VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Machine Learning (CO3117)

Assignment Report

Assignment 2: Machine Learning With Text Data

Advisor(s): Le Thanh Sach

Student(s): Cao Nhat Lam 2311815
Nguyen Thanh Toan 2313492
Doan Vo Viet Khoi 2311660
Group: CSEML04

HO CHI MINH CITY, OCTOBER 2025

Contents

Abstract

This report details the process and results of a text classification project on the AG News dataset. The primary objective was to classify news articles into four categories: World, Sports, Business, and Sci/Tech. Two distinct methodologies were compared: a traditional machine learning approach using TF-IDF feature extraction with models like Logistic Regression, SVC, and Random Forest, and a modern deep learning approach involving the fine-tuning of a pre-trained BERT model. The results demonstrate the superior performance of the fine-tuned BERT model, which achieved a test accuracy of 91.3%, significantly outperforming the best traditional model (SVC) that scored 87.1%. The findings highlight the effectiveness of transformer-based models in understanding language context, while also acknowledging the value of traditional methods as strong, computationally efficient baselines.

1 Introduction

Text classification is a fundamental task in Natural Language Processing (NLP) with wide-ranging applications, from spam detection to sentiment analysis and topic labeling. This project tackles the problem of news topic classification using the AG News dataset, which contains headlines and descriptions for four distinct categories.

The objective is to implement, evaluate, and compare two different modeling paradigms:

1. **Traditional Machine Learning Models:** Leveraging classical algorithms that rely on statistical features extracted from text, such as Term Frequency-Inverse Document Frequency (TF-IDF).

2. **A Modern Deep Learning Model:** Employing a state-of-the-art transformer-based model, BERT, which is pre-trained on a massive corpus of text and can be fine-tuned to understand the contextual nuances of language.

This comparative analysis aims to quantify the performance trade-offs between these approaches in terms of accuracy and complexity.

2 Exploratory Data Analysis (EDA)

Before modeling, an exploratory data analysis was conducted to understand the dataset's characteristics. A subset of 10,000 training samples and 1,000 test samples

was used for this analysis.

2.1 Class Distribution

The distribution of articles across the four classes was examined. As shown in the output of the notebook, the dataset is well-balanced, with each category having a similar number of samples. This is ideal as it prevents model bias towards a majority class.

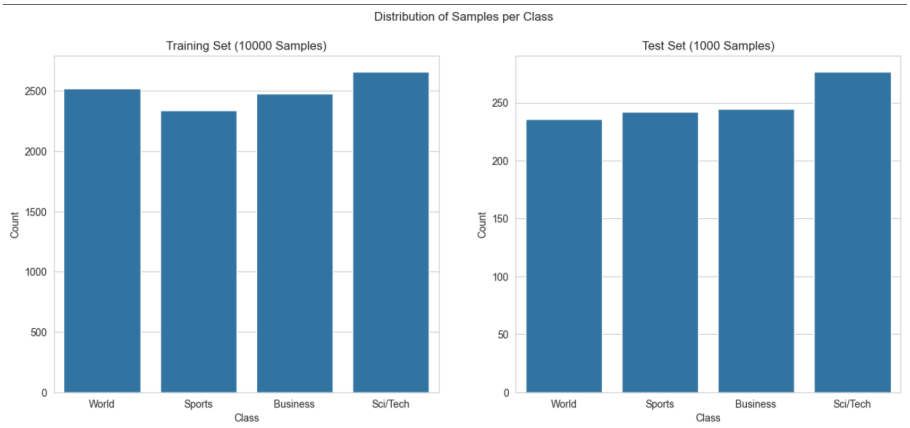

Figure 2.1: Distribution of Samples per class

2.2 Text Length and Word Count

The distribution of text length (character count) and word count revealed a right-skewed distribution. Most articles fall within a moderate length range. This insight is valuable for setting the `max_length` parameter for the BERT tokenizer, helping to balance information retention and computational efficiency.

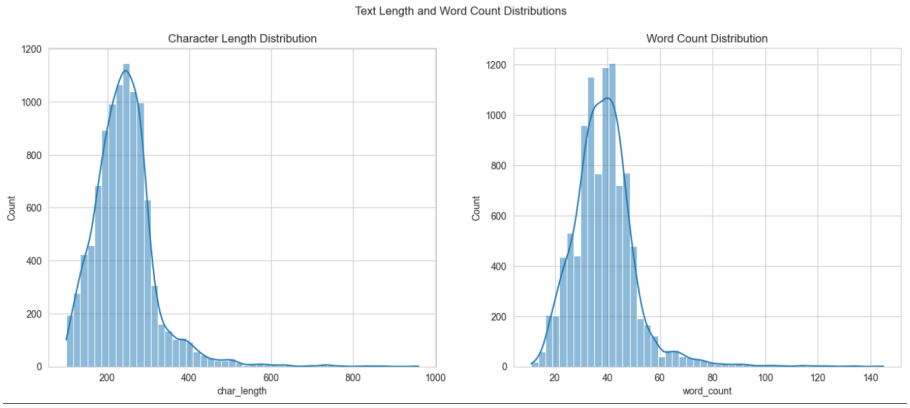

Figure 2.2: Distribution of text length and word count

2.3 Word Frequency

After cleaning the text (lowercase conversion, punctuation removal) and filtering out common English stop words, the frequency of the most common words was analyzed. This provides a glimpse into the important terms that might help distinguish between news categories.

