IT1244 Project Report on Sentence Sentiment Analysis

By Daniel Cheng, Lee JunYuan, Li Yihan and Wang Lifu

Abstract

Sentence sentiment analysis is a powerful artificial intelligence method that can detect the sentiment, or emotional tone of a sentence. It has a wide variety of uses in social media, business models, and other fields of work where the sentiments of the customers are important. In this project, a natural language processing (NLP) model to perform sentence sentiment analysis was designed by implementing five different methods, including logistic regression, k-Nearest Neighbors (KNN), Support Vector Machine (SVM), Naive Bayes (NB), and Long Short-Term Memory (LSTM). The methods were compared and evaluated using the confusion matrix. The LSTM model exhibited good consistency and achieved a high accuracy of 73-76%. Overall, while a successful NLP model for sentence sentiment analysis was implemented, the model could be further improved by expanding the dataset and including more varied sentences.

1. Introduction

Sentence sentiment analysis is a powerful artificial intelligence method that can detect the sentiment, or emotional tone of a sentence. As online comments and reviews are becoming an increasingly important source of information, identifying and classifying sentiments carried by these sentences can provide valuable insights and data, which have a variety of uses. For instance, businesses can use sentiment analysis to gain insights from customer feedback about products or services they provide, allowing them to make more informed decisions. Another use of sentiment analysis is in social media, to determine how people feel about a particular brand or topic. This information can then be used to identify trends and opportunities, or in targeted advertising. Other uses of sentiment analysis include political polling to gauge public opinion on various issues, or in market research to understand consumer behavior and preferences. This makes sentence sentiment analysis a useful tool to improve products, services and overall customer experience.

Prior works on sentence sentiment deduction implemented traditional supervised machine learning algorithms such as logistic regression, k-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Naive Bayes (NB) (Poornima and Priya, 2020). However, these models have some drawbacks. Firstly, KNN and SVM suffer from requiring too much memory space, making them unsuitable for analyzing large datasets. Secondly, when the dimensionality of the data increases, SVM suffers severely from overfitting bias. Thirdly, logistic regression and SVM can only perform binary classification, failing to classify sentences into more finely divided categories. Lastly, these algorithms did not take the order of words into account, which plays an impotent role in sentence sentiment.

On the other hand, more advanced deep learning models, such as Long Short-Term Memory (LSTM), allows longer and more complex sentences to be analyzed, making it a more appropriate model to address real life situations. It can also overcome overfitting by increasing the size of training data, which is readily available due to influx of data in a modern Internet world. It is also able to perform multi-class classification, allowing more than two labels beyond merely positive and negative. Additionally, it takes the sequence of words appearing in the sentence into account, enabling more accurate classification.

In this project, our group aims to develop a natural language processing (NLP) model that is able to classify sentences based on their sentiments into two groups: positive and negative. Five different sentiment analysis models based on content learnt in IT1244 lectures were implemented and tested, namely logistic regression, KNN, SVM, NB, and LSTM. The methods were then evaluated and compared using the confusion matrix. Other factors, such as variability and overall runtime of the model were also considered in the comparison.

2. Dataset

2.1 Sources of Dataset

The dataset used is the *3.2.2 Sentence Sentiment* dataset provided by IT1244 project coordinator. It is a CSV file that contains sentences labelled by binary sentiment labels, with 1 being positive and 0 being negative.

2.2 Preprocessing of Dataset

Pre-processing of the dataset was required before it was passed to the classification algorithm. For all models, stopwords were removed using nltk library. All characters were set to lowercase. Additionally, special characters and punctuations were removed to ensure that the dataset of words would be relevant and were more likely to carry either positive or negative sentiments. The data was then randomly split into training and testing data in an 8:2 ratio in order to assess the variability of the model across different datasets.

