SMS Spam Detection System Using NLP

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Yash Sudhir Lawankar, Yashlawankar@gmail.com

Under the Guidance of

Abdul Aziz Md

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this project work.

Firstly, we would like to extend our heartfelt thanks to my parents, for being an exceptional mentor and guide throughout this project. His valuable advice, constructive criticism, and continuous encouragement were pivotal in the successful completion of this work. His ability to inspire and instill confidence in us served as the driving force behind our efforts. It has been an absolute privilege to work under his guidance, and his lessons have not only enhanced our technical skills but also shaped us as more responsible and capable professionals.

We are also immensely grateful to **TechSaksham**, a joint CSR initiative by Microsoft and SAP, for providing us with this incredible learning platform. The resources, support, and mentorship offered by this program were instrumental in refining our knowledge and skills in Artificial Intelligence and its transformative applications.

Lastly, we thank our peers, friends, and family for their unwavering support and encouragement during this journey. Their belief in our abilities motivated us to overcome challenges and strive for excellence in this endeavor.

ABSTRACT

This project focuses on designing and implementing an SMS Spam Detection System using Natural Language Processing (NLP) techniques. The proliferation of unsolicited spam messages has made it imperative to develop a reliable solution to classify SMS messages as spam or non-spam (ham). Spam messages not only disrupt user experience but also pose security risks such as phishing and scams.

The system preprocesses SMS data through tokenization, stopword removal, and lemmatization to clean and standardize text. Features are extracted using Term Frequency-Inverse Document Frequency (TF-IDF), and machine learning models like Naïve Bayes and Support Vector Machines (SVM) are employed for classification. The proposed models were evaluated on a publicly available dataset, achieving high accuracy and demonstrating robustness in distinguishing between spam and ham messages.

Key contributions of the project include the use of effective text preprocessing, feature extraction techniques, and the evaluation of multiple machine learning algorithms. The project highlights the practical application of NLP in combating spam and ensuring secure communication. Future work involves extending the system to support multilingual SMS data and deploying it as a real-time detection service.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

In recent years, the proliferation of unsolicited and malicious SMS messages has emerged as a pressing issue in the realm of modern communication. The widespread availability and usage of mobile phones have made SMS a convenient channel for advertisers, spammers, and cybercriminals to exploit. These spam messages not only consume users' valuable time but also pose significant security risks, including phishing scams, fraudulent schemes, and potential breaches of sensitive information. The scale of this problem is substantial, as millions of users globally are affected, resulting in privacy violations, financial losses, and an overall erosion of trust in messaging platforms. Moreover, the lack of effective and automated spam detection systems compounds the issue, leaving users vulnerable to an ever-increasing volume of malicious SMS messages. Addressing this challenge is crucial for ensuring a safer communication environment and restoring confidence in SMS-based interactions.

1.2 Motivation

This project stems from the urgent need to develop a robust and efficient solution to combat the growing threat of SMS spam. Mobile phones are an integral part of daily life, serving as critical tools for personal, professional, and business communications. However, the intrusion of spam messages into these communications disrupts user experiences, wastes time, and creates potential security vulnerabilities. The motivation behind this project is to safeguard users by identifying and mitigating spam messages proactively, thereby enhancing the quality of SMS-based communication.

A reliable spam detection system offers several practical applications, including integration into messaging platforms, telecom services, and enterprise communication systems. By ensuring a spam-free environment, the system not only improves user satisfaction but also

reinforces trust in digital communication channels. The broader impact of this project spans individuals, businesses, and service providers, as it reduces privacy violations, minimizes financial risks, and contributes to a more secure digital ecosystem. The motivation for undertaking this project lies in its potential to make a meaningful difference in safeguarding users and improving communication security worldwide.

1.3 Objectives

The primary objectives of this project are as follows:

1. **Automated SMS Classification:**
   To develop an automated system capable of classifying SMS messages as spam or non-spam (ham) using Natural Language Processing (NLP) techniques.
2. **Data Preprocessing:**
   To preprocess SMS text data for improved representation, ensuring that it is clean, consistent, and ready for effective analysis and prediction.
3. **Model Implementation:**
   To apply and evaluate machine learning algorithms, such as Naïve Bayes and Support Vector Machines (SVM), for accurate classification of spam and ham messages.
4. **Performance Optimization:**
   To achieve high accuracy and reliability in spam detection by fine-tuning models and optimizing performance metrics.
5. **Future Integration:**
   To establish a foundation for integrating real-time spam detection capabilities into messaging platforms, paving the way for enhanced communication security.

