Grenze International Journal of Engineering and Technology, Jul Issue

Impact of Preprocessing and Feature Representation in the Classification of Documents with Abusive Content

Divya Ann Kurien[1], Namitha S[2], Vinaya Elma Givson[3] and Dr. Ansamma John[4]
[1-4]TKM College of Engineering, Kollam, India
Email: {divya.kurien, namithasadeeshbabu, vinayaelma12}@gmail.com, ansamma.john@tkmce.ac.in

***Abstract*—The use of social media has been increasing over the past decade and has skyrocketed last year due to the pandemic. With the increase in engagement, the amount of abusive text generated has also increased massively, that manually identifying and removing it has become a difficult undertaking. Though automated systems are built, its performance in categorizing text into abusive is highly affected by noise and outliers in unstructured text from social media. The performance of abusive language systems can be improved, by the appropriate structured representation of input text documents with suitable preprocessing stages. The selection of preprocessing steps to improve the overall efficiency of abusive text detection systems depends on the nature of words, language based features and abbreviations which are specific to the dataset. This work focuses on illustrating the appropriateness of different preprocessing steps and its individual or combinational selection based on the features of the dataset used, by transforming the document into a structured form with bag of words or term frequency-inverse document frequency representation methods. It is observed that by combining various preprocessing methods, overall computational efficiency is improved.**

***Index Terms*— Natural Language Processing, Abusive language detection, Preprocessing, Bag of Words, Term Frequency - Inverse Document Frequency.**

I. INTRODUCTION

With the emergence of better communication technology and handheld devices, the number of users who can access the Internet has increased, resulting in greater interaction on social media. The pandemic has accelerated this process, making Internet services and social media an essential tool to obtain emergency assistance, disseminate news and express opinions. Billions of people engage in social media, having multiple identities across different platforms. A huge part of the data generated as part of the interaction belongs to the abusive spectrum, and can possibly cause emotional turbulence, incite violence, degrade the reputation of the target user or organization using the platform. An appropriate abusive language detection system can be used to report such content and remove them from social media. Due to the large scale of data that needs to be handled, manual moderation has become inefficient, time-consuming and labour intensive. Hence, it has been replaced by automatic classification systems.

The performance of automated systems in identifying abusive content is highly affected since the text from social media is unstructured. Each platform has its own design and purpose for interaction, making the format of the data extracted to be different. The lack of proper grammar, usage of informal abbreviations, spelling mistakes, unwanted punctuations, special characters act as noise and outliers if provided directly to the classifier.

Grenze ID: 01.GIJET.7.2.14_1

Moreover, the subjective nature of the text makes it difficult to validate if the text has been correctly categorized. Hence, a common architecture for such a system across all the platforms, languages and demographics is not feasible. If the data is properly preprocessed, it can be represented in a structured manner before being given as input data to the classifier, which could lead to better predictions.

Most of the research on classifying abusive content on social media focuses on achieving better classifiers by employing keyword-based, machine learning and deep learning approaches. Large data requirements of deep learning methods can be a bottleneck in obtaining performance, whereas, for machine learning and keyword based methods, feature selection is the deciding factor. Preprocessing aids feature selection by properly structuring the data and reducing the amount of noise present. In reality, however, relatively less work has been going on in this area. Therefore, this work focuses on finding the impact of preprocessing, selection of preprocessing steps based on the analysis of the dataset and text representation methods on categorizing documents.

The rest of the paper is organized as follows. Section II describes the existing research in the field, section III on methodology, section IV focuses on the experimental analysis and results, and section V on conclusion.

II. Related Works

Natural language processing research has been going on for a long time, and is now primarily focused on ways to improve the performance of existing methods. One of the most effective ways to achieve this is by focusing on preprocessing, which has a huge impact on the final outcome. The irregularities and subjective nature of the text makes this phase difficult and time-consuming. Kadhim [1], discussed the impact of common preprocessing techniques and performs a comparison between the efficiency of chi-square and TF-IDF with cosine similarity score methods for text classification using Machine learning algorithms, proving the performance of TF-IDF with cosine similarity score to be better. Sergienko, Shan and Minker [2], performed a comparison between preprocessing techniques (term weighting and dimensionality reduction methods) and investigated the effect of novel approaches (collectives of term weighting methods and the novel feature transformation method) for user utterance classification. They focussed on term weighting methods, including Inverse Document Frequency(IDF), Gain Ratio(GR), Confident Weights(CW) and Relevance Frequency(RF).

