enable potential users to visualize opinions about specific features of products. Thus, it is highly desirable to develop a robust sentiment analysis tool capable of performing sentiment analysis for reviews considering various accuracy measures.

Since the past decade sentiment analysis for online customer reviews has attracted attention from researchers of machine learning domain [4]. The fundamental problem in here revolves around aspect detection [5]. Aspects are entities on which opinions are expressed. They are important because without knowing them opinions expressed in review are of limited use. The aspect detection is critical to sentiment analysis because its effectiveness affects performance of opinion word detection and sentiment orientation identification. Product reviews have always influenced customers' more than website information [6]. Investigating this relation between company and consumer generated information helps to improve company sales [7, 8]. Opinions stated on Web have become resource for companies. However, in order to achieve fine grained information for analyses, various aspects of product must be first recognized in text. Several methods have been proposed in product review mining. This involves broad range of fields from document to aspect level sentiment analysis for different reviews. Some of the notable research in recent past includes works by [8–12].

In this paper, robust hierarchical bidirectional recurrent neural network (HBRNN) is proposed for semantic analysis of DBS Text Mining Challenge 2015 data [13] which contains customer reviews of different hotels. HBRNN takes full advantage of deep recurrent neural network (RNN) towards modelling long-term contextual information of temporal sequences in data. The prediction of aspect sentiments vector is done by HBRNN at review level. The performance of HBRNN is improved by fine tuning parameters of network. It is compared with other methods such as long short term memory (LSTM) and bidirectional LSTM (BLSTM). The experiments are performed and evaluation is done on highly biased data. The aspect information content is increased through mini-batch sampling. The models are evaluated using precision, recall and $F1$ scores. The promising results on experimental dataset indicate superiority of HBRNN over other methods. This paper is organized as follows. In Sect. 2 computational method of HBRNN is highlighted. This is followed by experiments and results in Sect. 3. Finally in Sect. 4 conclusions are given.

2 Computational Method

In this section mathematical framework of proposed HBRNN model [14] is presented.

2.1 Problem Description

The customer reviews of different hotels spread across the world are considered to extract entity level sentiments. The hotel reviews are analyzed by (a) extracting most important features of hotel and (b) assigning an overall score for each of them. This allows us to structure information from reviews by summarizing them in a comprehensive and concise form. The problem can thus be formulated as: Given a review as form of sentence s_i, the sentiment scores $ss_{a,i}$ of relevant features or aspects a are to be identified.

2.2 Datasets

The experimental data is taken from DBS Text Mining Challenge 2015 which consists of reviews of 1500 hotels [13]. Each hotel review is contained in separate file which contains hotel's name and identification. The content tag contains an individual review. The review is followed by date tag. Each of these sentences in dataset has reviewed entity and level of sentiment involved. After performing experiments with dataset more than 7000 reviews are labelled manually. The total dataset contained 150,175 labelled reviews with 7 aspects viz comfort, excellent, hospitality, delicious, superb, cheap, features. Aspect sentiments are labelled on scale from $-7, \ldots, 7$. Here only aspect sentiments on scale from $-1, \ldots, 1: s_i = -1, 0, 1$ are considered. The data is pre-processed to fit aspect mentioned in aspect buckets.

2.3 Recurrent Neural Network and Long Short Term Memory Neuron

RNNs are deep learning artificial neural networks (ANNs) [15] where connections between different computational units form directed cycle. This creates an internal network state that exhibits its dynamic temporal behavior. RNNs use internal memory to process arbitrary input sequences. This makes them suitable for non-segmented handwriting recognition tasks. RNNs are more efficient than traditional ANNs and support vector machines (SVM) [14, 15] because they can be trained in either supervised or unsupervised manner. The network learns something

