A Transfer Learning Study on Email and SMS Messages for Enhanced Spam Detection using Naive Bayes as a Baseline

Shyamal Gajadha
Department of Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida, USA
sgajadha2016@fau.edu

Mindy Knowles
Department of Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida, USA
mknowles2021@fau.edu

John Renne
Department of Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida, USA
jrenne@fau.edu

***Abstract*— Naïve Bayes classifiers are known to be highly effective for text-based classification problems. One such problem that Naïve Bayes is excellent at solving is to identify spam emails from normal emails. However, as commonly used forms of communication evolve, cell phone usage has become more frequent. Much information that may have formerly been conveyed in an email may now be a quick text or SMS message. These messages, like email, have become subject to potential spam. This necessitates adaptation of spam detection methods to effectively and accurately handle SMS messages. This paper proposes a Naïve Bayes email spam classification method that performs well on email along with transfer learning via use of a small percentage (5%) of text messages to determine if the similarity in the domains can allow us to adapt and improve upon the baseline of pure transfer, that is predicting the spam of texts using only an email trained classifier. In addition, this study incorporates methods of data cleaning that take into account the nature of text messages, which tend to be shorter than emails, contains more typos, abbreviations, and emojis. SMS messages are also more informal and less structured than emails. By acknowledging and working with these differences, this study seeks to implement a model that improves upon the baseline.**

Keywords—SMS messages, text messages, email, spam detection, machine learning, Naive Bayes, transfer learning

I. Introduction

The widespread and increasing use of mobile smartphones has shifted personal communications, especially among younger generations, from talking to communicating via texting, also known as Short Message Service (SMS). Similar to the proliferation of email and unwanted spam over the past 30 years, marketing firms and those with malicious motivations have also taken advantage of SMS as a target for spamming the population.

Spam SMS annoys users and poses potential safety and security risks to individuals [1]. Spammers exploit SMS to disseminate unsolicited messages, resulting in a recent surge in SMS spam [2]. Unlike email, SMS's unique characteristics, such as its brevity, informal nature, and lack of a folder structure, make it a prime target for spamming, necessitating the development of robust spam detection techniques [3].

A. Research Problem

This paper focuses on devising a model to detect spam SMS messages, which differ in structure and content from emails [3]. The challenges posed by SMS spam are multifaceted, stemming from unique limitations. SMS messages are typically short, often containing 160 or fewer characters, restricting the amount of contextual information available for analysis [4]. Moreover, SMS language often includes abbreviations, slang, and emoticons, making it difficult for traditional spam detection methods to effectively identify and filter out spam messages [5].

B. Motivation

The motivation for this research arises from the proliferation in communications, where SMS has become increasingly utilized due to its convenience and immediacy [4]. The increasing proliferation of mobile devices has made SMS a primary mode of communication for personal and professional purposes [6]. However, in recent years, users have experienced an alarming increase in spam SMS, which inconveniences users and exposes them to scams, phishing attempts, and privacy breaches [5]. The consequences of SMS spam extend beyond mere annoyance, as it can lead to financial losses, identity theft, human trafficking, and other dangerous criminal activity [7].

The urgency to address the SMS spam problem is underscored by the growing sophistication of spamming strategies, which are rapidly evolving to bypass traditional detection methods [7]. Spammers employ various tactics to evade detection, such as using URL shorteners, obfuscating words, and leveraging social engineering techniques to manipulate users to click on malicious links or divulge sensitive information [8]. The dynamic nature of SMS spam necessitates the development of adaptive and resilient detection mechanisms that can keep pace with the ever-changing nature of spammers [9].

C. Existing Solutions and Limitations

Current spam detection strategies are predominantly tailored for email and utilize methodologies such as Naïve Bayes classifiers, which have proven effective in identifying spam in emails due to their ability to process large and structured datasets [8]. However, the unique challenges SMS spam poses render these traditional methods less effective [9]. Email spam filters often rely on analyzing header information, sender reputation, and content patterns, which are not readily available or applicable in the context of SMS [10]. Moreover, the informal and condensed nature of SMS language and the limited contextual information make it difficult for conventional spam detection techniques to accurately distinguish between legitimate and spam messages [11].

D. Proposed Method

To address the challenges associated with SMS spam detection, this research proposes a novel approach by adapting the well-established Naïve Bayes classifier to the SMS context and introducing a transfer learning component. The study aims to leverage the strengths of the Naïve Bayes method, renowned for its efficacy in text classification, by training it not only on traditional email data but also incorporating a small dataset (5%) of SMS messages [8]. This hybrid approach is designed to explore the feasibility of cross-domain application between emails and SMS, enabling the classifier to learn and adapt to the unique characteristics of SMS language [8].

Furthermore, the proposed method integrates advanced Natural Language Processing (NLP) techniques, such as tokenization, stop word removal, and vectorization, which are crucial for handling the SMS-specific challenges like message brevity and informal text [8]. Tokenization involves breaking down the SMS messages into individual words or tokens, allowing for more granular analysis of the content [12]. Stop word removal eliminates commonly occurring words that do not contribute significantly to the meaning of the message, thereby reducing noise and improving computational efficiency [13]. Vectorization assigns weights to the remaining tokens based on their frequency and importance, enabling the classifier to identify distinctive features that are indicative of spam messages [14].

To evaluate the effectiveness of the proposed approach, three models have been developed and compared. The first model serves as a control, a Naïve Bayes classifier trained solely on the email dataset using a 10-fold cross-validation. This model achieves an accuracy of approximately 94% when applied to the email dataset but only 44% when applied to the SMS dataset, highlighting the poor performance of a model trained on a different domain. The second model is the same as the first model except it includes 5% sample of the SMS data in addition to the email data.

