Fake Review Classification and Topic Modelling

Project Report

By

Durga Kannan

Objective:

The goal of this project is to develop a system that can classify reviews as either fake or real by leveraging a combination of traditional machine learning, deep learning models, and Transformer-based approaches. Additionally, the system will cluster similar reviews together to group related feedback, making it easier to analyse customer insights. Furthermore, it will identify underlying topics within the reviews, providing a deeper understanding of customer sentiments and issues.

Step-by-Step Approach for the Project:

1.Data Collection:

Collect a Fake Review Dataset that includes product reviews along with metadata such as ratings, helpful votes, and other relevant information.

2. NLP-Data Preprocessing:

Text Cleaning: Remove special characters, numbers, and stop words from the reviews.

Tokenization: Break down the review text into individual words or tokens.

- **Stop word Removal**: The process of removing common words (like "the", "is", "in") that don't carry significant meaning in text analysis.
- **Stemming**: Reducing words to their root form (e.g., "running" to "run") using simple rules.
- **Lemmatization**: Similar to stemming but more sophisticated; it converts words to their base form (e.g., "better" to "good").
- **Vectorization**: Converting text data into numerical representations (like word counts or word embeddings) for use in machine learning models.

- **TF-IDF Vectorization:** Convert the review text into a matrix of numerical features, capturing the importance of words in each review.

3.Topic Modelling (Unsupervised Learning):

LDA and NMF for topic modelling

Latent Dirichlet Allocation (LDA): Apply this probabilistic model to assign each word in the reviews to one or more topics. LDA assigns each word in a document to a topic based on probabilities. Non-Negative Matrix Factorization (NMF): Use NMF to factorize the word occurrence matrix and identify topics across all reviews. NMF decomposes the document-term matrix (DTM) into two smaller matrices (one representing documents and one representing terms), with non-negative constraints. This means all values in the matrices are non-negative.

4.Clustering (Unsupervised Learning):

K-Means Clustering: Group the reviews into K clusters based on similarities, such as product categories or review types. Randomly choose **K** initial cluster centroids. Each data point is assigned to the nearest centroid based on a distance metric (usually Euclidean distance). Recalculate the centroids by averaging the data points in each cluster. The assignment and update steps are repeated until convergence (i.e., when cluster assignments no longer change).

PCA Visualization: It's a powerful technique for visualizing complex, high-dimensional data For a dataset with many features (e.g., 100 features), PCA reduces the dimensionality to a few key components.

2D PCA plot: The x and y axes will represent the first two principal components. Data points that are similar or close in the high-dimensional space will be clustered together in the 2D space.

t-SNE (t-Distributed Stochastic Neighbor Embedding) is another dimensionality reduction technique used to visualize high-dimensional data in 2D or 3D. Unlike PCA, which captures global structures (variance) of data, t-SNE focuses on preserving local relationships (similarities) between data points, making it particularly useful for visualizing clusters or groups in high-dimensional datasets.

5.Fake Review Classification (Supervised Learning):

Traditional Machine Learning Models: Use algorithms like Logistic Regression, Random Forest, and Support Vector Machine (SVM),K-nearest neighbours(KNN),Decision tree, Naïve Bayes to classify reviews as fake or real.

ML-Models Evaluation: Measure performance using classification metrics such as accuracy, precision, recall, and F1 score.

Model Accuracy Report

- Support Vector Machines (SVM): 63.10%
- K-Nearest Neighbors (KNN): 63.10%
- Decision Tree Model: 73.81%
- Random Forest Model: 84.70%
- Multinomial Naive Bayes: 84.75%
- Logistic Regression: 86.55%

Conclusion & Recommendations

Best Performer: Logistic Regression achieved the highest accuracy, followed closely by Multinomial Naive Bayes and Random Forest. These models should be considered as the primary candidates for deployment.

Underperforming Models: Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) performed poorly with only 63.1% accuracy, suggesting that these models are not ideal for this dataset without further tuning or feature engineering.

Deep Learning Models: applied models like RNN,LSTM,BILSTM,BERT.

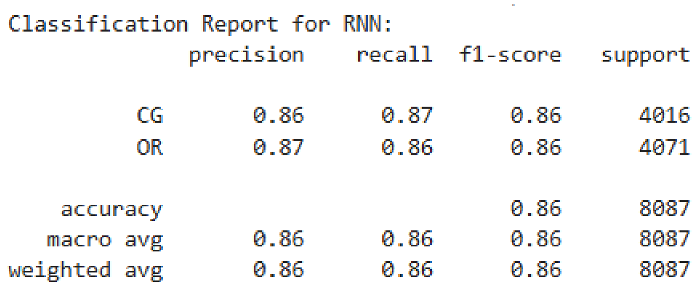
RNN (Recurrent Neural Network): RNN is a type of neural network designed for sequence data (e.g., time series, text), where the output from previous steps is fed as input to the current step, allowing the model to maintain memory of prior information.

LSTM (Long Short-Term Memory): LSTM is a special kind of RNN designed to overcome the limitation of vanilla RNNs, particularly the vanishing gradient problem, allowing it to remember long-term dependencies.

