Turkish Journal of Computer and Mathematics Education Vol.12No.13 (2021), 3273-3288
Research Article

Transfer Learning For Prediction Of Sentiment In Hotel Reviews

Supriya B N[a] and C.B. Akki[b]
Department of ISE, SJBIT, Affiliated to VTU, Bangalore 560060, India
Registrar, IIIT, Dharwad 580029, India
Bnsupriya@gmail.com, akki.channappa@gmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 4 June 2021

Abstract
The web platforms have made people share thoughts, write reviews and make a huge source of information. These web platforms can be online news, blogs, community's discussion forum. People visited any hotel write their reviews on these forums. Understand manually all the text written becomes complex because people express their views in the different and complex ways. For instance, the online reviews given on hotel services and quality, it is difficult to understand the reviews manually. To make certain decision on improving the quality and service of the hotel it will be inconvenient to read the reviews manually. In this concern, the paper aims to develop a deep learning technique and transfer learning with word embeddings to analyse hotel review for identifying the response strategies. We have also proposed a new combined model, which integrates machine learning and convolutional neural network models with GloVe Embeddings to analyse the text. The obtained results show that proposed new model can outperform compare to other machine learning techniques.

Keywords: Hotel reviews, machine learning, convolutional neural network, extreme gradient decent, word embeddings._

1. INTRODUCTION

In present days a company, a business organization or a service-based sector which requires feedback from its customers to improve the business knowledge and make policies to develop organization. Feedbacks are expressed in online portals, blogs from the customers through reviews and ratings. For example, in case of hotel business, the reviews and ratings will be given on quality of food, hospitality, price, location, cleanliness of rooms, wi-fi facility inside the hotel, staff with multilingual, wheel chairs and so on. These reviews play an important role in recommending the hotel for other customers. The major problems in understanding the reviews posted on online portals are the data is huge, unstructured, spelling mistakes, usage of special symbols, usage of words will vary from one customer to another customer [1]. The sample reviews and rating given for hotel is shown in figure 1.

Figure 1. Sample Hotel review [9]

In this paper we have mainly worked towards the introduction of word embeddings with GloVe and implemented transfer learning with convolution neural network. We have also combined weighted convolutional neural network model with weighted XGBoost model to improve the prediction accuracy and compared the obtained results with existing models such as KNN, SVM, DT, RF, XGBoost.

This paper is ordered as follows: section 2 represents the prior work carried out on sentiment analysis for hotel reviews, section 3 explores on the proposed approaches, section 4 gives detailed description of implementation and results obtained from proposed approach and comparative study, section 5 provides the conclusion and future work

2. PRIOR WORK

In this section, brief literature survey and the problem in analysis of the sentiment of hotel reviews has been indicated. User-generated content (UGC), is a platform for the user to express their thoughts, emotions, views and give their rating on the services and quality of the product [2]. In the current technology advancement UGC place an important role in continuous upgrading the process in hotel. Customers spend more time on social media to understand the rating and reviews given in UGC platforms based on that decision will be taken to visit the place, hotel. Advancement in the web 2.0 technologies the application such as Trip Advisor, facebook, twitter and so on provides access to share the opinions and write reviews on the web pages. These will generate a huge amount of unstructured data and becomes complex to analyse the data [3]. To overcome the complexity in analysing unstructured data, several authors have explored Natural Language Processing (NLP) techniques. Zhang et al [4], classified reviews of cantonese restaurant into positive and negative using 3-grams features and applied Support Vector Machine (SVM) and Naïve Bayes (NB). Tsai et al. [5], classified sentiment based on hotel aspects such as service, ambience, quality of food, price and computed aspect features. Trained and built a multi-class SVM classifier for aspect opinion features. Govindarajan et al [6], performed sentiment analysis and proposed weighted voting structure hybrid classification technique for yelp restaurant dataset. Xiang et al. [7], examined the hotel guest experience and ratings given by text analytical technique, focused mainly on the attributes such as services and rooms quality. Al-smadi et al. [8], addressed the challenge of aspect-based analysis of sentiment for Arabic hotel using two approaches namely Recurrent Neural Network (RNN) and SVM. A.Sharma et a.[10], proposed Back-Propagation Neural Network (BPANN) for hotel and movie reviews classification. Nirkhi et al. [11] used Self-Organizing Map (SOM) for the feature extraction and stored in the data vectors for analysis of hotel reviews. Chang et al [14], collected TripAdvisor reviews data and analysed the reviews using deep learning and visual analytics. Nguyen et al. [15], proposed a new methodology to understand the sentiment by combining hotel rating and reviews written. Tsai et al [16], extracted relevant features from text and summarised the hotel reviews.

The fact that hotel review data is unstructured is the key issue. It includes data that has been reviewed several times, spelling errors, and data that is irrelevant, unlabelled, and imbalanced. Unstructured data is difficult to analyse since distinguishing between correlation and causation is difficult. Direct analysis is not possible using traditional approaches, which are optimised for well-structured, quantitative data.

The existing work falls short of accurately reflecting the sense of hotel reviews. Within the current work, there are a number of limitations.

- Lexicon-based techniques do not require any training data, but they perform poorly in terms of accuracy due to lexicon coverage.
- While some studies extract feedback at the word level, it is preferable to handle them at the sentence level.
- Choosing features and extracting sentiment features is more difficult.
- Additional data processing and transformation tasks are needed for machine learning models, which may increase the computation's complexity.

3. PROPOSED METHODOLOGY

To overcome these limitations, transfer learning is proposed, which optimises pre-trained convolution neural network models using unique domain data. The proposed model combines transfer learning with GloVe word embedding and a CNN model that has already been trained. The proposed methodology consists of different phases such as data acquisition, exploratory data analysis, data pre-processing, building features, embedding transfer learning, prediction and evaluation of the models. The proposed methodology is shown in figure 2.

