Performance Evaluation of Machine Learning Models for Sentiment Analysis on Hotel Reviews

Dev Kalavadiya
Zaeem Shahzad

Abstract

Customer feedback is crucial for maintaining the reputation of a product as it is for a business. It is used to identify the subject's neutral, positive, and negative emotions. Like no other sector, hospitality heavily depends on consumer satisfaction and favorable ratings. Reviews can ultimately decide the fate of a hotel as well as the hoteliers. In this paper, we explore a few machine-learning models to analyze sentiments on hotel reviews. We use Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) models and compare them to see which one is more accurate.

1 Introduction

Sentiment analysis involves textual mining to extract valuable subjective data and comprehend societal attitudes towards a brand to assure quality and help improve the company and its products. Depending on different individuals, opinions on the product's utility vary.

Like no other sector, hospitality depends on consumer satisfaction and favorable ratings. Reviews are fundamental for hoteliers, given their power to make or break the establishment - meaning that it's crucial for hotel owners and managers to ensure that they receive as many positive online hotel reviews as they can. In a large-scale business, it is important to be able to efficiently parse through thousands of reviews and determine the overall sentiment around the business.

To analyze user sentiments, a variety of technologies are used. In this project, we look into deep learning techniques to analyze sentiments on hotel reviews. We explore Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) models and their efficiency in sentiment analysis. The final results are shown and discussed in the sections: Experiments and Results.

2 Related Work

In his paper "Sentiment Analysis using Neural Network and LSTM", Akana Srinivas applied deep learning techniques to perform sentiment analysis on tweets. Simple Neural Network, Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN) methods are applied for the sentiment analysis and their performances are evaluated. The LSTM was the best-performing model among all the proposed techniques with the highest accuracy of 87% in the paper.

Whereas Fengdong Sun, Na Chu, and Xu Du used LSTM in their paper "Sentiment Analysis of Hotel Reviews Based on Deep Learning" to analyze sentiment tendencies. Firstly, word2vec and word segmentation techniques were used to process the short comment text into the LSTM network and a dropout algorithm was added to prevent overfitting to get the final classification model. Using the unique characteristics of short-term memory of LSTM network, they were able to achieve a good accuracy rate of more than 95% on sentiment classification of hotel reviews.

In the study by Mullen and Collier (2004), the authors used SVMs for sentiment analysis by incorporating diverse information sources, such as word lists and contextual features, into the feature space. They found that using multiple information

sources improved the performance of the SVM classifier compared to using a single source.

Paltoglou and Thelwall (2010) conducted a study on the impact of different information retrieval weighting schemes on the performance of SVMs for sentiment analysis. They compared several schemes, including term frequency-inverse document frequency (TF-IDF), information density, and document-level sentiment, and found that the TF-IDF sentiment scheme performed the best.

3 Data

To evaluate the efficiency of the different deep learning techniques we want to explore, we will employ two data sets we found on Kaggle.com (**Set A**, **Set B**). Both Set A and Set B contain sentiment measures for reviews left on hotel booking websites. There are multiple columns in both datasets, but we are interested in two of them. For every row in the set, there is one column for the review text and another column for the sentiment measure. Set A has 38932 rows, while Set B has 20491.

There is one difference between the sets. Set A has Sentiment Analysis recorded as either "Happy" or "Not Happy," while Set B has Star Ratings ranging from 1 (Highly Negative) to 5 (Highly Positive). Thus, in pre-processing the data, we will standardize the first column in both data sets - which is the Sentiment Analysis measure - to only include a boolean value (0 - Not Happy, 1 - Happy). Set A can be converted to this format easily; however, for Set B, we will select Ratings of 2 and lower as Unhappy and 4 and higher as Happy. We will ignore the rows with ratings of 3 in Set B since we do not consider neutral sentiment scores in this experiment.

We then merge Set A and Set B into one large data set. This larger data set is split in the following proportions to create the Training, Development, and Test Corpus:

- 80% allocated to Training/Development.
- 20% allocated to Test

In the above splits, Training and Development are considered one corpus since we implement a K-fold cross-validation in the training process. Whereas for the 20% allocated to Test, we create two files: one to retain the Sentiment Analysis column while the second one excludes it. This is because we want to compare our system's output of Sentiment Analysis to the analysis in the original data set to evaluate its performance.

4 Methodology

To efficiently tackle the problem, we are following a rigorous approach starting with data pre-processing and moving on to feature engineering, training the models, cross-validation, and then scoring to evaluate the efficiency of every model separately.

