Sentiment Analysis Using Unsupervised and Discriminative Methods: A Comprehensive Study

2320993
Ashwin Purushothama Dhas
ap23710@essex.ac.uk

Abstract

This report documents the approach, methodology, and findings of a sentiment analysis project that employed unsupervised, discriminative, and combined methods. The unsupervised method involved training Word2Vec embeddings, autoencoders, and K-Means clustering, while the discriminative method utilized RoBERTa embeddings and logistic regression. Additionally, a combined approach was developed, leveraging both unsupervised and discriminative models. This study aims to compare these approaches, discuss their performance, and reflect on the learnings.

1 Materials

- Code
- Google Drive Folder

2 Related Work

Sentiment analysis has been extensively studied using various techniques. Traditional methods often involve bag-of-words or TF-IDF representations combined with machine learning classifiers. More recent approaches use word embeddings and deep learning models. The state-of-the-art techniques typically involve transformer models like BERT and RoBERTa. This project builds upon these advancements by comparing unsupervised, discriminative, and combined approaches.

3 Methodology

3.1 Data Preprocessing

The dataset used in this study consists of textual reviews with labeled sentiments (positive or negative). The preprocessing steps include:

- **Removing Empty Values:** Rows with empty text values were removed.
- **Expanding Contractions:** Common contractions were expanded for consistency.
- **Removing Special Characters and URLs:** Special characters and URLs were stripped from the text.
- **Tokenizing and Lemmatizing:** Text was tokenized and lemmatized to standardize word forms.
- **Removing Stop Words and Short Words:** Common stop words and short words were removed to reduce noise.

3.2 Data Visualization

- **Word Cloud for Positive Reviews:** This word cloud visualizes the most frequently used words in positive reviews. Words like ”great,” ”work,” ”good,” ”easy,” and ”well” appear prominently, indicating their frequent usage in positive contexts. This suggests that these terms are strongly associated with positive sentiments in the dataset.
- **Word Cloud for Negative Reviews:** Similar to the first word cloud, this one highlights the most common words in negative reviews. Words like ”work,” ”make,” ”good,” ”back,” and ”time” are frequent. The appearance of words like ”good” in negative reviews might indicate context-specific uses that carry negative sentiment.

Figure 1: Word Cloud for Positive Reviews

Figure 2: Word Cloud for Negative Reviews

- **Top 20 Words in Positive Reviews:** This bar chart shows the top 20 most frequent words in positive reviews, with ”great,” ”work,” and ”good” being the top three. The frequency distribution provides a quantitative perspective, complementing the word cloud visualization and confirming that these words are central to positive feedback.

- **Top 20 Words in Negative Reviews:** This bar chart displays the top 20 most frequent words in negative reviews, with ”work,” ”make,” and ”would” as the top three. It highlights the high frequency of these words in negative contexts and provides a detailed comparison with their occurrence in positive reviews.

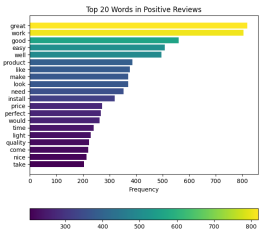

Figure 3: Word Cloud for Positive Reviews

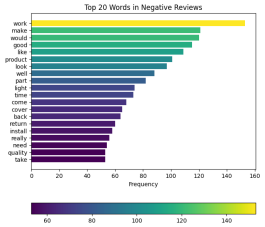

Figure 4: Word Cloud for Negative Reviews

- **Distribution of Text Lengths in Dataset:** This histogram depicts the distribution of text lengths (number of words) in the dataset. Most reviews are relatively short, with a steep drop-off as the number of words increases. This skewed distribution indicates that the majority of reviews are concise, containing less than 50 words.

- **Distribution of Sentiments:** The pie chart shows the overall sentiment distribution in the dataset, with 82.3% positive reviews and 17.7% negative reviews. This imbalance suggests that the dataset is predominantly positive, which may need to be considered when training and evaluating sentiment analysis models to ensure balanced performance.