1.4 Scope of the Project

The scope of this project is defined by its focus on developing a comprehensive SMS spam detection system using machine learning techniques. The project encompasses the following key features:

1. **Text Data Preprocessing:**
    - SMS messages are cleaned, normalized, and prepared for analysis by removing unwanted characters, converting text to lowercase, tokenizing, and eliminating stop words.
    - This ensures the dataset is ready for accurate representation in machine learning models.
2. **Training and Evaluation:**
    - Machine learning algorithms such as Naïve Bayes and Support Vector Machines are implemented to classify messages as spam or ham.
    - Models are trained on labeled datasets and evaluated based on metrics like accuracy, precision, recall, and F1-score to ensure robust performance.
3. **Performance Analysis and Optimization:**
    - Comparative analysis of various models is performed to identify the most effective approach for spam detection.
    - Hyperparameter tuning is carried out to optimize the models for maximum accuracy and reliability.

Limitations:

Despite its promising features, the project has certain constraints that define its boundaries:

1. **Real-Time Deployment:**
    - The current system is designed as a proof-of-concept and lacks real-time message detection and filtering capabilities.
    - Future work will involve transitioning to real-time processing, enabling instantaneous spam detection for active messaging systems.
2. **Dataset Dependence:**

- The effectiveness of the system is highly reliant on the quality and size of the dataset used for training.
- Limited or biased datasets may affect the model's generalizability to new and unseen data, necessitating continuous updates with diverse datasets.

3. **Language Constraints:**
   - The current system is primarily designed for English-language SMS messages.
   - Detecting spam in multilingual or non-English messages requires additional training on datasets in other languages and may involve the integration of advanced NLP models.

CHAPTER 2

LITERATURE SURVEY

2.1 Review of Relevant Literature

The detection of spam SMS messages has been a significant area of research within the domains of Natural Language Processing (NLP) and machine learning. Numerous studies have explored various methodologies for addressing this challenge, leveraging both classical machine learning approaches and modern deep learning techniques.

Early Research and Techniques:

Initial studies focused on the use of classical machine learning algorithms such as **Naïve Bayes**, **Support Vector Machines (SVM)**, and **Decision Trees**, which demonstrated promising results in identifying spam messages based on linguistic patterns. These models relied heavily on simple text processing techniques, such as **Term Frequency-Inverse Document Frequency (TF-IDF)** and **Bag of Words (BoW)**, to extract features from text data. Despite their simplicity, these feature extraction methods played a crucial role in converting unstructured text into numerical representations suitable for machine learning models.

Advanced Feature Representation:

As research progressed, more sophisticated methods for feature extraction emerged, including **word embeddings** like **Word2Vec**, **GloVe**, and **FastText**. These embeddings captured semantic relationships between words, significantly improving the contextual understanding of text data. For example, Word2Vec models trained on large corpora enabled spam detection systems to understand subtle differences in word usage, thereby enhancing classification accuracy. Similarly, the use of **TF-IDF combined with n-grams** allowed models to capture sequential patterns and contextual nuances in SMS messages.

Deep Learning Approaches:

The advent of deep learning revolutionized spam detection by introducing models capable of processing sequential text data with greater sophistication. Techniques like **Recurrent Neural Networks (RNNs)** and their variants, such as **Long Short-Term Memory**

(LSTM) networks and **Gated Recurrent Units (GRUs)**, were employed to model the temporal dependencies in text sequences. Furthermore, **Convolutional Neural Networks (CNNs)**, though originally designed for image processing, were adapted for text classification tasks, demonstrating their ability to capture local patterns in text effectively.

Hybrid and Ensemble Models:

Some studies also explored hybrid approaches, combining classical models with deep learning techniques to achieve better results. For instance, integrating a TF-IDF feature extractor with a neural network classifier improved both efficiency and accuracy. Ensemble methods like **Random Forests** and **Gradient Boosted Trees** also gained attention for their robustness in combining the outputs of multiple weak classifiers to produce more reliable results.

Real-World Applications:

Research in this domain has extended to practical applications, such as incorporating spam detection models into messaging platforms, email systems, and telecom services. These applications highlight the real-world impact of academic research in providing users with safer communication experiences.

