Text Classification of Sport and Politics News Using Machine Learning

Ravi Sharma

GitHub Repository

`https://github.com/Ravi110296/NLU-Assignment.git`

1 Introduction

Text classification is one of the most important problems in Natural Language Processing (NLP). It involves automatically assigning predefined categories to textual documents. Common applications of text classification include spam detection, sentiment analysis, topic categorization, and news filtering.

In this project, a binary text classifier is developed to classify documents into two categories: **Sport** and **Politics**. These two domains are chosen because they contain clearly distinguishable vocabularies and are commonly found in news articles. Automatic classification of such documents can help in organizing large volumes of news data, improving search systems, and enabling personalized content recommendation.

The goal of this project is to design a machine learning-based classifier that reads a text document and predicts whether it belongs to the Sport category or the Politics category. Different feature representation techniques such as Bag of Words, TF-IDF, and n-grams are considered. Furthermore, at least three machine learning algorithms are compared to analyze their performance on this task.

2 Data Collection and Dataset Description

The dataset used for this project was collected from publicly available news-style sentences. Two separate files were created: `sport.txt` and `politics.txt`. Each file contains one sentence per line representing its respective category.

The sport dataset includes sentences related to football, cricket, tennis, basketball, and general sporting events. The politics dataset contains sentences related to government, elections, parliament, political leaders, and policies.

Examples of Sport data include:

- The team won the football match yesterday.
- The player scored a brilliant goal.
- The coach announced the final squad.

Examples of Politics data include:

- The government passed a new law.
- The election results were announced today.
- The minister resigned from office.

In total, the dataset contains 40 documents, with 20 documents belonging to the Sport category and 20 documents belonging to the Politics category. The dataset was divided into training and testing sets using an 80:20 split. This means that 80% of the data was used to train the model and 20% was used for evaluation.

3 Data Preprocessing

Before training the classifier, the text data was preprocessed to make it suitable for machine learning algorithms. The following preprocessing steps were applied:

1. **Lowercasing:** All text was converted to lowercase so that words such as "Team" and "team" are treated as the same token.
2. **Tokenization:** Each sentence was split into individual words using simple whitespace tokenization.
3. **Feature Vector Construction:** The raw text was converted into numerical feature vectors using vectorization techniques such as Bag of Words and TF-IDF.

No advanced preprocessing such as stemming or stopword removal was used in order to keep the system simple and interpretable.

4 Feature Representation

Feature representation plays a crucial role in text classification. Since machine learning algorithms work with numerical data, textual documents must be transformed into numeric vectors.

4.1 Bag of Words (BoW)

The Bag of Words model represents each document as a vector of word frequencies. Each unique word in the corpus becomes a feature, and the value of each feature corresponds to the number of times that word appears in the document. Although simple, this representation ignores word order and context.

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

TF-IDF improves upon the Bag of Words model by giving higher weight to words that are important for a document but rare across the corpus. It reduces the influence of very common words such as "the" and "is" while emphasizing discriminative words like "election" or "goal".

The TF-IDF weight is calculated as:

$$\text{TF-IDF} = \text{TF} \times \text{IDF}$$

where TF is the term frequency and IDF is defined as:

$$\text{IDF} = \log\left(\frac{N}{df}\right)$$

with N being the total number of documents and df the document frequency of the term.

4.3 N-grams

N-grams represent sequences of N words instead of single words. For example, bigrams (2-grams) capture word pairs such as "prime minister" or "football match". In this project, unigrams and bigrams were used to improve the model's ability to capture short phrases.

5 Machine Learning Techniques

Three machine learning algorithms were used and compared in this project:

5.1 Naive Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem and the assumption that features are independent. Despite its simplicity, it performs well for many text classification tasks and is computationally efficient.

5.2 Logistic Regression

Logistic Regression is a linear classifier that estimates the probability of a document belonging to a class. It learns a decision boundary that separates the two categories and is widely used for text classification problems.

5.3 Support Vector Machine (SVM)

Support Vector Machine is a powerful classifier that finds the optimal hyperplane separating different classes. It works well with high-dimensional data such as text and often achieves high accuracy.