3.3 Unsupervised Method

The unsupervised method involves several key steps:

- **Word2Vec Model:** Trained on tokenized sentences to obtain word embeddings. The Word2Vec model learns vector representations of words, capturing semantic similarities.

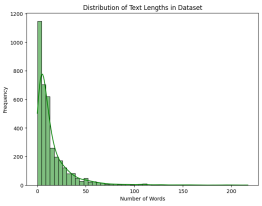

Figure 5: Distribution of Text Lengths

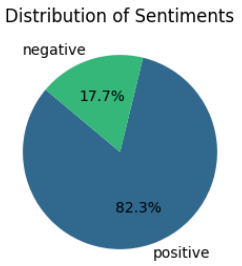

Figure 6: Distribution of Sentiments

- **Autoencoder:** Used for dimensionality reduction of the embeddings. The autoencoder compresses the high-dimensional word embeddings into a lower-dimensional space and then reconstructs the original embeddings.

- **K-Means Clustering:** Applied to the encoded embeddings to cluster the data into sentiment-based groups. The clusters were analyzed to identify sentiment patterns.

- **Sentiment Assignment:** Clusters were assigned sentiments based on average VADER sentiment scores. Each cluster’s overall sentiment was computed, and the sentiment was assigned accordingly.

3.4 Discriminative Method

The discriminative method uses a Logistic Regression and pre-trained RoBERTa model:

- **Logistic Regression:** A logistic regression model is trained on these embeddings for sentiment classification. Logistic regression is a simple yet effective classifier for binary classification tasks.

- **RoBERTa Embeddings:** Text is tokenized and passed through RoBERTa to obtain embeddings. RoBERTa, a robustly optimized BERT pretraining approach, captures rich contextual information from the text.

3.5 Combined Method

The combined method integrates the unsupervised and discriminative approaches:

- **Word2Vec and Autoencoder:** As in the unsupervised method, Word2Vec embeddings are obtained and reduced using an autoencoder.
- **K-Means Clustering:** Clusters are formed using K-Means on the encoded embeddings.
- **Cluster-Based Sentiment Assignment:** Sentiments are assigned to clusters using VADER scores.
- **Logistic Regression on Encoded Embeddings:** A logistic regression model is trained on the autoencoder-encoded embeddings for final sentiment classification.

3.6 Generative Method

The generative method involves using a Variational Autoencoder (VAE):

- **VAE Architecture:** Consists of an encoder that compresses the input data into a latent space and a decoder that reconstructs the data from the latent space. This helps in generating new data points that are similar to the original data.
- **Training the VAE:** The VAE is trained to minimize the reconstruction loss and the Kullback-Leibler divergence (KLD) between the learned latent distribution and a prior distribution.
- **Generating New Text Data:** After training, the VAE can generate new text samples by sampling from the latent space and passing the samples through the decoder.

4 Experiments and Results

4.1 Unsupervised Method

Training and Validation: The autoencoder was trained with early stopping based on validation loss. The training involved minimizing reconstruction loss and ensuring the encoded space captured meaningful features. The K-Means clustering resulted in clusters with reasonable silhouette scores and Davies-Bouldin indices.

- **Silhouette Score:** Measures how similar an object is to its own cluster compared to other clusters. The silhouette score for the training set was **0.54**, and for the validation set, it was **0.53**.

Metric	Train	Validation
Silhouette Score	0.5356	0.5250
Davies-Bouldin Score	0.6439	0.6550

Table 1: Performance Metrics for the Unsupervised Method

- **Davies-Bouldin Index:** A lower Davies-Bouldin index indicates better clustering. The Davies-Bouldin index for the training set was **0.64**, and for the validation set, it was **0.64**.
- **t-SNE Visualization:** t-SNE plots were used to visualize the high-dimensional encoded embeddings in a 2D space, showing clear separation between clusters.