The third model incorporates NLTK for data pre-processing and is trained on the entire email dataset and 5% of the SMS dataset. It employs 10 repetitions of a 10-fold cross-validation, exceeding the initial instructions. This model achieves an accuracy of around 86% on the remaining 95% of the SMS dataset, demonstrating the benefits of incorporating a small portion of the target domain data and applying NLP techniques.

The fourth model utilizes SpaCy for data pre-processing and follows the same training approach as the second model, with 10 repetitions of a 10-fold cross-validation. This model achieves an impressive accuracy of 90-94% on the remaining 95% of the SMS dataset, further highlighting the importance of advanced NLP techniques in improving the performance of the classifier.

The incorporation of these NLP techniques is expected to enhance the model's ability to discern and classify spam messages accurately by focusing on linguistic patterns that are typical of spam [8]. By learning from both email and SMS datasets, the classifier can identify common spam indicators, such as the presence of specific keywords, phrases, or patterns, while also adapting to the unique characteristics of SMS language [15]. The transfer learning component allows the model to leverage the knowledge gained from email spam detection and apply it to the SMS domain, thereby improving its performance and generalization capabilities [16].

The proposed approach holds promise for addressing the pressing issue of SMS spam detection, as it combines the strengths of established techniques with novel adaptations tailored to the specific challenges of the SMS medium. By developing a robust and effective SMS spam detection system, this research aims to contribute to the ongoing efforts to safeguard users from the growing menace of mobile spam and enhance the overall trustworthiness and reliability of SMS communication.

II. Related Work

Naïve Bayes classifiers have been used to identify spam detection and utilize transfer learning based on training and test datasets to monitor performance. Transfer learning has become a crucial technique in machine learning, particularly effective when adapting models from email to SMS. In Naïve Bayes classifiers, which are fundamentally probabilistic, transfer learning can involve modifying the model’s priors or likelihoods based on insights from a source domain like email spam detection to improve performance in a target domain like SMS spam detection.

As an example, Ning, Junwei and Feng [17], classify spam using a Naïve Bayes model, which is presented by the following equations from their paper:

For each classification C_k in the classification set C, the posterior probability of test text d_e against C_k can be calculated according to the formula [#4] as follows [17]:

$$P(d_e \mid c_k) = \sum_{w_i \in W} p(w_i \mid c_k)|$$

For each classification C_k in the classification set C, using the calculation result of formula (4), the posterior probability of test text d_e can be calculated by the Bayes formula [#6] [17]:

$$P(c_k \mid d_e) = \frac{P(d_e \mid c_k)P(c_k)}{P(d_e)}$$

The transfer of methodologies from email spam detection to SMS spam detection involves leveraging common spam indicators while adapting to the challenges of SMS. Techniques often include adapting feature extraction processes to accommodate SMS-specific elements such as abbreviations and slang. A review of adaptation techniques reveals how models trained on emails can be fine-tuned with minimal SMS data to improve detection accuracy [18].

Transfer learning for text classification frequently uses model-based or feature-based approaches. Model-based strategies may involve fine-tuning a pre-trained model on a new task, which has shown success in text classification tasks [19]. Feature-based transfer involves creating a shared feature space that can effectively represent both source and target domains, utilizing techniques such as embedding layers trained on large quantities [20].

Detecting spam in SMS poses unique challenges due to the constantly evolving nature of spam techniques and the informal, abbreviated text used in SMS. Traditional spam detection methods often fail to keep pace with these changes, suggesting a need for adaptive models that learn continually [21]. Moreover, the integration of unsupervised learning to handle unlabeled data effectively can significantly enhance the detection systems [22].

A foundational study tackles the issue of performance deterioration when training and test datasets come from different distributions, a common challenge in machine learning applications [23]. The authors propose the Naive Bayes Transfer Classifier (NBTC), which integrates a transfer learning approach with Naive Bayes classifiers, enhanced by the Expectation-Maximization (EM) algorithm. This method begins by estimating an initial model using labeled data from a source distribution and subsequently adapts this model to a target distribution using unlabeled data. The adaptation process utilizes the EM algorithm to adjust the model's parameters to fit the new distribution better, leveraging the Kullback-Leibler (KL) divergence to measure and adjust to the differences between the source and target distributions [23].

The effectiveness of the NBTC is demonstrated through experiments showing significant performance improvements over traditional supervised and semi-supervised methods, especially when there are considerable differences between the training and testing data distributions. This success is attributed to the NBTC's ability to dynamically adjust its learning process based on the distribution differences, using KL divergence not just as a metric but as an active component in parameter tuning. This approach allows the NBTC to effectively learn from the unlabeled data in the target domain while initially being informed by the labeled data from a related but distinct source domain, providing a robust solution to the problem of distribution shifts in practical machine learning scenarios [23].

A recent study focuses on the informal and short nature of SMS messages, which often contain typos, abbreviations, and slang, making traditional Naïve Bayes classification techniques challenging. The AstNB method leverages a transfer learning framework where SMS-specific training data are augmented by random sampling and combining data from related sources, such as emails, to enhance the training set. This augmented dataset then trains multiple base classifiers, whose predictions are subsequently stacked to form a new feature space that a final classifier uses to predict spam in SMS messages [24].