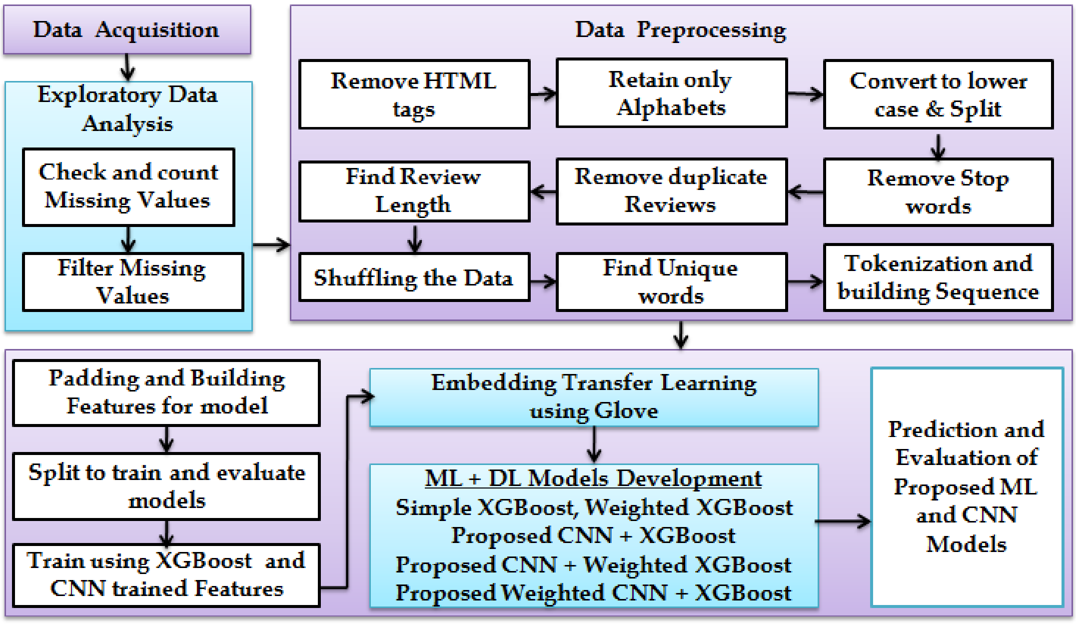

Figure 2. Proposed Methodology

3.1 Data acquisition

To collect data on hotel reviews, we downloaded dataset from Kaggle openly available web resource for research. The dataset downloaded consist of many fields such as city, address, country, latitude, longitude, review date, review text, review user and review rating. Sample hotel review dataset is shown in figure 3. Hotel review dataset downloaded is an unstructured and unlabelled. Based on the hotel rating, reviews are labelled into promising and non-promising. If the rating are greater than or equal to 3 then it labelled as promising reviews and rating less than 3 labelled as non-promising reviews.

reviews	reviews.text
4	Pleasant 10 min walk along the sea front to the Water Bus. restaurants etc. Hotel was comfortable breakfast was good - quite a variety. Room aircon didn't work very well. Take mosquito repelant!
5	Really lovely hotel. Stayed on the very top floor and were surprised by a Jacuzzi bath we didn't know we were getting! Staff were friendly and helpful and the included breakfast was great! Great location and great value for m
5	Ett mycket bra hotell. Det som drog ner betyget var att vi fick ett rum under taksarna dr det endast var full sthjd i 80 av rummets yta.
5	We stayed here for four nights in October. The hotel staff were welcoming, friendly and helpful. Assisted in booking tickets for the opera. The rooms were clean and comfortable- good shower, light and airy rooms with windo
5	We stayed here for four nights in October. The hotel staff were welcoming, friendly and helpful. Assisted in booking tickets for the opera. The rooms were clean and comfortable- good shower, light and airy rooms with windo
5	We loved staying on the island of Lido! You need to take a water is from Venice to get there. From the train station, a boat ride takes 45 minutes but has beautiful views along the way. Hotel is an EASY walk from the boat doc
4	Lovely view out onto the lagoon. Excellent view. Staff were welcoming and helpful.
4	ottimo soggiorno e ottima sistemazione nei giorni frenetici di inaugurazione della Biennale. Le signore alla reception sono efficientissime e squisite e non sono da meno le ragazze che servono la prima colazione. Da tornarci
3	Gnstiger Ausgangspunkt fr Venedig Besuche. Ruhige Lage auf dem Lido. Flugplatz Lido und Bootsanlegestellen fulufig erreichbar. Zimmer ziemlich eng, aber alles vorhanden. Frhstck fr Italien ausgesprochen reichhaltig. Hotel sc
4	Lidoen er perfekt til et par dages ro og afslapning, skn strand, lkkert omrde og lille hyggeligt familiehotel med et sdt personale
4	Accueil chaleureux, en franais Changement du linge de lit tous les jours, lit confortable, salle de bain de bonne taille et bien quipe. Petit djeuner copieux et vari.
3	It was ok hotel is nice from in and out but room was small we paid for double bed bat they attached 2 single bed
4	Klasse Frhstck, freundliches und aufmerksames Personal, gute Anbindung nach Venedig, Zimmer vllig ok und sauber, Parkplatz in der Nhe gut verfgbar. Wrden wir wieder whlen!
4	Bardzo sympatyczna obsuga, klimat hotelu. wietna azienka, widok na zatok. Bardzo dobry stosunek cena/jakosc. Polecam!
4	Bra o lugnt lge. Stor terrass. Nra till den hrliga Lidostranden.Bara en TV-kanal. Bra frukost. Litet opraktiskt badrum.Trevlig och kunnig personal.
4	The hotel staff was very friendly and helpful. The room was clean and comfortable. My wife and I had a room with a terrace over looking the water...it was a great view! I will look to stay at this hotel my next time in Venice.

Figure 3: sample hotel review dataset Collected

3.2 Exploratory Data Analysis (EDA)

To understand how the collected data is distributed, EDA is performed. In this work, the maximum length of the reviews, number of promising and non-promising reviews, unique words are identified to understand the most important words used in writing the reviews. Figure 4 shows the graphical representation of EDA made on the obtained dataset.