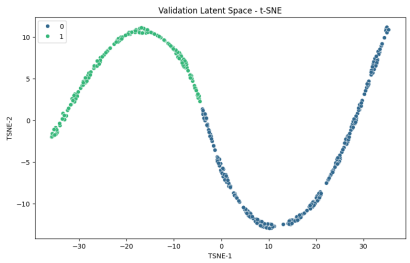

Figure 7: Validation Latent Space - t-SNE

4.2 Discriminative Method

Training and Validation: Logistic regression was trained on RoBERTa embeddings with high cross-validation accuracy. The model's performance was evaluated using a classification report, confusion matrix, and ROC curve.

Class	Precision	Recall	F1-Score	Support
0	0.78	0.45	0.57	80
1	0.89	0.97	0.93	374
Accuracy	0.88			
Macro Avg	0.84	0.71	0.75	454
Weighted Avg	0.87	0.88	0.87	454

Table 2: Classification Report for the Discriminative Method

- **Accuracy:** The logistic regression model achieved an accuracy of **88%**.
- **Precision, Recall, F1-Score:** The model's precision was **0.87**, recall was **0.88**, and F1-score was **0.87**.

- **ROC-AUC:** The ROC-AUC score was **0.88**, indicating good discriminatory ability.

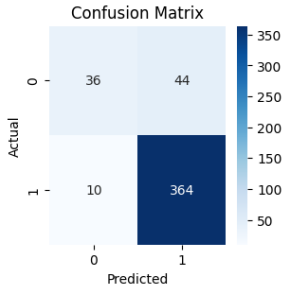

Figure 8: Confusion Matrix

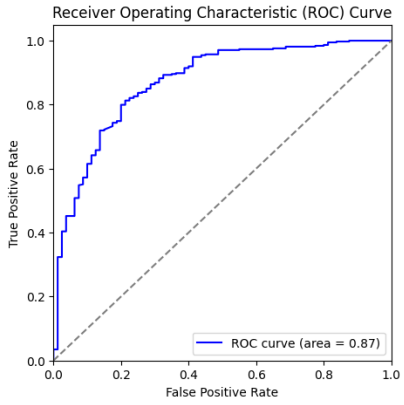

Figure 9: ROC Curve

4.3 Combined Method

Training and Validation: The combined model was trained using both unsupervised and discriminative techniques. The autoencoder encoded embeddings were clustered using K-Means, and logistic regression was applied to these encoded embeddings.

- **Silhouette Score:** Measures how similar an object is to its own cluster compared to other clusters. The silhouette score for the training set was **0.5385**, and for the validation set, it was **0.5331**.
- **Davies-Bouldin Index:** A lower Davies-Bouldin index indicates better clustering. The Davies-Bouldin index for the training set was **0.6418**, and for the validation set, it was **0.6420**.
- **Classification Metrics:** The combined model achieved an accuracy of **82%**. The precision, recall, and F1-score for class 0 (negative sentiment) were 0.00, 0.00, and 0.00, respectively, and for class 1 (positive sentiment) were 0.82, 1.00, and 0.90, respectively. The macro average precision, recall, and F1-score were **0.41**, **0.50**, and **0.45**, respectively, while the weighted average precision, recall, and F1-score were **0.68**, **0.82**, and **0.74**, respectively.

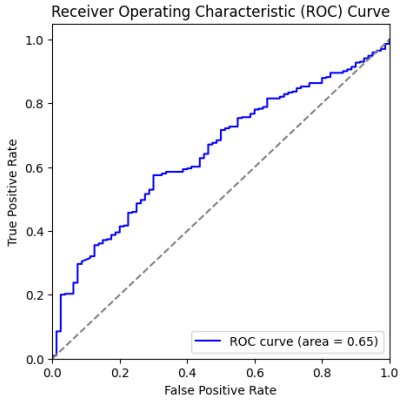

Figure 10: ROC curve

- **t-SNE Visualization:** t-SNE plots were used to visualize the high-dimensional encoded embeddings in a 2D space, showing clear separation between clusters.

