Semantic Analysis, Continuous Bag of Words and Random Forest for SMS Spam Detection

[redacted]

1 Problem statement and motivation

In the past decade, cellphone usage has skyrocketed. As of 2023, an estimated 6.8 billion people worldwide now use smartphones (Oberlo, 2023). Short Messaging Service (SMS) is the primary form of text messaging used by these devices. Because of the rise of cellphone usage, there has also been a significant rise in SMS spam. Spam is commonly associated with unwanted marketing or adult content messages sent in bulk. However, spam can also be used maliciously to steal personal information such as bank details or login credentials via phishing tactics. As such, it is important to understand the dangers of SMS spam and take necessary precautions to safeguard personal information. By detecting spam, we can enhance the user experience of mobile phone usage while also improving its security. For instance, detecting and filtering spam can help prevent users from receiving unsolicited and potentially harmful messages, which can lead to a more enjoyable and safer mobile phone experience. Additionally, by detecting spam, users can avoid falling victim to phishing attacks and other scams that aim to steal their personal information. This can provide a sense of security and peace of mind for users.

2 Research hypothesis

Can SMS spam messages be accurately classified using feature extraction with Continuous Bag of Wording(CBOW)? SMS spam messages are often written in a distinct style with certain words appearing in a large percentage of spam messages and many sentences having a similar grammatical structure. For instance in the UCI SMS Spam Collection Data Set (Almeida et al., 2011) used for this research we can see Figure 1 illustrates that apart from "ur", the 10 most frequent words in spam messages appear far less frequently in the non-spam messages or "ham messages" although the number of spam messages is only 13% of the dataset. The structure of a spam message often follows a similar pattern as well. Sousa et al. (2021) who worked with the same dataset note that if a message contains the word "text" the message can be classified as spam 53% of the time. Furthermore, if a message is in the format "Text BLANK to BLANK" the message can be classified as spam 96% of the time. This paper proposes that feature extraction using CBOW is an effective strategy for classifying SMS spam messages, particularly due to the importance of context in SMS spam messages. By evaluating how different preprocessing combinations such as Named Entity Recognition, POS tagging and dependency tagging perform with CBOW I was able to fine-tune my Random Forest model and deliver competitive results.

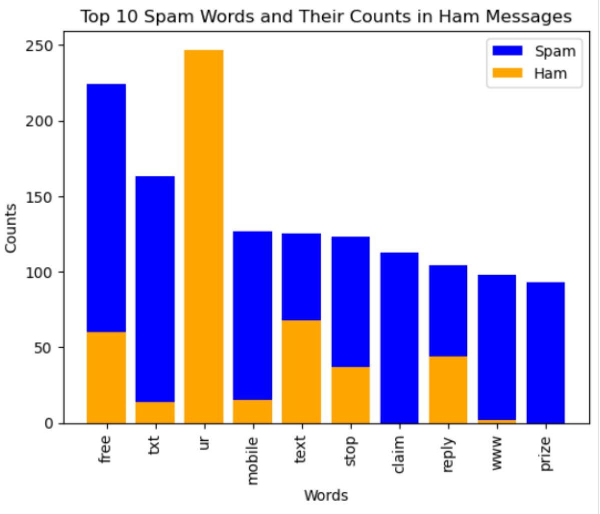


Figure 1: Word frequency in spam messages

3 Related work and background

Because the problem of SMS spam affects nearly everyone there has been an array of different methods used to successfully classify spam. (Almeida et al., 2011) give insight into the dataset used and demonstrate that SVM should be used for baseline models given its past success in spam classification. Abayomi-Alli et al. (2019) give a review of the soft techniques most commonly used and how successful they are. In Figure 2 they show which classifiers are most commonly used for spam detection.

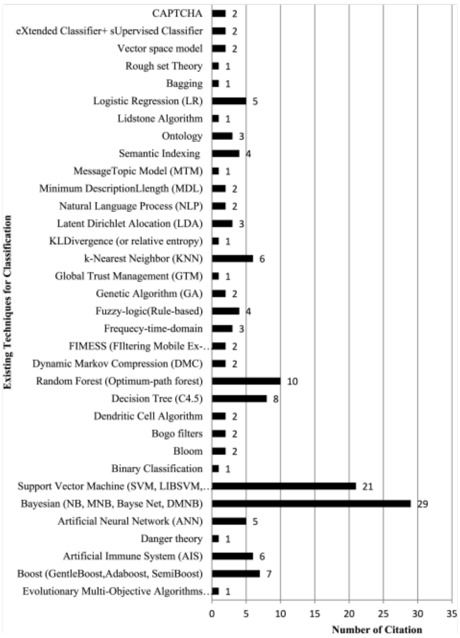


Figure 2: Frequency of citations for different classifiers.

