SENTIMENT ANALYSIS ON INDIAN TOURIST PLACES REVIEWS USING MACHINE LEARNING TECHNIQUES

Jitendra Soni

*Institute of Engineering & Technology, Devi Ahilya Vishwavidyalaya, Indore, M.P.

Abstract

The objective of this research paper is to compare the effectiveness of different natural language processing techniques in classifying Indian tourist place reviews as positive, negative, or neutral. The techniques studied are LSTM based classifier, and Naive Bayes classifier, and the dataset used comprises textual reviews obtained from various online platforms. To determine which technique performs better, we evaluated the performance of these classifiers using a range of metrics, including precision, recall, F1 score, and accuracy. Our findings reveal that outperforming both the LSTM and Naive Bayes classifiers. This study provides valuable insights into the effectiveness of different natural language processing techniques in sentiment analysis and can assist businesses in making informed decisions based on customer feedback.

Keywords: Sentiment Analysis, NLP, LSTM, Deep Learning, Naïve bayes, Machine learning.

Introduction

India is renowned for its diverse landscape and rich cultural heritage, making it a highly sought-after tourist destination. Travelers can experience everything from the scenic beauty of the Himalayan Mountains to the sandy beaches of Goa, as well as the cities of Delhi and Mumbai and the serene backwaters of Kerala. As online platforms and social media continue to play a significant role in trip planning, travelers are increasingly relying on reviews and ratings to inform their decisions. These reviews offer valuable insights into the experiences of previous visitors, aiding potential tourists in planning their itinerary and accommodations. However, analyzing a large volume of reviews manually can be a daunting and time-consuming task.

Consequently, there has been a surge of interest in natural language processing (NLP) techniques to automate this task. Sentiment analysis is one of the most prevalent tasks in NLP, as it involves determining the sentiment or emotion conveyed in a piece of text. By classifying reviews as positive, negative, or neutral, sentiment analysis can enable tourism businesses to enhance their services based on customer feedback. Additionally, sentiment analysis has diverse applications, including customer service, market research, and social media monitoring. It can provide valuable insights into how customers feel about a product or service, allowing businesses to make informed decisions to enhance customer satisfaction.

The purpose of this research paper is to conduct a comparative analysis of NLP techniques - LSTM based classifier, and Naive Bayes classifier, with the aim of accurately classifying Indian tourist place reviews as positive, negative, or neutral. Our dataset comprises of textual reviews from diverse online platforms, and we seek to enhance the existing knowledge on NLP techniques by examining their efficacy in analyzing Indian tourist reviews. This research can aid tourism businesses in improving their services based on customer feedback, and contribute to the development of more effective NLP techniques for sentiment analysis, which can be extended to other domains beyond tourism.

The following paragraph provides an overview of some of the existing literature in the field related to our research. the author proposed a hybrid CNN-LSTM deep learning model for tourism destination management using sentiment analysis. Naïve Bayes algorithm was used by the author to classify tweets related to tourist destination into positive or negative. The algorithm worked pretty well and achieved the desired goal. A comparative study was also done by the author where they compared the performance of Random Forest classifier and Support Vector Machine on a dataset of Reviews of Indian tourism. They found out that Random Forest outperformed SVM in terms of accuracy and execution time.

Materials and Methods

Data Collection

The data we used is extracted from various travel sites like trivago, TripAdvisor etc. in the form of Reviews. The dataset was taken from the data science website called Kaggle. A total of 142761 reviews were collected

The following table shows the structure of the raw dataset.

Table 1. Structure of the extracted dataset

COLUMN NAME	DESCRIPTION
City	City of the tourist place
Place	The Name of tourist place
Review	Textual review of the place
Rating	Rating given by the user
Name	Name of the reviewer
Date	Date of review posting

From this dataset we will drop the columns city, place, Name and Date, since we only need content and Rating for the sentiment analysis.

Pre-processing

The Text posted on the internet contains a lot of noise and information which of almost no use [1]. This raises the dimensionality of the problem and makes the classification problem more difficult.

The algorithms which are most used to polish and prepare data extracted from online sources includes, lowering the case, punctuation removal, lemmatization or stemming, tokenization as shown in [2] and [3].