4.1 Pre-processing:

4.1.1 Text cleaning:

To avoid any inconsistency, text cleaning is required. All words are converted to lowercase to avoid any difference that may result from the case sensitivity. Special characters alongside punctuation and numbers were removed since our model's purpose is to analyze the meaning conveyed by words. Removing special characters automatically eliminates "emotion symbols" that may be included in reviews (i.e, use of punctuation to convey emotion).

4.1.2 Text Tokenization:

As a second step, text tokenization allows us to break down a given text into smaller units. In this project, we have used **tokenization** combined with **stemming** to not only split the sentence into tokens but also to reduce these tokens into their stem. To do that, we opted for the Porter Stemming algorithm, which is advantageous in terms of space and time complexity.

4.2 Feature Engineering:

The next step consists of feature engineering. Using the **TF-IDF Vectorizer**, we transformed the cleaned text into a feature vector. Choosing the TF-IDF was based on the fact that it would account for the importance of every word in a given sentence and then improve our sentiment analysis whereas using a simple Count Vectorizer would

only store the count of the words. This allows us to maintain TF-IDF scores for each word in all the reviews based on the Training/Development cross-validated corpus. We used TF-IDF vectors as input to SVM whereas we transformed each review text to sequence of integers for CNN and LSTM.

4.3 Machine Learning Models

We will evaluate the efficiencies of the following deep learning models.

4.3.1 Support Vector Machines (SVM):

Support Vector Machines is a type of supervised machine learning algorithm that can be used for classification or regression tasks. SVM works by finding the hyperplane in a high-dimensional space that maximally separates the different classes (binary classification: positive or negative sentiment values in our case). This hyperplane is known as the decision boundary, and the points closest to it are called support vectors. The SVM algorithm aims to maximize the distance between the decision boundary and the support vectors, called the margin, in order to improve the model's generalization ability.

SVMs can handle nonlinear data by using kernel functions, which transform the data into a form where it can be linearly separated. This allows SVM to perform well on complex, real-world datasets. Additionally, SVM has the ability to handle high-dimensional data, such as text or image data, by using the kernel trick.

4.3.2 Long Short-Term Memory (LSTM):

Long Short-Term Memory (LSTM) neural networks are a type of recurrent neural network that is capable of learning long-term dependencies in data. This is achieved through the use of "memory cells" that can retain information for an extended period of time and gates that regulate the flow of information into and out of the cells.

LSTMs have been successfully applied to a variety of tasks, including natural language processing and speech recognition. They have also been shown to perform well on problems with long-term dependencies, such as predicting the next word in a sentence. We believe that this persistence of information will help us generate sentiments by understanding the context of each sentence in a review.

4.3.3 Convolutional Neural Networks (CNN):

Convolutional neural networks (CNNs) are a type of artificial neural network that is commonly used in computer vision tasks. Unlike traditional neural networks, which operate on a flat, fully-connected layer, CNN's use a series of layers that are designed to mimic the structure of the visual cortex in the human brain.

Each layer in a CNN is made up of a set of "filters" or "kernels" that are used to detect specific features in an image. For example, a filter might be designed to detect edges in an image, shapes, or colors. As the image passes through the network, each filter generates a "feature map" that encodes the presence of the corresponding feature in the input image.

The output of each layer is then fed into the next layer, where the filters are used to detect more complex features based on the information encoded in the previous layer. This hierarchical structure allows CNNs to automatically learn complex, high-level features from raw image data without the need for manual feature engineering.

In this way, a "convolution" is a window that slides over a larger input data with an emphasis on a subset of the input matrix. This is how CNN's are applicable for our task of sentiment analysis in NLP - model training over the emphasized text and its sentiment.

5 Experiments:

5.1 Model Configuration

We used scikit-learn and TensorFlow Keras libraries to configure our models.

To configure the SVM model, we used the Radial Basis Function (RBF) kernel, which computes the similarity between two points by measuring how close they are with each other. It can be represented as following:

$$K(X_1, X_2) = exp(-\frac{||X_1 - X_2||^2}{2\sigma^2})$$

We configured the CNN model, as a Sequential model, comprised of the following layers:

- Embedding Layer
- 1-D Convolution Layer
- Max Pooling Layer 1-D
- Flatten Layer
- Dense Layer (10 units)
- Dense Layer (1 unit)

The inputs for the CNN model are a sequences of integers transformed from the review text. We used this instead of the TF-IDF vectors because the text sequences resulted in a higher cross-validation and test accuracy.

To configure the LSTM model, we used the following layers in the Sequential model:

- Embedding Layer
- LSTM Layer
- Dropout Layer
- Dense Layer (1 unit)

The inputs for the LSTM model are same as the CNN model.