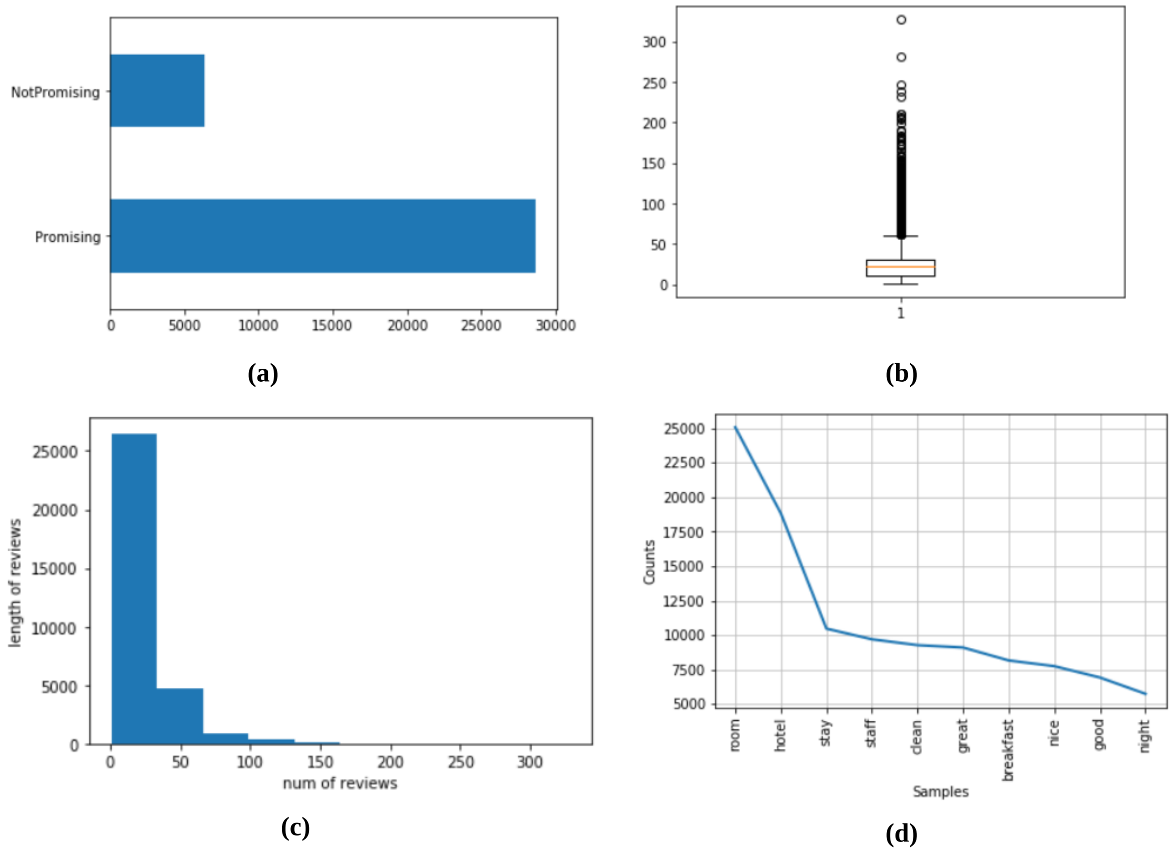

Figure 4 (a) Count of promising and non-promising data (b) Length of the hotel review varies from 1 to 327 (c) Maximum review range (d) Unique words and its count

3.3 Data Pre-processing

We downloaded the raw hotel review dataset from online web resources. Hotel review dataset consist of different fields such as address, categories, city, country, latitude, longitude, name postal Code, province, reviews date, reviews date Address, reviews do recommend, reviews id, reviews text, reviews title, reviews user City, reviews username reviews user Province. We are more interested to analyse reviews text hence extracted only reviews text. This reviews text needs to be pre-processed because it contains irrelevant and duplicate contents. Following steps are performed to clean and pre-processing of reviews texts:

- Removing html tags
- Retaining only alphabets.
- Converting to lower case and splitting
- Remove stopwords using wordnet lemmatizer.
- Removal of duplicate reviews.

Data pre-processing is done to obtain cleaner data which in turn will provide ease of processing the data further to obtain meaningful results. Figure 5 shows the cleaned data from data pre-processing

```
pleasant min walk along sea front water bus restaurant etc hotel comfortable breakfast good quite variety room aircon work well
take mosquito repelant


really lovely hotel stayed top floor surprised jacuzzi bath know getting staff friendly helpful included breakfast great great
location great value money want leave

stayed four night october hotel staff welcoming friendly helpful assisted booking ticket opera room clean comfortable good show
er light airy room window could open wide bed comfortable plenty choice breakfast spa hotel nearby used


loved staying island lido need take water venice get train station boat ride take minute beautiful view along way hotel easy wa
lk boat dock room clean breakfast plentiful would definitely recommend hotel
```

Figure 5: Insights of hotel reviews after data pre-processing

3.4 Embedding Transfer Learning with GloVe

Tokenization of words are performed after the data pre-processing. Word index is built to create the input features. The main functionality of Natural Language Processing (NLP) is to encode the word or sentence into a computer understandable format. Representing the words in form of vector brings NLP into the learn the meaning of the word. To represent the word meaning, GloVe model is applied. GloVe model derives the co-occurrence probabilities of words within a texts corpus for embedding the meaning to words. The word j occurred in the sequence of the word i all in the text's corpus. Let X be matric representing cooccurrence of word-word and $\mathbf{X_{ij}}$ be the count of number of times the word j occurred in the sequence of word j. The co-occurrence probabilities can be calculated using equation 1:

$$P_{ij} = P(j|i) = \frac{X_{ij}}{\sum_{k\in context} X_{ik}} \quad \text{----- (1)}$$

The GloVe model computes function F given in equation 2 that can predict the ratio of given two vectors of word w_i and w_j and a context word vector w_k as inputs

$$F(w_i, w_j, \widetilde{w}_k) = \frac{P_{ik}}{P_{jk}} \quad \text{------(2)}$$

Here, we have two input vectors of F, to reduce the complexity the GloVe model uses the dot product of the input two vectors. In the word "cool" is a context of the word "chill" can be considered as in same context. This symmetry of the **X** matrix (our co-occurrence matrix) has to be taken into account when building ***F***, we must be able to switch $\mathbf{w_i}$ and $\mathbf{w_k}$. First, we need ***F*** to be a homomorphism (F(a+b) = F(a)F(b)).

$$F((w_i - w_j)^T \widetilde{w}_k) = \frac{F(w_i^T \widetilde{w}_k)}{F(w_j^T \widetilde{w}_k)} \text{ , which gives } F(w_i^T \widetilde{w}_k) = P_{ik} = \frac{X_{ik}}{\sum_{w\in context} X_{iw}}$$

To restore the symmetry, a bias b_k is added for the vector w_k.