Step 1 – Convert the text to lower case. Upper case text might increase the intensity of the text [4].

Step 2 – Remove the numbers in the text.

Step 3 – Remove all the punctuations from the text. They generally don't contribute to the sentiment of a text. We will remove them to reduce noise. [4]

Step 4 – Remove the stop words from the text. These can result in less accurate classifier model. [4]

Step 5 – Handle the Emoticons in the data. Emoticons, though they may seem useless but can contribute to the sentiment of the text. [5]

Step 6 – Perform Lemmatization on the text. Lemmatization is the process of reducing a word to its base or root form. The goal of lemmatization is to group together different forms of a word so they can be analyzed as a single item. [6]

Step 7 – Negation Handling. Dealing with negation is a critical step in sentiment analysis [4]. We will convert words like wouldn't, can't etc. to would not, cannot etc.

We know that, our reviews have rating associated with them, the following table and chart shows the number of reviews belong to each rating

Table 2. Number of tweets per rating

RATING	NUMBER OF REVIEWS
5	33938
4	27629.
3	33106
2	39267
1	8821

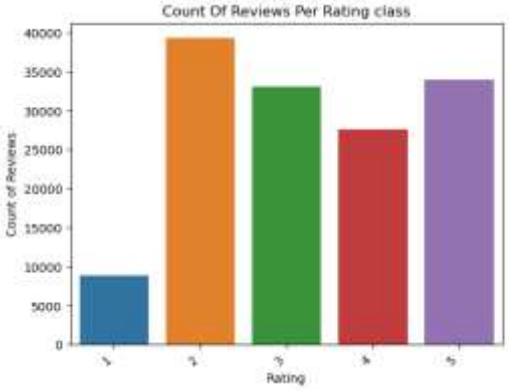


Figure 1. Bar Graph Showing Number of tweets per rating

In order to conduct sentiment analysis, reviews must be classified into one of three categories: positive, negative, or neutral. In our case we will consider reviews with a rating below 3 as negative, those with a rating above 3 as positive, and those with a rating of 3 as neutral. The resulting graph below illustrate the distribution of reviews among these new categories.

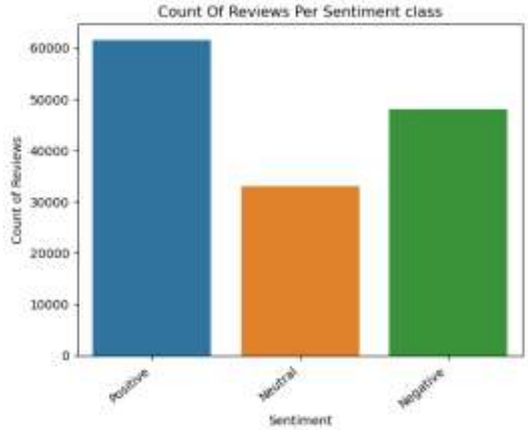


Figure 2. Bar Graph Showing Number of tweets per sentiment

As we can see there is a class imbalance and it is going to affect our training. To overcome this, we will now perform Data Augmentation.

Data augmentation

So why are we dealing with this? Class imbalance degrades the performance and accuracy of the machine learning technique as the overall accuracy and decision making will be biased towards the majority classes and the minority classes are wrongly identified .

To overcome this, we will oversample our minority classes to match our majority class using a python library NLPAug. The technique we will be utilizing is called Synonym replacement, which has been proven in to increase performance significantly.