This approach effectively transfers knowledge from the source domain (emails) to the target domain (SMS), which is especially valuable when labeled messages in the target domain are scarce. The paper compares the AstNB method against traditional classifiers and demonstrates its superior performance in terms of accuracy and adaptability. By augmenting SMS data with email content, the model benefits from a richer linguistic context, thus better capturing the nuances of spam indicators in SMS. Furthermore, the model's stacking component helps refine the predictions by integrating insights from multiple base models, thereby improving the overall accuracy of spam detection. This research contributes significantly to the field by providing a robust methodology for enhancing spam detection in SMS through innovative applications of transfer learning, model stacking, and data augmentation [24].

In summary, the application of transfer learning in Naïve Bayes classifiers for SMS spam detection has shown promising results in addressing the unique challenges posed by the informal and evolving nature of SMS communication. Recent studies have further advanced these techniques, such as a study that uses a hybrid approach combining Naïve Bayes with Support Vector Machines and K-Nearest Neighbors, utilizing transfer learning to adapt the model to new datasets [25]. Another introduces a semi-supervised approach that leverages unlabeled data to enhance the performance of the Naïve Bayes classifier in SMS spam detection [26]. Finally, another explores the potential of deep learning-based transfer learning, using pre-trained word embeddings to improve the accuracy of SMS spam classifiers [27]. These recent advancements demonstrate the ongoing efforts to refine and optimize transfer learning techniques for Naïve Bayes classifiers in the context of SMS spam detection.

III. Main Body

A. *Motivation of Design*

The proposed approach to transfer learning for Naïve Bayes classifiers is motivated by its simplicity, ease of interpretation, and effectiveness in handling text data. Naïve Bayes classifiers are simple and computationally efficient, which is the reason they are popular for text classification tasks, especially where resources are scarce. Transferring knowledge from a Naïve Bayes classifier trained on a source dataset to a target dataset will use the inherent language patterns and features captured by the classifier to enhance its performance on the target dataset.

The four different models explored in this study include 1. a simple Naïve Bayes trained on the email dataset (see Figure 1), 2. a simple Naïve Bayes trained with an additional 5% of the SMS data (see Figure 2), 3. NLTK (see Figure 3), and 4. spaCy models (see Figure 4). Each model exhibits different characteristics and methodologies in their approach to text classification tasks. The simple Naïve Bayes, which serves as this study's control, utilizes a traditional approach, using the Multinomial Naïve Bayes classifier trained on a dataset of email documents. It relies on basic preprocessing techniques and a standard bag-of-words representation for feature extraction. The simple Naïve Bayes trained with an additional 5% of the SMS data serves as a reference for how a simple Naïve Bayes would perform given some SMS training data. The NLTK model enhances the preprocessing phase by leveraging the Natural Language Toolkit (NLTK) library, allowing advanced functionalities like tokenization, stop word removal and lemmatization. This model aims to improve text representations' quality and classification performance. Lastly, the spaCy model introduces a different preprocessing pipeline, utilizing the spaCy library, known for its efficient and linguistically informed text processing capabilities. By using spaCy's tokenization, stop word removal and lemmatization functionalities, the spaCy model can refine text representations and enhance the classifier's ability to capture subtle nuances in the text data.

The Natural Language Toolkit offers a competitive solution for spam text classification because of its broad range of tools for natural language processing. NLTK excels in cleaning and normalizing text data, enhancing the quality of features extracted for analyses. Moreover, NLTK offers access to wide-ranging corpora, lexical resources, and linguistic data, enabling accurate modeling and analysis across diverse domains. NLTK's modular design and thorough documentation further contribute to its appeal for this project. These attributes allow the development of customized preprocessing pipelines and feature extraction methods for a wide range of project requirements. Additionally, the active and engaged community surrounding NLTK ensures community support, updates, and contributions, solidifying its position as a leading choice for text classification tasks.

On the other hand, spaCy emerges as a modern and efficient alternative for text classification, highly considered for its emphasis on speed, scalability, and ease of use. Designed with performance in mind, spaCy boasts optimized algorithms and data structures that enable quick and scalable text processing, even with larger datasets. Leveraging linguistic knowledge within its processing library, spaCy offers features such as named entity recognition, dependency parsing, and sentence segmentation, which elevates the quality of text representations and enhances the overall classification accuracy. SpaCy offers a user-friendly API and vast library of documentation which helps simplify integration for new or existing projects, which further streamlines the development and deployment of text classification models. Additionally, spaCy provides pre-trained models for various natural language processing (NLP) tasks, offering developers a jump-start in their projects which further strengthens its appeal for use as a text classification method.

These approaches hold significant potential to address the challenges of data scarcity and dataset adaptation in text classification. By reusing knowledge learned from one dataset to improve performance on another dataset, we can achieve better overall classification of normal and spam datasets, thus enhancing the scalability and application of Naïve Bayes classifiers across diverse domains.

B. Algorithm Details

The details of the proposed algorithm involve a multiple-step process encompassing data preprocessing, model training, and evaluation:

1) Data Preprocessing:

- Utilize advanced text preprocessing techniques like tokenization, stop word removal, and lemmatization provided by NLTK or spaCy.
- Transform raw text data into a structured format suitable for training a Naïve Bayes classifier.

2) Model Training:

- Train a Naïve Bayes classifier on a source dataset, in this case, the email dataset, using the preprocessed features.
- Utilize the trained classifier's parameters and features as the basis for transfer learning.
- The simple Naïve Bayes classifier is trained only using the email dataset.
- The Naïve Bayes classifier for the NLTK and spaCy model is trained on the entire email dataset and limited to 5% training on the SMS.csv dataset.

