applied sciences

MDPI

Article

Automatic Vulgar Word Extraction Method with Application to Vulgar Remark Detection in Chittagonian Dialect of Bangla

Tanjim Mahmud [1,2,*], Michal Ptaszynski [1,*] and Fumito Masui [1]

[1] Text Information Processing Laboratory, Kitami Institute of Technology, Kitami 090-8507, Japan; f-masui@mail.kitami-it.ac.jp
[2] Department of Computer Science and Engineering, Rangamati Science and Technology University, Rangamati 4500, Bangladesh
* Correspondence: tanjim_cse@yahoo.com (T.M.); michal@mail.kitami-it.ac.jp (M.P.)

Abstract: The proliferation of the internet, especially on social media platforms, has amplified the prevalence of cyberbullying and harassment. Addressing this issue involves harnessing natural language processing (NLP) and machine learning (ML) techniques for the automatic detection of harmful content. However, these methods encounter challenges when applied to low-resource languages like the Chittagonian dialect of Bangla. This study compares two approaches for identifying offensive language containing vulgar remarks in Chittagonian. The first relies on basic keyword matching, while the second employs machine learning and deep learning techniques. The keyword-matching approach involves scanning the text for vulgar words using a predefined lexicon. Despite its simplicity, this method establishes a strong foundation for more sophisticated ML and deep learning approaches. An issue with this approach is the need for constant updates to the lexicon. To address this, we propose an automatic method for extracting vulgar words from linguistic data, achieving near-human performance and ensuring adaptability to evolving vulgar language. Insights from the keyword-matching method inform the optimization of machine learning and deep learning-based techniques. These methods initially train models to identify vulgar context using patterns and linguistic features from labeled datasets. Our dataset, comprising social media posts, comments, and forum discussions from Facebook, is thoroughly detailed for future reference in similar studies. The results indicate that while keyword matching provides reasonable results, it struggles to capture nuanced variations and phrases in specific vulgar contexts, rendering it less robust for practical use. This contradicts the assumption that vulgarity solely relies on specific vulgar words. In contrast, methods based on deep learning and machine learning excel in identifying deeper linguistic patterns. Comparing SimpleRNN models using Word2Vec and fastText embeddings, which achieved accuracies ranging from 0.84 to 0.90, logistic regression (LR) demonstrated remarkable accuracy at 0.91. This highlights a common issue with neural network-based algorithms, namely, that they typically require larger datasets for adequate generalization and competitive performance compared to conventional approaches like LR.

Keywords: vulgar remark detection; vulgar term extraction; low-resource language; logistic regression; recurrent neural network

Citation: Mahmud, T.; Ptaszynski, M.; Masui, F. Automatic Vulgar Word Extraction Method with Application to Vulgar Remark Detection in Chittagonian Dialect of Bangla. *Appl. Sci.* **2023**, *13*, 11875. https://doi.org/10.3390/app132111875

Academic Editor: Rocco Zaccagnino

Received: 29 August 2023
Revised: 6 October 2023
Accepted: 25 October 2023
Published: 30 October 2023

1. Introduction

Bangladesh has seen a remarkable increase in its use of the Internet over the past two decades. There were more than 125 million Internet users in Bangladesh as of November 2022, according to the Bangladesh Telecommunication Regulatory Commission (BTRC) [1]. Additionally, with the help of the implementation of the Digital Bangladesh initiative [2], the vast majority of people in Chittagong [3], Bangladesh's second-largest city, now have access to the Internet and actively use social media. According to a survey [4], the number of Facebook users in Bangladesh is the highest among social media (see

Figure 1). Moreover, with the benefit of Unicode being widely used on most communication devices, such as tablets or smartphones, speakers of underrepresented languages, such as Chittagonian, can express their thoughts in their native languages and dialects. Many people in Chittagong now use social media on a regular basis, regularly using platforms like Facebook [5], imo [6], various blogs, and WhatsApp [7]. These platforms offer a venue where people can express themselves freely and informally. However, the pervasiveness of social media has also resulted in unfavorable influences that are difficult to shake. Excessive use of social media has the potential to cause addiction [8], which as a result could cause young people to spend more time on these platforms than they spend with their family and friends [9]. Their general health and social interactions may suffer as a result of this addiction. Additionally, social media witnesses the growing problem of the increase in online abuse and cyberbullying, which can have a negative impact on a person's self-esteem and even violate their privacy [10]. The spread of misinformation and hatred online has also contributed to an uptick in violent crimes in society [11]. Receiving messages with vulgar language is a startling realization of this unwelcome and damaging phenomenon. The likelihood of encountering such vulgar remarks rises as social media use increases.

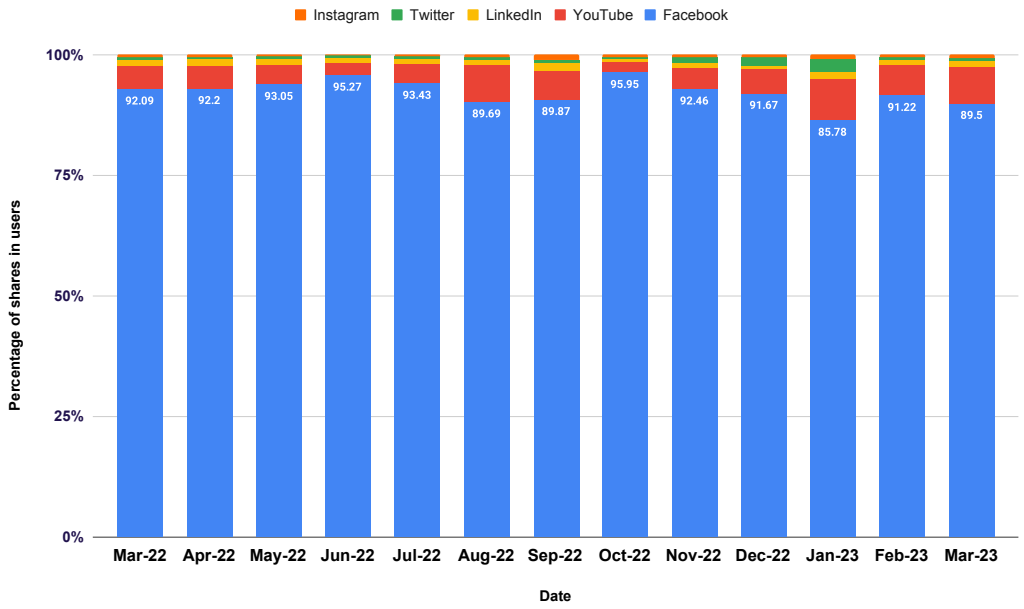

Figure 1. Yearly social media usage statistics in Bangladesh as of March 2023 [4].