One machine learning-based approach to SMS spam classification is presented by Shirani and Alizadeh (Shirani-Mehr), who use Naive Bayes and SVM classifiers. Similarly, Khan and Ahmed (Delvia Arifin et al., 2016) propose using FP-Growth and Naive Bayes classifiers for SMS spam detection. Kuruvilla and Issac (Mathew and Issac, 2011) introduce novel features such as message length, sender's reputation, and keywords. In addition, Kour and Bhatia (Fernandes et al., 2015) utilise Optimum-Path Forests for classification and include additional features. A deep learning-based approach is proposed by Almeida et al. (Jain et al., 2018), who use a semantic LSTM model for SMS spam detection. Meanwhile, a statistical approach is proposed by Amir Sjarif et al. (Amir Sjarif et al., 2019), who use TF-IDF for vectorization and achieve their best results using Random Forest.

Research that inspired the approach taken in this paper was Sousa et al. (2021) who proposes a machine learning-based approach for SMS spam detection that uses skip-gram embeddings and shallow neural networks. The authors provide valuable insight into the common format used in spam messages and present a strong case for using contextual embeddings for spam detection. Amir Sjarif et al. (2019) use TF-IDF(Term Frequency-Inverse Document Frequency) for vectorization and when comparing classifiers achieve their best results using Random Forest which is the classifier used and optimized in this research.

3.1 Accomplishments

In the proposal for this paper, I proposed using the following 4 steps:

- **Problem Review:** This step was crucial in understanding the problem and finding where a contribution could be made. This step was completed by reviewing and understanding the different approaches to SMS spam detection and text classification in general. By doing this I was able to formalize a unique approach to the problem.

- **Preprocessing, Tokenization and Vectorization**: Here is where the most time was spent during the experimentation phase. While this step was completed, it looked quite different in practice than initially planned. For preprocessing and tokenization rather than picking one strategy and only using that approach, I continuously tested how different techniques worked on different models. For example I compared how the model performed when using techniques like Named Entity Recognition, POS tagging, Dependency tagging, and lemmatization.

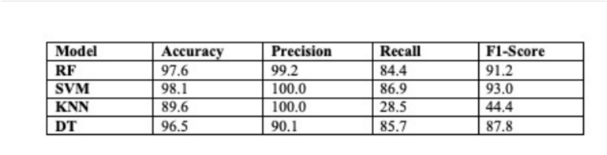
- **Train and Compare Classifiers**: In this step, I proposed comparing how Naive Bayes, Support Vector Machines, Random Forest, K Nearest Neighbour, and Convolutional Neural Networks(CNN) performed. Except for CNN, all of these classifiers were compared.

- **Optimize Selected Classifiers**: I also proposed optimizing one selected classifier. This was completed after comparing how the other classifiers performed, I decided to optimize the Random Forest, as it performed the best when paired with the chosen vectorization strategy.

4 Approach and Methodology

The approach I ended up taking in this paper was to focus on how different preprocessing techniques contributed to the CBOW and optimized Random

Forest model. I started with the most simple type of tokenization and observed how small changes to preprocessing influenced the model. The main ideas behind this approach were based on two observations. SMS spam messages have unique semantics and this can be exploited. The context surrounding a word in a text message is important in the meaning of that word. Several baseline models were created using Count Vectorizer and various classification algorithms. Below in Figure 3, we can see how Random Forest(RF), K Nearest Neighbor(KNN), Support Vector Machine(SVM) and Decision Tree(DT) perform.



Model	Accuracy	Precision	Recall	F1-Score
RF	97.6	99.2	84.4	91.2
SVM	98.1	100.0	86.9	93.0
KNN	89.6	100.0	28.5	44.4
DT	96.5	90.1	85.7	87.8

Figure 3: Baseline performances

Notice in Figure 3 the large gap between recall and precision. My approach aims to bring these stats closer together while increasing accuracy. I did this and created an implementation using POS tagging, CBOW, and Random Forest algorithms. The key components of my implementation are as follows:

- **Data preprocessing:** Using the spaCy library I started by preprocessing and analysing the data by cleaning and tokenizing the text, followed by POS tagging to identify the parts of speech for each word in the text. This will be discussed in further detail in Section 5.

