**MGTA 415 Homework 1**

**Rachel Wang**

**Problem 1: Text Pre-Processing**

**1.1 Data Preprocessing**

The dataset was preprocessed with the following steps:

1. Tokenization → Splitting text into words

2. Lowercasing → Converting text to lowercase.

3. Punctuation Removal → Removing punctuation.

4. Stemming → Reducing words to their root forms using PorterStemmer.

**1.2 The first 5 sentences using Tokenization**

0 [technopolis, plans, develop, stages, area, le...

1 [international, electronic, industry, company,...

2 [new, production, plant, company, would, incre...

3 [according, company, 's, updated, strategy, ye...

4 [financing, aspocomp, 's, growth, aspocomp, ag...

**1.3 The first 5 sentences using Stemming**

0 [technopoli, plan, develop, stage, area, less,...

1 [intern, electron, industri, compani, elcoteq,...

2 [new, product, plant, compani, would, increas,...

3 [accord, compani, 's, updat, strategi, year, 2...

4 [financ, aspocomp, 's, growth, aspocomp, aggre...

**Problem 2: Bag Of Words**

We train Logistic Regression classifiers using three different document representations:

1. Binary Representation → Presence of words as 1/0 values.

2. Frequency Representation → Raw word counts.

3. TF-IDF Representation → Importance-weighted word features.

**2.1 Model Training Process**

The model training process follows a structured approach consisting of four main steps: data splitting, hyperparameter tuning, model training, and evaluation. The objective is to ensure that the model achieves optimal performance while avoiding overfitting or underfitting.

**Step 1: Data Splitting**

The dataset is divided into three subsets:

• Training Set (80%) → Used to train the logistic regression model.

• Validation Set (10%) → Used for hyperparameter tuning, allowing us to determine the best values for regularization strength (C) and vocabulary size (dictionary\_size).

• Test Set (10%) → Used for final evaluation to assess the model’s performance on unseen data.

**Step 2: Hyperparameter Tuning**

To optimize the logistic regression model, we tune two key hyperparameters:

1. Regularization strength (C) → Determines how much the model is penalized for complexity. Lower values of C increase regularization, reducing overfitting. Higher values decrease regularization, allowing the model to capture more variance. Multiple values (C = 0.1, 1, 10) are tested, and the one yielding the highest Macro-F1 score on the validation set is selected.

2. Vocabulary size (dictionary\_size) → Defines how many words are included in the feature space. Using a very small vocabulary may remove important words, while a very large vocabulary may introduce noise and increase computational cost. Different vocabulary sizes (10%, 25%, 50%, 100% of total words) are tested, and the best one is chosen based on validation performance.

Each document representation—Binary, Frequency, and TF-IDF—is tuned separately. The model is trained using multiple values for C and dictionary\_size, and performance is measured using the validation set. The best hyperparameters are selected based on the highest **Macro-F1 score**, ensuring balanced performance across all sentiment classes.

The best hyperparameters were selected separately for each document representation (Binary, Frequency, and TF-IDF) using **the validation set**. The selection was based on **Macro-F1 score**, ensuring balanced classification across sentiment categories.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Representation** | **Dictionary Size** | **Regularization (C)** | **Validation Accuracy** | **Macro-F1** | **Micro-F1** | **AUROC** |
| **Binary** | 8981 | 1.0 | 0.7872 | 0.7378 | 0.7872 | 0.8623 |
| **Frequency** | 2245 | 1.0 | 0.7769 | 0.7299 | 0.7769 | 0.8526 |
| **TF-IDF** | 2245 | 10.0 | 0.7748 | 0.7300 | 0.7748 | 0.8569 |

**Step 3: Model Training**

Once the optimal hyperparameters are identified, the model is retrained using both the training and validation sets. This ensures that the model has access to more data, improving its generalization ability. Separate models are trained for each document representation:

• Binary Representation → Each word is represented as 1 if present and 0 if absent.

• Frequency Representation → Each word is represented by its count within the document.

• TF-IDF Representation → Each word is assigned a weight based on its importance in the documnt and across the dataset.

**Step 4: Model Evaluation**

The final trained models are evaluated using the test set, which consists of unseen data. Performance is measured using three key metrics:

• AUROC (Area Under the Receiver Operating Characteristic Curve) → Measures the model’s ability to correctly rank different sentiment categories. A higher AUROC indicates better class separation.

• Macro-F1 Score → Calculates the F1-score for each class independently and averages them. This metric ensures that all classes (positive, neutral, and negative) are treated equally.

• Micro-F1 Score → Aggregates predictions across all instances and evaluates overall model accuracy.

Each document representation (Binary, Frequency, TF-IDF) is evaluated separately, and results are compared to determine which representation provides the best classification performance.

**2.2 Model Performance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Representation** | **AUROC** | **Macro-F1** | **Micro-F1** |
| **Binary** | 0.899281 | 0.740409 | 0.797938 |
| **Frequency** | 0.895827 | 0.732831 | 0.793814 |
| **TF-IDF** | 0.895387 | 0.721844 | 0.785567 |

**Conclusion**

• **Binary representation** is the best approach for sentiment classification in this dataset.

• TF-IDF underperformed compared to Binary and Frequency, likely due to its tendency to downweigh common words that may be important in sentiment prediction.

• Future improvements could explore deep learning models (BERT, LSTMs) and feature engineering techniques (bigram/trigram models, sentiment lexicons).