Vulgarity or obscenity with regards to language refers to terms used to describe the use of vulgar language, such as swearing, taboo words, or offensive expressions [12,13]. Unfortunately, such language has become more and more common in contemporary culture [14], especially on social media sites like Twitter [15]. The majority of the time, however, vulgar language is used in the context of online harassment and negativity. While there are some instances where vulgar language may be used in a positive context to convey informality, express anger, or establish a sense of belonging with a particular group [16], these are usually rare in comparison to its use in negative contexts or in closed message groups. Therefore, detecting such use of language quickly and effectively is necessary to allow social media platforms to efficiently moderate their contents.

Therefore, the goal of this research was to find and evaluate vulgar remarks in the Chittagonian dialect of Bangla. Chittagonian is a member of the Indo-Aryan language family [17] and is spoken by between 13 and 16 million people, most of whom live in Bangladesh [18]. Although some linguists categorize it as a distinct language, Chittagonian is frequently used together with Bengali and has its own pronunciation, vocabulary, and grammar [19]. In this paper, we present a system for automatic detection of vulgar remarks in an effort to combat the growing problem of vulgar language online. To achieve

this we apply various machine learning (ML) and deep learning (DL) algorithms, such as logistic regression (LR) or recurrent neural networks (RNN). We also use a variety of feature extraction techniques to expand the system's functionality. The performance of these ML and DL algorithms in detecting vulgar remarks was thoroughly investigated through rigorous experimentation, which is particularly important in a low-resource language scenario, such as Chittagonian, where linguistic resources are scarce.

The goal of this research is to advance the field of vulgar remark detection for the Chittagonian dialect by achieving the following key objectives:

1. Collect a comprehensive dataset of 2500 comments and posts exclusively from publicly accessible Facebook accounts in the Chittagonian dialect.
2. Ensure the dataset's quality and reliability through rigorous manual annotation and validation using established metrics like Cohen's Kappa statistics [20] and Krippendorff's alpha [21].
3. Develop a simple keyword-matching-based baseline method for vulgar remark detection using a hand-crafted vulgar word lexicon.
4. Create a method for automatically expanding the vulgar word lexicon to ensure future-proofing of the baseline method.
5. Implement various matching algorithms to detect sentences containing vulgar remarks, beginning with a simple method using a manually crafted vulgar word lexicon, an automatic method using simple TF-IDF statistics for vulgar term extraction with no additional filtering of non-vulgar words, as well as a more robust method applying additional filtering of words with a high probability of being non-vulgar.
6. Evaluate various ML- and DL-based approaches to identify vulgar remarks in Chittagonian social media posts, aiming for over 90% accuracy for practical applicability.
7. Conduct a thorough comparison between the keyword-matching baseline method and machine learning (ML) and deep learning (DL) models to achieve the highest possible performance in vulgar remark detection.

The subsequent sections of this paper are organized as follows: In Section 2, we present a comprehensive review of related works on vulgar word detection and related content filtering in the languages of Bangladesh. Section 3 elaborates on the dataset collection and preprocessing techniques specific to the Chittagonian dialect as well as outlines the proposed automatic vulgar word extraction method, detailing the NLP, ML, and DL techniques employed. In Section 4, we present the experimental results and performance evaluation of our approach. Finally, Section 5 summarizes the contributions and discusses potential future directions for expanding this research.

2. Literature Review

This section presents a thorough analysis of prior work on the identification and categorization of vulgarity. We have also included studies from closely related fields because vulgar language in user-generated content on social media frequently includes expressions of sexism, racism, hate speech, and other types of online abuse [12]. Table 1 shows research in Bengali on topics related to detecting vulgarity.

Traditionally, vulgar expression lexicons have been developed as a means of vulgarity detection [22]. These lexicon-based approaches need to be updated frequently to remain effective, however. In contrast, machine learning (ML) techniques provide a more dynamic approach by classifying new expressions as either vulgar or non-vulgar without relying on predetermined lexicons. Deep learning has made significant contributions to the field of signal and image processing [23], diagnosis [24], wind forecasting [25] and time series forecasting [26].

Beyond lexicon-based techniques, vulgarity detection has been the subject of several studies. Moreover, numerous linguistic and psychological studies [27] have been carried out to comprehend the pragmatic applications [13] and various vulgar language forms [28].

For machine learning-related studies, for example, Eshan et al. [29] ran an experiment in which they classified data obtained by scraping the Facebook pages of well-known

celebrities using the traditional machine learning classifiers multinomial naive Bayes, random forest, and SVM (support vector machine). They gathered unigram, bigram, and trigram features and weighted them using *TF-IDF* vectorizers. On datasets of various sizes, containing 500, 1000, 1500, 2000, and 2500 samples. The results showed that when using unigram features, a sigmoid kernel had the worst accuracy performance, and SVM with a linear kernel had the best accuracy performance. However, MNB demonstrated the highest level of accuracy for bigram and trigram features. In conclusion, TfidfVectorizer features outperformed CountVectorizer features when combined with an SVM linear kernel.

Akhter et al. [30] suggested using user data and machine learning techniques to identify instances of cyberbullying in Bangla. They used a variety of classification algorithms, such as naive Bayes (NB), J48 decision trees, support vector machine (SVM), and k-nearest neighbors (KNN). A 10-fold cross-validation was used to assess how well each method performed. The results showed that SVM performed better than the other algorithms when it came to analyzing Bangla text, displaying the highest accuracy score of 0.9727.

Holgate et al. [16] introduced a dataset of 7800 tweets from users whose demographics were known. Each instance of vulgar language use was assigned to one of six different categories by the researchers. These classifications included instances of aggression, emotion, emphasis, group identity signaling, auxiliary usage, and non-vulgar situations. They sought to investigate the practical implications of vulgarity and its connections to societal problems through a thorough analysis of this dataset. Holgate et al. obtained a macro F1 score of 0.674 across the six different classes by thoroughly analyzing the data that were gathered.

Emon et al. [31] created a tool to find abusive Bengali text. They used various deep learning and machine learning-based algorithms to achieve this. A total of 4700 comments from websites like Facebook, YouTube, and Prothom Alo were collected in a dataset. These comments were carefully labeled into seven different categories. Emon et al. experimented with various algorithms to find the best one. The recurrent neural network (RNN) algorithm demonstrated the highest accuracy among the investigated methods, achieving a satisfying score of 0.82.