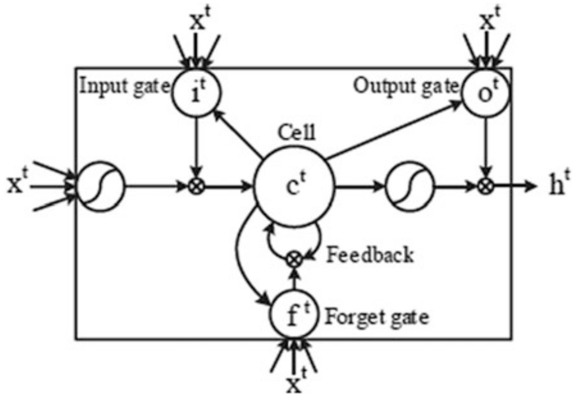

Fig. 1 Long short term memory block with one cell

intrinsic about data without help of target vector and is stored as network weights. The unsupervised training in network has identical input as target units. In deep learning optimization routine applied to network architecture itself. The network is directed graph where each hidden unit is connected to other hidden units. Each hidden layer going further into network is non-linear combination of layers because of combination of outputs from all previous units' with their activation functions. When optimization routine is applied to network, each hidden layer becomes optimally weighted and non-linear layer. When each sequential hidden layer has fewer units than one below it then each hidden layer becomes low dimensional projection of layer below it. With recurrent structure, RNN models contextual information of temporal sequence. Generally it is very difficult to train RNNs with commonly used activation functions due to vanishing gradient and error blowing up problems [15]. To solve this LSTM architecture is used [14] which replaces non-linear units in traditional RNNs. Figure 1 illustrates LSTM memory block with single cell. It contains one self-connected memory cell and three multiplicative units viz input gate, forget gate and output gate which can store and access long range contextual information of temporal sequence. The activations of memory cell and three gates are available in [15]. In order to utilize past and future context, BRNN is used through forward and backward sequence [15] to two separate recurrent hidden layers. These two recurrent hidden layers share same output layer.

2.4 *Bidirectional Recurrent Neural Network for Semantic Analysis*

Here BRNN [14] is evaluated in terms of RNN which is used to develop HBRNN. Instead of providing output for each word, the model gives only outputs as final prediction at end of sentence. To capture the entire context, backpropagation-through-time parameter is selected so that it exceeds sentence length. The customer reviews from DBS Text Mining Challenge dataset is expressed through 150,175 labelled reviews with 7 aspects such as comfort, excellent, hospitality, delicious, superb, cheap, features [14]. For each of these 7 aspects, there is −1, 0, 1 so that there is one-hot vector of 3 elements for each one; −1 (most negative), 1 (most positive) and 0 (neutral). If review does not mention an aspect it is assumed neutral. For 7 different aspects prediction is $\hat{z} \in R^{21}$. Considering $y^{(1)}, y^{(2)}, \ldots$ forward propagation is:

$$\boldsymbol{p}^{(t)} = \sigma\left(\boldsymbol{W}_p \boldsymbol{p}^{(t-1)} + \boldsymbol{W}_y \boldsymbol{y}^{(t)}\right) \tag{1}$$

The final output for each aspect results in:

$$\hat{z} = \boldsymbol{softmax}\left(\boldsymbol{W}_s \boldsymbol{h}^{t=T}\right) \tag{2}$$

Here $\hat{z}$ is concatenation of single predictions for each aspect of product:

$$\hat{z} = \begin{pmatrix} \hat{z}_1 & \hat{z}_2 & \hat{z}_3 & \hat{z}_4 & \hat{z}_5 & \hat{z}_6 & \hat{z}_7 \end{pmatrix}^{\mathrm{T}} \tag{3}$$

