Assignment 1

LEEECS4412

tiqbal02, Talha Iqbal, 217 967 951, tiqbal02@my.yorku.ca

melg23, Mariam El Ghargomi , 216 980 161, melg23@my.yorku.ca

Contents

Objective 3

Bag of Words (BOW) Approach 3

Preprocessing 3

Classifier Selection and Cross-validation 3

Results 3

Text Embeddings Approach 4

Preprocessing 4

Classifier Selection and Cross-validation 4

Results 4

Discussion and Conclusion 5

Running the Programs 5

Objective

Given a dataset of reviews from Yelp, classify the reviews into positive, neutral, and negative using 2 different methods; Bag of Words (BOW) and Text Embeddings. BOW represents documents in a numerical vector format using TF-IDF features, while the Text Embedding method uses pre-trained embeddings from a transformer-based model (BERT) for feature representation. Ultimately, the objective is to compare the performance of various classifiers within each approach and draw conclusions based on validation results.

Bag of Words (BOW) Approach

Preprocessing

- Data: Yelp review datasets for training and testing are used.
- Data Splitting: There are 2 datasets; a training dataset and a test dataset. The training dataset is split into a training and a validation set with the test size set to 20% of the original training dataset.
- Text Vectorization: The reviews are transformed into TF-IDF numerical vectors using the TfidfVectorizer. This helps capture the importance of words in the corpus while considering their frequency and inverse document frequency. The parameter lowercase = True is passed which converts all text to lowercase. Additionally, the parameter stop_words=English is passed which removes common stop words in the English language from the documents as they usually do not contribute to the meaning.

Classifier Selection and Cross-validation

- Feature Selection: The best k features are evaluated using SelectKBest with chi-squared scoring. Chi-squared measures the importance of the feature. The range of k is 600, 900, 1500, 1800, 2100. Due to the length of time it takes to train SVM and Random forests, The range of k was limited to 5 and the values were kept small. After 2100 features, the accuracy does not change much.
- Classifier Evaluation: Three classifiers (Multinomial Naive Bayes, Random Forest, SVM) are compared using cross-validation to find the best-performing combination of the classifier and the value of k from the given range. 3-fold cross-validation is used. The classifier with the highest mean accuracy is selected.

Results

The selected classifier is trained on the training dataset with the k best features, and predictions are made on the validation and test sets. The classification report on the validation set is printed and the predictions on the test dataset are saved as a CSV file.

The higher the number of features used, the better the accuracy.

The highest performing classifier was SVM followed by Random Forests, and then Multinomial Naïve Bayes.

The accuracy of SVM was 84%

The full report can be seen in the notebook file.

Text Embeddings Approach

Preprocessing

- Data Loading: Yelp review datasets for training and testing are loaded.
- Data Splitting: There are 2 datasets; a training dataset and a test dataset. The training dataset is split into a training and a validation set with the test size set to 20% of the original training dataset.
- Text Embeddings: Reviews are converted into pre-trained embeddings using the SentenceTransformer with the bert-base-nli-mean-tokens model.

Classifier Selection and Cross-validation

- Classifier Evaluation: Three classifiers (Logistic Regression, Random Forest, SVM) are compared using cross-validation based on the embeddings. 3-fold cross-validation is used. The classifier with the highest mean accuracy is selected.

Results

The best-performing classifier is selected based on cross-validation results and trained on the training data. Predictions are made on the evaluation dataset and the report is outputted. Predictions are then made on the test dataset, and the results are saved as a CSV file.

Logistic Regression was the best-performing classifier followed by SVM and then Random Forests.

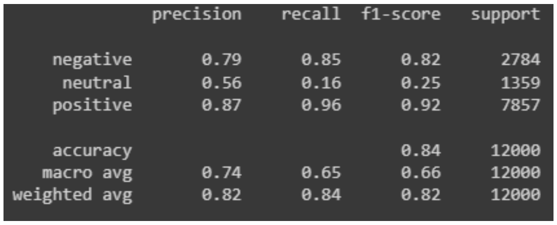
The accuracy of Logistic Regression was 84%

The full reports can be seen in the notebook file.

Discussion and Conclusion

This project uses the BOW and Text Embedding method for classifying reviews. Classifiers such as SVM, Logistic Regression, Multinomial Naïve Bayes, and Random Forests were used. The BOW method used SVM as the classifier and the Text Embeddings used Logistic Regression as the classifier. The results show that the BOW and Text Embedding methods have the same accuracy to the nearest percent. However, if one were to look at the details when it comes to classifying the different types of reviews, we can see that the BOW method yielded a higher precision in neutral reviews whereas Text Embeddings has a higher precision when it comes to classifying positive reviews. Ultimately if one has the resources to use the Text Embeddings method then they should as the BOW method with SVM takes a longer time to train compared to Logistic Regression.

BOW METHOD – SVM

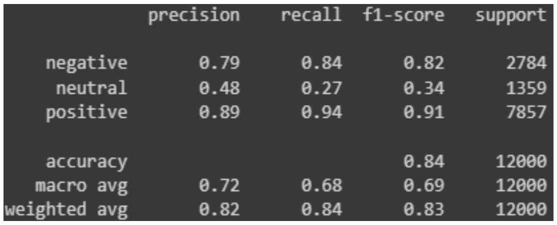


```
              precision    recall  f1-score   support

    negative       0.79      0.85      0.82      2784
     neutral       0.56      0.16      0.25      1359
    positive       0.87      0.96      0.92      7857

    accuracy                           0.84     12000
   macro avg       0.74      0.65      0.66     12000
weighted avg       0.82      0.84      0.82     12000
```

TEXT EMBEDDINGS METHOD – LOGISTIC REGRESSION



```
              precision    recall  f1-score   support

    negative       0.79      0.84      0.82      2784
     neutral       0.48      0.27      0.34      1359
    positive       0.89      0.94      0.91      7857

    accuracy                           0.84     12000
   macro avg       0.72      0.68      0.69     12000
weighted avg       0.82      0.84      0.83     12000
```

Running the Programs

Ensure that the required libraries (pandas, sci-kit-learn, sentence-transformers) are installed. Additionally, make sure the pre-trained BERT model is available for the Text embedding approach. The paths to the training and test datasets should be correctly specified before running the code.