Report: Advanced Text Classification for Multi-Class Document Categorization

1. Introduction:

Text classification is a fundamental task in natural language processing (NLP), with applications ranging from sentiment analysis to document categorization. In this report, we present our approach to building a text classification model capable of predicting the class of unseen documents from a set of 91 categories. We detail our methodology, including data preprocessing, feature encoding, model selection, and evaluation.

2. Methodology:

Data Preprocessing:

We began by preprocessing the training dataset, which consisted of 91 folders, each representing a document class. We followed these steps:

Word Tokenization: We split the text into uni-gram tokens using standard tokenization techniques.

Token Normalization: We employed Porter's Stemmer to normalize tokens, reducing them to their base forms.

Vocabulary Set Extraction: We extracted a vocabulary set from the preprocessed tokens to build our feature space.

Word Stemming: Alongside token normalization using Porter's Stemmer, we applied word stemming to further reduce tokens to their base forms.

Removing digits: We removed numeric digits from the text before tokenization to enhance the accuracy of our model. Numeric digits often lack significant semantic meaning and can introduce noise into the data. This preprocessing step helps the model focus on relevant textual information, improving classification performance.

Feature Encoding:

For feature encoding, we explored two approaches:

TF-IDF Feature Encoding: We used the Term Frequency-Inverse Document Frequency (TF-IDF) method to encode our features, which assigns weights to terms based on their frequency and inverse document frequency.

Word Embeddings: We utilized pre-trained word embedding model (word2vec) to represent words in a continuous vector space.

Model Selection and Training:

We experimented with various machine learning and deep learning algorithms:

Machine Learning Algorithms: We tried Naïve Bayes, Support Vector Machine (SVM), logistic regression , and Random Forest classifiers.

Deep Learning Models: We implemented Long Short-Term Memory (LSTM) networks with embedding layers.

We evaluated each model's performance using the macro-averaged F1-score and selected the logistic regression model with both TF-IDF and word embedding features as the best-performing model.

Model Evaluation:

Our evaluation focused on comparing the performance of different algorithms and models. The logistic regression model with TF-IDF and word embedding features consistently outperformed others, achieving the highest F1-score.

3. Results and Discussion:

In conclusion, we developed a text classification model capable of predicting document classes from a diverse set of categories with a 0.29% F1-score. Our approach involved extensive experimentation with various preprocessing techniques, feature encodings, and modeling algorithms. The logistic regression model, augmented with both TF-IDF and word embedding features, emerged as the most effective solution for this task.