$$w_i^T \widetilde{w}_k + b_i + \widetilde{b}_k = log(X_{ik})$$

In the training phase, the GloVe model will learn the appropriate word vectors $\mathbf{w_i}$ and $\mathbf{w_j}$ to reduce the problem of weighted least square. The weighted function *$f(X_{ij})$ is used to make the rare cooccurrence and cooccurrence which are most common with same importance:*

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \widetilde{w}_j + b_i + \widetilde{b}_j - log(X_{ij}))^2$$

3.5 Building Machine Learning and Convolutional Neural Network models

In this section, we present machine learning (ML) and convolutional neural network (CNN) built. The main task is feature learning, to perform this task pretrained GloVe model is used with transfer learning. We have built Convolutional Neural Network model. The CNN was designed by Geoffery Hinton, one of the inventors of ML. CNN mainly consists of convolution layers, pooling layers and fully connected layer. The kernel conceded over the input matrix to produce a feature maps to the next convolutional layers. To reduce the dimension max pooling is used to take the average and create input matric for next convolutional layer. The results of the max pooling layers is fed into fully connected layers to derive the classification of reviews. The architecture of CNN is shown in figure 6.

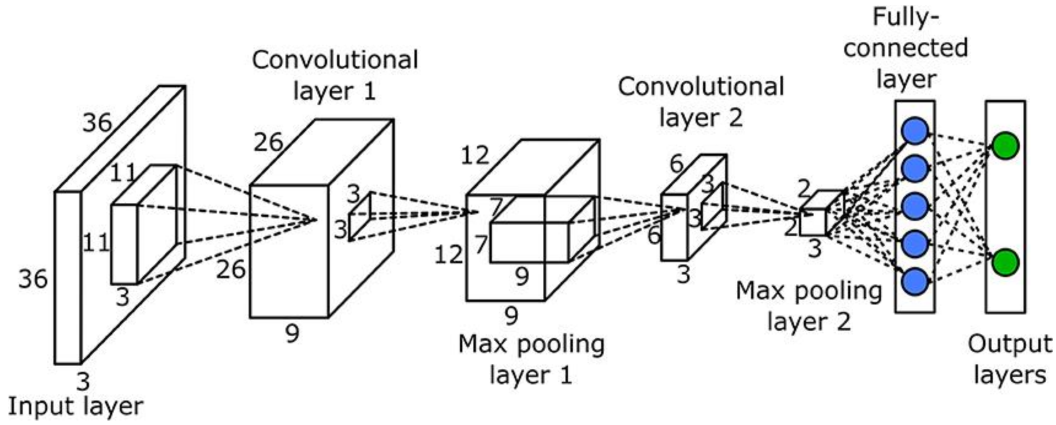

Figure 6: Architecture of Convolutional Neural Network

For classification of the promising and non-promising reviews, initially we experimented with simple XGBoost model and weighted XGBoost models. XGBoost is an ensemble machine learning built based on decision tree that uses gradient boosting [13]. Ensemble machine learning combines the predictive output of multiple learned models. The aggregated models can be either same algorithm learnt or different learning algorithms. Bagging and boosting are the most commonly used in ensemble learning techniques. In bagging technique many decision trees are computed in parallel from the initial learners. Data patters with replacement are provided to the learners during the training. Boosting technique consists of three steps: Initial built model P0 is determined to predict target parameter 't'. This model will be correlated with an residual (t- P0). An new generated model m1 is fitted with residual in previous step. Now, P0 and m1 gives the P1 , the mean square error of P1 will be lesser than P0. These steps can be made in 'n' iterations until the residual errors are minimized as shown in below equation.

$$P_n(x) < P_{n-1}(x) + m_n(x)$$

For gradient boosting following steps are followed. P0 (x) with initial model are determined and function to minimise the Mean Square error in this case is:

$$P_0(x) = \arg\min_{\phi} \sum_{i=1}^{n} S(\phi_i - \phi)^2$$

The loss function f_{in} in gradient are determined iteratively, where δ is a rate of learning:

$$f_{in} = -\delta \left[\frac{\partial(S(\phi_i, P(x_i))}{\partial P(x_i)} \right]_{P(x)=P_{n-1}(x)}$$

We combined weighted CNN and XGBoost and for predicting the hotel reviews classes. The proposed algorithm is given below:

WConvXGB Learning Algorithm:

$Let\ \ I = \{(p_i, q_i) \mid 1 \le i \le D\}$, where D is the given size of the training data

Set, $p_i = \{p_1, p_2, ...p_n\}$ be a set of N features vectors in F^N and q_i is the label of vector p_i.

Let E = $\{e_1, e_2..........e_n\}$, be a set of word embedding from Global Vector (GloVe)

Initialise the parameters of the convolution layers:

Number of convolution layers *CL*

Convolutional layer output depth z, for each layer set size of the filter, T and strides S_k

Calculate the convolutions to generate the Y for layer, l:

$$\mathscr{Y}_i^{(l)} = \phi\left(B_i^{(l)} + \sum_{j=1}^{f^{(l-1)}} K_{i,j}^{(l)} * \mathscr{Y}_j^{(l-1)}\right)$$

The parameters are initialised for predicting

(a) Count of the trees, T

(b) regularization values g and l,

(c) subsampling column,

(d) depth of max tree and

(e) rate of learning.

For each filter number, filter size, pooling size in conv layers:

x = Conv1D(filter number, filter size)(x)

use x = Activation('relu')(x)

if pooling size != -1:

x = MaxPooling1D(pool size=pooling size)(x)

x = Flatten()(x)

Fully connected layers

for each dense size in fully connected layers:

x = Dense(dense size, activation='relu')(x)

x = Dropout(dropout p)(x)

Output Layer

predictions = Dense(num_of_classes, activation='softmax')(x)

Determine the class labels for output

$$\widehat{y}_i = \varphi(YY_i) = \sum_{k=1}^{K_K} f_k(x_i), \quad f_k \in F,$$

To derive the best solution, we have divided the proposed methodology in to two parts namely training, validation and predicting phase. The training, validating and predicting phase is discussed below:

Algorithm: Training and Validation Phase

Input: Hotel reviews without labels

Output: relationship of customer with hotel services

METHOD:

1. Read the hotel_reviews.csv file for training phase
2. Remove not necessary patterns
3. Data segmentation and create a sequence for train and test
4. Encoding input and output features
5. Split the preprocessed data for training (Ratio: 80:20 and 70:30)
6. Feed input and output features standard weighted CNN models for analyzing input and output features. Fine tune the hyper parameters such as activation function, fully
7. Validate standard using categorical cross entropy loss function.
8. Evaluate the proposed weighted CNN+XGBoost models for obtaining performance results in terms of precision, recall, accuracy and F1-score.