The sentiment is calculated at end. The matrices W_y, W_p, W_s and M word vectors are required to be learned. The idea behind this structure is that RNNs accumulate the sentiment over whole sentence. Post word context is not considered as sentence is observed only in one direction. In order to determine aspect sentiment BRNN is used. In BRNN accumulation task is performed in two directions which allow more flexibility. The model runs through sequence in reverse order with different set of parameters that is updated. In order to specify backward channel sequence of words are inverted and the same RNN is performed as done before on other direction. The final output is calculated concatenating p_g and p_h from both directions:

$$\boldsymbol{p}_g^{(t)} = \sigma\left(\boldsymbol{W}_{p_g} \boldsymbol{p}_g^{(t-1)} + \boldsymbol{W}_y \boldsymbol{y}^{(t)}\right) \tag{4}$$

$$\boldsymbol{p}_h^{(t)} = \sigma\left(\boldsymbol{W}_{p_h p_h^{(t-1)}} + \boldsymbol{W}_y \boldsymbol{y}_{inverted}^{(t)}\right) \tag{5}$$

$$\hat{z} = \boldsymbol{softmax}\left(\boldsymbol{W}_{s,brnn} \begin{pmatrix} \boldsymbol{p}_g \\ \boldsymbol{p}_h \end{pmatrix} + \boldsymbol{b}_s\right) \tag{6}$$

In order to capture aspects context in more granular way LSTM version of RNN is deployed here. Instead of just scanning word sequence in order the model stores information in gated units in an input gate $i^{(t)}$ with weight on current cell, a forget gate $f^{(t)}$, an output gate $o^{(t)}$ to specify relevance of current cell content and new memory cell $\tilde{cc}^{(t)}$. For time series tasks of unknown length LSTM are capable of storing and forgetting information better than their counterparts [14, 16].

$$\boldsymbol{i}^{(t)} = \sigma\left(\boldsymbol{W_i y^t} + \boldsymbol{V_p p}^{(t-1)}\right) \tag{7}$$

$$\boldsymbol{f}^{(t)} = \sigma\left(\boldsymbol{W_f y^t} + \boldsymbol{V_f p}^{(t-1)}\right) \tag{8}$$

$$\boldsymbol{o}^{(t)} = \sigma\left(\boldsymbol{W_o y^t} + \boldsymbol{V_o p}^{(t-1)}\right) \tag{9}$$

$$\widetilde{\boldsymbol{cc}}^{(t)} = \boldsymbol{tanh}\left(\boldsymbol{W_{cc} y^t} + \boldsymbol{V_{cc} p}^{(t-1)}\right) \tag{10}$$

$$\boldsymbol{cc}^{(t)} = \boldsymbol{f}^{(t)}\boldsymbol{cc}^{(t-1)} + \boldsymbol{i}^{(t)}\widetilde{\boldsymbol{cc}}^{(t)} \tag{11}$$

$$\boldsymbol{p}^{(t)} = \boldsymbol{ot}^{(t)}\boldsymbol{tanh}\left(\boldsymbol{cc}^{(t)}\right) \tag{12}$$

Here, $\boldsymbol{fs}^{(t)}$ and $\boldsymbol{hv}^{(t)}$ are final and hidden vectors. The prediction now becomes:

$$\hat{z} = softmax(W_z p + b_z) \tag{13}$$

The model is implemented using MATLAB. The bidirectional LSTM version of RNN scans sequence of words in reverse order using second set of parameters. The final output is concatenation of final hidden vectors from original and reversed sequence:

$$\hat{\boldsymbol{z}} = \boldsymbol{softmax}\left(\boldsymbol{W_z}\begin{pmatrix}\boldsymbol{p_g^T}\\ \boldsymbol{p_h^T}\end{pmatrix} + \boldsymbol{b_z}\right) \tag{14}$$

The standard version of RNN performs below expectations as most reviews do not contain detectable aspects with positive or negative sentiment. Prior distribution of dataset is biased towards 0 class (neutral class). The model tends to always predict 0 and is not capable to predict –1 or 1.

