CSL 7640: NATURAL LANGUAGE UNDERSTANDING

Assignment 1

Github repository link:
https://github.com/Shruti03052/NLU-assignment-1

Introduction:

Text classification is a fundamental problem in Natural Language Processing (NLP) where the objective is to automatically assign predefined categories to textual documents.
In this assignment, we design and evaluate a binary text classifier that categorizes documents into Sports or Politics. We explore multiple feature representation techniques like bag of words(BoW), TF-IDF, and N- gram based TF-IDFand compare the performance of these across three machine learning techniques: Multinomial Naive Bayes, Logistic Regression, Support Vector Machine.The goal is to analyze how different representations and classifiers affect classification performance.

Data Collection and Description:

Dataset Source:

The dataset used in this assignment is the 20 Newsgroups dataset. It's a widely used benchmark dataset for text classification tasks. The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics. The dataset was accessed programmatically using the fetch_20newsgroups function from the scikit-learn library.

Category Selection:

To formulate a binary classification problem for the given category, a subset of categories was selected:

Sports Categories

- rec.sport.baseball
- rec.sport.hockey

Politics Categories

- talk.politics.guns
- talk.politics.mideast
- talk.politics.misc

All documents belonging to sports-related categories were labeled as “sport”, while documents from political categories were labeled as “politics”.

Data Cleaning

To reduce noise and prevent overfitting to metadata, the following components were removed during dataset loading:

- Email headers
- Footers
- Quoted text

This ensured that the classifier learned primarily from the semantic content of the documents.

Dataset Statistics:

- Total documents: 4618
- Classes: 2 (Sport, Politics)
- Class balance: Sport: 1993, Politics: 2625
- Train–Test Split: Training set: 80% Test set: 20%

Text Preprocessing

Raw text data contains noise such as punctuation, URLs, stopwords, etc. Effective preprocessing is essential to improve model generalization.

The following preprocessing steps were applied:

- Lowercasing: Converts all text to lowercase to ensure uniformity.
- Noise Removal: URLs. Email addresses, Non-alphabetic characters, Extra whitespace
- Tokenization: Splitting text into individual word tokens
- Stopword Removal: Common English stopwords (e.g., the, is, and) were removed using NLTK
- Lemmatization: Converts words to their base form (e.g., running -> run)
- Stemming: Further reduces words to root forms (e.g., political -> politic)

These steps significantly reduce vocabulary size and help models focus on meaningful linguistic patterns.

Feature Representation Techniques:

Bag of Words (BoW):

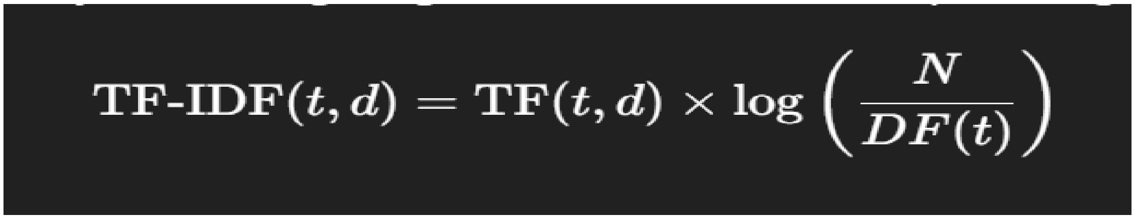
Bag of Words represents documents as vectors of word frequencies. Each dimension corresponds to a unique word in the vocabulary.

Characteristics:

- Ignores word order
- Simple and effective
- High-dimensional sparse vectors

TF-IDF (Term Frequency–Inverse Document Frequency):

TF-IDF improves upon BoW by down-weighting common words and emphasizing rare but informative terms. It reduces impact of frequent but uninformative words



$$\textbf{TF-IDF}(t,d) = \textbf{TF}(t,d) \times \log\left(\frac{N}{DF(t)}\right)$$

TF-IDF with N-grams:

To capture limited contextual information, unigrams and bigrams (1–2 grams) were used. This helps the model differentiate phrases that may have distinct meanings in political or sports contexts.

Machine Learning Models:

To evaluate the effectiveness of different feature representations for text classification, three supervised machine learning models were implemented and compared.

Multinomial Naive Bayes:

It is a probabilistic generative classifier based on Bayes' theorem. It assumes that features (words) are conditionally independent given the class label and that word frequencies follow a multinomial distribution.

Despite its strong independence assumptions, Multinomial Naive Bayes performs remarkably well in text classification tasks due to the high dimensionality and sparsity of textual data.

Logistic Regression:

It is a linear discriminative classifier that directly models the posterior probability of a document belonging to a particular class. It learns a weighted combination of features to make predictions and is particularly well suited for sparse, high-dimensional data such as text.

Unlike Naive Bayes, Logistic Regression does not assume feature independence, allowing it to better capture correlations between words.

Support Vector Machine (Linear SVM):

Support Vector Machines aim to find an optimal hyperplane that maximizes the margin between classes. Linear SVMs are especially effective for text classification because text data often becomes linearly separable in high-dimensional feature spaces.

Linear SVM focuses on maximizing class separation rather than probability estimation, which often leads to superior generalization performance.

Results:

Feature Type	Model	Accuracy
BoW	Naive Bayes	0.94
BoW	Logistic Regression	0.92
BoW	SVM	0.90

Feature Type	Model	Accuracy

TF-IDF	Naive Bayes	0.932
TF-IDF	Logistic Regression	0.934
TF-IDF	SVM	0.944

Feature Type	Model	Accuracy
TF-IDF(n-gram)	Naive Bayes	0.94
TF-IDF(n-gram)	Logistic Regression	0.93
TF-IDF(n-gram)	SVM	0.93

Analysis and Observations:

Performance with BoW:

Using the Bag of Words representation, Multinomial Naive Bayes achieved the highest accuracy of 0.94, outperforming Logistic Regression (0.92) and SVM (0.90).

This result aligns with the theoretical strengths of Naive Bayes, which performs well when word frequency information is highly discriminative. Sports-related documents often contain distinctive vocabulary (e.g., game, team, season), making BoW features particularly effective for Naive Bayes.

However, the comparatively lower performance of SVM suggests that raw word counts without weighting may not provide sufficient discriminative power for margin-based classifiers.

Performance with TF-IDF Features:

When TF-IDF features were used, overall performance improved across all classifiers. The Linear SVM achieved the best accuracy of 0.944, followed by Logistic Regression (0.934) and Naive Bayes (0.932).

TF-IDF reduces the influence of frequent but uninformative words and highlights domain-specific terms, which benefits discriminative models such as SVM and Logistic Regression.

Performance with TF-IDF N-gram Features:

Using TF-IDF with n-grams, Multinomial Naive Bayes achieves strong performance of 0.94, while Logistic Regression and SVM achieved accuracies of 0.93.

Although n-grams capture limited contextual information, the results suggest that adding bigrams did not significantly improve performance over unigram TF-IDF for this dataset. This indicates that single-word features were already sufficiently expressive for separating the two domains.

Limitations:

- The system only distinguishes between Sport and Politics. Real-world applications often require multi-class.
- Traditional feature representations such as BoW and TF-IDF do not capture semantic meaning or contextual relationships between words. Synonyms and polysemy are not handled effectively.
- The high accuracy achieved is partly due to the structured and topic-focused nature of the 20 Newsgroups dataset. Performance may degrade on noisier real-world data such as social media posts.