ALGORITHM: TRAINING AND VALIDATION PHASE ENDS

3.6 Prediction and Evaluation of the models

To measure the performance of the proposed models, we used performance metrics such as Precision, Recall, F1-Score. True Positive (TP): promising and non-promising hotel review correctly classified. True Negative (TN): promising and non-promising hotel review not correctly classified. False Positive (FP): promising reviews are classified as non-promising and non-promising reviews are classified as promising. False Negative (FN): non promising but predicted as promising. Following formulas are used to compute precision, recall and f1-score

Precision = TP/TP+FP

Recall = TP/TP+FN

F1-Score = 2*(Recall * Precision) / (Recall + Precision)

4. IMPLEMENTATION

Implementation of the work is done using python 3.8 version in anaconda version 3. Libraries such as numpy, sklearn, pandas and plotly are used in the development. Numpy library in the python packages provides an scientific computing functionality for numerical analysis. Scikit learn is an open source software ML package in the python programming language. The XGBoost parameters are initialised with following values.

Table1: XGBoost paramter initialization

Parameters	**Value**
Eta	0.1
max_depth	10
n_estimators	300
Silent	1
n_job	-1
num_class	2

Layer (type)	Output Shape	Param #
input (InputLayer)	(None, 328)	0
embedding_1 (Embedding)	(None, 328, 100)	2792300
conv1d_1 (Conv1D)	(None, 322, 256)	179456
activation_1 (Activation)	(None, 322, 256)	0
max_pooling1d_1 (MaxPooling1	(None, 107, 256)	0
conv1d_2 (Conv1D)	(None, 101, 128)	229504
activation_2 (Activation)	(None, 101, 128)	0
max_pooling1d_2 (MaxPooling1	(None, 33, 128)	0
conv1d_3 (Conv1D)	(None, 31, 128)	49280
activation_3 (Activation)	(None, 31, 128)	0
max_pooling1d_3 (MaxPooling1	(None, 10, 128)	0

____

flatten_1 (Flatten) (None, 1280) 0

____

dense_1 (Dense) (None, 64) 81984

____

dropout_1 (Dropout) (None, 64) 0

____

dense_2 (Dense) (None, 32) 2080

____

dropout_2 (Dropout) (None, 32) 0

____

dense_3 (Dense) (None, 2) 66

===

Total params: 3,334,670

Trainable params: 3,334,670

Non-trainable params: 0

Model: "model_2"

____

Layer (type) Output Shape Param #

===

input (InputLayer) (None, 328) 0

____

embedding_2(Embedding) (None,328,100) 2792300

____

conv1d_4 (Conv1D) (None, 322, 256) 179456

____

activation_4 (Activation) (None, 322, 256) 0

____

max_pooling1d_4 (MaxPooling1 (None, 107, 256) 0

____

conv1d_5 (Conv1D) (None, 101, 128) 229504

____

activation_5 (Activation) (None, 101, 128) 0

____

max_pooling1d_5 (MaxPooling1 (None, 33, 128) 0

____

conv1d_6 (Conv1D) (None, 31, 128) 49280

____

activation_6 (Activation) (None, 31, 128) 0 max_pooling1d_6 (MaxPooling1 (None, 10, 128) 0

flatten_2 (Flatten) (None, 1280) 0

dense_4 (Dense) (None, 64) 81984

dropout_3 (Dropout) (None, 64) 0

dense_5 (Dense) (None, 32) 2080

dropout_4 (Dropout) (None, 32) 0

dense_6 (Dense) (None, 2) 66

===

Total params: 3,334,670

Trainable params: 3,334,670

Non-trainable params: 0

5. RESULTS

To analyse the proposed algorithm, initially experimented with simple XGBoost model with 70:30 training and testing split ratio and then tried with 80:20 split ratio. The hyper parameters such as estimator and rate of learning of the XGBoost model are initialised with default values and then varied to analyse the changes in precision, recall and F1-score of each class label 0 (Non-promising), 1 (Promising). The obtained result after applying simple XGBoost for 70:30 and 80:20 split ratio is tabulated in table 2 and table 3. Since the data is imbalance and skewed, F1 score of non-promising class is less compare to promising class. The accuracy of simple XGBoost gives better result when learning rate is 0.1 and estimator value is 300.

Table 2: Simple XGBoost results with 70:30 training and testing split ratio.

Estimator, Learning Rate	Accuracy	Precision				Recall				F1-Score			
		0	1	Macro avg	W-avg	0	1	Macro avg	W-avg	0	1	Macro avg	W-avg
200,0.075	82.9	59.8	83.8	71.8	79.5	11.7	98.3	54.9	82.9	19.5	90.5	55.0	77.9
300,0.1	**83.3**	**61.9**	**84.03**	**72.9**	**80.1**	**13.1**	**98.2**	**55.6**	**83.2**	**21.6**	**90.6**	**56.1**	**78.4**
400, 0.25	83.2	58.8	84.6	71.7	80.0	17.3	97.4	57.4	83.2	26.8	90.5	58.6	79.2
500,0.5	82.9	54.4	84.9	69.7	79.8	20.9	96.6	58.3	82.8	29.5	90.3	59.9	79.5
600,0.75	82.3	49.9	85.1	67.5	78.9	22.4	95.1	58.7	82.3	30.8	89.9	60.4	79.4

Table 3: Simple XGBoost results with 80:20 training and testing split ratio.