Considerable work is being conducted on improving the efficiency of sentiment analysis using better preprocessing approaches. Haddia, Liua and Shib [3], attempted to explore the role of text preprocessing and feature representation in improving sentiment analysis, and reports on the results. They followed a computational frame consisting of three stages: extraction of relevant features, classification and analysis and finally a performance comparison with other approaches. Their experiments proved that with appropriate preprocessing methods, including text transformation and filtering, can significantly enhance classifier's performance. Oza and Mishra [4], Tang, Li, Cao and Tang [5], Etaiwi and Naymat [6], analysed dataset specific preprocessing techniques on data obtained from web pages, emails and review respectively.

Billal, Fonseca and Sadat [7], present an efficient pipeline for preprocessing the data for NLP, consisting of tweet normalization, hashtag decomposition and named entity recognition. The approach is also based on graph theory and focus on grouping words of tweets using semantic relations of WordNet and connected components. They compared the various results and analysed the effects of these preprocessing steps on the data.

Given the diverse characteristics of each dataset, a pre-existing pipeline of preprocessing techniques might not generate efficient results for all. Ramachandran and Parvathi [8], discussed the dataset specific nature of preprocessing by analysing the impact of Twitter specific preprocessing methods and general techniques on tweets**.** Their work compared the accuracy scores of the two and proved that both work equally competent but for certain stages, the twitter specific techniques had visible advantages.

Munka and Drlíka [9], studied the importance of preprocessing phase in mining to make an efficient pattern analysis on the data extracted from educational systems. Their research focussed on specifying the steps required for gaining valid data representations from the stored logs.

The creation of a stopword list is an essential step in preprocessing, which requires manual analysis and investigations on the dataset. This problem is addressed by Choy [10], by proposing a novel method for automatically generating stopword lists by using the combinatorial nature of the words in speeches. The author analysed the method's effectiveness on nine different datasets, proving it to be comparable with other existing methods, and better in certain cases. Munkov´a, Munk, and Voz´ar [11], also studied the effect of stopwords removal on sequence pattern identification. Similarly, Brill and Moore [12], analysed the effect of another preprocessing, spelling correction on model performance.

III. METHODOLOGY

Due to the widespread engagement and application of social media, developing effective abusive language detection systems has become both a societal need and a difficult undertaking. Hence, this work proposes an abusive language detection system by analysing the dependence of classification performance on preprocessing. The overall architecture for a general classification system is given in Fig. 1. The efficiency of the system can be improved by the selection of appropriate preprocessing steps depending upon the input document, thereby improving the selected features, or by the modification of classifiers. This work gives emphasis for the translation of documents by the application of preprocessing methods.

Text obtained from various social media platforms are unstructured, and different from each other in its format since the design, purpose and people using each platform are different. Therefore, a generic abusive language classifier would not perform sufficiently and has to be modified according to the nature of the data. Preprocessing and text representation phases are available for the appropriate transformation of input documents, and they must be in structured form when it is given as input to the classifier. The transformations necessary for this are determined by the document's format and noise level. Based on the input document analysis, necessary noise removal, out of vocabulary cleansing and text transformations are performed to obtain a modified dataset. Further, this dataset is converted into a format that can be processed directly by the classifier. The new format should account for all the important words in the document and the frequency of those words. This can be addressed by the use of two popular count based vector representations, BoW and TF-IDF. All the necessary steps that should be followed for the appropriate document transformation require the critical analysis of the input dataset, and are described in the next section.

A. Dataset Analysis

Identification of abusive language is a popular research area owing to the importance of the task. A suitable data set from the pool of several available data sets is chosen for this work. Hate speech and Offensive Content identification (HASOC) dataset, made available as part of the FIRE(Forum for Information Retrieval Evaluation workshop 2019) consist of posts sampled from both Facebook and Twitter. The posts are labelled as Hate Offensive (HOF) and Not Hate Offensive (NOF). A detailed analysis is required to uncover the underlying structure and maximize insights into the dataset. Hence, the analysis of the data from several perspectives, starting from the computation of various entities like class and sentence length distribution, occurrence of non-English words, spelling errors and unwanted information which is further extended to manual analysis. Each selected entity influences the performance of the system in different ways. Calculating the class distribution is critical for determining the performance parameter, whereas sentence length variance increases the processing complexity. The use of non-English words in the dataset to express real emotions increases the complexity in the preprocessing phase due to the diversity in the way these can be handled. Since HASOC dataset consists of a lot of punctuation, erroneously spelt words and the mixture of upper and lower case words to express the intensity of usage of some words and context, manual analysis is inevitable due to the complexity of developing automated systems. Most of the said complexities associated with the dataset can be handled by the proper preprocessing of this document.