3) Evaluation:

- Evaluate the transferred Naïve Bayes classifier's performance on a target dataset, such as an SMS dataset, using reliable evaluation methods such as 10 iterations of 10-fold cross-validation.
- Assess key performance to gauge the accuracy and the effectiveness of the transfer learning approach.

FIGURE 1: MODEL FRAMEWORK FOR A SIMPLE NAÏVE BAYES CLASSIFIER

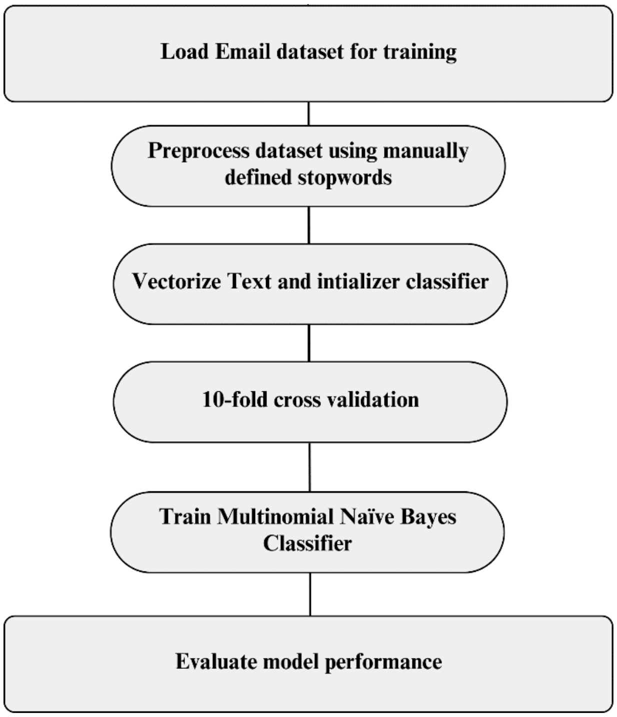

Load Email dataset for training
Preprocess dataset using manually defined stopwords
Vectorize Text and intializer classifier
10-fold cross validation
Train Multinomial Naïve Bayes Classifier
Evaluate model performance

FIGURE 2: MODEL FRAMEWORK FOR A SIMPLE NAÏVE BAYES CLASSIFIER WITH 5 PERCENT SMS SAMPLE

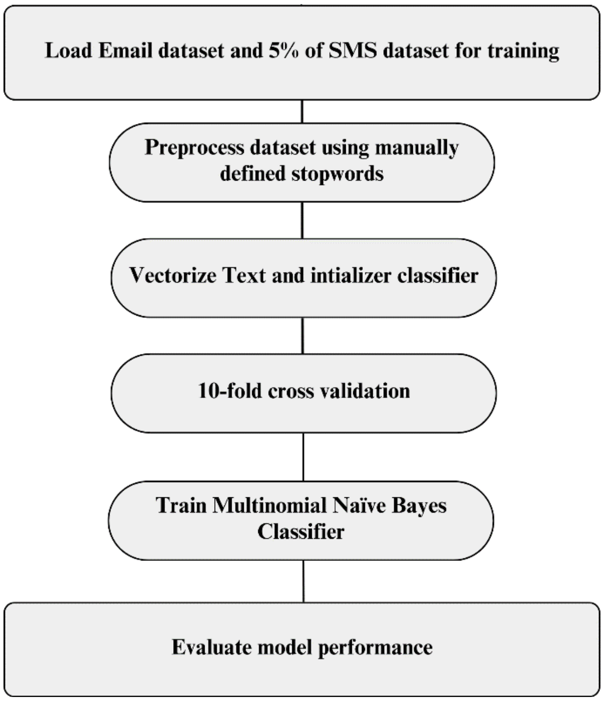

Load Email dataset and 5% of SMS dataset for training
Preprocess dataset using manually defined stopwords
Vectorize Text and intializer classifier
10-fold cross validation
Train Multinomial Naïve Bayes Classifier
Evaluate model performance

FIGURE 3: MODEL FRAMEWORK FOR NLTK

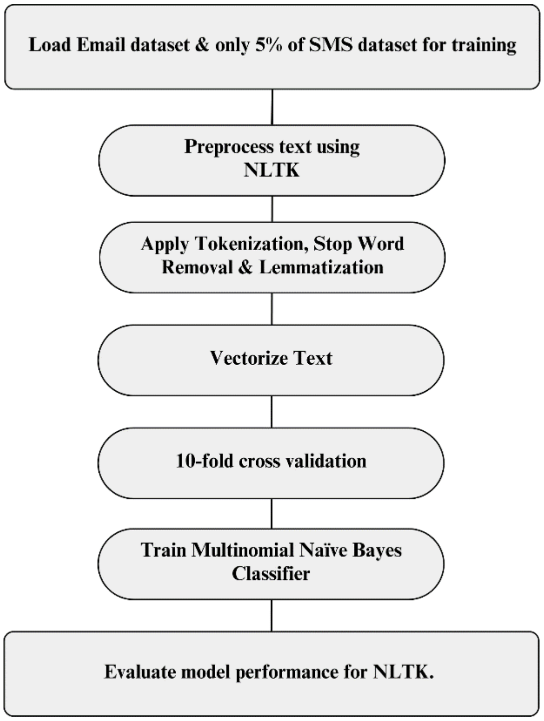
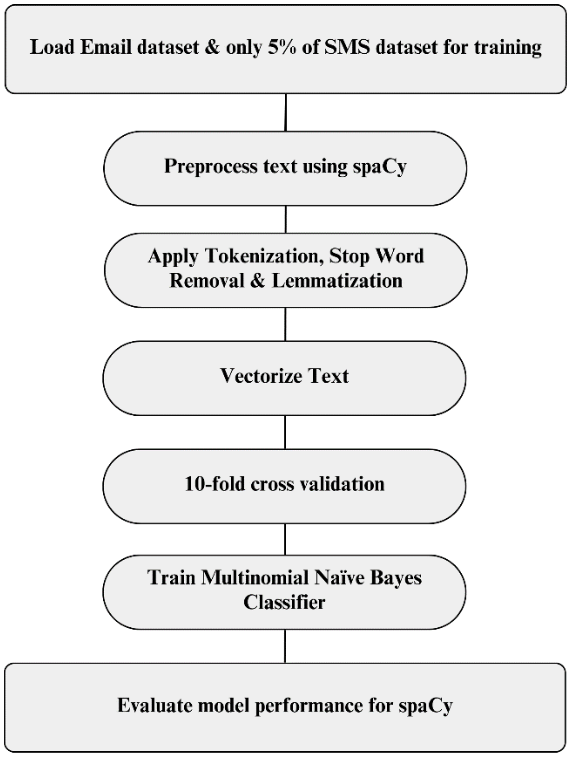