Estimator, Learning Rate	Accuracy	Precision				Recall				F1-Score			
		0	1	Macro avg	W-avg	0	1	Macro avg	W-avg	0	1	Macro avg	W-avg

200,0.075	83.1	60.5	84	72	79.8	12.7	98.2	55.5	83.1	21.1	90.5	55.9	78.2
300,0.1	81.3	46.4	86.8	66.6	79.7	35.7	91.1	63.4	81.3	40.3	88.9	64.7	80.4
400, 0.25	83.2	58.8	84.6	71.7	80.0	17.3	97.4	57.4	83.2	26.8	90.5	58.6	79.2
500,0.5	82.4	50.4	86.1	68.2	79.8	29.9	93.6	61.8	82.4	37.5	89.7	63.7	80.5
600,0.75	82.3	49.9	85.0	67.5	78.9	22.3	95.1	58.7	82.3	30.8	89.9	60.4	79.4

To overcome the data imbalance problem, the average of non-promising and promising instances is computed. The computed results, are given as weight (W=0.22) to the XGBoost model and named as weighted XGBoost. The table 4 and table 5 shows the results obtained after applying weighted XGBoost with split ratio of 70:30 and 80:20 respectively. The accuracy of weighted XGBoost gives better result when learning rate is 0.5 and estimator value is 500 in case of 70:30 split ratio. The accuracy of weighted XGBoost gives better result when learning rate is 0.075 and estimator value is 200 in case of 80:20 split ratio.

Table 4: Weighted XGBoost (W= 0.22) results with 70:30 training and testing split ratio.

Estimator, Learning Rate	**Accuracy**	**Precision**				**Recall**				**F1-Score**			
		0	1	Macro avg	W-avg	0	1	Macro avg	W-avg	0	1	Macro avg	W-avg
200,0.075	78.9	40.3	86.8	63.5	78.6	38.2	87.8	62.9	78.9	39.1	87.3	63.2	78.7
300,0.1	80.8	44.3	86.4	65.3	78.9	33.4	90.9	62.1	80.8	38.1	88.6	63.3	79.7
400, 0.25	82.2	49.5	85.9	67.7	79.5	28.6	93.7	61.1	82.2	36.8	89.7	62.9	80.2
500,0.5	81.9	48.1	85.9	67.7	79.2	28.9	93.3	61.1	81.9	36.0	89.4	62.7	80.6
600,0.75	81.1	44.9	85.8	65.3	78.5	28.6	92.4	60.5	81.1	34.9	88.9	61.9	79.4

Table 5: Weighted XGBoost (W= 0.22) results with 80:20 training and testing split ratio.

Estimator, Learning Rate	**Accuracy**	**Precision**				**Recall**				**F1-Score**			
		0	1	Macro avg	W-avg	0	1	Macro avg	W-avg	0	1	Macro avg	W-avg
200,0.075	83.1	62.3	84	73	80.0	12.0	98.4	55.5	83.1	20.1	90.5	55.2	78.2
300,0.1	81.3	46.4	86.8	66.6	79.7	35.7	91.1	63.4	81.3	40.3	88.9	64.7	80.4
400, 0.25	82.2	49.5	85.9	67.7	79.5	28.6	93.7	61.1	82.2	36.8	89.7	62.9	80.2
500,0.5	82.9	54.5	84.9	69.7	79.5	20.9	96.3	58.8	82.9	30.0	90.2	60.1	79.6
600,0.75	81.2	44.9	85.8	65.5	78.6	29.3	92.3	60.7	81.1	35.3	88.9	62.2	79.4

To analyse the data, we tried to build simple convolutional neural network with GloVe embedding layer and experimented with different activation function and fully connected layers. The table 6 shows the result obtained from applying simple CNN. The accuracy is better when simple CNN with activation function in FCL is Relu and activation function in Output Layer (OL) is Softmax. Simple CNN is extended with weighted value for CNN and embedding layers with GloVe to analyse the data. Table 7 shows the result obtained from applying weighted CNN with GloVe embedding layers. The accuracy is better when weighted CNN with GloVe embedding layers and activation function in FCL is Relu and activation function in OLis Softmax.

Table 6: Simple CNN results with different activation function in fully connected layer (FCL) and output layer (OL).

FCL	**OL**	**Accuracy**	**Precision**				**Recall**				**F1-Score**			
			0	1	M-avg	W-avg	0	1	M-avg	W-avg	0	1	M avg	W-Avg
Relu	Softmax	87.8	69.3	90.9	80.1	87.1	55.9	94.7	75.3	87.8	61.9	92.7	77.3	87.3
Linear	Sigmoid	82.3	0.01	82.3	41.6	67.7	0.01	100	50.4	82.3	0.03	90.3	45.1	74.4
Sigmoid	Relu	17.7	17.7	0.01	8.9	3.2	100	0	50.0	17.7	30.0	0	15.0	5.3
Relu	Elu	17.7	17.7	0	8.9	3.1	99.9	0	49.9	17.7	30.0	0	15.0	5.32
Elu	Swish	18.14	17.7	83.7	50.7	72.0	99.4	0.67	50.0	18.1	30.0	1.34	15.7	6.4

Table 7: Weighted CNN results with different activation function in fully connected layer (FCL) and output layer (OL) [Weight for class 0: 2.83, Weight for class 1: 0.61].

FCL	**OL**	**Accuracy**	**Precision**				**Recall**				**F1-Score**			
			0	1	M-avg	W-avg	0	1	M-avg	W-avg	0	1	M avg	W-avg
Relu	Softmax	86.3	66.05	89.1	77.6	85.0	46.4	94.8	70.6	86.3	54.5	91.9	73.2	85.3
Linear	Sigmoid	85.7	57.11	94.6	75.8	87.9	76.9	87.6	82.2	85.7	65.5	91.0	78.2	86.5
Sigmoid	Relu	17.7	17.7	0.01	8.9	3.1	100	0	50.0	17.7	30.07	0	15.0	5.3
Relu	Elu	17.7	17.7	0	8.9	3.1	100	0	50.0	17.7	30.0	0	15.0	5.32
Elu	Swish	17.7	17.7	83.7	50.7	72.0	99.4	0.67	50.0	18.1	30.0	1.34	15.7	6.4

The two combined new models are proposed, which combines CNN models with simple XGBoost and weighted XGBoost model. The accuracy results obtained from combined model are shown in table 8 and table 9. Finally, another new model is proposed by combining Weighted CNN and Weighted XGBoost with GloVe word embedding layers and achieved a best accuracy of 88.4%. The accuracy obtained from new combination model Weighted CNN and Weighted XGBoost is shown in table 10 and all the model accuracy comparison is given in table 11. Graphical representation of accuracy obtained from all the seven build XGBoost CNN models are shown in figure 7.