Awal et al. [32] demonstrated a naive Bayes system made to look for abusive comments. They gathered a dataset of 2665 English comments from YouTube in order to evaluate their system. They then translated these English remarks into Bengali utilizing two techniques: (i) Bengali translation directly; (ii) Bengali translation using dictionaries. Awal et al. evaluated the performance of their system after the translations. Their system impressively achieved the highest accuracy of 0.8057, demonstrating its potency in identifying abusive content in the context of the Bengali language.

Hussain et al. [33] suggested a method that makes use of a root-level algorithm and unigram string features to identify abusive Bangla comments. They gathered 300 comments for their dataset from a variety of websites, including Facebook pages, news websites, and YouTube. The dataset was split into three subsets, each of which contained 100, 200, and 300 comments. These subsets were used to test their system, which resulted in an average accuracy score of 0.689.

Das et al. [34] carried out a study on detecting hate speech in Bengali and Romanized Bengali. They extracted samples from Twitter in order to gather the necessary information, producing a dataset with 5071 samples in Bengali and Romanized Bengali. They used a variety of training models in their study, including XML-RoBERTa, MuRIL, m-BERT, and IndicBERT. Following testing, they discovered that XML-RoBERTa had the highest accuracy, at 0.796.

Sazzed [35] collected 7245 YouTube reviews manually and divided them into two categories: vulgar and non-vulgar. The purpose of this process was to produce two benchmark corpora for assessing vulgarity detection algorithms. Following the testing of several methods, the bidirectional long short-term memory (BiLSTM) model showed the most promising results, achieving the highest recall scores for identifying vulgar content in both datasets.

Jahan et al. [36] created a dataset by using online comment scraping tools to collect comments from public Facebook pages, such as news and celebrity pages. SVM, random Forest, and AdaBoost were the three machine learning techniques used to categorize the comments for the detection of abusive content. Their approach, which was based on the random forest classifier, outperformed other methods in terms of accuracy and precision, scoring 0.7214 and 0.8007, respectively. AdaBoost, on the other hand, demonstrated the best recall performance, earning a score of 0.8131.

Ishmam et al. [37] collected a dataset sourced from Facebook, categorized into six distinct classes. The dataset was enriched with linguistic and quantitative features, and the researchers employed a range of text preprocessing techniques, including punctuation removal, elimination of bad characters, handling hashtags, URLs, and mentions, as well as tokenization and stemming. They utilized neural networks, specifically GRUs (gated recurrent units), alongside other machine learning classifiers, to conduct classification tasks based on the historical, religious, cultural, social, and political contexts of the data.

Karim et al. [38] used a combination of machine learning classifiers and deep neural networks to detect hate speech in Bengali. They analyzed datasets containing comments from Facebook, YouTube, and newspaper websites using a variety of models, including logistic regression, SVM, CNN, and Bi-LSTM. The researchers divided hate speech into four distinct categories: political, religious, personal, and geopolitical. With F1 scores of 0.78 for political hate speech, 0.91 for personal hate speech, 0.89 for geopolitical hate speech, and 0.84 for religious hate speech detection in the Bengali language, their results showed satisfying performance.

Sazzed [39] created a transliterated corpus of 3000 comments from Bengali, 1500 of which were abusive and 1500 of which were not. As a starting point, they used a variety of supervised machine learning methods, such as deep learning-based bidirectional long short-term memory networks (BiLSTM), support vector machines (SVM), logistic regression (LR), and random forest (RF). The SVM classifier displayed the most encouraging results (with an F1 score of 0.827 ± 0.010) in accurately detecting abusive content.

User comments from publicly viewable Facebook posts made by athletes, officials, and celebrities were analyzed in a study by Ahmed et al. [40]. The researchers distinguished between Bengali-only comments and those written in English or a mix of English and other languages. Their research showed that 14,051 initial comments in total, or approximately 31.9% of them, were directed at male victims. However, a significant number of the 29,950 comments, or 68.1% of the total, were directed at female victims. The study also highlighted how comments were distributed according to the different types of victims. A total of 9375 comments were directed at individuals who are social influencers. Among these, 5.98% (equivalent to 2633 comments) were aimed at politicians, while 4.68% (or 2061 comments) were focused on athletes. Additionally, 6.78% (about 2981 comments) of the comments were centered around singers, and the majority, which is 61.25% (totaling 26,951 comments), were directed at actors.

For the classification of hate speech in the Bengali language, Romim et al. [41] used neural networks, including LSTM (long short-term memory) and BiLSTM (bidirectional LSTM). They used word embeddings that had already been trained using well-known algorithms such as FastText, Word2Vec, and Glove. The largest dataset of its kind to date, the extensive Bengali dataset they introduced for the research includes 30,000 user comments. The researchers thoroughly compared different deep learning models and word embedding combinations. The outcomes were encouraging as all of the deep learning models performed well in the classification of hate speech. However, the support vector machine (SVM) outperformed the others with an accuracy of 0.875.

Islam et. al. [42] used large amounts of data gathered from Facebook and YouTube to identify abusive comments. To produce the best results, they used a variety of machine learning algorithms, such as multinomial naive Bayes (MNB), multilayer perceptron (MLP), support vector machines (SVM), decision tree, random forest, and SVM with stochastic gradient descent-based optimization (SGD), ridge classifier, and k-nearest neighbors (k-

NN). They used a Bengali stemmer for preprocessing and random undersampling of the dominant class before processing the dataset. The outcomes demonstrated that, when applied to the entire dataset, SVM had the highest accuracy of 0.88.

In their study, Aurpa et al. [43] used transformer-based deep neural network models, like BERT [44] and ELECTRA [45], to categorize abusive comments on Facebook. For testing and training, they used a dataset with 44,001 Facebook comments. The test accuracy for their models, which was 0.85 for the BERT classifier and 0.8492 for the ELECTRA classifier, showed that they were successful in identifying offensive content on the social media platform.

Table 1. Research on vulgarity detection or related topics in Bengali (Facebook (F), YouTube (Y)).