2.5 Hierarchical Bidirectional Recurrent Neural Network for Semantic Analysis

The hierarchical version of BRNN viz HBRNN [14] for semantic analysis of review data is proposed here in terms of BRNN. The computational benefits received from BRNN [14, 15] serve the major motivation. HBRNN is different from BRNN in terms of efficient classification accuracy based on similarities and running time when volume of data grows [14]. The architecture of proposed model is shown in Fig. 2 where temporal sequences in review are modeled by BRNNs which are combined together to form HBRNN. The model is composed of 6 layers viz $br_1 - br_2 - br_3 - br_4 - fc - sm$. Here, $br_i; i = 1, 2, 3, 4$ denote layers with BRNN

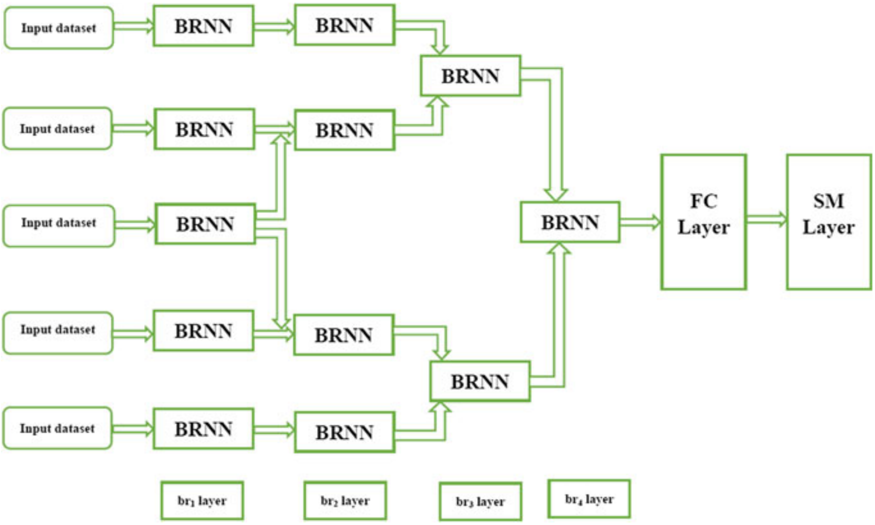

Fig. 2 The architecture of proposed HBRNN model

nodes, *fc* denotes fully connected layer and *sm* denotes softmax layer. In HBRNN each layer takes care of classification tasks [14] and plays a vital role in success of whole network. Each layer constitutes hierarchy of classifier. To recover any single hierarchy split BRNN is run on small subset of review data comprising of few words [14] to compute seed classification value. The subset of input dataset is produced randomly. This activity starts at layer br_1. Using initial classification value, remaining data is placed into seed class for which it is most similar on average. This results in classification of entire dataset using only similarities to words in small subset. By recursively applying this procedure to each class HBRNN is obtained using small fraction of similarities. The classification task proceeds till br_4. In this recursive phase no measurements are observed between classes at previous split. This results in robust HBRNN that aligns its measurements *mt* to resolve higher resolution in the class structure. The pseudo code for HBRNN is shown in Algorithm 1.

Algorithm 1: HBRNN$\left(\text{BRNN}, mt, \{y_i\}_{i=1}^{Wr_j}, Cs_j\right)$

if $Wr_j < mt$ **then return** $\{y_i\}_{i=1}^{Wr_j}$

Select $V \subseteq \{y_i\}_{i=1}^{Wr_j}$ of size v uniformly at random

$C_1^{'}, \ldots, C_{Cs_j}^{'} \leftarrow BRNN\left(V, Cs_j\right)$

Set $C_1 \leftarrow C_1^{'}, \ldots, C_{Wr_j} \leftarrow C_{Wr_j}^{'}$

for $y_i \in \{y_i\}_{i=1}^{Wr_j} \backslash V$ **do**

$\forall k \in [Cs_j],\ \alpha_k \leftarrow \frac{1}{|C_j'|} \sum_{y_s \in C_j'} S(y_i, y_s)$

$C_{\text{argmax}_{k \in [Cs_j]}} \alpha_k \leftarrow C_{\text{argmax}_{k \in [Cs_j]}} \alpha_i \cup \{y_i\}$

end for

output $\{C_k, \text{HBRNN}(\text{BRNN}, mt, C_k, Cs_j)\}_{j=1}^{Cs_j}$

HBRNN is characterized in terms of probability of success in recovering true hierarchy Cs^*, measurement and runtime complexity. Some restrictions are placed on similarity function S such that similarities agree with hierarchy up to some random noise:

S1 For each $\boldsymbol{y_i} \in \boldsymbol{Cs_j} \in \boldsymbol{Cs}^*$ and $\boldsymbol{j}' \neq \boldsymbol{j}$:

$$\min_{y_p \in Cs_j} \mathbb{E}\mathbb{x}\mathbb{p}\left[S(y_i, y_p)\right] - \max_{y_p \in Cs_j'} \mathbb{E}\mathbb{x}\mathbb{p}\left[S(y_i, y_p)\right] \geq \gamma > 0$$

Here expectations are taken with respect to the possible noise on $\boldsymbol{S}$.

S2 For each $\boldsymbol{y_i} \in \boldsymbol{Ct_j}$, a set of $\boldsymbol{V_j}$ words of size $\boldsymbol{v_j}$ drawn uniformly from $\boldsymbol{Cs_j}$ satisfies:

$$\mathbb{P}\mathbb{r}\mathbb{o}\mathbb{b}\left(\min_{y_p \in Cs_j} \mathbb{E}\mathbb{x}\mathbb{p}\left[S(y_i, y_p)\right] - \sum_{y_p \in V_j} \frac{S(y_i, y_p)}{v_j} > \epsilon\right) \leq 2e^{\left\{\frac{-2v_j\epsilon^2}{\sigma^2}\right\}}$$

Here $\sigma^2 \geq 0$ parameterizes noise on similarity function $\boldsymbol{S}$. Similarly set $\boldsymbol{V_{j'}}$ of size $\boldsymbol{v_{j'}}$ drawn uniformly from cluster $\boldsymbol{Cs_{j'}}$ with $\boldsymbol{j} \neq \boldsymbol{j}$ satisfies:

$$\mathbb{P}\mathbb{r}\mathbb{o}\mathbb{b}\left(\sum_{y_p \in V_{j'}} \frac{S(y_i, y_p)}{v_{j'}} - \max_{y_p \in C_{j'}} \mathbb{E}\mathbb{x}\mathbb{p}\left[S(y_i, y_p)\right] > \epsilon\right) \leq 2e^{\left\{\frac{-2v_{j'}\epsilon^2}{\sigma^2}\right\}}$$

The condition **S1** states that similarity from word y_i to its class should be in expectation larger than similarity from that word to other class. This is related to tight classification condition [14] and is less stringent than earlier results. The condition **S2** enforces that within-and-between-class similarities concentrate away from each other. This condition is satisfied if similarities are constant in expectation perturbed with any subgaussian noise. From the viewpoint of feature learning stacked BRNNs extracts temporal features of sentiment sequences in data. After obtaining features of sentiment sequence, fully connected layer *fc* and softmax layer *sm* performs classification. The LSTM architecture effectively overcomes vanishing gradient problem [14]. The LSTM neurons are adopted in last recurrent layer *br*4. The first three BRNN layers use *tanh* activation function. This is trade-off between improving representation ability and avoiding over fitting. The number of weights in LSTM is more than that in *tanh* neuron. It is easy to overfit network with limited data training sequences. The algorithm has certain shortcomings for practical applications. Specifically if *Cs* is known and constant across splits in hierarchy, above assumptions are violated in practice. This is resolved by fine tuning the algorithm with heuristics. The eigengap is employed where *Cs* is chosen such that eigenvalues gap of Laplacian is large. All subsampled words in data are discarded with low degree when restricted to sample with removes underrepresented classes from sample. In averaging phase if words in data are not similar to any represented class, new class for the word is created.