Load Email dataset & only 5% of SMS dataset for training
Preprocess text using NLTK
Apply Tokenization, Stop Word Removal & Lemmatization
Vectorize Text
10-fold cross validation
Train Multinomial Naïve Bayes Classifier
Evaluate model performance for NLTK.

FIGURE 4: MODEL FRAMEWORK FOR SPACY


Load Email dataset & only 5% of SMS dataset for training
Preprocess text using spaCy
Apply Tokenization, Stop Word Removal & Lemmatization
Vectorize Text
10-fold cross validation
Train Multinomial Naïve Bayes Classifier
Evaluate model performance for spaCy

An essential step in the preprocessing stage is the stop word removal where common, every day, and uninformative words such as "the", "a", or "is" are filtered out of the text. Through this process of filtering out and omitting these words during our preprocessing step, the focus of the model is shifted towards more relevant content that increases the overall quality of the text representation for better training. Besides this, lemmatization is also used that reduces words to their base form or lemma. This action helps to standardize all of the various forms of words that, in turn, improves the consistency of the text representation across documents.

The text is then transformed into a numerical feature vector, which uses a technique called bag-of-words. The process requires several steps. First, a vocabulary is created containing all the unique words across the entire collection of documents. A document's frequency count of words from the vocabulary is then computed. These count vectors will be the numeric representation of the documents. Each document will be a fixed-length vector.

In a bag-of-words representation, the order of words is ignored, and the emphasis is on word frequency within the documents. This is a very simple approach, making it an excellent choice for numerically representing text data. It is, however, limited to issues where word relationships and context are ignored and where the model is very sensitive to stop words and common words that have little meaning.

With the vectorized features now in hand, a Multinomial Naïve Bayes classifier is trained on that information to learn the underlying patterns and relationships within the data. This classifier will use vectorized features to make predictions on class labels for unseen data instances. Once trained, the classifier's effectiveness in spam classification is evaluated on the remaining 95% of the SMS dataset. This thorough evaluation will be a strong indication of the model's effectiveness in accurately classifying spam messages—a useful indicator of its value in real-world applications and scalability.

In addition to the high-level steps outlined above, each preprocessing method deserves further exploration and delving into the intricacies of the method and its implication for the classification task. For example, Tokenization is much more than breaking text into words or sentences; the methodology is very sensitive to language-specific nuances, including special characters and punctuations. Ambiguities in tokenization can, hence, lead to a problem in downstream processes and, in turn, affect the overall quality of the classification model.

Stop word removal is another essential preprocessing step that needs deeper exploration. On the one hand, removing commonplace, uninformative words sounds very simple, but the cutoff line between which words to exclude depends on the context and domain of the dataset. What may be considered a stop word in one context may carry significant meaning in another. Therefore, the choice of stop words needs careful consideration and domain expertise to retain relevant information without noise effectively being filtered out.

Lemmatization is a method that plays a very critical role in reducing words to their base forms. However, the process is not without its difficulties. The algorithms for lemmatization need to be able to consider the sometimes irregular forms of words, morphological variations, and linguistic complexities to derive the base form of a word correctly. Another difficulty is the computational cost: which can be an enormous cost when dealing with large volumes of text data. Thus, optimizations and trade-offs must be carefully considered to balance accuracy and efficiency.

The problem of the vectorization process, especially in the bag-of-words representation, needs to be further studied in its consequences for text classification. Although bag-of-words provides simplicity and speed, it inherently neglects important information concerning the relations between words and the semantic context. That might lead to problems when the model is trying to capture the underlying semantics of text data, especially when the task is one where context is of utter importance, like sentiment analysis or document summarization. Investigating alternative ways of vectorization, like word embeddings or n-gram models, might provide a more proper representation of text data, potentially leading to better results in the classification model.

In general, understanding the preprocessing methods and vectorization techniques used in text classification is important for building robust and reliable classification models. Researchers and practitioners should explore the subtleties of each step to make informed decisions and optimizations toward better results in the model's performance.