- **Feature Extraction:** The CBOW algorithm was used to extract features from the preprocessed data. The CBOW algorithm generates word embeddings by predicting a target word given its context words. These embeddings are used as features for the Random Forest algorithm. This step was done using word2vec from the Gensim library.

- **Model Training:** The Random Forest algorithm was used to train a classification model using the extracted features. The Random Forest algorithm is an ensemble learning method that builds multiple decision trees and combines their predictions to produce the final output. The model was trained using labelled data, and the performance was evaluated using 20% of the dataset that was set aside for testing. The Random Forest was optimized using hyperparameter tuning to find the optimum number of trees and max depth. The Random Forest model was implemented using the sklearn ensemble library.

By combining the above components, I was able to develop a working model that accurately classified a high percentage of spam messages. The models implemented share the same file.

- **Baseline Models:** As discussed above several baseline models were implemented at the start of the research. These can be found first in the smsSpam.ipynb file.

- **Primary Model:** The primary model that is the focus of this paper using an optimized Random Forest model can be found after the baseline models in the smsSpam.ipynb file.

During the development of the primary model, several challenges were encountered. One of the issues was the fluctuating accuracy of the model's label predictions. Upon investigation, it was discovered that the cause of this problem was the failure to set the seed for word2vec. Another challenge faced was that not all of the POS tags were contributing to the model's improved performance. To address this issue, an analysis of the dataset was conducted to determine the most frequently occurring POS tags, and spaCy's displacy was used to examine the role that various POS tags played in spam messages. Based on the findings, a list of the most significant POS tags was compiled, and tokens with POS tags appearing in the list were tagged with the corresponding POS.

5 Dataset

5.1 Introduction to the dataset

Examining the dataset is an important task in any machine learning task as it helps in understanding the underlying patterns, trends, and relationships present in the data. The first thing I did was inspect the dataset. The UCI SMS spam collection dataset provided by (Almeida et al., 2011) is composed of 5,574 messages with only 13% (747) of these being spam. Some examples of a spam messages can be seen below.

- URGENT! Your Mobile No. was awarded £2000 Bonus Caller Prize on 5/9/03 This is our final try to contact U! Call from Landline 09064019788 BOX42WR29C, 150PPM

- FREE RING TONE just text "POLYS" to 87131. Then every week get a new tone. 0870737910216yrs only £1.50/wk.

- You are guaranteed the latest Nokia Phone, a 40GB iPod MP3 player or a £500 prize! Txt word: COLLECT to No: 83355! IBHltd LdnW15H 150p/Mtmsgrcvd18

Many of the spam messages in the dataset follow a similar style as these messages. Notice that most of the time a spam message contains either text or call contains abnormal sequences of characters such as "LdnW15H", contains a different number to text, and often contains a reference to money.

This particular dataset presents some of the following problems:

- Imbalanced class distribution: The number of spam messages is usually much lower than non-spam messages, resulting in an imbalanced dataset. This can lead to biased predictions, where the model may predict non-spam for all messages to achieve high accuracy which was seen in the baseline models.

- Variability in spam messages: Spam messages can be highly diverse in terms of content, language, and style. They can contain misspelt words, special characters, and irrelevant information, making it difficult for the model to accurately identify them.

- Limited training data: Collecting a large number of labelled SMS messages for training a machine learning model is challenging, especially when attempting to capture the diversity of spam messages. This can also result in overfitting, where the model may memorize the training data instead of learning the underlying patterns.

5.2 Dataset preprocessing

Through the course of this research, I have tested various preprocessing techniques. After careful evaluation of their impact on the performance of the models, I discovered that retaining punctuation and numerical characters in the messages contributed to improved classification accuracy. However, the following preprocessing techniques contributed positively to the performance of the model:

- Lowercasing: Lowercasing is a necessary step for almost any NLP task.

- Stop-word removal: Although stop-words account for a large amount of the vocabulary removing them still improved the model.

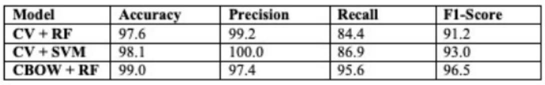
- Selective POS tagging: I found that if the following POS were tagged the model achieved higher accuracy than if none or all the POS were tagged: ’ADJ’, ’NUM’, ’SCONJ’, ’NOUN’,’CCONJ’ ,’PROPN’ ,’ADP’, ’PRON’, ’AUX’.

- CBOW vectorization: As stated a CBOW model was trained and utilised for feature extraction.

- Balancing: I used upsampling on the vectorised training data to allow my model to see more spam messages which improved its accuracy and recall.