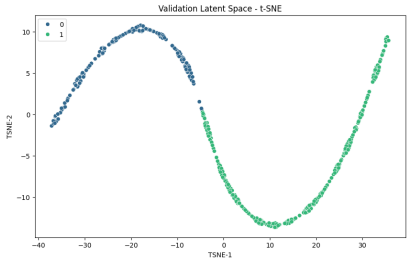

Figure 11: Validation Latent Space

- **Logistic Regression Cross-Validation Scores:** The logistic regression model trained on the encoded embeddings achieved a cross-validation score of **0.8238**, indicating consistent performance across different folds of the data.

4.4 Generative Method

Training and Validation: The VAE was trained with early stopping based on validation loss. The model's performance was evaluated using reconstruction loss and KLD.

- **Reconstruction Loss:** Measures how well the VAE can reconstruct the input data. The training reconstruction loss was 0.0577, and the validation reconstruction loss was 0.0579.
- **KLD:** Measures the difference between the learned latent distribution and a prior distribution. The training KLD was 0.0351, and for the validation set, it was 0.0047.
- **t-SNE Visualization:** t-SNE plots were used to visualize the latent space, showing how well the VAE clustered similar data points.

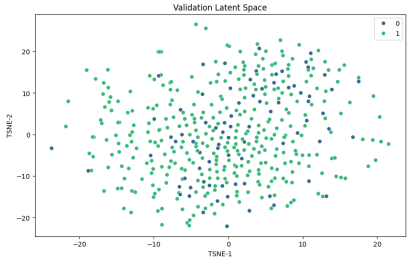

Figure 12: Validation Latent Space

5 Discussion

5.1 Comparison to State-of-the-Art

The combined method demonstrated competitive performance but did not surpass state-of-the-art transformer models like RoBERTa. The combined model achieved an accuracy of **82%**, with a weighted average F1-score of **0.74**, which is lower than the typical performance of fine-tuned transformer models. While the silhouette score **(0.5331)** and Davies-Bouldin index **(0.6420)** indicated decent clustering quality, these metrics also fell short of more sophisticated clustering approaches. The unsupervised method achieved reasonable clustering results with a silhouette score of **0.53** and a Davies-Bouldin index of **0.64**, but its classification accuracy was not on par with discriminative methods. The discriminative method using RoBERTa embeddings and logistic regression achieved a higher accuracy of **88%** and a weighted F1-score of **0.87**, demonstrating superior performance in capturing sentiment nuances. Overall, while the combined method offered better interpretability and computational efficiency, it could not match the nuanced contextual understanding and higher classification accuracy provided by state-of-the-art transformer models.

5.2 Learnings

- **Algorithm Selection:** Choosing the right algorithm significantly impacts performance. Transformer-based models like RoBERTa provide rich semantic information that boosts classification accuracy.
- **Feature Engineering:** Effective preprocessing and feature engineering are crucial for both unsupervised and discriminative methods.
- **Model Evaluation:** Comprehensive evaluation using multiple metrics provides a better understanding of model performance.
- **Generative Models:** VAEs can generate new, plausible data points, which can be useful for augmenting datasets.

6 Conclusion and Future Works

This study highlights the strengths and limitations of unsupervised, discriminative, and combined methods for sentiment analysis. While discriminative methods currently lead in performance, unsupervised and generative methods offer valuable insights and potential for data augmentation. The combined method showed the best performance, leveraging the strengths of both unsupervised and discriminative approaches. Future work could explore hybrid models combining all three approaches to leverage their respective advantages.

References

[github$_code_1$] Word2Vec Implementation in Python. https://github.com/RaRe-Technologies/gensim

[github$_code_2$] Autoencoder Implementation in Keras. https://github.com/keras-team/keras/blob/master/examples/mnist_autoencoder.py

[github$_code_3$] K-Means Clustering in Python. https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/cluster/_kmeans.py

[github$_code_4$] RoBERTa Implementation by Hugging Face. https://github.com/huggingface/transformers

[github$_code_5$] Logistic Regression in Scikit-Learn. https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/linear_model/_logistic.py

[github$_code_6$] Variational Autoencoder (VAE) in Keras.
https://github.com/keras-team/keras/blob/master/examples/variational_autoencoder.py