C. *Differentiation from Existing Work*

This design distinguishes itself from existing work in several key aspects. Firstly, while transfer learning has been extensively explored within neural networks and deep learning models, this approach focuses on applying transfer learning to Naïve Bayes classifiers. Additionally, this design integrates either NLTK or spaCy, leveraging the robust text preprocessing capabilities of these libraries. This integration enhances the quality of feature extraction and prepares the data for efficient training. Moreover, this study places emphasis on scalability and applicability. Through comprehensive evaluations across diverse datasets and domains, this paper aims to demonstrate the scalability of this approach and its potential for real-world deployment. By combining the simplicity of Naïve Bayes classifiers with the advanced preprocessing capabilities of NLTK or spaCy, this design offers a practical and effective solution for transfer learning in text classification.

IV. Experiments

A. *Experimental Settings*

The primary objective of these experimental studies is to determine if a Naïve Bayes classifier trained to detect spam on email data can be effective in a similar domain of SMS text message spam identification by incorporating 5% of text message data into the training model. Using existing trained models as a base to train the model for a new purpose is a type of transfer learning.

The primary tools used are the Python libraries scikit-learn, NLTK, and spaCy. The Naïve Bayes classifier used for all the experiments is the MultinomialNB provided by scikit-learn. It is commonly used in text classification and will be fed with the features represented as word vectors. In addition, the experimental models will compare two commonly used natural language processing libraries: NLTK and spaCy.

First, a baseline Naïve Bayes classifier trained only on email data (BaselineNB) will be used to test its ability to classify SMS text messages accurately. The second method will include 5% of the SMS data in the initial model (NB5%). The third method will include 5% of the SMS data into the training model, use NLTK libraries and processing methods, and 10-fold cross-validation (NLTK). The fourth method will include 5% of the SMS data into the training model, use spaCy libraries and processing methods, and 10-fold cross-validation (spaCy).

B. *Benchmark Data*

The data includes two separate datasets: emails and SMS texts (see Table 1). The email dataset is the base domain to be trained with, and the domain to transfer learning to is SMS texts. Emails are a balanced set of data, comprised of 250 email messages, of which 125 (50%) are normal and 125 (50%) are spam. The length of an email instance ranges from 48 to 6051 characters, with an average length of 783 characters. The SMS text data includes 5572 sample texts, of which 4824 (86.5%) are normal, and 747 (13.5%) are spam. The SMS text data is unbalanced with regard to class outcomes. The length of an SMS text instance ranges from 5 to 913 characters, with an average length of 84 characters. The datasets differ greatly across all measures. The minimum length of an instance for emails is 48 characters, while the minimum length of an instance for SMS text is only 5 characters. The maximum length of an instance for emails is 6051, while the maximum length of an SMS text is only 913 characters. The average length of an email is 783 characters, and the average length of an SMS text is much shorter at only 84 characters. These differences are to be expected, due to the nature and use cases of emails and SMS text. Emails tend to be longer, more formal, used in more professional settings and where a response can take longer. SMS text is generally shorter, more informal and used casually and with the expectation of quick responses.

TABLE 1: COMPARISON OF DATASETS

						Length in Characters		
Data	Total	Normal	%Normal	Spam	%Spam	Min	Max	Avg
Emails	250	125	50%	125	50%	48	6051	783
SMS text	5572	4824	86.5%	747	13.5%	5	913	84

C. *Baseline Methods*

The baseline method used was two-fold. First, the results of the simplest possible transfer method were demonstrated. This method trained the scikit-learn Multinomial Naïve Bayes classifier MultinomialNB() only on the email dataset with only a simple stopwords list and vectorization. The SMS test data was used for testing. This gave a baseline for the inefficacy of unassisted transfer learning from the two domains, despite their similarities in both being text classified as spam or normal.

The second baseline was to use a small amount of the SMS text data (5%) to help train the scikit-learn Multinomial Naïve Bayes email model. Incorporating a small amount of text domain data into the training model mimics the transfer learning processes seen in more complex methods. This model also only used the simple stopwords list and vectorization. This allowed a comparison between the first and second baseline methods.

The second baseline model proved to be better, as expected, due to the incorporation of a small amount of SMS data. This model was then further adjusted to compare two commonly used natural language processing libraries, NLTK, and spaCy. The training data in both cases are the full dataset from email and 5% of the data from SMS texts. The variable here is the NLP library that was incorporated into the data pre-processing.

The NLTK model used NLTK library functions for stopwords, tokenization, and lemmatization. The length of tokens considered were all tokens greater than 2 characters in length. Tokens were converted to lowercase and punctuation was stripped or removed.

The spaCy model used the spacy.lang.en.STOP_WORDS for stop words. Similarly, tokenization and lemmatization were used. Tokens were converted to lowercase and punctuation was stripped or removed.

D. *Results*

Results were reviewed using the metrics of overall accuracy and precision. Overall accuracy gives a quick way to see if the model is performing well in general. Precision is a measure of the proportion of SMS text that are correctly identified as their corresponding class (see Figure 5).

1) *BaselineNB vs NB5%:*

The results from the BaselineNB model and the NB5% model shows that adding 5% of the SMS text data to the second model improved the overall accuracy by over 30% when testing on the SMS text data. However, the precision for detecting spam remained very low, only increasing from 16 to 19 percent.

2) *NB5% vs NLTK:*

The results for the NLTK model show that adding the natural language processing techniques included with NLTK dramatically improved overall accuracy and precision. Accuracy improved 18 percent, from 78% to 96%. More importantly, the precision on spam detection rose from under 20% to 95%.

