**SMS Spam Detection and Text Generation Using Natural Language Analysis**



The increasing prevalence of SMS Spam Detection and Text Generation has recently become a significant concern. Detecting and filtering spam messages is crucial for ensuring effective communication and protecting users from malicious content. This report will explore various techniques and models in natural language analysis to develop an SMS spam detection system. Additionally, we will implement a text generation model to generate "SPAM-like" emails and evaluate the performance of the detection models on the generated samples.

Problem Definition: The task is to classify SMS messages as either spam or legitimate (HAM). We have a dataset comprising 4,827 HAM messages and 747 SPAM messages. The dataset needs an imbalance problem, with approximately 86.6% of the SMS being HAM. The primary challenge is to develop models that can accurately identify spam messages while handling the imbalance issue.

Proposed Solutions: To address the problem, we will employ various techniques and models as follows:

Feature Extraction Techniques: a) Term Frequency-Inverse Document Frequency (TF-IDF): We will implement TF-IDF to convert text documents into numerical representations, assigning weights to words based on their frequency and inverse document frequency. We will explain the working principle, advantages, and disadvantages of TF-IDF. b) Word2vec: We will utilize word embeddings generated by Word2vec, a neural technique, to capture semantic relationships between words. We will discuss the benefits and limitations of Word2vec.

Standard Machine Learning Algorithm: We will employ a standard Machine Learning algorithm, such as Support Vector Machines (SVM), Logistic Regression, or Random Forest, using the features extracted in the previous step. We will train the algorithm and evaluate its performance on the testing set, analyzing its accuracy, precision, recall, and F1 score.

**SMS Spam Detection and Text Generation: Deep Learning Model**



We will train a Deep Learning model, such as Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), or Convolutional Neural Network (CNN), using the Word2vec features extracted earlier. We will explain the architecture of the chosen model, hyper-parameter settings, and the loss function used. Justification for these choices will be provided for you if you are okay with it.

Performance and Training Time Analysis: We will compare and analyze the performance and training time results of both the Machine Learning and Deep Learning models. The evaluation metrics will include accuracy, precision, recall, and F1 score. We will discuss the strengths and weaknesses of each model in the context of SMS spam detection.

Text Generation Model: We will build a text generation model that generates new "SPAM-like" emails using the training data of the 'SPAM' class. We will explain the working principle of the model and how it creates text. The model will produce 100 samples, which will be used for further evaluation.

Evaluation of Generated Samples: Using the 100 generated samples from the previous step, we will test the performance of the Machine Learning and Deep Learning models developed in steps 2 and 3. We will report and discuss the results, analyzing how well the models classify the generated samples and their performance in the presence of artificially generated spam-like content.

Analysis of Results: We will present the results of each step, including the performance metrics of the Machine Learning and Deep Learning models on the testing set. We will compare and contrast the results, identifying the strengths and weaknesses of each approach. Also, we will evaluate the performance of the models on the generated spam-like samples and discuss any observed differences.

Discussion: In the discussion section, we will delve deeper into the findings, addressing the challenges encountered during the project and the limitations of the proposed solutions. We will explore possible strategies to mitigate the class imbalance issue and improve the performance of the models. We will also consider potential ethical implications and discuss the impact of our work on real-world applications.

Conclusion: This report comprehensively analyzes SMS spam detection using natural language analysis techniques. We have implemented feature extraction techniques, employed Machine Learning and Deep Learning models, and evaluated their performance on a provided dataset. Additionally, we have developed a text generation model to test the detection models on spam-like samples. The findings and insights gained from this study can contribute to developing robust SMS spam detection systems and further advancements in natural language analysis.

https://youtu.be/YncZ0WwxyzU

**References:**

"SMS Spam Filtering: A Review": This survey paper provides an overview of various techniques and approaches for SMS spam filtering. It discusses feature extraction methods, machine learning algorithms, and evaluation measures. [Link: <https://ieeexplore.ieee.org/document/7467755>]"SMS Spam Detection using Machine Learning Techniques": This research paper presents a comparative study of various machine learning algorithms for SMS spam detection. It evaluates the performance of algorithms such as Naive Bayes, Support Vector Machines, and k-Nearest Neighbors. [Link: <https://ieeexplore.ieee.org/document/7754811>]

"Language Models are Unsupervised Multitask Learners": This paper introduces OpenAI's GPT (Generative Pre-trained Transformer) model, a state-of-the-art language model for text generation tasks. It explains the architecture, training procedure, and various applications of the model. [Link: <https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf>]

"GPT-2: Language Models are Unsupervised Multitask Learners": This paper describes the GPT-2 model, an advanced version of GPT, which significantly improves text generation. It discusses model architecture, training objectives and presents evaluation results. [Link: <https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf>]