B. Preprocessing

The unstructured nature of the document and the presence of non-English words, grammatical errors, poorly arranged and incomplete sentences prevent the direct usage of the input document by the classifier. The transformation of this document to an appropriate format by the separation of words can be done in multiple steps which are either popular or specific to the dataset and is decided by the analysis of the dataset. Fig. 2. depicts the various preprocessing steps used for this work. The main focus of this document transformation is to represent the document in a structured form to reduce the space and the time complexity in further processing and enhance the overall efficiency of the system.

As the first step, the tokenisation can be performed to decompose the document into words or tokens using any one of the popular methods like regular expression based, split function-based or by the NLTK and Gensim depending on the content of the document and can be selected by proper analysis. On the tokenized output, necessary processing must be performed in order to clean the tokenised document and remove noises, as well as words that do not convey any emotion. Detection and removal of such noises are another challenging task due to the absence of a proper pre-defined format for the documents in social media, and the presence of platform-dependent information that is irrelevant from the abusive classification point of view. Moreover, usage of non-English words by non-native speakers can be removed blindly or tactically handled by assigning valid scores

depending upon the nature of the document. The presence of punctuations should be handled carefully, as they may act as an intensifier for expressing emotions using some tokens. From the document analysis, it is observed that, out of 32 popular punctuations that can be present in the document, only exclamation (!) and question (?) marks can be considered as valid sentiment intensifiers. All other punctuation can be removed to reduce the size of the document.

Further size reduction can be achieved by the removal of stop words that do not convey any meaning from the sentiment point of view. Some language-specific stopwords [13-14], are accessible and can be used directly, while others can be defined through proper document analysis. Once the document size is reduced, the next step is to convert it into the appropriate format by recognizing multiple forms of the same root word and correcting misspelled words. Since NLP representation methods are case-sensitive, words must be translated to a uniform case prior to further processing to reduce document transformation complexity. Abbreviations and words that have full uppercase, which express the intensity of excitement or anger, must be handled without changing its original form. Even by considering all the measures specified above, the size of the transformed document may not decrease significantly. This problem can be addressed by converting the different forms of the same word into its root word by stemming [15-16] process. Many standard stemming algorithms such as Lancaster, Porter and Snowball stemmer are available and the performance of these methods are document dependent. Hence, appropriate selection of a stemmer is possible only through the proper analysis of the input document.

Occurrence of erroneous words can be due to the ignorance of the people using the platform, or intentionally used shorthand representation to express spontaneous emotions. A few character changes are manageable and may be fixed using spell checkers and dictionaries. These corrections may sometimes adversely affect the performance when the input contains purposefully considered words that deviate from their correct forms by the repetition of specific characters to convey the intensity of emotions. The presence of a substantial amount of these words in the dataset needs to be normalized by removing the repeating characters and should be handled separately. All of these stages are too complex to be handled by a single preprocessing method, and may necessitate the use of multiple preprocessing methods. After pre-processing, documents must be converted to a format by considering the significant words or features in the dataset on which the classifier can work.

C. Feature Representation

Although there exist several methods for the representation of documents, this work focuses on two count based approaches, BoW and TF-IDF. The dataset is restructured as a document-term matrix, with rows representing the number of documents and columns representing the vocabulary set of the dataset. As a result of preprocessing, the vocabulary set is compressed, containing only the relevant words or features. The values in the matrix vary according to the representation technique followed.

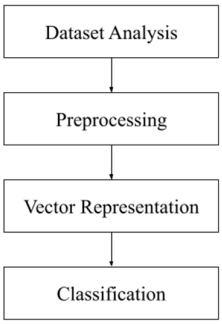

Figure 1. Overall Architecture

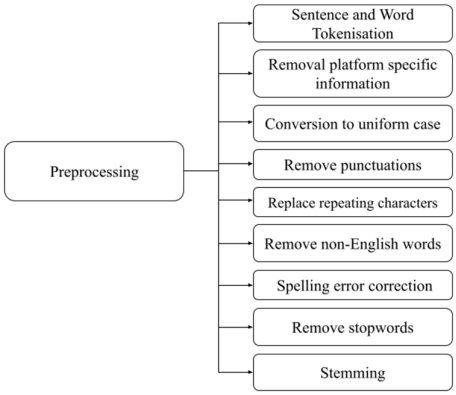

Figure 2. Stages of preprocessing

The BoW method represents a document as a vector in terms of the number of times a specific word appears in it, rather than considering its relevance. TF-IDF is another technique available to restructure documents which take into account the relevance of a word in a document. Each value in the document term matrix is proportional to the word count in a document and inversely proportional to the number of times the word appears across the documents, by which frequently appearing words that do not have high significance, like conjunctions, are given