Table 8: Results obtained from combination of simple CNN + simple XGBoost

Estimator Learning Rate	FCL	OL	Accuracy	Precision				Recall				F1-Score			
				0	1	M-avg	W-avg	0	1	M-avg	W-avg	0	1	M avg	W-avg
200,0.075	Relu	Softmax	87.3	65.15	91.4	78.6	86.8	59.4	93.8	77.6	86.9	62.5	92.9	77.2	86.9
300,0.1	Linear	Sigmoid	86.3	66.11	89.6	77.6	85.0	45.9	95.0	70.4	86.2	54.2	91.9	73.0	85.2
400, 0.25	Sigmoid	Relu	85.3	62.9	88.3	75.6	83.8	41.4	94.8	68.1	85.3	50	91.4	70.7	84.1
500,0.5	Relu	Elu	85.6	62.8	88.9	75.9	84.4	45.8	94.1	70.0	85.6	53	91.5	72.3	84.7
600,0.75	Elu	Swish	85.5	63.8	88.5	76.1	84.1	42.4	94.9	68.6	85.5	50.9	91.5	71.2	84.35

Table 9: Results obtained from combination of simple CNN + Weighted XGBoost

Estimator Learning Rate	FCL	OL	Accuracy	Precision				Recall				F1-Score			
				0	1	M-avg	W-avg	0	1	M-avg	W-avg	0	1	M avg	W-avg
200,0.075	Relu	Softmax	86.9	64.5	91.4	77.6	86.8	57.4	93.2	76.6	86.9	61.5	92.1	76.4	86.5
300,0.1	Linear	Sigmoid	86.3	66.1	89.6	77.8	85.9	46.9	95.0	70.2	86.7	54.5	91.9	73.2	85.5
400, 0.25	Sigmoid	Relu	84.9	61.0	88.1	74.6	83.3	41.2	94.3	67.7	85.0	49.2	91.6	70.1	83.7
500,0.5	Relu	Elu	85.0	60.8	88.5	74.9	83.6	43.8	94.1	68.6	85.0	50.5	91.1	71.3	84.0
600,0.75	Elu	Swish	85.5	63.7	88.5	76.1	84.1	43.4	94.9	68.9	85.5	51.4	91.5	71.2	84.45

Table 10: Results obtained from combination of Weighted CNN + Weighted XGBoost

Estimator Learning Rate	FCL	OL	Accuracy	Precision				Recall				F1-Score			
				0	1	M-avg	W-avg	0	1	M-avg	W-avg	0	1	M avg	W-avg
200,0.075	Relu	Softmax	88.4	68.5	92.4	80.6	88.1	63.4	93.7	78.6	88.4	65.8	92.9	79.4	88.5
300,0.1	Linear	Sigmoid	85.3	62.9	89.	75.8	83.9	41.9	94.7	68.2	85.7	49.8	91.4	70.6	84.1
400, 0.25	Sigmoid	Relu	85.0	61.0	88.5	74.6	83.6	43.2	94.0	68.7	85.0	50.5	91.2	70.8	83.9
500,0.5	Relu	Elu	86.0	64.5	89.5	77.9	84.8	46.9	94.4	70.7	86.0	54.5	91.7	73.1	85.1
600,0.75	Elu	Swish	85.7	65.7	88.2	76.7	84.1	40.4	95.9	68.1	85.7	50.4	91.6	71.0	84.45

Table 11: Accuracy Obtained from build ML + CNN models

Built ML+CNN Models	**Accuracy**
Simple XGBoost	82.84
Weighted XGBoost	81.81
Simple CNN	87.83
Weighted CNN	85.69
Simple CNN + Simple XGBoost	87.20
Simple CNN + Weighted XGBoost	86.86
Weighted CNN + Simple XGBoost	**88.37**

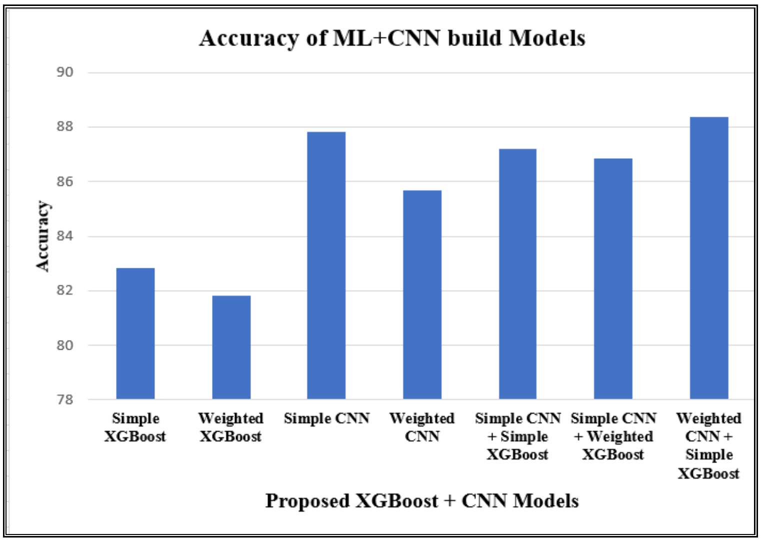

Figure 7: Graphical representation of accuracy obtained from build XGBoost + CNN models

A few existing combinations such as KNN, SVM , DT, Random forest algorithms has been used for the comparison with the results of the proposed method. The table below shows the result obtained and also the graphical view of the same has been displayed.