3) *NB5% vs spaCy:*

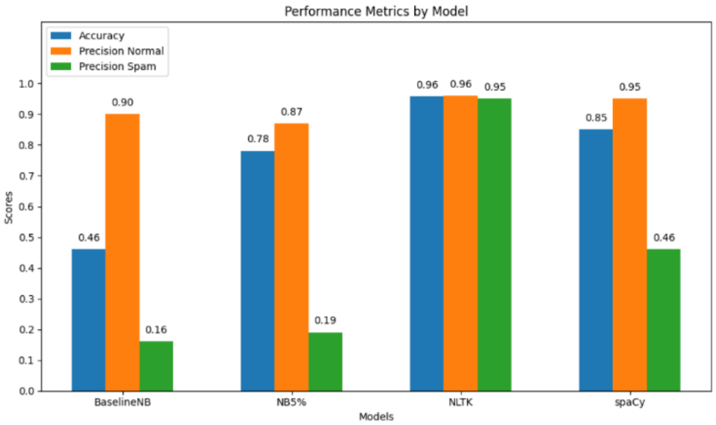
The results for the spaCy model show that adding the natural language processing techniques included with spaCy improves the overall accuracy, as well as the spam precision. Accuracy improved by 7% over the NB5%, from 78% to 85%. Spam precision improved by 27%, however this still only gives a 46% precision on spam detection.

4) *NLTK vs spaCy:*

The results comparison between NLTK and spaCy for the task of spam detection on SMS text data shows that NLTK outperformed spaCy on both overall accuracy and spam precision. Overall accuracy of NLTK was over 10% higher, and the performance on spam precision was nearly 50% higher, with spaCy only 46% precision on spam and NLTK 95%.

Precision on normal messages remained relatively high for all models, likely due to the fact that 86.5% of the messages were normal; however, spaCy and NLTK both performed with 95% and 96% precision on normal messages, respectively.

FIGURE 5: PERFORMANCE METRICS BY MODEL



E. *Analysis*

Due to the class imbalance in the SMS text data, accuracy should not be the sole measure on which a model's performance is based. Precision on the spam class must be carefully considered. Imbalanced classes can cause accuracy to be misleading in the performance of spam detection. A model with high overall accuracy could have a low spam precision and indicate a model that is poor at identifying spam.

The low accuracy and precision of spam on the BaselineNB model are not surprising due to the methodology used. The model was trained only on email data and applied directly to the SMS text data. Introducing just 5% of the SMS text data to the next version of the model, NB5%, showed an overall increase in accuracy. However, this model did not really improve upon the precision of spam measure, indicating that it is also not a good model.

The NLTK model was the best model across all measures: accuracy, precision on spam class and precision on normal class. This suggests that the NLTK libraries used for data preparation, such as stopwords, tokenization, and lemmatization were very effective at reducing the noise in the SMS text data. This is particularly impressive considering the model was only allowed to use 5% on the SMS text data to train on.

The spaCy model was middle-of-the-road. Its performance was disappointing on spam precision, relegating it to be unsuitable for this task under these conditions. The parallel data preparation, including stopwords, tokenization, and lemmatization, was used as in the NLTK model, but the results fell short.

F. *Implications*

The implications of this research are that transfer learning can be successfully applied using Naïve Bayes based models to cross over from the email to SMS text domains with the injection of a small amount of the target domain data and good pre-processing techniques.

G. *Suggestions for Future Research*

Future research could include comparing additional models, such as ones based on neural networks and deep learning techniques. Additionally, comparisons among different datasets used for training and testing could test the generalizability of the methodology.

V. CONCLUSIONS

This research studied the problem of detecting spam in SMS text messages by adapting a Naive Bayes classifier trained on email data. SMS spam presents unique challenges compared to email spam due to the short, informal nature of text messages which often contain abbreviations, slang, and typos. Directly applying an email-trained spam filter to SMS results in poor performance, with only around 44% accuracy.

To address this, transfer learning was employed by incorporating a small amount (5%) of SMS data into the training of the email-based classifier. This simple form of transfer learning improved accuracy to around 78% on the SMS test set. The classifiers were further enhanced with more sophisticated NLP techniques from the NLTK and spaCy libraries for tokenization, lemmatization, and removing stop words. The NLTK-based classifier achieved the best performance, with around 96% accuracy and 95% precision on identifying SMS spam. The spaCy-based model attained 85% accuracy and 46% spam precision.

In conclusion, this study demonstrates that a Naive Bayes classifier trained primarily on email data can be effectively adapted to the SMS domain using transfer learning with a small amount of SMS data. Advanced NLP techniques, particularly from the NLTK library, can substantially boost performance on this challenging task. The methods explored here offer promising solutions for combating the growing problem of SMS spam.

References

[1] S. J. Delany, M. Buckley, and D. Greene, "SMS spam filtering: Methods and data," Expert Systems with Applications, vol. 39, no. 10, pp. 9899-9908, 2012. DOI: https://doi.org/10.1016/j.eswa.2012.02.053

[2] A. Chandra and S. K. Khatri, "Spam SMS Filtering using Recurrent Neural Network and Long Short Term Memory," in 2019 4th International Conference on Information Systems and Computer Networks (ISCON), 2019, pp. 118-122. DOI: https://doi.org/DOI:10.1109/ISCON47742.2019.9036269

[3] P. Navaney, G. Dubey, and A. Rana, "SMS Spam Filtering using Supervised Machine Learning Algorithms," in 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2018, pp. 43-48. DOI: https://doi.org/10.1109/CONFLUENCE.2018.8442564

[4] B. Ning, W. Junwei, and H. Feng, "Spam Message Classification Based on the Naïve Bayes Classification Algorithm," IAENG International Journal of Computer Science, vol. 46, no. 1, pp. 11-19, 2019.