Paper	Classifier	Highest Score	Language	Sample Size	Class and Ratio	Data Sources
[29]	Multinomial Naive Bayes, Random Forest, Support Vector Machines,	80% (Accuracy)	Bengali	2.5K	-	F
[30]	Support Vector Machines, Naive Bayes, Decision Tree, K-Nearest Neighbors	97% (Accuracy)	Bengali	2.4 K	Non-Bullying Bullying (10%)	F, T
[31]	Linear Support Vector Classification, Logistic Regression, Multinomial Naive Bayes, Random Forest Artificial Neural Network, RNN + LSTM	82.2% (Accuracy)	Bengali	4.7 K	Slang (19.57%), Religious, Politically, Positive, Neutral, violated (13.28%), Anti-feminism (0.87%), Hatred (13.15%), Personal attack (12.36%)	F, Y, News portal
[32]	Naive Bayes	80.57% (Accuracy)	Bengali	2.665 K	Non-Abusive, Abusive (45.55%)	Y
[33]	Root-Level approach	68.9% (Accuracy)	Bengali	300	Not Bullying, Bullying	F, Y News portal
[35]	Logistic Regression, Support Vector Machines, Stochastic Gradient Descent, Bidirectional LSTM	89.3% (F1 Score) 82.4% (F1 Score)	Bengali	7.245 K	Non Vulgar, Vulgar	Y
[36]	Support Vector Machines, RF, Adaboost	72.14% (Accuracy) 80% (Precision)	Bengali	2 K	Non Abusive, Abusive (78.41%)	F

Table 1. *Cont.*

Paper	Classifier	Highest Score	Language	Sample Size	Class and Ratio	Data Sources
[37]	Gated Recurrent Units, Support Vector Classification, LinearSVC, Random Forest, Naive Bayes	70.1% (Accuracy)	Bengali	5.126 K	Religious comment (14.9%), Hate speech (19.2%), Inciteful (10.77%), Communal hatred (15.67%), Religious hatred (15.68%), Political comment (23.43%)	F
[38]	Logistic Regression, Support Vector Machines, Convolutional Neural Network, BIdirectional LSTM, BERT, LSTM	78% 91% 89% 84% (F1 Score)	Bengali	8.087 K	Personal (43.44%), Religious (14.97%), Geopolitical (29.23%), Political (12.35%)	F
[39]	Logistic Regression, Support Vector Machines, Random Forest, Bidirectional LSTM	82.7% (F1 Score)	Bengali	3 K	Non abusive, Abusive (10%)	Y
[41]	Long Short-term Memory, Bidirectional LSTM	87.5% (Accuracy)	Bengali	30 K	Not Hate speech, Hate speech (33.33%)	F, Y
[42]	Multinomial Naive Bayes, Multilayer Perceptron, Support Vector Machines, Decision Tree, Random Fores, Stochastic Gradient Descent, K-Nearest Neighbors	88% (Accuracy)	Bengali	9.76 K	Non Abusive, abusive (50%)	F, Y
[43]	ELECTRA, Deep Neural Network, BERT	85% (Accuracy) (BERT), 84.92% (Accuracy) (ELECTRA)	Bengali	44.001 K	Troll (23.78%), Religious (17.22%), Sexual (20.29%), Not Bully (34.86%), Threat (3.85%)	F

Based on our comprehensive analysis of papers related to vulgarity detection and related topics like abusive and bullying detection, as well as detection in the low-resource language Bengali, several critical research gaps emerge. These gaps include the absence of a clear problem definition in some papers, the prevalence of small-sized datasets without a well-defined annotation process, and the lack of benchmarking efforts to assess dataset

quality. Additionally, class imbalance in datasets remains an issue, and limited attention has been given to vulgarity detection in low-resource language Bengali, with only a single work [35] addressing this area. Many papers fail to specify the source of their datasets and conduct limited experiments. Field surveys are often superficial or nonexistent. Furthermore, none of the papers considered ethical considerations in data collection, such as preserving user privacy through dataset anonymization. Addressing these research gaps is essential for advancing the field of vulgarity detection and related areas, ensuring the development of more robust, ethical, and well-defined detection systems.

Although many of the above-mentioned studies focus on the detection of bullying or hate speech, which often contain vulgar remarks, the presence of vulgarities specifically in the Chittagonian dialect of Bangla has not previously been investigated. By concentrating on information taken from posts on social media that were written in the Chittagonian dialect, this study seeks to close this gap. It is the start of an effort to accurately identify and gauge the frequency of vulgar language used in these social media posts.

3. Proposed Methodology

The experimental procedures are depicted in Figure 2, and the methodology is explained as follows:

3.1. Data Collection

Due to the lack of a good quality dataset designed specifically for vulgar text detection in the Chittagonian dialect, gathering data posed one major challenge in this study. The ML models need to be trained and tested on a sizable dataset in order to produce trustworthy results for classifying vulgar remarks. The data collection procedure used for the vulgar word detection in the paper is described as follows (see also Figure 3):

1. The dataset used in this study was made up of text excerpts from social media sites and open comment sections. A dataset with a wide range of topics, writing styles, and user demographics was intended to be both diverse and representative.
2. Facebook comments were manually gathered from a variety of independent sources, such as the public profiles and pages of well-known people.
3. Random sampling was used to guarantee a balanced and representative dataset. Each data source's popularity, user activity, and content suitability had to be taken into account when selecting random text samples from it. The objective was to gather a significant amount of information while keeping a variety of vulgar words and their context.

Figure 4 shows six examples from the dataset.

3.2. Data Annotation Process

Annotating data is required for the creation of machine learning models, such as those used to detect vulgar remarks. To properly train the model, the data must be labeled or annotated with relevant information. Below, we provide an outline of the data annotation process for vulgar remark detection.

Data annotators: Three native speakers of the Chittagonian dialect were hired, two of whom had Bachelor of Science degrees in engineering and one of whom had a master's degree in linguistics.

Experts responsible for preparing annotation standard and guidelines: Two people work in NGO organization [46], where they work on, among others, Internet-based surveys about harassment. One of them was a male, and the other one was a female, both with higher education (master's degree, sociology and social work).