6 Results, error analysis

My model was able to achieve competitive results and outperform the baseline SVM in all categories except for precision. The results of using the optimized Random Forest alongside CBOW and conditional POS tagging can be seen below in Figure 4.



Model	Accuracy	Precision	Recall	F1-Score
CV + RF	97.6	99.2	84.4	91.2
CV + SVM	98.1	100.0	86.9	93.0
CBOW + RF	99.0	97.4	95.6	96.5

Figure 4: CBOW Model compared to baselines

Notice this model performs particularly better in terms of recall. Recall refers to the number of spam messages that the model incorrectly classifies as a negative result or ham in this case. The SVM and other baseline models have a clear bias towards predicting ham because ham messages make up 87 % of the dataset. Some of the messages that the baseline models struggle to identify as spam are often messages in the format discussed previously where the message tells the destination to "text BLANK" to a particular number or to call a particular number as seen in the following misclassified messages:

- CALL 09090900040 & LISTEN TO EXTREME DIRTY LIVE CHAT GOING ON IN THE OFFICE RIGHT NOW TOTAL PRIVACY NO ONE KNOWS YOUR [sic] LISTENING 60P MIN 24/7MP 0870753331018+

- Will u meet ur dream partner soon? Is ur career off 2 a flyng start? 2 find out free, txt HORO followed by ur star sign, e. g. HORO ARIES

While the primary model performed better in these aspects and overall misclassified fewer messages, it still struggled at classifying spam messages written in a less obvious tone, particularly messages containing subtle adult content. In the cases that the model misclassified ham messages as spam, these were typically messages containing various types of numbers such as the following:

- Your bill at 3 is £33.65 so thats not bad!
- MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*

While removing numbers in the preprocessing step is common I found that this significantly decreased the overall performance. Overall despite this, I was able to achieve the performance goals with my model and gain valuable insight on a different approach to predicting SMS spam.

7 Lessons learned and conclusions

While I was pleased with the results of this study, there are some areas I would like to see explored in future work. For instance, using POS weights to add weights to more prominent POS. Also creating a model that is more familiar with the slang and abbreviations commonly associated with text messages would be a promising area to study. One of the difficulties faced was finding that many of the preprocessing strategies had little or no effect on the model's performance such as Named Entity Recognition, number normalization and dependency tagging. Rather than try and force these things to work I opted for using only the POS tags that proved to help the model's performance. In conclusion, there are some valuable findings and lessons learned from the research conducted. I was able to find which POS are most important in predicting spam, successfully use CBOW for feature extraction, and optimize a Random Forest that outperformed the baseline models.

References

Olusola Abayomi-Alli, Sanjay Misra, Adebayo Abayomi-Alli, and Modupe Odusami. 2019. A review of soft techniques for sms spam classification: Methods, approaches and applications. *Engineering Applications of Artificial Intelligence*, 86:197–212.

Tiago A. Almeida, José María G. Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: New collection and results. DocEng '11, page 259–262, New York, NY, USA. Association for Computing Machinery.

Nilam Nur Amir Sjarif, Nurulhuda Firdaus Mohd Azmi, Suriayati Chuprat, Haslina Md Sarkan, Yazriwati Yahya, and Suriani Mohd Sam. 2019. Sms spam message detection using term frequency-inverse document frequency and random forest algorithm. *Procedia Computer Science*, 161:509–515. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia.

Dea Delvia Arifin, Shaufiah, and Moch. Arif Bijaksana. 2016. Enhancing spam detection on mobile phone short message service (sms) performance using fp-growth and naive bayes classifier. In *2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, pages 80–84.

Dheny Fernandes, Kelton A.P. Da Costa, Tiago A. Almeida, and João Paulo Papa. 2015. Sms spam filtering through optimum-path forest-based classifiers. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 133–137.

Gauri Jain, Manisha Sharma, and Basant Agarwal. 2018. Optimizing semantic lstm for spam detection. *International Journal of Information Technology*, 11.

Kuruvilla Mathew and Biju Issac. 2011. Intelligent spam classification for mobile text message. In *Proceedings of 2011 International Conference on Computer Science and Network Technology*, volume 1, pages 101–105.

Oberlo. 2023. How many people have smartphones? (2023 data and statistics). Accessed: March 8, 2023.

Houshmand Shirani-Mehr. Sms spam detection using machine learning approach.

Gustavo Sousa, Daniel Carlos Guimarães Pedronette, João Paulo Papa, and Ivan Rizzo Guilherme. 2021. SMS spam detection through skip-gram embeddings and shallow networks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4193–4201, Online. Association for Computational Linguistics.