NLP-Take Home Assignment

Project Title: Topic Assignment

Your Name: Srinivasulu Devanuru
Date: 26-07-2024

Brief Overview of the Problem:

Problem Statement:

- The submission should allow the user to run the code and ask the user for a list of topics as an input.
- How are you presenting the results back to the user?
- How do you handle reviews that do not fall within one of the categories?
- How are you evaluating the quality of your model?

Objectives of the Assignment:

- To classify reviews into predefined topics.
- To present the classified results to the user.
- To handle and report reviews that do not fit into any of the predefined categories.
- To evaluate and report the quality of the classification model.

Data Description:

Source of the Data:

The data was sourced from a Kaggle dataset containing user reviews, which include the following key features:

- **reviewId**: Unique identifier for the review.
- **userName**: Name of the user who wrote the review.
- **review**: Text of the review.
- **score**: Rating given by the user.
- **thumbsUpCount**: Number of thumbs-up the review received.
- **reviewCreatedVersion**: Version of the app when the review was written.
- **at**: Date and time when the review was posted.
- **appVersion**: Version of the app.

Key Features and Their Descriptions:

- **review**: The main text data used for topic classification.
- **topic**: The target variable indicating the topic of the review.

Data Preprocessing:

Steps Taken to Clean and Preprocess the Data:

1. **Handling Missing Values**: Filled NaN values in the 'review' column with empty strings.
2. **Text Cleaning**:
    - Removed non-alphabetic characters.
    - Converted text to lowercase.
    - Removed punctuation.
    - Removed stopwords.
    - Applied lemmatization to reduce words to their base forms.

Feature Engineering:

- **Word2Vec Embeddings**: Created embeddings for each review using a Word2Vec model trained on the cleaned review texts.

Methodology:

Description of the Algorithms Used:

- **Word2Vec**: Used to create dense vector representations of words and aggregate them to form review-level embeddings.
- **XGBoost Classifier**: Used for classifying the reviews into predefined topics.

Justification for the Chosen Methods:

- **Word2Vec**: Effective in capturing semantic meanings of words and their context in the text.
- **XGBoost**: A powerful and efficient gradient boosting algorithm known for its performance on structured data

Implementation:

Tools and Libraries Used:

- **pandas**: For data manipulation.
- **numpy**: For numerical computations.
- **nltk**: For text preprocessing.
- **gensim**: For training the Word2Vec model.
- **xgboost**: For training the classifier.
- **sklearn**: For model evaluation and hyperparameter tuning.

Key Steps in the Implementation Process:

1. Data loading and preprocessing.
2. Training Word2Vec model to create embeddings.
3. Training XGBoost classifier on the embeddings.
4. Hyperparameter tuning using GridSearchCV.
5. Evaluating the model on validation and evaluation datasets.
6. Implementing a user interface for review classification.

Evaluation:

Metrics Used to Evaluate the Models:

- **Accuracy**: The ratio of correctly predicted instances to the total instances.
- **Precision, Recall, F1-Score**: Evaluated per class and averaged (macro and weighted).

Performance Results:

- **Training Accuracy**: 63%
- **Validation and Evaluation Accuracy**: Results from the evaluation phase.**Results and Discussion:**

Key Findings from the Analysis:

- Word2Vec embeddings combined with XGBoost classifier provided reasonable performance.
- Hyperparameter tuning improved the model's performance.

Interpretation of Results:

- The model performed well on training data but had lower performance on the evaluation provided dataset.

Conclusion:

Summary of What Was Accomplished:

- Successfully implemented a text classification model to categorize reviews into predefined topics.
- Developed a user interface for interacting with the model.
- Evaluated the model's performance and identified areas for improvement.

Main Takeaways:

- Preprocessing and feature engineering are critical for text classification tasks.
- Word2Vec embeddings are effective for capturing semantic information in text data.

Suggestions for Future Work:

- Experiment with other embeddings like BERT for potentially better performance.
- Implement additional preprocessing steps like handling negations and using domain-specific stopwords.
- Use more sophisticated imbalance handling techniques like SMOTE.