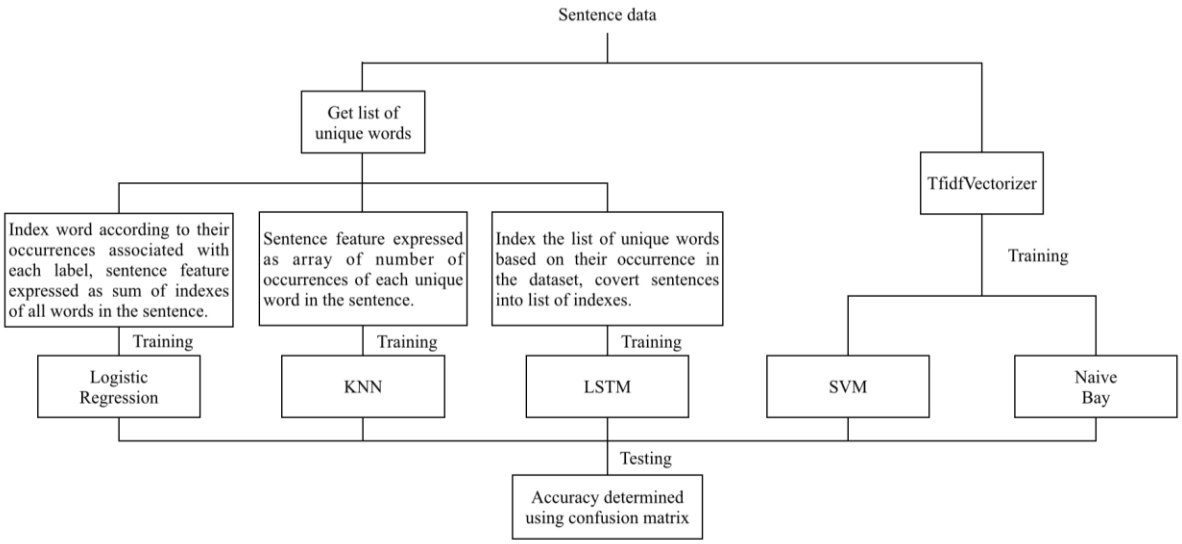

Figure 1: Flow cart of the five models implemented in the project

3. Models

Five different classifier algorithms including logistic regression, KNN, SVM, NB and LSTM were implemented and their metrics based on the test data were compared. Among them, only SVM and NB were not taught in the module.

3.1 Logistic Regression Model

The logistic regression model (Mashalkar, 2021) was implemented and tested as it is the foundation of many sophisticated models that utilize neuron networks (Jurafsky et al., 2009). A dictionary containing the number of occurrences of each word in the dataset under each label was created. Under each label, the numbers of occurrences of all the words in the sentence were summed to be used as the feature for the sentence. The training data was then passed to a logistic regression model, and the testing data was used to assess the accuracy of the model.

3.2 KNN model

The KNN model (Butt, 2022) was chosen as it is straightforward and does not require pre-training. For this model, a list of unique words from the preprocessed sentences was obtained. Each sentence was represented as an array of the number of times of each unique word appearing in the sentence. The training data part of this feature matrix was passed to the KNN model from the Scikit-learn package, and the testing data was classified using the model. The accuracy of the model was plotted against the value of k used.

3.3 SVM model

SVM learns from labelled vectors to generate a hyperplane to classify vectors based on their labels (Ray, 2023). With its ability to handle high dimension data, it does not require feature extraction and dimensionality reduction, allowing it to preserve more information for more accurate classification. Each word in the sentences were vectorized using Term frequency-Inverse Document Frequency (TF-IDF). The training data was passed to the SVM model from the sklearn package and the accuracy was assessed by testing data.

3.4 NB model

The NB model was chosen due to its low demand of training data, high computational speed and excellent ability to handle high dimensional data, which is usually the case for NLP (Yildirim, 2020). TF-IDF was used to vectorize each word in the sentences. The training data was passed to the NB model from the Sklearn package and the testing data was used to calculate the accuracy.

3.5 LSTM model

The LSTM model was chosen as it takes order of words into consideration, and performs especially well in classifying long texts. A list of unique words from the preprocessed sentences was obtained and sorted according to their frequencies in the dataset in a decreasing order (Team). Each word in the sentences were then indexed using this frequency list. The training data and testing was then passed to the LSTM model from Keras and the accuracy of the model was plotted against each epoch.

4. Results and Discussion

4.1 Details for each model

4.1.1 Logistic Regression Model

For the logistic regression model, the reported accuracy fluctuated between 50-75%, depending on the training and testing datasets generated. The total runtime of the gradient descent algorithm was less than 1 second, which meant that the logistic regression was the fastest of all the models used here, probably due to its low amount of computation required.