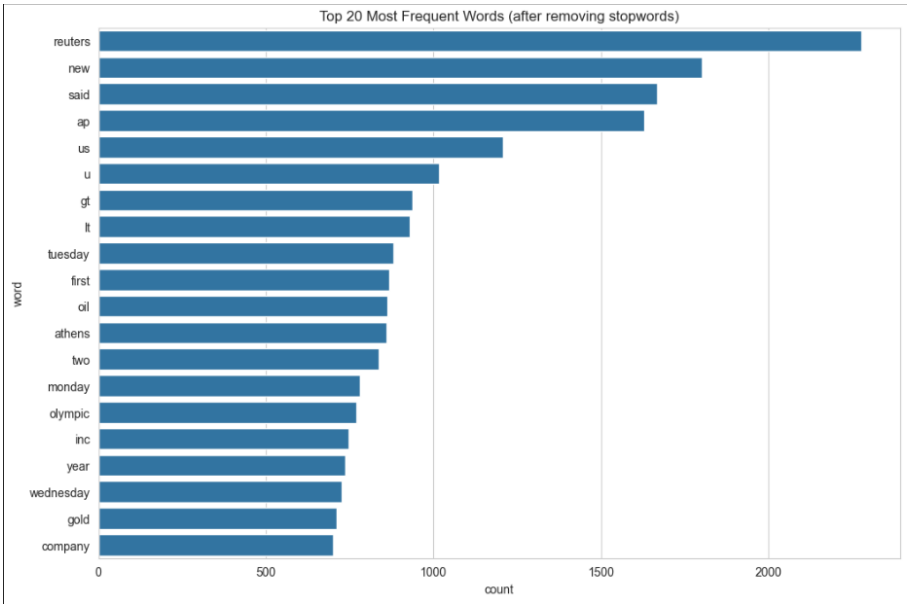

Figure 2.3: Top most frequent words

3 Model Training

The project was structured around two main pipelines: a traditional ML approach and a deep learning approach.

3.1 Theoretical Overview of Traditional Models

The models selected for the traditional approach represent a diverse set of algorithmic principles, from probabilistic methods to linear models, margin-based classifiers, and ensemble techniques.

- **Multinomial Naive Bayes:** This is a probabilistic classifier based on **Bayes' Theorem**. It is "naive" because it makes a strong (and often unrealistic) assumption that the features (in this case, the presence of words) are conditionally independent of each other, given the class. Despite this simplification, it is highly efficient and performs remarkably well in text classification, where it models the frequency of words in each category. It calculates the probability of a document belonging to a class based on the probabilities of the words it contains.

- **Logistic Regression:** Despite its name, Logistic Regression is a model used for classification, not regression. It is a **linear model** that predicts the probability of an instance belonging to a particular class by passing a linear combination of its features through a logistic (or sigmoid) function. The output is a probability value between 0 and 1, which is then mapped to a discrete class. For multi-class problems, it can be extended using a "one-vs-rest" or a "multinomial" approach. It is simple, interpretable, and computationally inexpensive.

- **Support Vector Classifier (SVC):** This is a powerful and versatile classifier that works by finding an optimal **hyperplane** in a high-dimensional space that best separates the data points of different classes. The "support vectors" are the data points that lie closest to the decision boundary (the hyperplane). The goal of the algorithm is to maximize the **margin**—the distance between the hyperplane and the nearest support vectors. For non-linearly separable data, SVC uses the "kernel trick" (e.g., the Radial Basis Function or RBF kernel) to map the data into a higher-dimensional space where a linear separation becomes possible.

- **Random Forest Classifier:** This is an **ensemble learning** method that operates by constructing a multitude of decision trees at training time. A decision tree is a simple, flowchart-like model that makes predictions based on a series of feature-based splits. The "forest" is created by training each tree on a random subset of the training data (a technique called bagging) and considering a random subset of features for each split. To make a prediction, the Random Forest aggregates the votes from all individual trees (e.g., by majority vote) and outputs the class that gets the most votes. This ensemble approach reduces overfitting, which is a common problem with single decision trees, and generally leads to higher accuracy.

3.2 Traditional Model Approach

This pipeline involved two main stages:

1. **Feature Extraction:** The raw text was converted into numerical vectors using the **TF-IDF (Term Frequency-Inverse Document Frequency)** method. This technique creates a feature vector for each document where each feature represents a word, and the value is its TF-IDF score.