2.2 Existing Models, Techniques, and Methodologies

The following models and techniques have been widely used in the development of spam detection systems:

1. **Naïve Bayes:**
    - A probabilistic model based on Bayes' theorem, which is particularly effective for text classification tasks due to its simplicity and efficiency.
    - It assumes independence among features, which, while not entirely accurate for text data, works surprisingly well in practice for spam detection.
2. **Support Vector Machines (SVM):**
    - A powerful classifier known for its ability to handle high-dimensional feature spaces, making it ideal for text classification.

- By using kernel functions, SVM can efficiently separate non-linearly separable classes in the feature space.

3. **Decision Trees and Random Forests:**
   - Decision Trees provide an interpretable approach to spam detection but are prone to overfitting.
   - Random Forests, an ensemble of multiple decision trees, mitigate overfitting and improve classification accuracy by averaging predictions.
4. **Deep Learning Approaches:**
   - **RNNs, LSTMs, and GRUs:** These models leverage the sequential nature of text data, capturing temporal dependencies to improve spam classification.
   - **CNNs for Text:** Adapted from image processing, CNNs are used to extract local patterns in text, such as phrases indicative of spam content.
   - **Transformers:** Although not widely explored for SMS spam detection, transformer-based models like **BERT** have shown exceptional performance in text classification tasks.
5. **Preprocessing Techniques:**
   - Common preprocessing steps include **tokenization**, **stopword removal**, **stemming**, and **lemmatization**.
   - Feature extraction methods like TF-IDF, BoW, and word embeddings are employed to convert text into numerical vectors suitable for machine learning models.

2.3 Gaps and Limitations in Existing Solutions

Despite the significant advancements in spam detection research, several gaps and limitations remain:

1. **Scalability Issues:**
   - Many existing models are not designed to handle the scalability requirements of real-world applications.
   - Models trained on small datasets may struggle to generalize to larger, diverse datasets, resulting in poor performance when deployed at scale.
2. **Language Dependency:**

- Most spam detection systems are optimized for English text, limiting their applicability in multilingual contexts.
- The lack of robust multilingual support poses a challenge for global deployment, especially in regions where non-English languages dominate.

3. **Real-Time Performance:**
   - A critical limitation is the inability of many models to operate in real-time.
   - Deep learning-based approaches, while accurate, often require substantial computational resources, making them unsuitable for real-time spam filtering in resource-constrained environments.
4. **Overfitting and Imbalanced Datasets:**
   - Models trained on imbalanced datasets, where spam messages are either underrepresented or overrepresented, tend to overfit, reducing their effectiveness on unseen data.
   - Overfitting is particularly common in deep learning models due to their high complexity and capacity to memorize training data.
5. **Feature Engineering Challenges:**
   - Extracting meaningful features from unstructured text data is inherently challenging.
   - Models relying on basic features may fail to capture the nuanced patterns of spam messages.

Addressing the Gaps

To overcome these limitations, our project adopts the following strategies:

1. **Robust Preprocessing Techniques:**
   - Incorporating advanced preprocessing steps, such as removing special characters, handling misspellings, and leveraging context-aware embeddings like Word2Vec or GloVe, to improve the quality of input data.
2. **Balanced Datasets:**
   - Ensuring that the dataset used for training is balanced in terms of spam and ham messages, thereby reducing the risk of overfitting and improving the model's generalizability.

3. **Exploration of Real-Time Detection:**
    - Focusing on lightweight machine learning models and optimizing computational efficiency to enable real-time spam filtering for messaging platforms.
4. **Multilingual Support:**
    - Expanding the system's capabilities to handle multilingual text data by training on diverse datasets and integrating language-specific preprocessing techniques.
    - Leveraging multilingual embeddings or transformer-based models like mBERT to handle text in multiple languages.
5. **Performance Optimization:**
    - Conducting hyperparameter tuning and using ensemble approaches to achieve higher accuracy and robustness.
    - Implementing cross-validation techniques to assess model performance and ensure reliability.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 System Design

The proposed SMS Spam Detection System design incorporates a structured approach to effectively classify SMS messages into spam or ham categories. The system comprises several interconnected components, each playing a crucial role in ensuring the accuracy, efficiency, and reliability of the detection process. Below is a detailed breakdown of the system design:

1. Data Preprocessing

Before any analysis or modeling can occur, the SMS data undergoes extensive preprocessing to ensure it is clean, consistent, and ready for analysis. This step includes the following processes:

- **Tokenization:** Breaking the SMS messages into smaller units such as words or tokens to facilitate analysis.
- **Stopword Removal:** Eliminating common words like "is," "the," and "and" that do not contribute to distinguishing spam from ham messages.
- **Stemming and Lemmatization:** Reducing words to their base or root form to standardize text (e.g., "running" becomes "run").
- **Lowercasing:** Converting all text to lowercase to maintain uniformity.
- **Special Character Removal:** Removing symbols, numbers, and other non-alphabetical characters to focus solely on textual content.