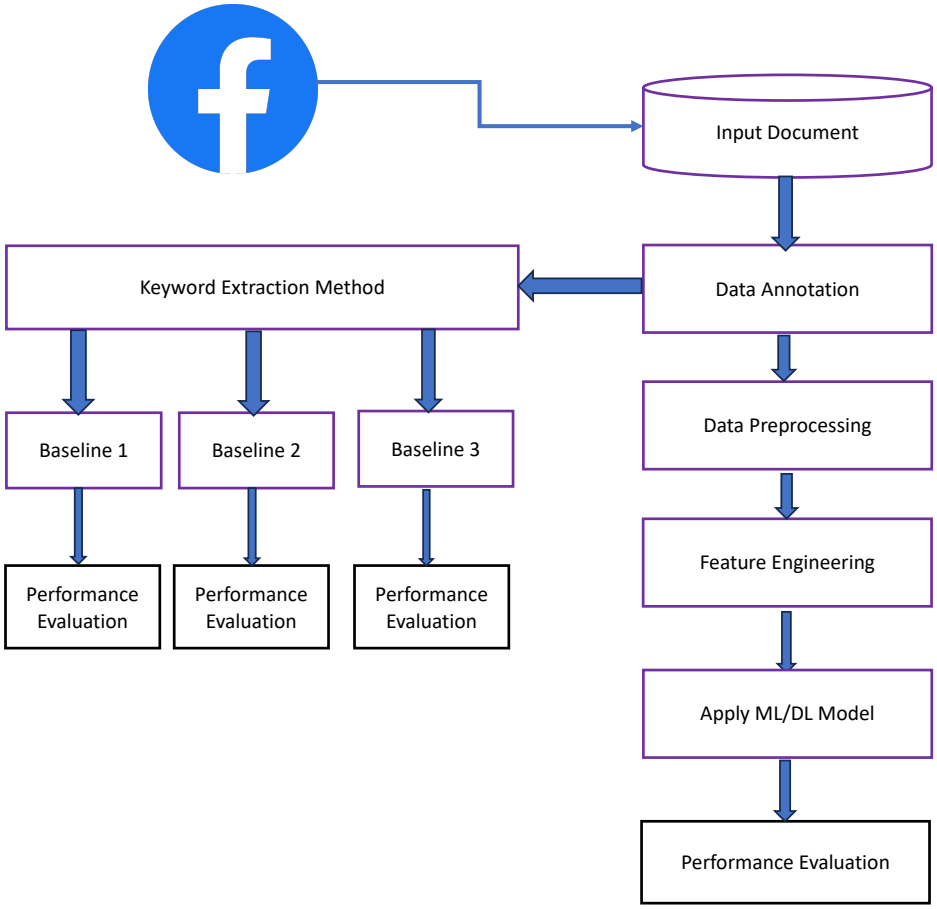

Figure 2. Outline of performed experiments.

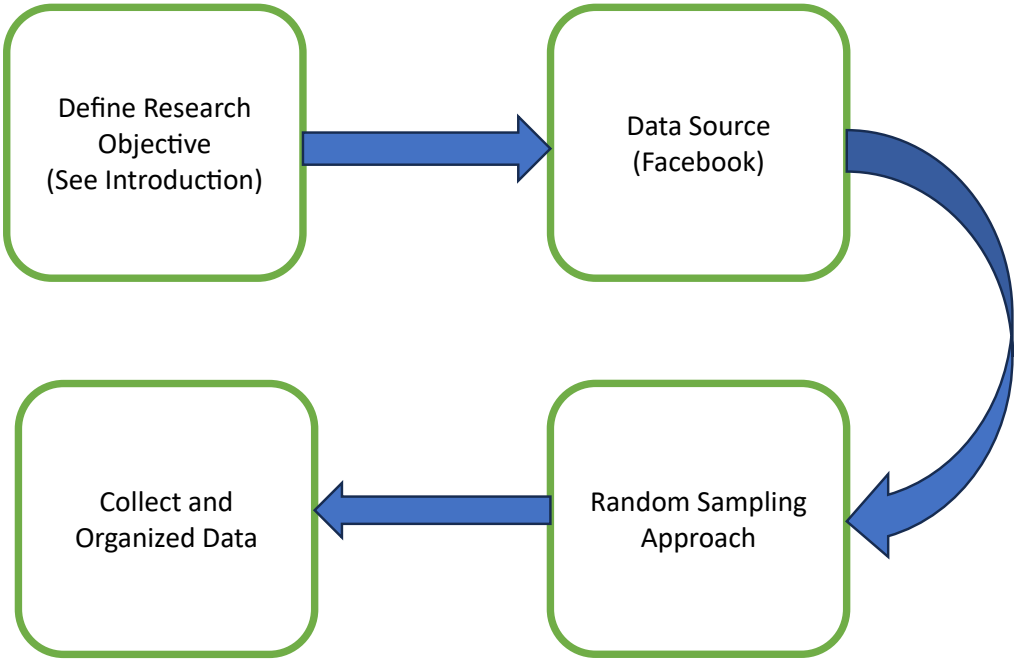

Figure 3. Steps that were used to collect and organize data.

Chittagonian Dialect	English Translation
অডা তুই শুয়োরের বাচ্চা ।	You are P*glet.
তোরা বেগ্গুন খানকির ফুয়া	You are all son of wh*res
বাংলাদেশত নতুন বেইশ্যাদেহা যার।	New pr*stitutes are appearing in Bangladesh.
মাগির দালাল শিমুল্লে	Shimulleh is a sl*t's broker
কুত্তার বাচ্চা	Son of a b*tch
সানার দুধ এক্কান ১০ কেজি	Sana's t*t is 10kg

Figure 4. Examples from the dataset.

3.2.1. Standard and Guidelines

To identify vulgarities, a dataset required annotations adhering to a predefined set of guidelines [47,48]. Therefore, prior to beginning the annotation process, clear and detailed annotation guidelines were developed. Using these guidelines, human annotators were taught how to identify and label vulgar remarks in the text. We provided definitions and usage examples of vulgar words and procedures for dealing with ambiguous cases of vulgar expressions in the guidelines.

The following are the essential standards and recommendations used in this study during the annotation process:

1. Definition of vulgar words: In this research, we defined vulgar words as unpleasant words such as *sl*t, motherf*cker, b*tch*, etc., from the Chittagong dialect of Bangla used to harass other people, institutions, groups, and society.
2. Severity scale: A number between 1–100 was assigned to each vulgar word from the Chittagong dialect by three language experts.
3. Annotator training: Three annotators were trained in the interpretation of vulgar words from the Chittagong dialect, so that the annotation process could be conducted properly. This includes training to maintain professional attitude towards the annotated text in all annotations. This includes avoiding any personal bias or judgment.
4. Consideration of context: Depending on the context, vulgar words can mean different things and offend people in different ways. The context of the message as well as any cultural or social elements that might affect how a vulgar word is perceived should be taken into account when annotating the text.
5. Evaluation of annotation integrity: All data annotations were evaluated for their integrity using inter-rater agreement measures like Cohen's Kappa [20] and Krippendorff's alpha [21].
6. Respect of privacy: Treat any personally identifying information in the annotated text in accordance with any applicable laws or policies and respect the individual's privacy.

3.2.2. Data Annotation Evaluation

The dataset comprised 2500 samples, with each sample being manually annotated following the process depicted in Figure 5. Initially, three annotators independently annotated each review, generating a total of 7500 judgments. In case of any disagreements among the annotators, a majority voting approach was employed to resolve them. As a result, the raw dataset included 1009 samples marked as vulgar and 1476 samples marked as non-vulgar. Additionally, 15 conflicting samples were identified during the annotation process, and after discussion with the annotators, these were excluded and discarded from the final dataset. An example of a conflict was, e.g., a sentence like মঙ্গল শোভাযাত্রা অমঙ্গল (English translation: "*Mangal shovajatra* (Mass procession) is inauspicious"). Three annotators gave three different judgments to this comment, i.e., the first judged this comment as non-vulgar, the second as judged this comment as vulgar, and the third could not reach a

decision. Three people have given three types of judgments on this comment by looking at the word **Mangal** from a different religious point of view. Since this was a more difficult problem, surpassing the notion of vulgarity, we decided to not include it in the study this time but will consider it for separate research in the future. Figure 6 displays some of the most typical vulgar words in the dataset.