less emphasis. Despite the fact that these count-based methods ignore the document's semantics, context, or sentence structure, resulting in inferior performance than semantic-based methods, these approaches are used in this study due to the large reduction in size, which can improve overall performance. For each individual and combinational selection of preprocessing steps, both BoW and TF-IDF representations are constructed and given to the classifier. This work does not attempt to develop or implement classifiers, but rather, it tries to improve and analyse the input document's preprocessing steps. The efficiency of the preprocessing phases in terms of classification accuracy can be evaluated by selecting a suitable classifier. From similar classification works using machine learning techniques conducted on HASOC[17], SVM classifier gave the best performance and hence is used for this work. The following section describes the experimental analysis of the work implemented.

IV. Experimental Analysis and Results

Based on the literature review and analysis of the input document, a number of preprocessing methods are implemented using the input dataset. The performance of this classifier is analysed individually and in groups by applying them in successive stages. The dataset is carefully selected, providing the platform to test and analyse the preprocessing stages suggested and implemented. The next section outlines the characteristics of the dataset HASOC used for the abusive language detection.

A. Dataset

The HASOC dataset is a labelled dataset wherein all documents have been labelled as Hate Offensive(HOF) or Not Offensive(NOT). The dataset is skewed since HOF posts constitute only 38.64% of the entire dataset, which can be inferred from Fig. 3. This dataset should be transformed into an appropriate format so that the classifier can directly handle it by analysing the dataset from various viewpoints. To obtain the necessary information on the content of the document, words with high frequency counts are extracted. Fig. 4 represents the top words in the dataset and its relevance with the input document is observed and the co-relationship is justified. The tweets cover various themes related to Trump, ICC World cup, Doctors strike as the major ones. Facebook and Twitter dataset include user mentions, hashtags, slang phrases, URLs, emojis, non-English words and other entities which are irrelevant from the text document point of view and may express a high degree of emotions. Platform specific information (user mentions, usernames, URLs) are not relevant for the classification task, but will be useful in identifying abusers and cyberbullies. This dataset consists of 26% usernames and tags, which can be removed directly to reduce the overall size of the input dataset. 0.15% of the documents contain Non-English words, 0.05% contains punctuation, 0.18% have spelling errors, due to which the meaning of these documents can be altered. The train-test dataset distribution is as shown in Table I. The test data, which consists of 1153 tweets, is available separately and can be used to evaluate our model.

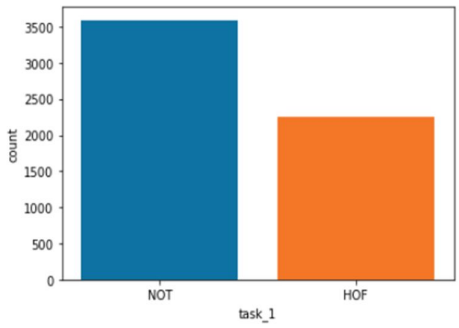

Figure 3. Class distribution

B. Analysis

The impact on classification by the application and without the application of various preprocessing techniques have been analysed, and the consolidated inferences are discussed below:

To represent the document in a computationally efficient format, the vocabulary set is extracted using tokenisation methods. It is observed that most of the tokenisers fail to handle the tokens like decimals, abbreviations, miscellaneous non-sentence-ending uses of punctuation, and escape characters like '\n' efficiently. For most Facebook and Twitter data, the different tokenisers shows random behaviour due to the freedom of users while expressing their views and sometimes formal rules are not sufficient for separating