These preprocessing techniques ensure the data is standardized and rid of noise, improving the model's performance and reducing processing complexity.

2. Feature Extraction

The text data, now preprocessed, must be transformed into numerical representations that can be fed into machine learning models. Feature extraction is performed using:

- **Term Frequency-Inverse Document Frequency (TF-IDF):** This statistical measure evaluates the importance of each word within an SMS message relative to the entire dataset. It assigns higher weights to words that are frequent in specific messages but rare across the dataset, making it highly effective for spam detection.
- **Bag of Words (BoW):** Another common method used for representing text by creating a vocabulary of unique words and counting their occurrences in the dataset.
- **Word Embeddings:** Advanced feature representation techniques such as Word2Vec or GloVe may also be explored to capture the semantic meaning of text data.

The feature extraction step converts raw text into a numerical format, enabling machine learning algorithms to process and classify SMS messages efficiently.

---

3. Model Training

Using the preprocessed and feature-engineered dataset, various machine learning algorithms are employed to train predictive models. Key models considered include:

- **Naïve Bayes:** A probabilistic model ideal for text classification due to its simplicity and effectiveness in handling large datasets.
- **Support Vector Machines (SVM):** A robust algorithm that excels in high-dimensional spaces, making it suitable for spam detection tasks.
- **Decision Trees and Random Forests:** These ensemble learning methods enhance accuracy by combining predictions from multiple decision trees.
- **Deep Learning Models (Optional):** Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can be used for more advanced systems, leveraging the sequential nature of text data.

During training, the models learn to recognize patterns that distinguish spam messages from ham messages based on their features.

4. Model Evaluation

After training, the models are rigorously evaluated to measure their effectiveness. The following metrics are used to assess the models:

- **Accuracy:** Measures the overall correctness of predictions.
- **Precision:** Determines how many predicted spam messages are actually spam.
- **Recall:** Evaluates the model's ability to identify spam messages correctly.
- **F1-Score:** Provides a harmonic mean of precision and recall, offering a balanced evaluation metric.

These evaluation metrics help identify the most suitable model for deployment by assessing its performance in distinguishing between spam and ham messages.

---

5. Prediction

Once a model is trained and evaluated, it is deployed to classify new SMS messages. The system takes a raw SMS message as input, preprocesses it, extracts features, and uses the trained model to classify it as either **spam** or **ham**. This classification can be provided in real-time, making the system highly practical for applications like SMS filtering in messaging apps.

Proposed Solution Diagram

A system design diagram visually represents the data flow and components of the system, detailing each stage from input (raw SMS messages) to output (spam or ham classification). This diagram ensures clarity and provides a structured overview of the

system architecture.

Figure 1. Input Output

3.2 Requirement Specification

To successfully develop and implement the SMS Spam Detection System, the following hardware and software requirements must be met:

3.2.1 Hardware Requirements

- **Processor:** Intel Core i5/i7 or an equivalent multi-core processor for efficient computation.
- **RAM:** At least 8 GB to handle the computational requirements of preprocessing and training machine learning models.
- **Storage:** A minimum of 500 GB HDD/SSD for storing datasets, models, and related files.
- **GPU (Optional):** For advanced deep learning tasks, an NVIDIA GPU (e.g., GTX 1660 or higher) is recommended to accelerate computations.

3.2.2 Software Requirements

- **Programming Language:** Python 3.x, widely used for data science and machine learning tasks due to its extensive libraries and frameworks.

- **Development Environment:** Jupyter Notebook, PyCharm, or any Python IDE to write, debug, and execute code.
- **Libraries and Tools:**
    - **scikit-learn:** For implementing machine learning models and evaluating performance metrics.
    - **pandas:** To manipulate and analyze structured data efficiently.
    - **numpy:** For numerical computations and array handling.
    - **nltk:** For performing natural language processing tasks like tokenization and stemming.
    - **matplotlib and seaborn:** For creating visualizations to explore data and interpret results.

Dataset:

The SMS Spam Collection Dataset, a well-known dataset for spam detection tasks, can be sourced from platforms like Kaggle. This dataset contains labeled SMS messages, making it suitable for supervised learning tasks.