3 Experiments and Results

In this section experimental results are presented for opinion target extraction from reviews of DBS Text Mining Challenge 2015 datasets [13]. The train, development and test set splits are used to compare results with benchmark systems. The general performance of HBRNN is presented on datasets based on tenfold cross validation. The performance of HBRNN is evaluated with standard precision, recall and $F1$ score measures. The $F1$ score is equivalent to harmonic mean of recall and precision and has higher significance. For multiclass classification problem macro-averaged $F1$ score is selected as it gives equal weight to all classes and emphasises on rare classes [14, 16]:

$$F_i = \frac{2 \cdot pr_i \cdot re_i}{pr_i + re_i} \tag{15}$$

$$F_{macro} = \frac{\sum_i F_i}{n_{classes}} \tag{16}$$

Here *pr* and *re* are precision and recall. It is calculated from local categories and then averaged without considering data distribution. The micro-averaged $F1$ score is:

$$pr_{global} = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)} \tag{17}$$

$$re_{global} = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)} \tag{18}$$

$$F_{micro} = \frac{2 \cdot pr_{global} \cdot re_{global}}{pr_{global} + re_{global}} \tag{19}$$

In all experiments paired *t*-test are used on $F1$ scores to measure statistical significance. Highly biased data majority or duplicate minority classes are sub-sampled. Each non-trivial review is duplicated number of non-trivial aspects times contained in review. This method affects training data. The test is performed with and without combinations of other methods. The cost function is modified directly in model. The prediction problem is overcome by multiplying cost with weighting term $w_i > 1$ for non-trivial classes and $w_i < 1$ for 0 class. The cross entropy cost function for aspect results in:

$$\begin{aligned} EC_a &= \sum_i w_i z_i \log(\hat{z}_i) \\ \text{subject to: } &\sum_i w_i = 1 \wedge w_1 = w_3 \end{aligned} \tag{20}$$