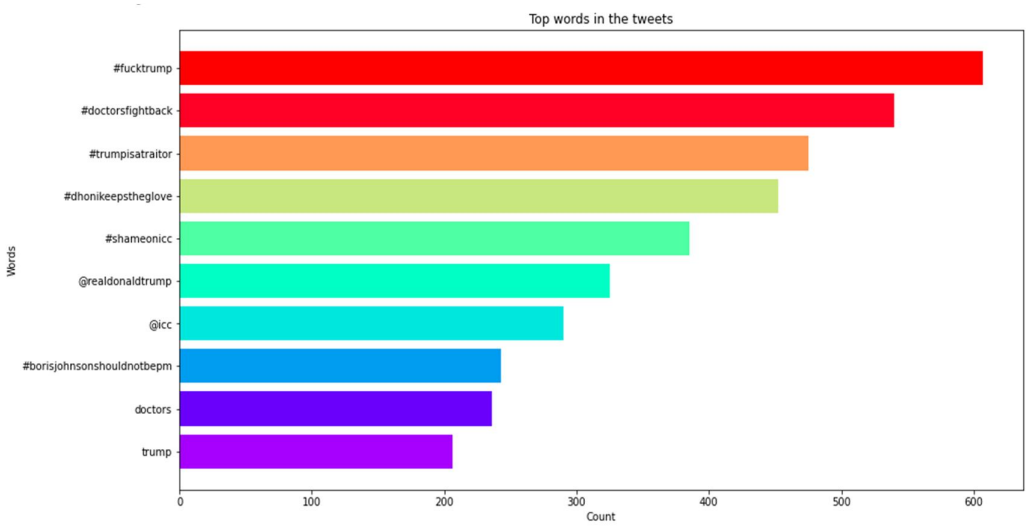

Figure 4. Top 10 words in the dataset

tokens. Hence, choosing an effective tokeniser depends on the task at hand and the input dataset chosen. In this work, we observed the behaviour of tokenisers like split function tokeniser, regular expression based tokeniser, Built-in tokeniser provided NLTK and Gensim. Split function tokeniser correctly tokenized documents with words separated by spaces or any other single character. The effectiveness of the regular expression based tokeniser depends on the construction of a strong rule which takes care of almost all the special cases present in the document. The built-in tokenisers handle punctuation and escape sequences differently, Gensim based tokeniser remove punctuation completely and NLTK based tokenisers remove them partially. In comparison to other tokenisers, the regular expression based tokeniser handled the majority of cases efficiently for this dataset.

Due to the speciality of the dataset considered, it comprises a number of tokens with spelling errors which is induced intentionally or occurred unknowingly. To handle this situation the spell check is applied, and it is observed that some words have changed completely while other words have been restored to their correct form.

The number of words available in the document and the tokens generated are nearly identical due to the presence of multiple forms of the same words that can be converged to its root form by stemming process. This operation considerably reduced the token count with respect to word count. To handle diverse aspects of the text document various standard stemming algorithms are available and for the analysis of this work, we consider Lancaster, and Porter stemmers. Among these methods, porter stemmer reduces the number of tokens noticeably, and its performance is better for this dataset, which is justified in Table II.

The preprocessing steps discussed above, aimed at decreasing the size of the words or features, and it is achieved to a certain extent. The original document must be represented using these reduced features appropriately. The best suitable method that is suggested by most of the researchers is considered here, which is the vector representation of the documents**.** The vector size is determined by the number of characteristics or words extracted after the preprocessing phase. As the number of words decreases, the size of the vector also decreases proportionately, thereby reducing the processing time. In the vector representation itself, two approaches are implemented, one using BoW and the other using TF-IDF. Analysis of the influence of various preprocessing methods on the input data, its effect on vector size and comparison of BoW and TF-IDF vector representation is conducted using a classifier for categorising the sentences in the document. Initially, without any preprocessing the document is represented as vectors using both BoW and TF-IDF separately and classification f1-score is observed. Following this, the vectors are constructed after the preprocessing steps to (i) remove platform dependent tokens (ii) convert tokens to uniform case. The classifier performance is also analysed for both vector representations and is indicated in the Table III. From Fig. 5 it is clear that by the application of the removal of platform specific features and conversion to uniform case preprocessing steps did not increase the performance of the classifier significantly due to the presence of very less non-uniform lettercase features and the irrelevance of platform dependent features for the classification considered. By the application of these preprocessing stages, vector size is reduced considerably. Without preprocessing, the document yields 20124 tokens, which is

decreased by 26% when platform-specific information is removed, resulting in a significant reduction in vector size, which can influence processing time.

By applying other preprocessing methods like removal of punctuation, repeating characters, non-English words, spelling errors, stopword, and stemming, document vector is constructed independently for each of the case separately and the performance of the classifier is observed, which is given in Table IV and Table V. Although the independent application of each of the preprocessing methods did not influence the classification process significantly, the vector size is minimised by 1.2% for transforming elongated words, 3.8% for spelling error correction, 0.86% for stopwords removal and 7.1% for stemming which decreases processing time.