4.1.2 KNN Model

For the KNN model, the highest accuracy value of around 60.2% was achieved at a k-value of 3, and steadily decreased as the k-values increased. However, the accuracy and the optimal k-value would also fluctuate depending on the training and testing datasets generated. It was also noted that all even k-values had a generally lower accuracy compared to the odd k-values, which was to be expected as a result of ties, causing the model to pick a class at random (Figure 2). The total runtime of the algorithm was 20 seconds for the optimal k-value, 3, indicating a low efficiency.

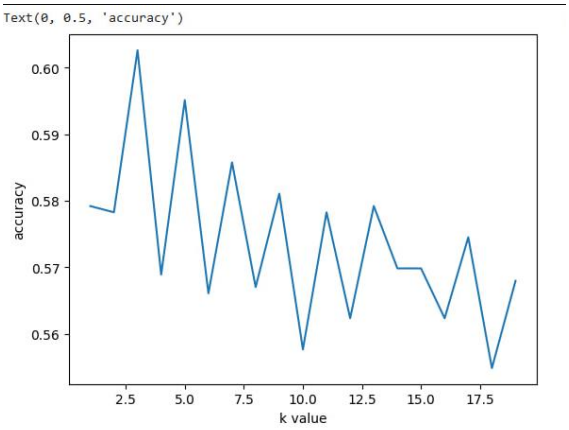

Figure 2: Graph of accuracy of KNN model against k-values. The model achieved the highest accuracy at k=3 with an accuracy of 60.3%.

4.1.3 SVM Model

For the SVM model, the accuracy was 78.9% when the kernel "linear" was used, and was close to 80% for the kernel "poly". Despite the minimal improvement in accuracy, training time was doubled. Kernel refers to the type of function used to adjust the dimensionality of the data to generate the hyperplane. In NLP, as the dimension of the data used usually was already high, further increasing the dimension does not help much in improving the accuracy. Therefore, it is more practical to use a linear function as a kernel as compared to a polynomial kernel.

4.1.4 NB Model

For the NB model, the accuracy was around 75% with little fluctuation, depending on the training and testing data given. The runtime was less than 1 second, comparable to that of logistic regression due to the small amount of calculation. Yet, the NB model was still able to exhibit a higher accuracy.

4.1.5 LSTM Model

For the LSTM model, it achieved an accuracy between 73-76%. With 3 epochs, the total runtime of LSTM was one minute, but it could perform fast prediction once after the model was trained. It was also noted that while the model reached a high accuracy of around 85-95% with the training dataset by epoch 3, the accuracy of the testing set did not improve, indicating that the model was overfitting as we increased the number of epochs(Figure 3). This was likely due to the limited size of the dataset given.

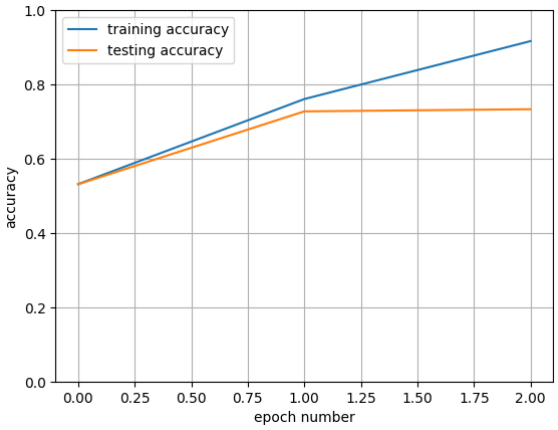

Figure 3: Graph of accuracy of LSTM model plotted against epoch value for a randomized training and testing dataset. Overfitting of the model to the training data could be observed at epoch 3, where the training accuracy was much higher than the testing accuracy.

4.2 Comparison between models

The performances of the four models were evaluated and compared using confusion matrices as presented in Table 1.

	Log Reg	KNN	SVM	NB	LSTM
Accuracy(%)	71.4	59.5	78.9	76.4	74.9
F1 score (%)	68.9	54.8	78.4	77.5	75.5
Recall (%)	77.5	63.8	78.4	77.5	75.6
Precision (%)	62.0	48.1	78.8	77.6	75.3

Table 1: Confusion metrics of the 5 models on the exact same set of training data and testing data.