5.2 Cross-validation

For every machine learning model, we used the K-fold cross validation technique to have an understanding of how the model performs in different splits of the training data.

We split the training set into K folds of equal size and for K times, we created a new model and trained it on K-1 folds in which one fold was used for testing (as the validation set).

The performance measure of the cross-validation, is the average of different metrics from all K iterations. After performing a 5-fold cross validation, we received the following results:

Model	F-measure	Accuracy	MSE
SVM	94.05%	91.19%	0.0881
CNN	89.91%	84.86%	0.114
LSTM	90.34%	84.90%	0.1139

6 Results:

We evaluated the performance of the three different models mentioned above for sentiment analysis on test data and the results of our experiments are summarized in the graph below.

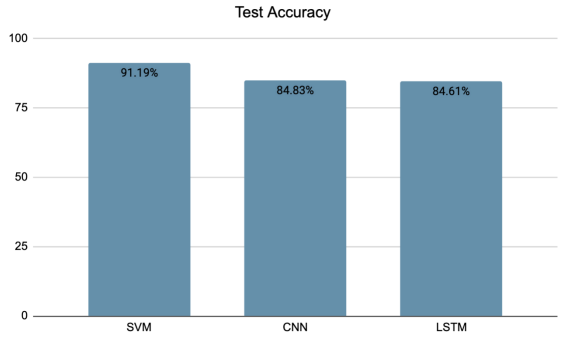

Figure 1: Test Accuracy for the models used.

As shown in the graph, the SVM model achieved the highest accuracy of 91.19%. The CNN and LSTM models had slightly lower accuracy, with scores of 84.83% and 84.61%, respectively.

A confusion matrix is a useful tool for evaluating the performance of a classifier. It allows you to see how well the classifier is able to predict the correct class for each instance in the test set.

One of the advantages of using a confusion matrix is that it provides a more detailed breakdown of the classifier's performance than just overall accuracy. For example, you can see how many true positive and true negative instances were correctly classified, as well as how many false positive and false negative instances were produced. This information can be helpful in identifying any weaknesses or biases in the classifier.

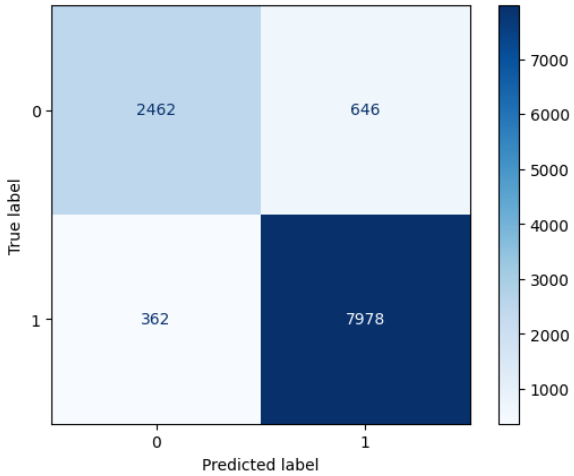

Figure 2: Confusion Matrix for the SVM model

The confusion matrix (in Figure 2) shows the number of true positive and true negative instances

that were correctly classified as positive and negative, respectively. The classifier achieved a positive accuracy of 7978 and a negative accuracy of 2462, for an overall accuracy of 91.9%. In addition to overall accuracy, we also calculated precision, recall, and F1 score to provide a more complete evaluation of the classifier's performance.

F1 score is a metric that can be used to evaluate the performance of a model for sentimental analysis of hotel reviews, particularly if the model is being used for a classification task. The F1 score is a combination of precision and recall, which are both important evaluation metrics in the context of classification tasks.

$$F_1 = \frac{Precision \times Recall}{Precision + Recall}$$

Mean Squared Error (MSE) is a commonly used evaluation metric in machine learning. It is a measure of the difference between predicted values and actual values, and it is commonly used to evaluate the performance of regression models. In the context of sentimental analysis, a regression model might predict a numerical value (such as a score on a scale from -1 to 1) that represents the sentiment of a given review. In this case, MSE could be used to measure how closely the model's predictions match the actual sentiment of the reviews.

Metric	Value
Accuracy	91.19%
Precision	92.51%
Recall	95.66%
F1	94.06%
MSE	0.087

Figure 3: SVM

Metric	Value
Accuracy	85.25%
Precision	87.73%
Recall	92.71%
F1	90.15%
MSE	0.111%

Figure 4: CNN

Metric	Value
Accuracy	85.90%
Precision	87.93%
Recall	93.48%
F1	90.62%
MSE	0.1030%

Figure 5: LSTM

Overall, our results demonstrate the effectiveness of using machine learning approaches, such as SVM, CNN, and LSTM in sentiment analysis of hotel reviews.