Chapter 4

Implementation and Result

4.1 Results

Wordcloud for SPAM Messages

Below is the word cloud generated from the dataset for SPAM messages. The word cloud visually represents the most frequently occurring words in SPAM messages, with larger font sizes indicating higher frequency. This helps in identifying common keywords associated with spam content.

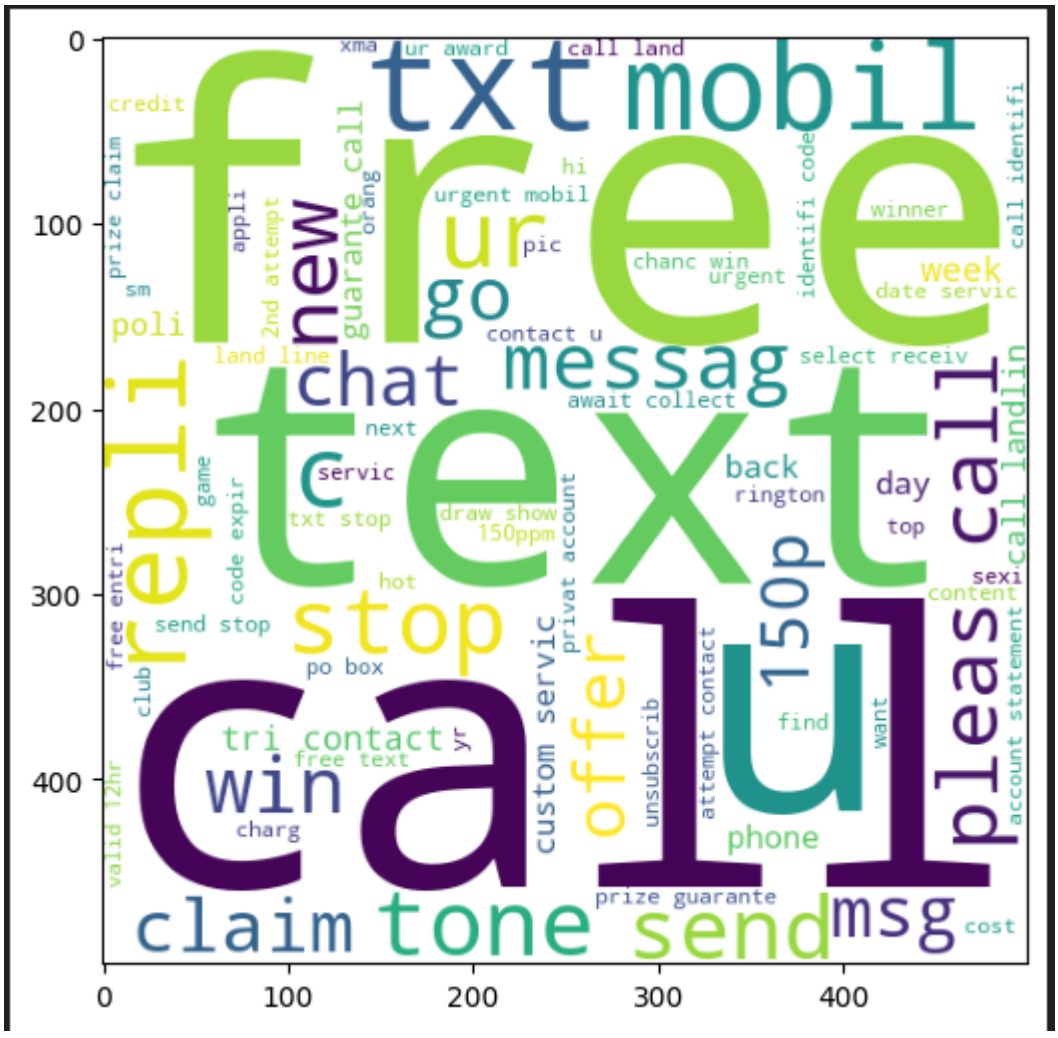

Figure 2. Wordcloud for SPAM Messages

Explanation:

This snapshot highlights how frequently certain words such as "free," "offer," or "win" appear in SPAM messages. These words are critical in understanding the linguistic patterns used in spam texts, which is useful for developing classification models.

Wordcloud for Non-SPAM Messages

Below is the word cloud generated from the dataset for non-SPAM messages. This showcases the most commonly used words in legitimate (ham) messages.