The three more sophisticated models, SVM, NB and LSTM, generally performed better than simpler models like logistic regression and KNN. However, the LSTM model showed a lower accuracy compared to SVM and NB, which was unexpected. This was likely caused by the dataset chosen, which is small in size (10662 sentences) and includes short sentences (maximum sentence length of 52 words) only. As SVM and NB are good at analyzing such datasets, they outperformed LSTM when this particular dataset was used.

However, we still expect the LSTM model to perform better under real life situations, as it is able to learn and remember long-term dependencies between words. This allows LSTM to analyze longer and more complicated texts with higher accuracy (Huilgol, 2019). Moreover, in real life context, as the size of the dataset increases, SVM will face difficulty due to the large amount of computation required, while LSTM will benefit from a larger training dataset. On the other hand, NB assumes that the use of each word in the sentence is independent, and each word is of the same importance, carrying the same weightage, which is usually not true when it comes to sentiment analysis, while LSTM can avoid these assumptions, making more accurate predictions.

Besides accuracy, time complexity and space complexity were also taken into consideration. Of all the models, KNN required the most memory space and runtime, reflecting its inefficiency. Logistic regression had the fastest runtime, possibly due to the relatively small amount of computation required, making it applicable for occasions where speed is prioritized over accuracy.

5. Conclusion and Future Work

In conclusion, all five models implemented were able to classify sentences based on the sentiment, with the LSTM model performing the best overall. The quality of the models could be improved by expanding the dataset and including more varied sentences, as well as including more class labels like “neutral”. Pre-trained embedding like Word2vec could also be used to further increase the accuracy. Furthermore, more complex models such as the transformer neural network could be implemented to further improve the accuracy of sentiment analysis (Jiang et al., 2019).

6. Acknowledgement

We thank Prof. Prabhu Natarajan for delivering the lecture content on basic machine learning and deep learning models. We thank Vasista Reddy for providing the sample model for SVM. We also thank Bharath Shankar for providing learning materials for various machine learning models.

References

Butt, M. M. 2022, June 16. Sentiment analysis of Twitter's US Airlines data using KNN Classification. Medium. https://towardsdatascience.com/sentiment-analysis-of-twitters-us-airlines-data-using-knn-classification-91c7da987e13. Accessed April 2, 2023

Google. (n.d.). Dataset.csv. Google Drive. https://drive.google.com/file/d/1A87CRUb-VrJOG0Q0d8ehOV9vOhhmRCM12/view?usp=drivesdk. Accessed April 2, 2023

Huilgol, P. 2019, August 24. Accuracy vs. F1-score. Medium.https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2. Accessed April 2, 2023

Jiang, M., Wu, J., Shi, X., & Zhang, M. 2019. Transformer based memory network for sentiment analysis of web comments. *IEEE Access*, 7, 179942–179953. https://doi.org/10.1109/access.2019.2957192

Jurafsky, D., & Martin, J. H. 2009. *Logistic Regression . In Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (pp. 1–5). essay, Pearson Prentice Hall.

Mashalkar, A. (2021, March 7). Sentiment analysis using logistic regression and naive Bayes. Medium. https://towardsdatascience.com/sentiment-analysis-using-logistic-regression-and-naive-bayes-16b806eb4c4b. Accessed April 2, 2023

McCrae, J., & Lakshminarayanan, S. K. (n.d.). A Comparative Study of SVM and LSTM Deep Learning Algorithms for Stock Market Prediction. https://ceur-ws.org/Vol-2563/aics_41.pdf Accessed April 2, 2023

Poornima, A., & Priya, K. S. 2020. A comparative sentiment analysis of sentence embedding using machine learning techniques. *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*.https://doi.org/10.1109/icaccs48705.2020.9074312

Ray, S. 2023, March 24. Learn how to use support vector machines (SVM) for Data Science. Analytics Vidhya.https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/. Accessed April 2, 2023

Team, K. (n.d.). Keras Documentation: IMDB Movie Review Sentiment Classification Dataset. Keras. https://keras.io/api/datasets/imdb/. Accessed April 2, 2023

Yıldırım, S. 2020, May 12. Naive Bayes classifier - explained. Medium. https://towardsdatascience.com/naive-bayes-classifier-explained-50f9723571ed. Accessed April 2, 2023