TABLE I. TRAIN AND TEST DATASET CLASS DISTRIBUTION

	Train Data	Test data
HOF	2261	288
NOT	3591	865

TABLE II. OUTPUT COMPARISON OF VARIOUS STEMMERS

Original Word	Output from Porter Stemmer	Output from Lancaster Stemmer
all	all	al
very	very	veri
clearly	clear	cleari

TABLE III. RESULTS OF APPLYING PLATFORM SPECIFIC INFORMATION REMOVAL AND CONVERSION TO UNIFORM CASE

	Without preprocessing (P1)				Removal of platform specific information (P2)				Conversion to uniform case (P3)			
	Precision	Recall	F1-Score		Precision	Recall	F1-Score		Precision	Recall	F1-Score	
			Macro	Weighted			Macro	Weighted			Macro	Weighted
BoW	0.82	0.89	0.67	0.76	0.82	0.90	0.68	0.77	0.82	0.89	0.67	0.76
TF-IDF	0.83	0.87	0.69	0.77	0.83	0.87	0.69	0.77	0.83	0.87	0.69	0.77

TABLE IV. RESULTS OF APPLYING PUNCTUATION REMOVAL, REPLACING REPEATING CHARACTERS, NON-ENGLISH WORD REMOVAL

	Punctuation removal (P4)				Replacing repeating characters (P5)				Separate Non-English Words (P6)			
	Precision	Recall	F1-Score		Precision	Recall	F1-Score		Precision	Recall	F1-Score	
			Macro	Weighted			Macro	Weighted			Macro	Weighted
BoW	0.82	0.89	0.68	0.77	0.82	0.90	0.67	0.77	0.82	0.89	0.67	0.76
TF-IDF	0.84	0.87	0.68	0.78	0.83	0.87	0.69	0.77	0.84	0.87	0.69	0.78

TABLE V. RESULTS OF APPLYING SPELLING ERROR CORRECTION, STOPWORD REMOVAL AND STEMMING

	Fix spelling errors (P7)				Remove stopwords (P8)				Stemming (P9)			
	Precision	Recall	F1-Score		Precision	Recall	F1-Score		Precision	Recall	F1-Score	
			Macro	Weighted			Macro	Weighted			Macro	Weighted
BoW	0.82	0.89	0.68	0.77	0.82	0.93	0.70	0.79	0.81	0.90	0.66	0.76
TF-IDF	0.82	0.86	0.67	0.76	0.83	0.91	0.70	0.79	0.83	0.87	0.70	0.78

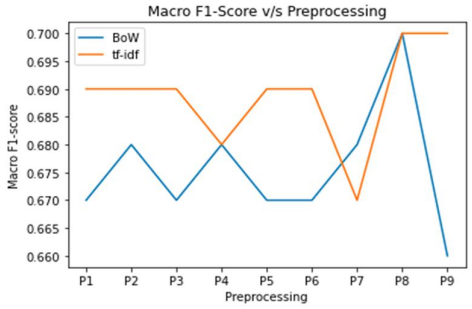

Figure 5. Experimental results in terms of Macro F1 Score on individual preprocessing steps

Since independent application of preprocessing stages did not contribute much to the classification performance, an additional analysis is carried out to see how the combination of preprocessing stages affects the classification process. While doing this analysis, platform specific information removal, conversion to uniform lowercase methods, removal of stopwords are performed along with all other combinations of preprocessing. Table VI presents the preprocessing combination applied, and the performance analysis. Visualisation of the same is

depicted in Fig. 6. As there are several preprocessing stages to consider, an experiment is carried out to see how the sequence in which the preprocessing processes are applied affects the outcome. However, punctuations were an exception, as their removal at different stages had no effect on performance, but a modest reduction in feature size was detected. A decline was observed each time the spelling correction was introduced into a combination. Without considering punctuation removal and by considering spelling correction, noticeable decrease is observed. The inclusion of stemming stage in conjunction with other preprocessing methods resulted in a considerable reduction in feature size by 41% while maintaining the performance. The size was decreased by 52 percent when all the processes were completed in order, and there was also a commensurate reduction in time. However, this resulted in the elimination of certain essential features that influence the classification process, resulting in a performance drop for both representations when compared to other combinations specified. Storage and time efficiency obtained while transforming the document into vector form by the application of various preprocessing stages is given in Table VII and the pictorial representation of the same is given in Fig. 7 and Fig. 8. From the analysis conducted with various preprocessing stages and the vector constructions, it is evident that TF-IDF vector representation performed better compared to BoW.

