Comparative Report on Text Classification Models

Ai22btech11013
Ai22btech11025
Ai22btech11006

March 1, 2025

Abstract

This report examines the performance of four text classification models on two datasets: Dataset-1 (Scientific Documents) and A2D2 (Entertainment, Healthcare, Sports, Technology, and Tourism). We evaluate the performance of Naïve Bayes, Support Vector Machine (SVM), Random Forest, and a deep learning LSTM model using similar preprocessing and feature extraction techniques. The report discusses the accuracy results and provides a comparative analysis of why the observed performance order emerged for each dataset.

1 Introduction

Text classification is a fundamental task in natural language processing. This report compares traditional machine learning models with a deep learning approach (LSTM) across two datasets with distinct characteristics. While Dataset-1 comprises scientific documents with technical language and an extensive vocabulary, the A2D2 dataset contains content from domains such as Entertainment, Healthcare, Sports, Technology, and Tourism with clearer domain separation.

2 Methodology

2.1 Data Preprocessing & Feature Extraction

For both datasets, we applied a similar preprocessing pipeline:

- **Text Normalization:** All text was converted to lowercase to ensure uniformity.
- **Tokenization and Vectorization:**

  - **Classical Models:** TF-IDF (Term Frequency-Inverse Document Frequency) was used to convert text into numerical feature vectors.
  - **LSTM Model:** A vocabulary was built from the training texts. Each text was converted into a sequence of token indices and then padded or truncated to a fixed length.

- **Label Encoding:** Categorical labels were transformed into numerical values.
- **Data Splitting:** A manual, per-class split was performed (50% for training, 30% for validation, 20% for testing) to maintain balanced class distributions.

2.2 Models Implemented

We implemented the following four models:

1. **Naïve Bayes (MultinomialNB):** A probabilistic classifier that assumes feature independence, particularly effective when using TF-IDF features.
2. **Support Vector Machine (SVM) – LinearSVC:** A linear classifier known for its performance in high-dimensional spaces, utilizing a hinge loss function.
3. **Random Forest Classifier:** An ensemble method that combines multiple decision trees to capture non-linear relationships and mitigate overfitting.
4. **LSTM (Long Short-Term Memory Network):** A deep learning model designed to capture sequential dependencies. The architecture consists of an embedding layer, a bidirectional LSTM layer, and a fully connected output layer. Early stopping based on validation loss was used to prevent overfitting.

3 Results and Analysis

3.1 Dataset-1:

Accuracy Results:

- Naïve Bayes: $\sim 78.7\%$
- SVM: $\sim 79.5\%$
- Random Forest: $\sim 77.1\%$
- LSTM: $\sim 71.1\%$

Analysis:

- **SVM and Naïve Bayes:** These models achieved the highest accuracies by effectively leveraging TF-IDF features to capture key domain-specific terms within the scientific documents. The SVM's margin maximization helps it generalize better in a high-dimensional feature space.
- **Random Forest:** Although robust, it slightly underperformed compared to SVM and Naïve Bayes, possibly due to the sparse nature of the TF-IDF features which can pose challenges for tree-based methods.
- **LSTM:** The deep learning model struggled with Dataset-1. The complex and nuanced vocabulary typical of scientific texts required more data and finer hyperparameter tuning, leading to the lowest accuracy among the models.

3.2 A2D2 Dataset: Entertainment, Healthcare, Sports, Technology, and Tourism

Accuracy Results:

- Naïve Bayes: $\sim 99.75\%$
- SVM: 100%
- Random Forest: $\sim 99.87\%$
- LSTM: $\sim 93.5\%$

Analysis:

- **Naïve Bayes, SVM, and Random Forest:** These classical models achieved near-perfect accuracy on the A2D2 dataset. The distinct and straightforward language across domains enabled TF-IDF to capture discriminative features effectively, resulting in a highly separable feature space.
- **LSTM:** Although the LSTM model was effective at modeling sequential data, it did not reach the performance level of the classical models on this dataset. The relatively simpler text structure and limited data per class meant that the additional complexity of the LSTM did not provide a substantial advantage.

3.3 Comparative Analysis and Observed Performance Order

Dataset-1 :

- **Performance Order:** SVM $\approx$ Naïve Bayes $>$ Random Forest $>$ LSTM

- **Discussion:** The nuanced and technical language of scientific documents required models to capture fine-grained details. Both SVM and Naïve Bayes excelled by leveraging TF-IDF features. SVM's margin maximization allowed for better generalization in a high-dimensional space, while Random Forest struggled slightly with the sparsity of the feature space. The LSTM model, although capable of capturing sequential patterns, was hampered by the complex vocabulary and limited dataset size.

A2D2 Dataset:

- **Performance Order:** SVM $\approx$ Random Forest $\approx$ Naïve Bayes $\gg$ LSTM
- **Discussion:** The clear domain separation and simpler language of the A2D2 dataset allowed classical models to achieve near-perfect accuracy. The straightforward feature space created by TF-IDF was effectively exploited by SVM, Random Forest, and Naïve Bayes. In contrast, the LSTM model, although robust, did not gain significant benefits from modeling sequential dependencies on this relatively simple dataset and was more sensitive to hyperparameter settings.

4 Conclusion

In summary, this report compared four text classification models across two datasets:

- For **Dataset-1 (Scientific Documents)**, the classical models (SVM and Naïve Bayes) outperformed the others by efficiently capturing domain-specific features through TF-IDF, while the LSTM lagged behind due to the high complexity and vocabulary diversity.
- For the **A2D2 Dataset**, the distinct and clear language allowed SVM, Random Forest, and Naïve Bayes to achieve near-perfect accuracy, whereas the LSTM did not perform as well because the sequential modeling advantage was less pronounced.

This comparative analysis demonstrates that while deep learning models like LSTM can capture sequential information, traditional machine learning models may offer superior performance when the text is less complex and more clearly separated by domain.