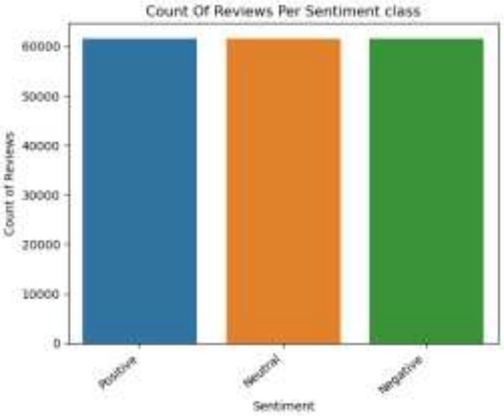


Figure 3. Bar Graph Showing Number of tweets per sentiment after Augmentation

LSTM based sentiment analysis

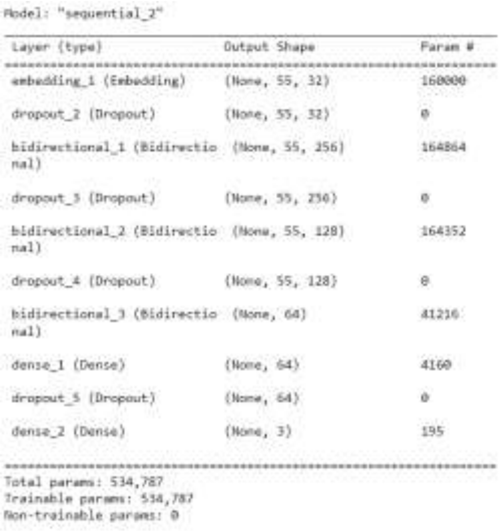
In recent times, the standard LSTM architecture networks "have become the state-of-the-art models for a variety of machine learning problems" [9].

Bidirectional LSTM (BiLSTM) is a type of recurrent neural network (RNN) that can process the input sequence in both forward and backward directions. This allows the network to capture not only the context preceding the current word but also the context following it. This is particularly useful for sentiment analysis because sentiment often depends on the context in which words are used.

To perform sentiment analysis with a BiLSTM, the first step is to preprocess the input data. This involves tokenizing the text and converting the tokens to numerical vectors. This can be done using techniques known as word embedding.

The next step is to feed the preprocessed data into the BiLSTM network. The network will learn to identify patterns in the input data that are associated with positive, negative, or neutral sentiment. The output of the BiLSTM network will be a probability distribution over the three sentiment categories.

Once the sentiment probabilities have been obtained, a decision can be made about the overall sentiment of the text. This can be done by choosing the sentiment category with the highest probability and can be classified as positive, negative, or neutral based on the probability.



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 55, 32)            160000

 dropout_2 (Dropout)         (None, 55, 32)            0

 bidirectional_1 (Bidirectio  (None, 55, 256)          164864
 nal)

 dropout_3 (Dropout)         (None, 55, 256)           0

 bidirectional_2 (Bidirectio  (None, 55, 128)          164352
 nal)

 dropout_4 (Dropout)         (None, 55, 128)           0

 bidirectional_3 (Bidirectio  (None, 64)               41216
 nal)

 dense_1 (Dense)             (None, 64)                4160

 dropout_5 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 3)                 195