TABLE VI. RESULTS OF SELECTED COMBINATIONAL PREPROCESSING STEPS IN VECTOR CONSTRUCTION

Combination		Precision	Recall	F1-score	
				Macro	Weighted
Removal of platform specific information + Conversion to lowercase + Removing stopwords (C1)	BoW	0.83	0.93	0.71	0.80
	TF-IDF	0.83	0.91	0.71	0.79
Removal of platform specific information + Conversion to lowercase + Punctuation removal + Removing stopwords (C2)	BoW	0.83	0.93	0.70	0.79
	TF-IDF	0.83	0.92	0.71	0.79
Removal of platform specific information + Conversion to lowercase+Punctuation removal +Fix spelling errors + Removing stopwords (C3)	BoW	0.82	0.93	0.70	0.79
	TF-IDF	0.82	0.92	0.70	0.79
Removal of platform specific information + Conversion to lowercase +Fix spelling errors + Removing stopwords (C4)	BoW	0.82	0.93	0.70	0.79
	TF-IDF	0.82	0.92	0.69	0.78
Removal of platform specific information + Conversion to lowercase + Removing stopwords + Punctuation removal (C5)	BoW	0.83	0.93	0.71	0.80
	TF-IDF	0.83	0.92	0.71	0.80
Removal of platform specific information + Conversion to lowercase + Removing stopwords + Punctuation removal + Stemming (C6)	BoW	0.83	0.92	0.71	0.79
	TF-IDF	0.84	0.91	0.71	0.79
All the mentioned steps combined (C7)	BoW	0.82	0.93	0.68	0.79
	TF-IDF	0.83	0.90	0.69	0.78

TABLE VII. FEATURE SIZE AND TIME (IN SECS) FOR TRAINING SVM WITH TF-IDF FOR INDIVIDUAL AND COMBINATION OF PREPROCESSING STEPS

	Size	Time (in secs)		Size	Time (in secs)
P1	20124	1081	P9	18695	922
P2	14818	833	C1	14809	790
P3	20124	1167	C2	14743	723
P4	20110	1050	C3	13334	655
P5	19878	1029	C4	13992	686
P6	20102	995	C5	14803	783
P7	19340	960	C6	11731	628
P8	19950	1035	C7	9638	505

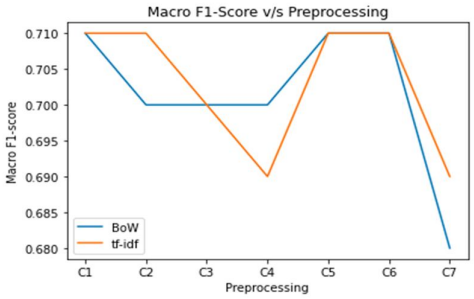

Figure 6. Experimental results in terms of Macro F1 score on combinational preprocessing steps

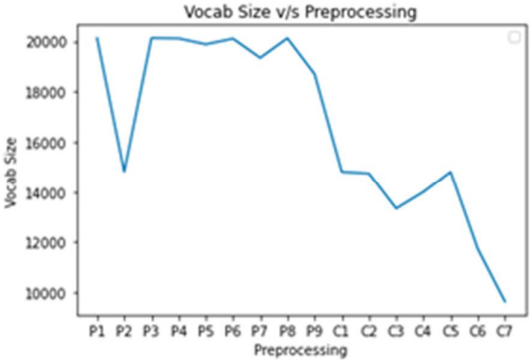

Figure 7. Experimental results in terms of vocabulary size

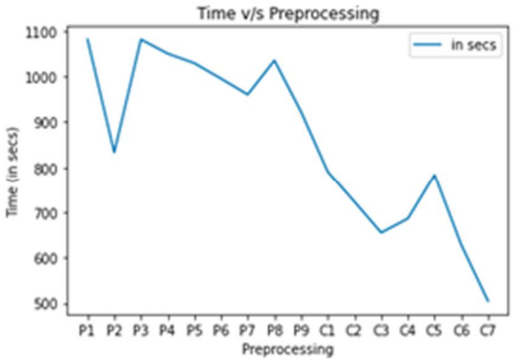

Figure 8. Experimental results in terms of time taken (in secs) for training SVM with TF-IDF feature set