3.2.3. Inter-Rater Agreement Evaluation

Cohen's Kappa:

After annotating the data, we conducted an examination of the inter-rater agreement. The analysis, employing Cohen's Kappa [20], revealed an impressive average value of 0.91 Kappa. This indicates very strong agreement among the annotators, as demonstrated in Table 2.

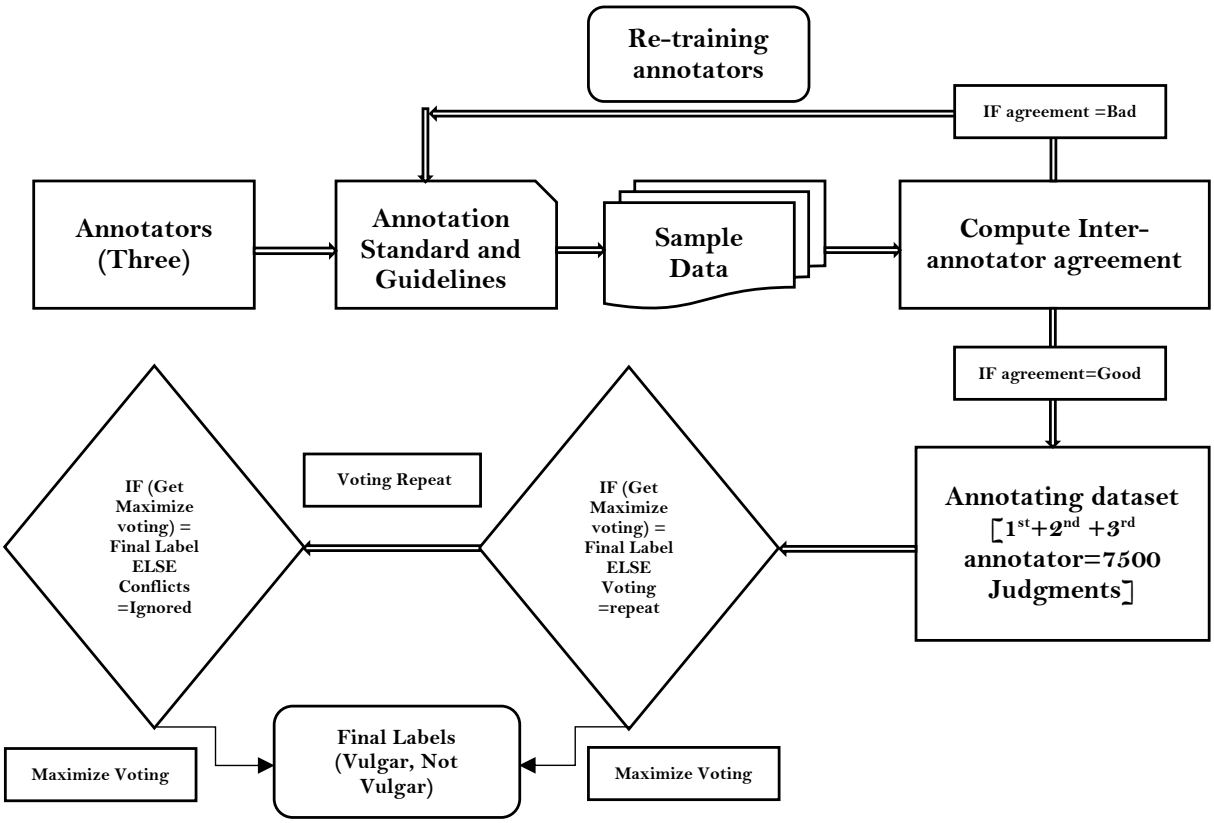

Figure 5. Data annotation process.

Chittagonian Dialect	English Translation	TF-IDF Score	Frequency
হানকি	Wh*re	0.003594	100
মাগি	Sl*t's	0.001797	40
হেডা	Female Genital Organ	0.001366	37
চুদি	F*ck	0.001078	35
মাদারচোদ	Mother F*cker	0.000791	34
চনু	P*nis	0.000647	20
চুইদ্দুম	F*cking	0.000575	18
বেশ্যা	H*oker	0.000575	18
দুধ	T*t	0.000431	14
কুত্তা	B*tch	0.000288	12

Figure 6. Top 10 most frequent vulgar words in the dataset.

Table 2. Cohen's Kappa score.

Annotator Pairs	Cohen's Kappa
1 and 2	0.92
1 and 3	0.90
2 and 3	0.91

Krippendorff's alpha: Using Cohen's Kappa score we can see that the annotators were in almost perfect agreement. To double-check the annotation agreements, we calculated the inter-rater agreement scores using Krippendorff's alpha [21]. As a result, we achieved an average agreement value of 0.914, demonstrating a significantly high agreement among annotators (refer to Table 3).

Table 3. Krippendorff's alpha.

Annotator Pairs	Krippendorff's Alpha
1 and 2	0.927
1 and 3	0.898
2 and 3	0.917

With the above two inter-rater agreements scores, we verified that there was a very high agreement between the annotators during the data annotation process. With the assurance of good quality guidelines and annotators with professional backgrounds, we can state that the created dataset can be considered high quality.

3.3. Baselines

It has been widely accepted that vulgar remark detection can be performed with sufficient accuracy using a basic keyword-matching method to find specific words or phrases of interest within a text [49]. By comparing the input text to a predetermined list of keywords, such a method searches for exact matches with a predetermined lexicon or a list of vulgar keywords [50]. Often, the input text is also preprocessed to remove any unnecessary words or information before keyword matching. Tokenization and removal of stop words or punctuation are a few potential strategies.

Numerous modifications of the keyword-matching method have been applied, including regular expressions [51], string matching algorithms [52], or pattern matching [53,54], all of which can be used to carry out the matching process. The algorithm analyzes each keyword in the text to determine whether it is used alone or as a component of a longer phrase.

Keyword matching has the advantage of being efficient, straightforward, and explainable [55]. Because the input text and keyword list are directly compared, the computational overhead is typically very low. The algorithm can handle a large number of keywords, making it suitable for tasks that call for the simultaneous identification of multiple specific terms.