=================================================================
Total params: 534,787
Trainable params: 534,787
Non-trainable params: 0
```

Figure 5. LSTM Model for sentiment analysis

Machine learning using naïve bayes

Vectorization of text before Naïve Bayes – We need to convert our text data into machine understandable numbers, for this we will utilize TF-IDF vectorizer [15] which will assign weights to each word based on their frequency. Now the data is ready to be analyzed. The following formula is used for this,

$$W_{x,y} = tf_{x,y} * \log(N/df_x)$$

Where,

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

Now we apply naïve bayes algorithm. The algorithm works by first calculating the probability of each word or phrase appearing in a positive or negative sentiment class based on the frequency of occurrence in a training dataset. Then, for a new text sample, the algorithm calculates the probability of the sample belonging to each sentiment class based on the frequency of occurrence of the words in the sample. The sample is then classified as the sentiment class with the highest probability. The probability is calculated with the following formula,

$$P(A|B) = \{P(B|A) * P(A)\}/P(B)$$

One of the advantages of using Naive Bayes for sentiment analysis is its simplicity and efficiency. It requires relatively little training data and computational resources, making it a useful tool for quickly analyzing large volumes of text data. Additionally, Naive Bayes can be easily updated as new data becomes available, allowing the model to adapt to changing sentiment patterns over time.

Overall, Naive Bayes is a useful algorithm for sentiment analysis, especially in situations where simplicity and efficiency are important considerations. However, it may not be the best choice for tasks that require more nuanced analysis of language as will observe in our results later.

Metrics for evaluation

To assess the effectiveness of our model, we will employ a set of evaluation metrics, including Accuracy, Precision, Recall, and F1-score.

Accuracy is calculated as the percentage of correctly predicted data, which is determined by dividing the total number of correct predictions by the total number of predictions made. The following equation demonstrates how to calculate Accuracy:

$$Accuracy = (Tp + Tn)/(Tp+Tn+Fp+Fn)$$

Recall is the percentage of correctly predicted positive data, the equation shows how to calculate it.

$$Recall = (Tp)/(Tp+Fn)$$

Precision is the percentage of positive data predicted as positive; the equation shows how to calculate it.

$$Precision = (Tp + Tn)/(Tp+Fp)$$

F-Score is a representation of recall and precision, the equation shows how to calculate it.

$$F1\text{-}score = (2*P*R)/(P+R)$$

Here,

Tn – True Negative, Tp – True Positive, Fp – False Positive, Fn – False Negative, P – Precision, R – Recall

Results and Discussion

After training our model on the data, we evaluate our model on the parameters mentioned earlier.

The accuracy of the models is shown below

Table 3. Accuracy measure of models

MODEL	ACCURACY
Naive Bayes	63.08%
LSTMs	86.6%

The other measures, namely Precision, Recall and F1-score of the models are indicated below.

Table 4. Precision, Recall and F1 score

MODEL	Precision	Recall	F1 score
Naive Bayes	0.67	0.63	0.61
LSTMs	0.86	0.89	0.87

Conclusions

In conclusion, our Paper compared the performances of popular algorithms for sentiment analysis: BiLSTM, and Naive Bayes.

These findings have important implications for the tourism industry, as sentiment analysis can provide valuable insights into the experiences of tourists at different destinations. By accurately classifying the sentiment of reviews, tourism boards and businesses can identify areas for improvement and tailor their offerings to better meet the needs and expectations of visitors.

Furthermore, our research highlights the importance of considering the specific context and language used in the text when selecting an approach for sentiment analysis. Indian tourist place reviews are likely to contain specific cultural references and nuances that may require algorithms trained on similar datasets.

References

[1] Haddi, E., Liu, X., Shi, Y. (2013), The role of text pre-processing in sentiment analysis. Procedia Computer Science 17, 26–32.

[2] Duncan, B., Zhang, Y. (2014), Neural networks for sentiment analysis on twitter. In: Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on. pp. 275–278. IEEE (2015).

[3] Meng, Xinfan, et al. "Cross-lingual mixture model for sentimentclassification." Proceedings of the 50th Annual Meeting of theAssociation for Computational Linguistics Volume 1,2012

[4] Pang and L. Lee. Opinion mining and sentiment analysis.Foundations and Trends in Information Retrieval, 2(1-2):1{135,2008.

[5] Socher, Richard, et al. "Recursive deep models for semanticcompositionality over a sentiment Treebank." Proceedings of theConference on Empirical Methods in Natural Language Processing(EMNLP). 2013.

[6] Peter Halacsy, 2006. Benefits of deep NLP-based lemmatization for information retrieval. Benefits of deep NLP-based lemmatization for information retrieval. CLEF.

[7] Shaza M. Abd Elrahman and Ajith Abraham, (2013), A Review of Class Imbalance Problem, Journal of Network and Innovative Computing ISSN 2160-2174, Volume 1

[8] Dmitry Davidov, Ari Rappoport." Enhanced Sentiment Learning Using Twitter Hashtags and Smileys". Coling 2010: Poster Volumepages 241{249, Beijing, August 2010

[9] Zhou Zhao, Hanqing Lu, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Microblog sentiment classification via recurrent random walk network learning. In IJCAI, volume 17, pages 3532–3538

[10] Duyu Tang, Furu Wei, Bing Qin, Nan Yang, Ting Liu, and Ming Zhou. 2016. Sentiment embeddings with applications to sentiment analysis. IEEE Transactions on Knowledge and Data Engineering, 28(2):496–509.

[11] Zhou Zhao, Hanqing Lu, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Microblog sentiment classification via recurrent random walk network learning. In IJCAI, volume 17, pages 3532–3538

[12] M. Cliche, BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs [J], 2017.