V. Conclusion And Future Works

Due to the increasing use of social media, it is essential to recognize and remove abusive language in order to provide a better user experience with the application of text processing. Preprocessing is an essential step in all text processing research, by which documents can be transformed into an efficient form. The selection of preprocessing methods, identifying the order in which these preprocessing method should be applied and determining the combinational preprocessing stages to be applied to improve the overall efficiency of the system in terms of time, space and performance is a challenging task since it depends upon the nature of the dataset and the computation task under consideration. A detailed analysis is performed on the effect of various preprocessing stages in the classification process using HASOC dataset by transforming the input document into two vector forms TF-IDF and BoW. TF-IDF method gives a better performance compared to BoW. In this work, by the application of some preprocessing methods, words with the repeated characters, emojis and non-English words maybe considered as noise and steps are taken to remove such noises and that decreased the efficiency of the system. Performance can be improved if these noises are considered as emotion intensifiers and to incorporate their presence as separate entities during the document vector construction phase. This study can be extended to develop new preprocessing methods to handle multilingual text dataset.

References

[1] Ammar Ismael Kadhim, “An Evaluation of Preprocessing Techniques for Text Classification”, International Journal of Computer Science and Information Security, 2018
[2] Roman Sergienko, Muhammad Shan and Wolfgang Minker, “A Comparative Study of Text Preprocessing Approaches for Topic Detection of User Utterances”, Tenth International Conference on Language Resources and Evaluation,2016
[3] Emma Haddia, Xiaohui Liua and Yong Shib, “The Role of Text Pre-processing in Sentiment Analysis”, Information Technology and Quantitative Management (ITQM2013)
[4] Alpa K. Oza, Shailendra Mishra, “Elimination of Noisy Information from Web Pages”, International Journal of Recent Technology and Engineering (IJ RTE 2013).
[5] Jie Tang, Hang Li, Yunbo Cao and Zhaohui Tang, “Email Data Cleaning”, SIGKDD international conference on Knowledge discovery in data mining 2005.
[6] Wael Etaiwi and Ghazi Naymat, “The Impact of applying Different Preprocessing Steps on Review Spam Detection”, The 8th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2017).
[7] Belainine Billal, Alexsandro Fonseca and Fatiha Sadat,“Efficient Natural Language Pre-processing for Analyzing Large Data Sets”, IEEE International Conference on Big Data 2016.
[8] Dharini Ramachandran and Parvathi R, “Analysis of Twitter Specific Preprocessing Technique for Tweets”, International Conference on Recent Trends in Advanced Computing, ICRTAC 2019.
[9] Michal Munka and Martin Drlíka, “Impact of Different Pre-Processing Tasks on Effective Identification of Users' Behavioral Patterns in Web-based Educational System”, International Conference on Computational Science, ICCS 2011.
[10] Murphy Choy, “Effective Listings of Function Stop words for Twitter”, International Journal of Advanced Computer Science and Applications 2012.
[11] Da˘sa Munkov´a, Michal Munk, and Martin Voz´ar, “Influence of Stop-Words Removal on Sequence Patterns Identification within Comparable Corpora”, International Conference on ICT Innovations 2013.

[12] Eric Brill and Robert C Moore, “An Improved Error Model for Noisy Channel Spelling Correction”, 38th Annual Meeting of the Association for Computational Linguistics 2000.
[13] Ansamma John, P. S. Premjith, and M. Wilscy, “Extractive Multi-document Summarization using Population-based Multicriteria Optimization”, Expert Systems with Applications 86 (2017): 385-397.
[14] Ansamma John, and M. Wilscy, “Random Forest Classifier based Multi-document Summarization System”, IEEE Recent Advances in Intelligent Computational Systems (RAICS) 2013.
[15] Premjith, P. S., Ansamma John, and M. Wilscy, “Metaheuristic Optimization using Sentence Level Semantics for Extractive Document Summarization”, International Conference on Mining Intelligence and Knowledge Exploration. Springer, Cham, 2015.
[16] Jasbeen, M. S., Ansamma John, and Manu J. Pillai, “Enhancing Sentiment Analysis using Rule Mining and Dual Processing”, AIP Conference Proceedings. Vol. 2222. No. 1. AIP Publishing LLC, 2020.
[17] Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, Aditya Patel(2019) “Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages”, FIRE '19: Proceedings of the 11th Forum for Information Retrieval Evaluation ,December 2019.