2. **Model Training:** A `GridSearchCV` was employed to systematically test several traditional classification algorithms with different hyperparameters. The models evaluated were:

   - Multinomial Naive Bayes
   - Logistic Regression
   - Support Vector Classifier (SVC)
   - Random Forest Classifier

   The best-performing model from this search was selected as the representative for the traditional approach.

3.3 Deep Learning Approach (Fine-tuning BERT)

This modern approach leveraged the power of a pre-trained transformer model.

1. **Tokenization:** The text was processed using the `bert-base-uncased` tokenizer, which converts sentences into tokens that the model can understand, including special tokens and attention masks.

2. **Fine-Tuning:** A pre-trained `BertForSequenceClassification` model was fine-tuned on our specific news dataset. This process adjusts the model's weights to specialize it for our classification task. The training was conducted for 2 epochs using the AdamW optimizer on a GPU-enabled environment.

4 Results

All models were evaluated on the same unseen test set. The primary metrics for comparison are Test Accuracy and the macro-averaged F1-Score.

4.1 Performance Comparison

The final performance of all tested models is summarized in Table 4.1. The fine-tuned BERT model achieved the highest performance across all metrics. Among the traditional models, Support Vector Classifier (SVC) with an RBF kernel and C=10 delivered the best results.

Table 4.1: Comprehensive Performance Comparison of All Models

Model	Parameters	Validation Acc.	Test Acc.	Test F1-Score (Macro)
BERT (Fine-tuned)	**bert-base-uncased**	**N/A**	**0.913**	**0.9133**
Logistic Regression	{'C': 10, 'solver': 'liblinear'}	0.857	0.872	0.8729
SVC	**{'C': 10, 'kernel': 'rbf'}**	**0.862**	**0.871**	**0.8712**
Logistic Regression	{'C': 1, 'solver': 'saga'}	0.861	0.869	0.8692
SVC	{'C': 1, 'kernel': 'rbf'}	0.862	0.869	0.8688
Logistic Regression	{'C': 1, 'solver': 'liblinear'}	0.861	0.868	0.8681
MultinomialNB	{'alpha': 0.1}	0.861	0.862	0.8625
Logistic Regression	{'C': 10, 'solver': 'saga'}	0.854	0.861	0.8618
MultinomialNB	{'alpha': 0.5}	0.862	0.860	0.8606
SVC	{'C': 1, 'kernel': 'linear'}	0.860	0.859	0.8597
MultinomialNB	{'alpha': 1.0}	0.861	0.857	0.8578
RandomForestClassifier	{'max_depth': None, ...}	0.823	0.825	0.8252
...	...	...	...	...

4.2 Classification Reports

The detailed classification reports for the BERT provide per-class precision, recall, and F1-scores, offering deeper insight into their performance on each news category.

```
================================================================================
                DETAILED REPORT FOR THE BEST PERFORMING MODEL
================================================================================
The best performing model is: BERT (Fine-tuned) with parameters bert-base-uncased

              precision    recall  f1-score   support

       World       0.91      0.92      0.92       236
      Sports       0.98      0.98      0.98       242
    Business       0.89      0.82      0.86       245
    Sci/Tech       0.87      0.92      0.89       277

    accuracy                           0.91      1000
   macro avg       0.91      0.91      0.91      1000
weighted avg       0.91      0.91      0.91      1000
```

Figure 4.1: Detailed report for BERT

5 Discussion

The results clearly indicate that the deep learning approach with a fine-tuned BERT model significantly outperforms all traditional machine learning models. With a test accuracy of **91.3%**, BERT demonstrates a strong capability to understand the semantic context of the news articles, leading to more accurate classifications.

The best traditional model, SVC, achieved a respectable accuracy of **87.1%**. This shows that with proper feature engineering (TF-IDF) and hyperparameter tuning, classical ML methods can still provide a powerful and effective baseline.

The performance gap highlights a key trade-off in NLP:

- **Performance:** BERT's ability to process words in context gives it a distinct advantage over "bag-of-words" models like TF-IDF, which lose word order and syntactic information.
- **Complexity and Cost:** Fine-tuning BERT is computationally intensive, requiring GPU resources and significantly more time than training traditional models. The traditional pipeline is much faster and can be run on a standard CPU.

6 Conclusion

This project successfully compared traditional and modern approaches for text classification on the AG News dataset. The fine-tuned BERT model was the undisputed top performer, showcasing the power of transformer architectures in natural language understanding.

For applications where maximizing accuracy is the primary goal, the fine-tuned **BERT model is the recommended solution**. However, for scenarios where computational resources are limited or a faster, simpler solution is required, a well-tuned **Support Vector Classifier (SVC)** with TF-IDF features serves as an excellent and highly effective baseline.