[5] C. Ulus, Z. Wang, S. M. A. Iqbal, K. M. S. Khan, and X. Zhu, "Transfer Naïve Bayes Learning using Augmentation and Stacking for SMS Spam Detection," in 2022 IEEE International Conference on Knowledge Graph (ICKG), 2022, pp. 275-282. DOI: https://doi.org/10.1109/ICKG55886.2022.00042

[6] G. Cormack, J. Hidalgo, and E. Sánz, "Feature engineering for mobile (SMS) spam filtering," in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007, pp. 871-872. DOI: https://doi.org/10.1145/1277741.1277951

[7] T. A. Almeida, J. M. G. Hidalgo and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in Proceedings of the 11th ACM symposium on Document engineering, 2011, pp. 259-262. DOI: https://doi.org/10.1145/2034691.2034742

[8] S. Ruan, H. Li, C. Li, and K. Song, "Class-specific deep feature weighting for Naïve Bayes text classifiers," IEEE Access, vol. 8, pp. 20151-20159, 2020. DOI: https://doi.org/10.1109/ACCESS.2020.2968984

[9] A. Alzahrani and D. B. Rawat, "Comparative study of machine learning algorithms for SMS spam detection," in SoutheastCon 2019, 2019, pp. 1-6. DOI: https://doi.org/10.1109/SoutheastCon42311.2019.9020530

[10] Q. Xu, E. Xiang, Q. Yang, J. Du, and J. Zhong, "SMS Spam Detection Using Noncontent Features," IEEE Intelligent Systems, vol. 27, no. 6, pp. 44-51, 2012. DOI: https://doi.org/10.1109/MIS.2012.3

[11] K. Yadav, P. Kumaraguru, A. Goyal, A. Gupta, and V. Naik, "SMSAssassin: Crowdsourcing driven mobile-based system for SMS spam filtering," in Proceedings of the 12th Workshop on Mobile Computing Systems and Applications (HotMobile '11), 2011, pp. 1-6. DOI: https://doi.org/10.1145/2184489.2184491

[12] O. Akande, O. Gbenle, O. Abikoye, R. Jimoh, H. Akande, A. Balogun and A. Fatokun. (2023). "SMSPROTECT: An automatic smishing detection mobile application." ICT Express, Vol. 9, no. 2, pp. 168-176, 2023. DOI: https://doi.org/10.1016/j.icte.2022.05.009

[13] M. Ikonomakis, S. Kotsiantis, and V. Tampakas, "Text Classification Using Machine Learning Techniques," WSEAS Transactions on Computers, vol. 4, no. 8, pp. 966-974, 2005.

[14] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," in Proceedings of the first instructional conference on machine learning, vol. 242, 2003, pp. 133-142.

[15] M. Popovac, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla, "Convolutional Neural Network Based SMS Spam Detection," in 2018 26th Telecommunications Forum (TELFOR), 2018, pp. 1-4. DOI: https://doi.org/10.1109/TELFOR.2018.8611916

[16] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, 2010. DOI: https://doi.org/10.1109/TKDE.2009.191M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[17] B. Ning, W. Junwei, and H. Feng, "Spam message classification based on the Naïve Bayes Classifcaiton Algorithm," IAENG International Journal of Computer Science, vol. 46, no. 1, 2019, pp. 46-53.

[18] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the Study of SMS Spam Filtering: New Collection and Results," in Proceedings of the 11th ACM Symposium on Document Engineering (DocEng '11), ACM, New York, NY, USA, 2011, pp. 259–262.

[19] L. Yao, C. Mao, and Y. Luo, "Graph Convolutional Networks for Text Classification," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 1, 2019, pp. 7370-7377.

[20] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, et al., "A Comprehensive Survey on Transfer Learning," Proceedings of the IEEE, vol. 109, no. 1, 2020, pp. 43-76.

[21] S. M. Abdulhamid, M. S. A. Latiff, H. Chiroma, O. Osho, G. Abdul-Salaam, A. I. Abubakar, and T. Herawan, "A Review on Mobile SMS Spam Filtering Techniques," IEEE Access, vol. 5, 2017, pp. 15650-15666.

[22] G. V. Cormack, J. M. G. Hidalgo, and E. P. Sánz, "Spam Filtering for Short Messages," in Proceedings of the 16th ACM Conference on Information and Knowledge Management, 2007, pp. 313-320.

[23] W. Dai, X. Gui-Rong, Y. Qiang, and Y. Yong, "Transferring naive bayes classifiers for text classification" AAAI, vol. 7, 2007, pp. 540-545.

[24] C. Ulus, Z. Wang, S. Iqbal, K. Khan, and X. Zhu, "Transfer Naïve Bayes Learning using Augmentation and Stacking for SMS Spam Detection." 2022 IEEE International Conference on Knowledge Graph (ICKG), 2022, pp. 275-282.

[25] D. C. Asogwa, S. O. Anigbogu, I. E. Onyenwe, and F. A. Sani, "Text classification using hybrid machine learning algorithms on big data," arXiv preprint arXiv:2103.16624, 2021

[26] I. Ahmed, D. Guan, and T. Chung, "A novel semi-supervised learning for SMS classification," in 2014 International Conference on Machine Learning and Cybernetics, vol. 2, IEEE, 2014, pp. 856-861.

[27] W. H. Gomaa, "The impact of deep learning techniques on SMS spam filtering," International Journal of Advanced Computer Science and Applications, vol. 11, no. 1, 2020.