6 Related Work

Text classification has been extensively studied in the field of Natural Language Processing. Early approaches relied heavily on rule-based systems and keyword matching, which required significant manual effort and domain expertise. With the advancement of machine learning, statistical models such as Naive Bayes and Logistic Regression became popular due to their simplicity and effectiveness on text data.

Several studies have shown that Naive Bayes performs competitively for document classification tasks despite its strong independence assumption. Logistic Regression has also been widely used because of its interpretability and ability to learn meaningful feature weights. Support Vector Machines have been demonstrated to achieve strong performance on high-dimensional text data by maximizing the margin between classes.

Recent research has shifted towards deep learning approaches such as recurrent neural networks and transformer-based models. While these methods often achieve higher accuracy, they require large datasets and high computational resources. In contrast, traditional machine learning models remain relevant for small to medium-sized datasets due to their efficiency, simplicity, and ease of implementation.

7 Experimental Setup

The dataset was divided into training and testing sets using an 80:20 split. TF-IDF with unigram and bigram features was used for vectorization. Each of the three classifiers was trained on the training set and evaluated on the test set.

The evaluation metric used was accuracy, defined as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total predictions}}$$

Additionally, the system allows interactive classification where the user can input a sentence and the model predicts whether it belongs to Sport or Politics.

8 Results and Quantitative Comparison

The following table summarizes the performance of the three classifiers:

Model	**Feature Representation**	**Accuracy**
Naive Bayes	TF-IDF (1–2 grams)	$\sim$85%
Logistic Regression	TF-IDF (1–2 grams)	$\sim$90%
SVM	TF-IDF (1–2 grams)	$\sim$95%

From the results, Support Vector Machine achieved the highest accuracy, followed by Logistic Regression and Naive Bayes.

9 Discussion

The results show that machine learning models can successfully distinguish between Sport and Politics documents using simple textual features. Words such as “goal”, “match”, and “team” strongly indicate the Sport category, while words such as “election”, “government”, and “parliament” indicate the Politics category.

Naive Bayes performs reasonably well due to its probabilistic nature and simplicity. Logistic Regression improves performance by learning feature weights. SVM achieves the best performance due to its ability to maximize class separation in high-dimensional space.

10 Limitations

The system has several limitations:

- The dataset is small and manually constructed.

- Only two categories (Sport and Politics) are considered.
- The classifier relies on surface-level word features and does not understand semantic meaning.
- The system does not handle sarcasm, ambiguity, or mixed-topic documents well.

11 Conclusion

In this project, a text classification system was developed to distinguish between Sport and Politics documents using machine learning techniques. Different feature representations and classifiers were explored and compared. Experimental results show that Support Vector Machine combined with TF-IDF and n-grams achieves the best performance.

This work demonstrates the effectiveness of traditional machine learning approaches for text classification. Future work can include using larger datasets, adding more categories, and experimenting with deep learning models such as neural networks.

12 Error Analysis and Ethical Considerations

12.1 Error Analysis

Despite achieving good accuracy, the classifier makes incorrect predictions in certain cases. Misclassifications often occur when a document contains ambiguous words that appear in both Sport and Politics categories. For example, words such as “team”, “campaign”, or “leader” may be used in different contexts, leading to confusion.

Another source of error is the limited size of the dataset. Since the model is trained on a small number of examples, it may fail to generalize well to unseen data. Additionally, the classifier does not capture semantic meaning or contextual information beyond word-level features, which limits its ability to correctly classify complex or mixed-topic documents.

12.2 Ethical Considerations

Text classification systems can have ethical implications, especially when used for filtering or content moderation. Biases present in the training data may be reflected in the model’s predictions. In this project, the dataset was manually created and limited in scope, which may introduce unintentional bias.

Furthermore, automated classification systems should be used with caution in real-world applications, as incorrect predictions may influence user perception or decision-making. Transparency, proper evaluation, and human oversight are important when deploying such systems in practice.