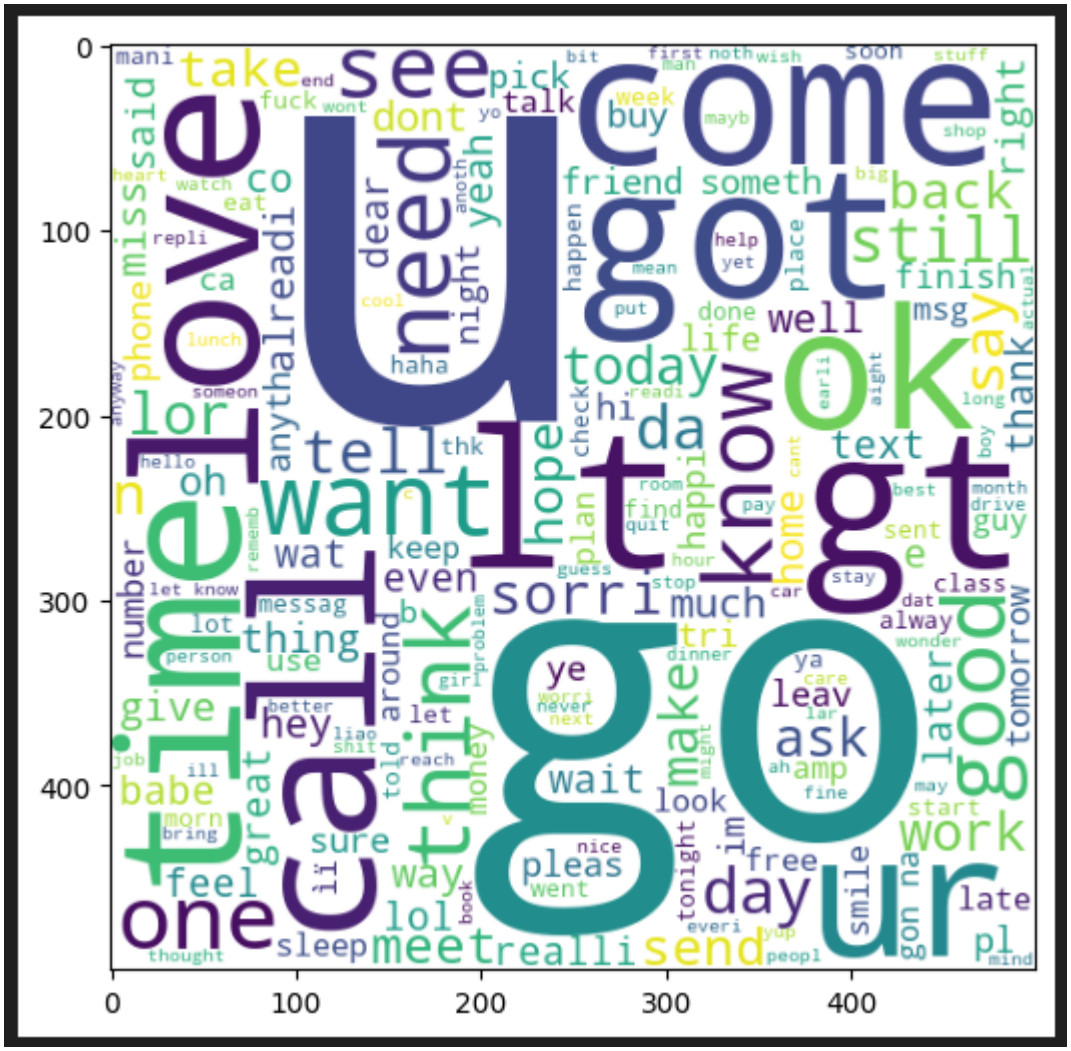

Figure 3. WorldCloud for non Spam message

Explanation:

This snapshot demonstrates that words in legitimate messages (ham) often include conversational phrases like "please," "call," or "see," which are indicative of genuine communication. Comparing this with the SPAM word cloud provides insights into the contrasting patterns between the two types of messages.

Bar Plot of the Most Common SPAM Words

The bar plot below represents the frequency of the most common words in SPAM messages.