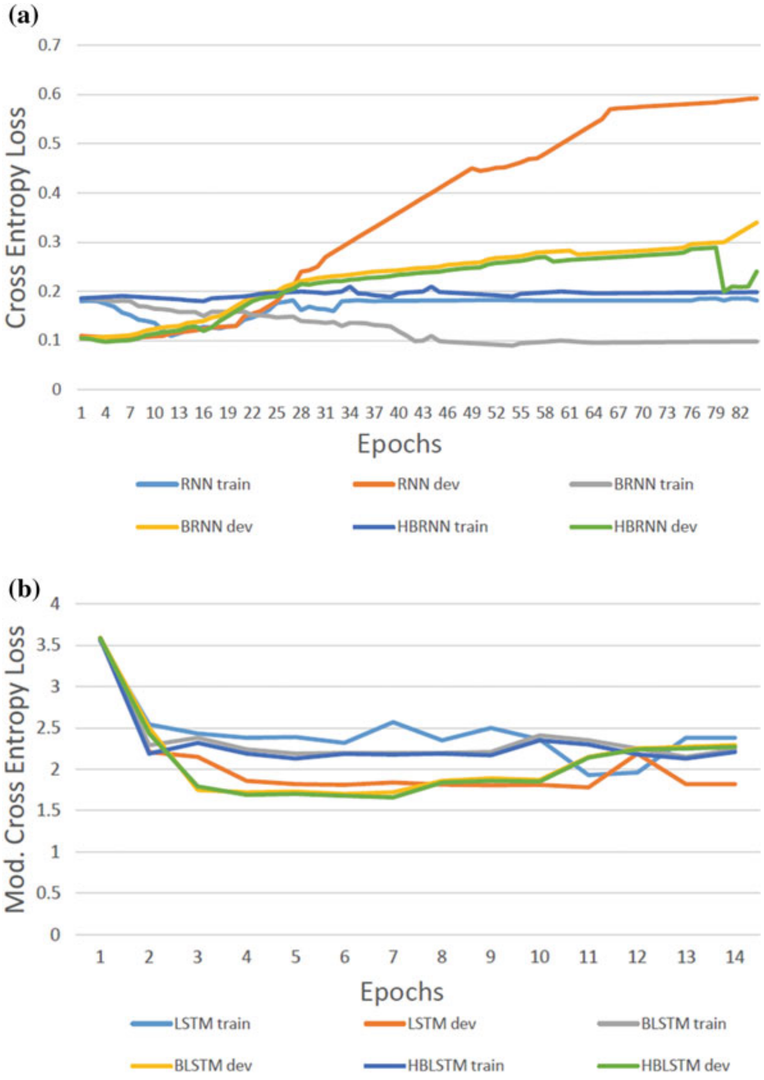

Fig. 3 **a** RNN, BRNN and HBRNN behaviour (cross entropy loss vs. epoch number). **b** RNN, BRNN and HBRNN behaviour (mod. cross entropy loss vs. epoch number)

The cost function is penalized more when existing sentiments are missed and it forces to look for sentiments. If weights assigned to w_1 and w_3 are smaller than w_2, the model predicts zeroes. If w_1 and w_3 are too large as compared to w_2 the network predicts with lower accuracy. The weights in network are initialized by sampling from small random uniform distribution $\mathbb{U}(-0.2, 0.2)$. The optimal weights are found using cross validation and $F1$ scores. The optimal weights for evaluation are $w = [1.1, 0.8, 1.1]$. The aspect information content is solved by increasing mini-batch sampling. This is based on mini-batch gradient descent augmented with information gain. Each mini-batch is created by randomly sampling data from training set. It is used to build mini-batch with highest possible entropy. On maximizing entropy it is likely to maximize information capability of each batch. The algorithm for batch creation has been adopted from [14, 16]. In order to find right parameters, datasets are divided into three subsets viz training set, development set for cross validation and optimization and test set for determination of final scores. The test set is prepared by separating randomly 10 % of available training data; remaining 70 and 20 % are used for training and cross validation. The time step is fixed to 5 on basis of validation set performance. Smaller values affect performance while larger values give no significant gains. The learning rate is fine tuned to reasonable value. The word vector dimension and number of epochs are adjusted jointly or marginally in terms of complexity and time. A fixed

Table 1 The results of different semantic analysis models in terms of precision, recall, F1 (macro) and F1 (micro) percentage scores

Semantic analysis models	Precision	Recall	F1 (macro)	F1 (micro)
RNN	82.2	90.2	31.2	81.2
RNN (duplicate)	84.5	89.5	31.5	86.5
Weighted RNN (duplicate)	86.5	90.2	32.5	85.5
BRNN	85.5	90.5	35.2	86.5
BRNN (duplicate)	86.0	90.0	43.0	87.5
Weighted BRNN (duplicate)	90.0	91.0	42.4	87.6
BRNN (mini-batches)	91.0	91.0	47.2	91.5
Weighted BRNN (mini-batches)	86.2	90.0	37.5	87.2
LSTM (duplicate)	82.0	90.2	32.5	91.0
Weighted LSTM (duplicate)	83.5	90.0	34.5	89.5
LSTM (mini-batches)	84.5	82.6	40.0	83.2
Weighted LSTM (mini-batches)	84.2	85.6	40.2	86.5
BLSTM (duplicate)	82.2	90.0	32.5	90.6
Weighted BLSTM (duplicate)	82.0	89.0	32.4	89.6
BLSTM (mini-batches)	84.7	88.6	40.0	89.0
Weighted BLSTM (mini-batches)	84.2	84.0	38.5	85.2
HBRNN (duplicate)	89.2	93.2	36.5	91.5
Weighted HBRNN (duplicate)	92.2	94.2	38.2	93.2
HBRNN (mini-batches)	93.5	94.5	40.2	94.2
Weighted HBRNN (mini-batches)	94.2	95.2	40.5	96.2

learning rate of 0.01 is used but batch size is changed depending on sentence length [14]. The process is repeated for 30 epochs and $F1$ score is calculated on validation set after each epoch. The size of context window is set to 3 based on validation set performance. Figure 3a, b show behaviour of RNN, BRNN, HBRNN, LSTM and BLSTM through different epochs. From epoch 10 models start overfitting so it is chosen for evaluation. For implementation word vector dimension is taken as 120 and LSTM hidden layer dimensions are set at 40. Table 1 shows performance metrics of models. RNN performs poorly and predicts only zeros. LSTM and BLSTM perform poorly as RNN. It is observed from Table 1 that BRNN combined with HBRNN based on augmented mini-batches performs best in all metrics. It can be taken as the best way to overcome biased distribution given lack of flexibility and high bias.