7 Discussion:

The paper "Sentiment Analysis using Neural Network and LSTM" (Srinivas et al, 2021) addresses the need for neural networks to improve performance in sentiment analysis and has found that LSTM performed the best. Although we expected similar results, our research indicates that SVM performed better than LSTM and CNN.

To verify our results were correct, we tried using TF-IDF scores (Paltoglou and Thelwall, 2010) as inputs for CNN and LSTM instead of text sequences, but the performance did not improve. This is because SVMs are a type of supervised learning algorithm that are most-suitable for classification tasks, and they work by finding a decision boundary that maximally separates different classes in the training data. CNN and LSTM networks, on the other hand, are particularly well-suited for processing sequential data, such as text.

In line with Karanasou et al, one of the biggest limitations of studying sentiment analysis on hotel reviews is the subjectivity of human opinions and the potential for different people to interpret the sentiment of a given review differently. This can make it challenging to accurately assess the sentiment of a review, particularly if the review contains mixed or nuanced sentiment via the use of figurative language such as metaphors, irony, and/or slang.

Another limitation, is the quality of the training data that can impact the performance of the model. For example, if the reviews in the dataset contain

spelling errors or are written in a language that is not well-represented in the training data, this could lead to lower performance.

Finally, there is the potential for error in the model itself, such as overfitting to the training data or underfitting to the test data. Careful model selection and hyperparameter tuning can help to mitigate these issues, but it is important to be aware of these potential sources of error and to carefully evaluate the performance of the model in order to ensure reliable and accurate results.

8 Conclusion:

In conclusion, the results of our experiments show that the SVM model outperformed the CNN and LSTM models in terms of accuracy, achieving a score of 91.19%. The CNN and LSTM models had slightly lower accuracy scores of 84.83% and 84.61%, respectively.

We also used a confusion matrix to provide a more detailed breakdown of the classifier's performance, including precision, recall, and F1 score. These results demonstrate the effectiveness of using machine learning approaches, such as SVM, CNN, and LSTM, in sentiment analysis of hotel reviews. However, it is worth noting that the relative performance of the models may vary depending on the characteristics of the dataset and the specific parameters used.

9 Future Work:

If computational resources allow it, the next step will be to add **hyperparameter tuning** to find the optimal parameters which define the model architecture on our dataset. This can be achieved using a specific hyperparameter optimization algorithm such as Grid Search, Random Search, and Bayesian Optimization. Using the right hyperparameter tuning algorithm could increase the accuracy of our models even further. Understanding the strengths and limits of the integration of each hyperparameter tuning technique will certainly be beneficial to our sentiment analysis.

Additionally, it is worth investigating more machine-learning algorithms and challenging their performance. As an example, **logistic regression** is worth being evaluated. Building a logistic regression model could give us insight into sentiment analysis depending on how accurately and rapidly it performs.

Valence Aware Dictionary for Sentiment Reasoning (VADER) is worth evaluating as well since it doesn't require any previous training given that it is a Natural Language Toolkit package.

10 Acknowledgements:

The authors would like to thank Professor Adam Meyers for providing us with the platform to work on this project i.e. the Special Topics: NLP class at NYU. We would also like to thank our mentor Ziyang Zeng for his helpful comments, feedback, and overall guidance.

References

[1] Akana Chandra Mouli Venkata Srinivas et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1074 012007 DOI 10.1088/1757-899X/1074/1/012007

[2] F. Sun, N. Chu and X. Du, "Sentiment Analysis of Hotel Reviews Based on Deep Leaning," 2020 International Conference on Robots & Intelligent System (ICRIS), 2020, pp. 627-630, doi: 10.1109/ICRIS52159.2020.00158.

[3] Maria Karanasou, Christos Doulkeridis, and Maria Halkidi. 2015. DsUniPi: An SVM-based Approach for Sentiment Analysis of Figurative Language on Twitter. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 709–713, Denver, Colorado. Association for Computational Linguistics.

[4] Georgios Paltoglou and Mike Thelwall. 2010. A Study of Information Retrieval Weighting Schemes for Sentiment Analysis. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 1386–1395, Uppsala, Sweden. Association for Computational Linguistics.

[5] Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2019. Sentiment Analysis Is Not Solved! Assessing and Probing Sentiment Classification.

In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 12–23, Florence, Italy. Association for Computational Linguistics.