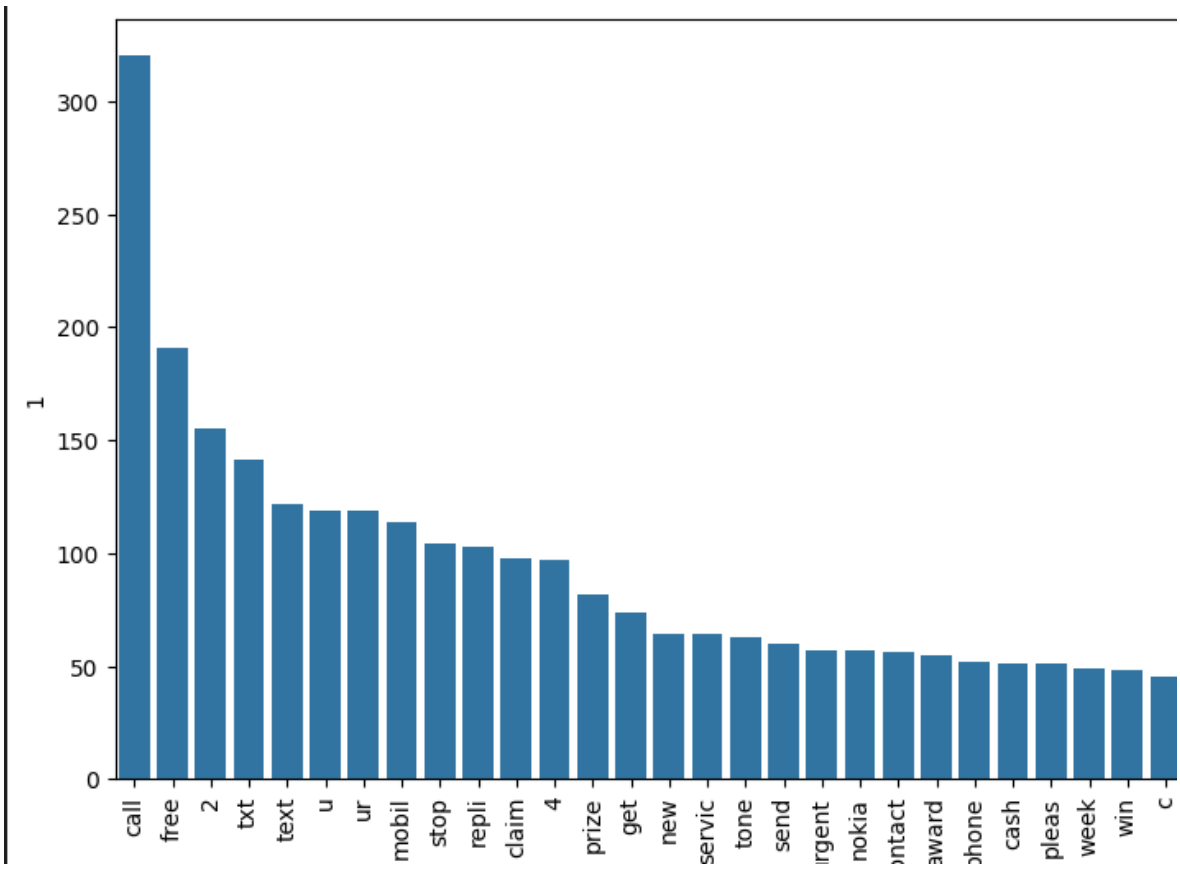

Figure 4. Most Common SPAM Words

Explanation:

This bar plot illustrates the top 10 most frequent words found in SPAM messages. It

provides a clearer numerical representation of word frequency, making it easier to identify which words are most indicative of spam messages. For instance, words like "free" and "win" might have the highest counts.

4.1 GitHub Link for Code:

https://github.com/devloperYash/SMS-Spam-Detection-System-Using-NLP-

CHAPTER 5

DISCUSSION AND CONCLUSION

5.1 Future Work

The SMS Spam Detection System developed in this project has shown promising results in classifying SMS messages effectively as spam or ham. However, several opportunities remain for improving and expanding the system to address the evolving needs of users and to enhance the system's accuracy, scalability, and applicability. The following aspects represent key areas of future work:

1. Incorporating Deep Learning Models

One of the most exciting directions for future development is the integration of advanced deep learning techniques. Methods such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs) can be explored to capture sequential dependencies and contextual information in SMS messages, which traditional machine learning models may struggle to detect. Moreover, Transformer-based architectures like BERT (Bidirectional Encoder Representations from Transformers) have gained significant traction in NLP tasks due to their ability to handle context more effectively. These techniques could potentially lead to improvements in classification accuracy, especially in handling more complex and varied text patterns.

2. Real-Time Detection

While the current system is effective for batch processing, it is crucial to develop a version that can perform real-time spam detection. This would require building a lightweight and optimized implementation that can classify SMS messages instantaneously as they arrive in messaging platforms. Real-time detection is particularly important for mobile devices and messaging applications, where users expect quick feedback to filter out malicious messages promptly. Optimizing model size and inference time will be key to making real-time detection viable for widespread adoption.