4 Conclusion

Aspect specific sentiment analysis for reviews of different products is gaining popularity among machine learning researchers. The problem becomes challenging when data volume grows. The entity level semantic analysis with robust HBRNN is proposed here. It is presented as general class of discriminative model based on RNN architecture and word embeddings. HBRNN is developed by extending RNN and BRNN so that accuracy and efficiency are improved. This optimization is achieved by fine tuning different parameters. The results are compared with LSTM and BLSTM also. The major challenges encountered here include: (a) lack of high quality labeled online review data and (b) high skewness in review data. The aspect information content which increased mini-batch sampling is used during experiments. All methods are evaluated using precision, recall and $F1$ scores. The experimental results have proved the fact that HBRNN has outperformed all other methods. As future work the proposed method would be applied to other fine grained text and opinion mining tasks with increasing data volumes. Also experiments are to be performed in order to determine to what extent these tasks be jointly modeled in this multitasking framework by incorporating soft computing tools.

References

1. Feldman, R.: Techniques and applications for sentiment analysis. Commun. ACM **56**(4), 82–89 (2013)
2. Li, M. X., Tan, C. H., Wei, K. K., Wang, K. L.: Where to place product review? An information search process perspective. In: 31st International Conference on Information Systems, Paper 60 (2010)
3. Liu, B.: Sentiment analysis and opinion mining. In: Synthesis Lectures on Human Language Technologies, vol. 16. Morgan and Claypool (2012)
4. Liu, B., Zhang, L.: A survey of opinion mining and sentiment analysis. In: Aggarwal, C.C., Zhai, C.X. (eds.) Mining Text Data, pp. 415–463. Springer, Heidelberg (2012)

5. Balahur, A., Hermida, J.M., Montoyo, A.: Detecting implicit expressions of emotion in text: a comparative analysis. Decis. Support Syst. **53**, 742–753 (2012)
6. Bickart, B., Schindler, R.M.: Internet Forums as Influential Sources of Consumer Information. J. Interact. Mark. **15**(3), 31–40 (2001)
7. Chen, Y., Xie, J.: Online consumer review: word-of-mouth as a new element of marketing communication mix. Manage. Sci. **54**(3), 477–491 (2008)
8. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: 19th National Conference on Artificial Intelligence, pp. 755–760. AAAI Press (2004)
9. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 339–346 (2005)
10. Wei, C.P., Chen, Y.M., Yang, C.S., Yang, C.C.: Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews. IseB **8**(2), 149–167 (2010)
11. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques. In: 3rd IEEE International Conference on Data Mining, pp. 427–434 (2003)
12. Zhu, J., Wang, H., Zhu, M., Tsou, B.K., Ma, M.: Aspect based opinion polling from customer reviews. IEEE Trans. Affect. Comput. **2**(1), 37–49 (2011)
13. DBS Text Mining Challenge 2015 Dataset: https://www.ideatory.co/challenges/dbs-text-mining-challenge-2015/
14. Chaudhuri, A.: Semantic analysis of customer reviews with machine leaning methods. Technical Report, Samsung R & D Institute, Delhi, India (2015)
15. Heaton, J.: Deep Learning and Neural Networks. In: Artificial Intelligence for Humans, vol. 3. CreateSpace Independent Publishing Platform (2015)
16. Ahres, Y., Volk, N.: Entity level sentiment analysis for amazon web reviews. Final Year Project Report, Stanford University, California (2015)