Table 12: Comparison of proposed method with Existing methods

Machine Learning Models	**Hyper Parameter Grid**	**Accuracy (%)**
KNN	Neighbours= 5, weights=uniform, algorithm=brute, metric= minkowski	78.45
SVM	Degree=3, kernel=rbf, gamma= 0.01, epsilon=0.1	81.8
DT	Maxdepth=7, max features=100, min sample split=12, max leaf nodes=8,	79.24
Random Forest	n_estimators=500, 800, 1000, max_features=auto,sqrt, max_depth=20,30,40 , min_sample_split=5,7,10,15	84.56
XGBoost	Estimator=300, Learning rate=001	83.3
WConvXGB (Proposed Model)	**Weighted CNN with GloVe embedding layers, Weighted XGBoost: eta=0.1, maxdepth:10, n_estimators=300, n_job=-1**	**88.3**

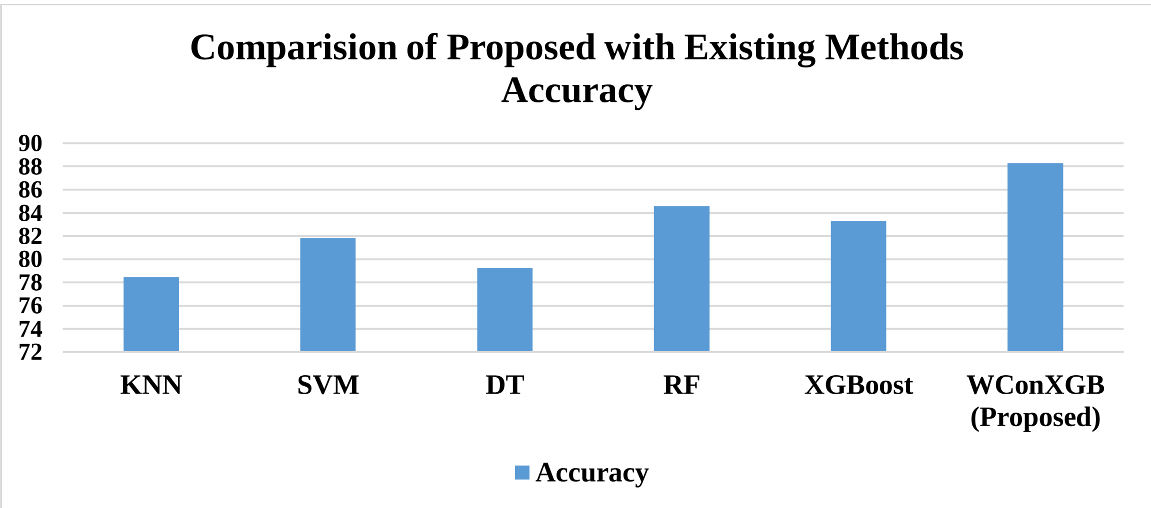

Figure 8: Graphical representation of accuracy obtained from Comparison of proposed method with Existing methods

6. CONCLUSION

This paper proposes and evaluates a new weighted convolutional neural network model with a weighted XGBoost model. GloVe word embedding into the convolution neural network can often be used for transfer learning to predict the hotel reviews outcomes like promising or not promising. The model is built form the scratch to analyse the hotel reviews. Model hyper parameters of CNN and XGBoost are experimented with different values to find the optimised solution. Seven different model combinations are tried to understand the impact of machine learning and deep learning with NLP for analysing the user behaviour on hotel reviews. Among different models weighted CNN and weighted XGBoost model gave significantly better result compared to other state-of-art models.

ACKNOWLEDGMENT

This research was supported by Visvesvaraya Technological University, Jnana Sangama, Belagavi-590018 for Grant of financial assistance.

REFERENCES

1. V. Dhanalakshmi, B. Dhivya and A.M. Saravanan, Opinion mining from student feedback data using supervised learning algorithms. *IEEE 3rd MEC International Conference on Big Data and Smart City.* , pp. 1-5, 2016.
2. Schmunk, S., H¨opken, W., Fuchs, M., Lexhagen, M.: Sentiment analysis: extracting decision-relevant knowledge from UGC. In: Xiang, Z., Tussyadiah, I. (eds.) Information and Communication Technologies in Tourism 2014, pp. 253–265. Springer, Cham (2013).
3. Chang, Y.C., Ku, C.H. and Chen, C.H., 2019. Social media analytics: Extracting and visualizing Hilton hotel ratings and reviews from TripAdvisor. *International Journal of Information Management*, *48*, pp.263-279.
4. Z. Zhang, Q. Ye, Z. Zhang, and Y. Li, .Sentiment classification of internet restaurant reviews written in cantonese,. Expert Syst. Appl., 2011.
5. Y. L. Tsai, Y. C. Wang, C. W. Chung, S. C. Su, and R. T. H. Tsai,. Aspect-category-based sentiment classification with aspect-opinion relation, in 2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2016, pp. 162–169.
6. M. Govindarajan, .Sentiment analysis of restaurant reviews using hybrid classification method,. 2014.
7. Xiang, Z., Schwartz, Z., Gerdes, J. H., Jr., & Uysal, M. (2015). What can big data and text analytics tell us about hotel guest experience and satisfaction? *International Journal of Hospitality Management, 44*, 120–130.
8. Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y. and Gupta, B., 2018. Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. *Journal of computational science*, *27*, pp.386-393.
9. TripAdvisor:https://www.tripadvisor.in/ShowUserReviews-g274887-d667272-r488566942-Carat_Boutique_Hotel-Budapest_Central_Hungary.html, 2020.
10. A.Sharma and S.Dey, A Document-Level Sentiment Analysis Approach Using Artificial Neural Network and Sentiment Lexicons, ACM SIGAPP Applied Computing Review, VOL. 12, pp. 67-75, ACM, 2012.
11. S.Nirkhi, Potential use of Artificial Neural Network in Data Mining, Computer and Automation Engineering, Vol. 2, pp. 339 – 343, IEEE, 2010.
12. B. Gokulakrishnan, P. Priyanthan, T. Ragavan, N.Prasath, A. Perera, Opinion Mining and Sentiment Analysis on a Twitter Data Stream, Advances in ICT for Emerging Regions, pp. 182 – 188, IEEE, 2012
13. Chen, Tianqi, and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp. 785-794. ACM, 2016
14. Chang, Yung-Chun, Chih-Hao Ku, and Chien-Hung Chen. Using deep learning and visual analytics to explore hotel reviews and responses. *Tourism Management* 80 (2020): 104129.
15. Nguyen, Jennifer, Jordi Montserrat-Adell, Núria Agell, Monica Sánchez, and Francisco J. Ruiz. Fusing hotel ratings and reviews with hesitant terms and consensus measures. Neural Computing and Applications 32, no. 19 (2020): 15301-15311.

16. Tsai, Chih-Fong, Kuanchin Chen, Ya-Han Hu, and Wei-Kai Chen.. Improving text summarization of online hotel reviews with review helpfulness and sentiment. *Tourism Management* 80 (2020): 104122.