However, one serious problem is that keyword lists are static, while the language on the Internet is constantly evolving. Therefore, the algorithm may have issues with newly emerging terms or terms that depend on context and are not on the keyword list. The keyword list, therefore, must be periodically updated if the method is to continue working effectively over time.

Despite these shortcomings, keyword matching is still a useful technique in many applications. It provides an efficient and flexible way to pick specific words or phrases of interest. When combined with other methods like machine learning, keyword matching can be a useful component of systems that perform more thorough content filtering [56].

To perform keyword matching, one first needs to prepare the list or lexicon of such keywords: in this case, a lexicon of vulgar words and phrases.

Although a variety of approaches can be designed for keyword extraction, such as rule-based linguistic approaches, statistical approaches, or even machine learning-based approaches, we can specify three general methods for extracting keywords or key phrases from text [55]. Namely, the required keywords can be extracted manually by a trained human designer or annotator, which assures high accuracy but is time-consuming and requires significant human effort. Additionally, the task of extracting vulgar words also poses a burden to the mental health of such human annotators. Secondly, the keyword list can be extracted fully automatically, which takes away all of the burden from the annotators. However, it is usually difficult to assure high accuracy of automatic extraction, since such methods rely on statistical properties of text (term occurrences, term frequencies, frequencies in the whole document, etc.). A third way is to improve the automatic extraction method to the point where it is as close to human judgment as possible and leave the remaining correcting work to the human annotator.

Consequently, in this paper firstly, we proposed a keyword-matching baseline method for vulgar remark detection, which we based on a vulgar keyword extraction. In the keyword-matching baseline, the lexicon of vulgar words provided to the method is considered as a list of features, while the matching procedure is treated as classification in the sense that if at least one vulgar word from the lexicon is matched in the input sentence, the sentence is considered vulgar.

Since the baseline method for classification is only based on simple keyword matching, we compared the three above-mentioned methods for keyword extraction to evaluate which of the keyword extraction methods would be the most effective and efficient and if we could find a method with sufficiently high efficiency and efficacy. For efficiency, we consider the amount of human effort put into preparing the lexicon, while for efficacy, we consider the method's accuracy in reclassifying the sentences into either vulgar or non-vulgar. The whole process of keyword extraction and comparison of all three methods is shown in Figure 7.

Firstly, in the purely manual extraction method, the human annotators read all sentences and manually extracted relevant vulgar expressions. This resulted in a total of 1010 vulgar words. The total accuracy was 0.648. Next, we aimed to propose a method capable of fully automatic (no human effort required) or semi-automatic (only a limited human effort required comparing to fully manual method) extraction of vulgar keywords.

Firstly, to initially extract vulgar keyword candidates, we applied *TF-IDF* and probability of occurrence. Here, *TF*, or term frequency, is calculated by dividing the occurrences of a specific word (or "term") in a document by the number of all terms in that document [57], as in Equation (1) .

$$TF = \frac{\text{Frequency of a certain word in the document}}{\text{Word count in the entire document}} \tag{1}$$

Next, *IDF*, or inverse document frequency, determines the importance of keywords in a text and is calculated as a logarithmically scaled inverted division of the number of documents containing the term and total number of documents, as shown in Equation (2).

$$IDF = \log_2\left(\frac{\text{Total documents}}{\text{Documents with a particular term}}\right) \tag{2}$$

Finally, in order to generate the *TF-IDF* measure, the *TF* and *IDF* are multiplied, as in Equation (3).

$$TF\text{-}IDF = TF \times IDF \tag{3}$$

By calculating *TF-IDF* for all words in the two groups of sentences (vulgar and non-vulgar) we obtain the list of words, where the higher *TF-IDF* score for the vulgar group

represents a higher probability of the word being the most representative of the vulgar group, which by assumption should be equivalent to the word being potentially vulgar. In the evaluation, we test the efficacy of this purely TF-IDF-based method without any additional filtering. However, in reality it is not always true, and many non-vulgar words also become included in the list. Specifically, one can observe how many actual vulgar words are included in the first ten, twenty, etc., words on the list, as represented in the result and discussion section.

To solve this problem, we added an additional method to delete the words that have the highest probability of being non-vulgar from the list of top TF-IDF vulgar word candidate terms. The method is explained as follows. The idea for this step was borrowed from Ptaszynski and Yagahara's (2021) method for the automatic extraction of technical terms from larger corpora [58].

To perform this, firstly, we calculate the probability of occurrence (PoO) for each word in either vulgar or non-vulgar class according to Ptaszynski et al.'s (2019) pattern extortion method [54], represented in Equation (4), which is a simplified sigmoid function normalizing the weighted score between 1 (completely vulgar) and −1 (completely non-vulgar).

$$PoO = \left(\frac{\text{Occurrence of Vulgar word}}{(\text{Occurrence of Vulgar word} + \text{Occurrence of Non-vulgar word})} - 0.5 \right) * 2 \quad (4)$$

From that list, we then take all words for which the weight was −1 (appeared only in the non-vulgar group). We use this list to additionally filter out potential non-vulgar terms which might appear on the list of TF-IDF extracted terms. In this manner, we can to some extent automatically eliminate potential non-vulgar words included in the TF-IDF lists by mistake. To test the coverage and usability of this method, we evaluate to what extent were the non-vulgar words eliminated from the list by looking at the ratio of the number of vulgar words in all automatically extracted words. We also verify this for various extraction spans, namely, top ten, top twenty, etc.

In this manner, we end up with three baseline methods for vulgar remark detection based on simple keyword matching, each based on a different keyword extraction procedure, as follows.

1. Automatic keyword extraction method with no additional filtering of non-vulgar words,
2. Automatic keyword extraction method with manual filtering of non-vulgar words,
3. Automatic keyword extraction method with additional automatic filtering of non-vulgar words.

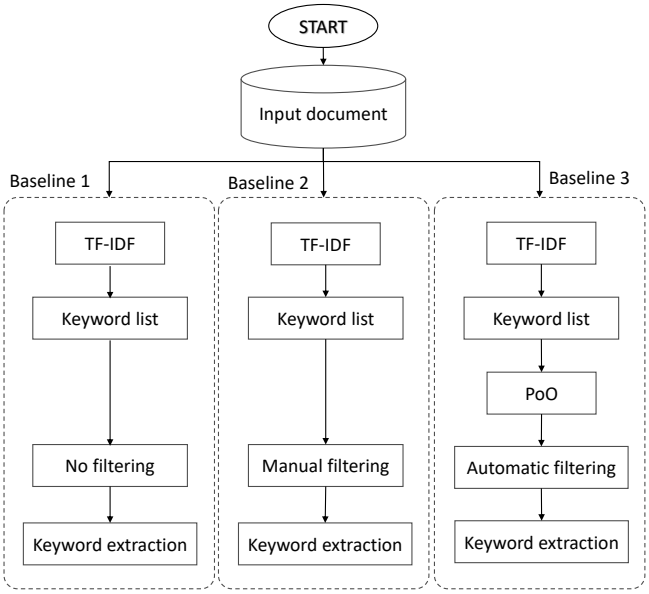

Figure 7. Mechanism of vulgar keyword extraction method.