BiLSTM (Bidirectional LSTM): BiLSTM is an extension of LSTM that processes data in both forward and backward directions (i.e., from past to future and future to past).

Evaluation metrics for RNN,LSTM,BILSTM:

Performance of RNN

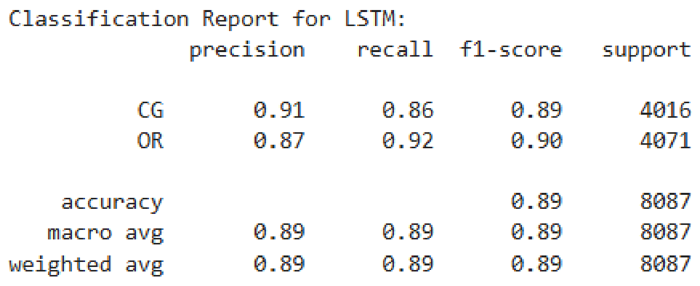


```
Classification Report for RNN:
              precision    recall  f1-score   support

          CG       0.86      0.87      0.86      4016
          OR       0.87      0.86      0.86      4071

    accuracy                           0.86      8087
   macro avg       0.86      0.86      0.86      8087
weighted avg       0.86      0.86      0.86      8087
```

Performance of LSTM

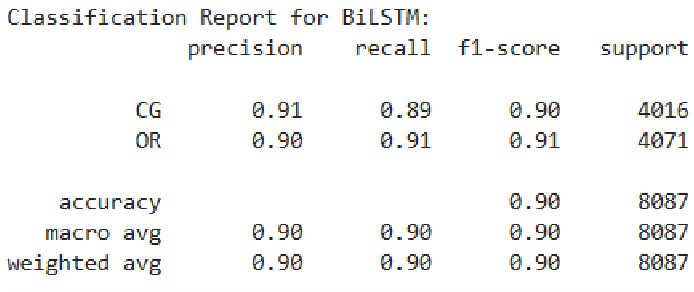


```
Classification Report for LSTM:
              precision    recall  f1-score   support

          CG       0.91      0.86      0.89      4016
          OR       0.87      0.92      0.90      4071

    accuracy                           0.89      8087
   macro avg       0.89      0.89      0.89      8087
weighted avg       0.89      0.89      0.89      8087
```

Performance of BILSTM



```
Classification Report for BiLSTM:
              precision    recall  f1-score   support

          CG       0.91      0.89      0.90      4016
          OR       0.90      0.91      0.91      4071

    accuracy                           0.90      8087
   macro avg       0.90      0.90      0.90      8087
weighted avg       0.90      0.90      0.90      8087
```

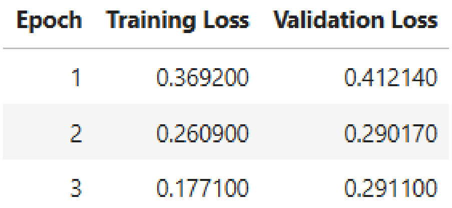
Conclusion:

The **BiLSTM** model outperforms both **LSTM** and **RNN**, with the highest accuracy of **90%** and balanced F1-scores

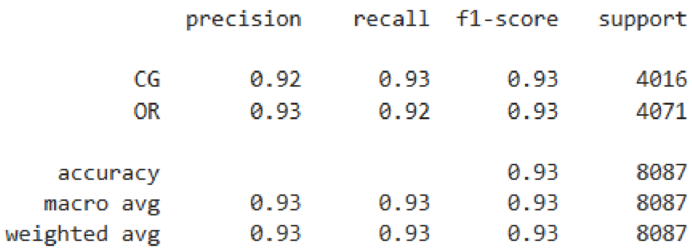
BERT (Bidirectional Encoder Representations from Transformers) model for sequence classification using the Transformers library by Hugging Face

BERT is a pre-trained transformer-based model developed by Google for Natural Language Processing (NLP) tasks. It uses a transformer architecture, which is fundamentally different from RNN, LSTM, and BiLSTM. BERT is designed to understand context from both directions in a text (left-to-right and right-to-left) simultaneously.

Performance Metrics of Bert model :



Epoch	Training Loss	Validation Loss
1	0.369200	0.412140
2	0.260900	0.290170
3	0.177100	0.291100



```
              precision    recall  f1-score   support

          CG       0.92      0.93      0.93      4016
          OR       0.93      0.92      0.93      4071

    accuracy                           0.93      8087
   macro avg       0.93      0.93      0.93      8087
weighted avg       0.93      0.93      0.93      8087
```

Summary & Insights:

- The model achieved **excellent performance**, with an overall accuracy of **93%**.
- **Precision**, **Recall**, and **F1-Score** for both classes (CG and OR) are all around **0.93**, indicating balanced and strong performance across both classes.
- The training loss decreased rapidly, showing that the model is converging well, though the validation loss slightly increased in the last epoch, which could suggest mild overfitting.

This model is performing well on the classification task and generalizes well to unseen data based on the validation results.