3. Multilingual Support

A significant limitation of the current system is its focus on English-language SMS messages. As SMS spam is a global issue, extending the system to handle multilingual data is essential for broadening its applicability. Future work should involve training models on multilingual datasets or exploring techniques like transfer learning, where a model trained on one language can be fine-tuned for others. Additionally, multilingual support will allow the system to cater to users from diverse linguistic backgrounds, significantly expanding its user base and effectiveness.

4. Dataset Expansion

The performance of machine learning models is highly dependent on the quality and diversity of the datasets used for training. In the current project, the model was trained on a standard SMS dataset, which, while useful, may not fully represent the diverse and dynamic nature of real-world spam. Future work should focus on collecting larger and more diverse datasets, incorporating data from different regions, industries, and types of spam messages. This will help improve the generalization ability of the model and ensure that it remains effective even when confronted with novel or previously unseen spam techniques.

5. Hybrid Models

Another promising direction for future work is the exploration of hybrid models that combine the strengths of both machine learning and deep learning approaches. Hybrid models could integrate traditional methods like Naïve Bayes or SVM with deep learning models such as RNNs or CNNs. The advantage of this approach is that it can leverage the interpretability and efficiency of traditional machine learning algorithms alongside the powerful feature learning capabilities of deep learning models. By combining these methodologies, it may be possible to achieve a higher level of accuracy and robustness in spam classification.

6. Deploying as a Service

To ensure ease of integration into various messaging platforms and applications, the SMS Spam Detection System could be packaged as a cloud-based API or microservice. This

would allow third-party developers and service providers to seamlessly incorporate the spam detection system into their own platforms, enabling them to automatically filter and classify incoming messages for their users. By offering the system as a service, it can be more widely adopted, helping to secure a larger user base and improve the overall effectiveness of spam detection across different platforms.

5.2 Conclusion

The SMS Spam Detection System developed in this project serves as a powerful tool for identifying and classifying SMS messages as spam or ham, leveraging Natural Language Processing (NLP) and machine learning techniques. Through effective preprocessing, feature extraction, and model training, the system demonstrated the ability to accurately classify SMS messages, ensuring reliable detection of spam messages. The primary machine learning models used in this project, such as Naïve Bayes and Support Vector Machines (SVM), have proven effective in providing a good balance between performance and computational efficiency.

The successful application of NLP and machine learning highlights the growing role of artificial intelligence in addressing real-world challenges in digital communication. Spam messages not only waste users' time but also pose significant security risks, including phishing attempts, scams, and potential data breaches. By developing an effective and automated solution, this project contributes to enhancing the security and trustworthiness of messaging platforms.

However, as demonstrated, there are several opportunities for future enhancement. Incorporating deep learning models, enabling real-time spam detection, extending multilingual capabilities, and expanding the dataset are just a few of the avenues that can further strengthen the system's effectiveness. Additionally, deploying the system as a service will increase its accessibility and adoption, making it a practical tool for a broader range of users and service providers.

Ultimately, this project sets the foundation for the development of more robust, intelligent, and scalable spam detection systems. With continuous improvements, the SMS Spam

Detection System could become an integral part of modern communication platforms, offering a reliable solution to users and businesses worldwide by minimizing the risks associated with SMS spam and enhancing overall user experience.

REFERENCES

1. Almeida, T.A., Hidalgo, J.M.G., & Yamakami, A. (2011). Contributions to the study of SMS spam filtering: New collection and results. Proceedings of the 2011 ACM Symposium on Applied Computing, 343-350.
2. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
3. Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys, 34(1), 1-47.
4. Kibriya, A.M., Frank, E., Pfahringer, B., & Holmes, G. (2004). Multinomial Naïve Bayes for text categorization revisited. Proceedings of the 17th Australasian Joint Conference on Artificial Intelligence, 488-499.
5. Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. Proceedings of the European Conference on Machine Learning, 137-142.
6. Islam, R., Tasnim, T., & Khan, A. (2020). A hybrid deep learning model for SMS spam detection. Journal of Information Security and Applications, 54, 102558.
7. Manning, C.D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
8. Chhabra, R., Verma, O.P., & Shukla, K.K. (2018). SMS spam classification using machine learning and content-based feature engineering approach. International Journal of Computers and Applications, 40(1), 37-49.
9. Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. Proceedings of the First International Conference on Machine Learning.