3.4. Data Preprocessing

Apart from the baselines, we also applied classic machine learning (ML) algorithms to classify vulgar remarks. However, the ML algorithms required a number of specific data preprocessing and feature engineering techniques to be applied before the classification.

Regarding data preprocessing, it is important to consider that the dataset comprises SNS comments, which may contain a significant amount of irrelevant information for the analysis. To minimize the impact of such unwanted and redundant features, we have carried out the following data preprocessing steps. Figure 8 shows each preprocessing step.

Removing punctuation: When punctuation is removed from a text, all quotation marks and other special characters are also removed. Examples of punctuation used to denote pauses, emphasis, and other grammatical functions in written language include periods, commas, question marks, exclamation points, hyphens, parentheses, quotation marks, and other non-alphanumeric characters. The text data used in this study include a variety of punctuation and special characters, some of which may not have a discernible effect on the meaning of a sentence. In earlier research, Mahmud et al. [59] showed that removing punctuation can improve text classification, including automatic cyberbullying detection, especially when using traditional ML algorithms. As a result, we also adopted this strategy and removed all punctuation from the text under analysis. Therefore, we also followed this approach and eliminated all punctuation from the analyzed text. The list of punctuation we considered for removal includes the following characters: ’ ’’!#()*+,-./:;<=>?@[]‘| , etc.

Removing emoji and emoticons: Eliminating these graphical representations of emotions, expressions, or symbols from a text involves using emoji and emoticon removal. Emoticons and emoji are frequently used in written communication to convey emotions and reactions or to provide additional context in online communications. There are two ways to deal with this type of information: either completely removing them from the text or replacing them with corresponding text representations. While emojis have been shown to aid text classification in some cases [60], this research primarily focuses on testing the baseline performance of simple ML classifiers on the dataset. Hence, we opted not to consider emojis and emoticons during the classification process.

Removing English characters: Despite the fact that in Bangladesh Bengali is the primary official language, it is common for people to incorporate English words into their speech. However, since we were only focused on the Chittagonian dialect for this task, we eliminated any English characters. We did this in order to make sure that our analysis of the Chittagonian language was precise and focused.

Removing English digits: Upon thorough examination of the dataset samples, we observed the presence of digits and numbers that did not carry specific semantic meaning. In standard practice, named entity recognition (NER) [61] tools are employed to identify and categorize such numerical entities, such as phone numbers, percentages, and currencies. However, for the Chittagonian dialect, no NER tool is currently available. To address this limitation and facilitate the initial experiment, we opted to remove the digits and numbers from the dataset. Nevertheless, we acknowledge the importance of handling numerical entities accurately in future experiments. Therefore, our future plans include developing a dedicated NER tool specifically designed for Chittagonian. This tool will significantly enhance the processing capabilities and enable more effective handling of such cases in subsequent research and applications.

Removing stopwords: Stopwords are frequently eliminated when text data are processed for tasks involving natural language processing. Stop words are frequent words in a language that do not influence the meaning of a sentence. In the Chittagonian dialect, such words include, e.g., 'কি(What)','তুই(You)','ইতি(Him)', 'তুঁই(You)', 'তোর(Yours)' 'তোয়ারে(You)' আরার/আরাঁর(Our),কেইনে(How)', etc. They are often removed in various natural language processing tasks, like text classification, to reduce the dataset's dimensionality [62]. Therefore, in order to decrease the dimensionality of the text data and increase the effectiveness and efficiency of the subsequent analysis, we opted to eliminate stopwords from the dataset.

Tokenization: The process of tokenizing involves separating the text into tokens, or individual words. To facilitate further analysis for vulgar remark detection in the Chittagong dialect, we tokenized the text into useful linguistic units, such as words or subwords similarly to other previous research studying the Chittagonian dialect [59].

3.5. Feature Engineering

The transformation of lexical features (words) into numerical representations is necessary to enable the application of machine learning algorithms on textual data. In order to perform this, in this research we used four different feature extraction techniques, including CountVectorizer, *TF-IDF* Vectorizer, Word2vec, and fastText. We also followed the methodologies used in previous studies [31,35,59,63]. We successfully converted the textual data, which initially consisted of strings of characters (words), into numerical features using these feature extraction techniques.

3.5.1. Count Vectorizer

In tasks involving natural language processing, the widely used text preprocessing method CountVectorizer is employed [64]. A group of text documents are transformed into a matrix that shows the frequency of each word's (term's) occurrence in each document. The matrix's columns represent distinctive terms in the corpus, while the rows represent documents. In order to convert text data into a numerical format suitable for machine learning algorithms and enable further analysis and modeling based on the term frequencies in the documents, we used the CountVectorizer function of the Scikitlearn library [65].

3.5.2. TF-IDF Vectorizer

Information retrieval and natural language processing tasks often use the TF-IDF Vectorizer [66], a popular text processing method. It converts a collection of text documents into numerical feature vectors, where each feature denotes the weight of a term in a given document in relation to the corpus as a whole. It determines the term frequency (TF) and inverse document frequency (IDF) for every term in the documents, multiplies them, and then produces the *TF-IDF* score, which is a numerical representation appropriate for machine learning algorithms and other statistical analyses. In addition to assisting with a variety of text-based tasks, such as document classification, this vectorization process [63] also helps us to capture the significance of terms within documents. In this research, we specifically used the implementation of TF-IDF from the Scikit learn library [67].

3.5.3. Word2vec

By representing words as dense vectors in a high-dimensional space, Word2vec [68] is a popular classic word embedding technique. It is a shallow neural network model that learns to translate words into continuous vector spaces based on their patterns of co-occurrence in a significant body of text. Words with similar meanings can be placed closer together in the vector space thanks to the word embeddings that are created. This enables tasks related to word similarity, analogies, and text classification, among other natural language processing operations [69].

3.5.4. fastText

The widely used fastText library [70] was created by Facebook AI Research [71] for text representation and classification [72]. It is especially helpful for dealing with out-of-vocabulary words because it uses word embeddings and character n-grams to effectively encode words and capture subword information. As a result of the model's quick execution and scalability, massive text datasets can be trained and inferred with efficiency. It is widely used in the research and business communities thanks to its success in a variety of natural language processing tasks, such as text classification and language modeling.

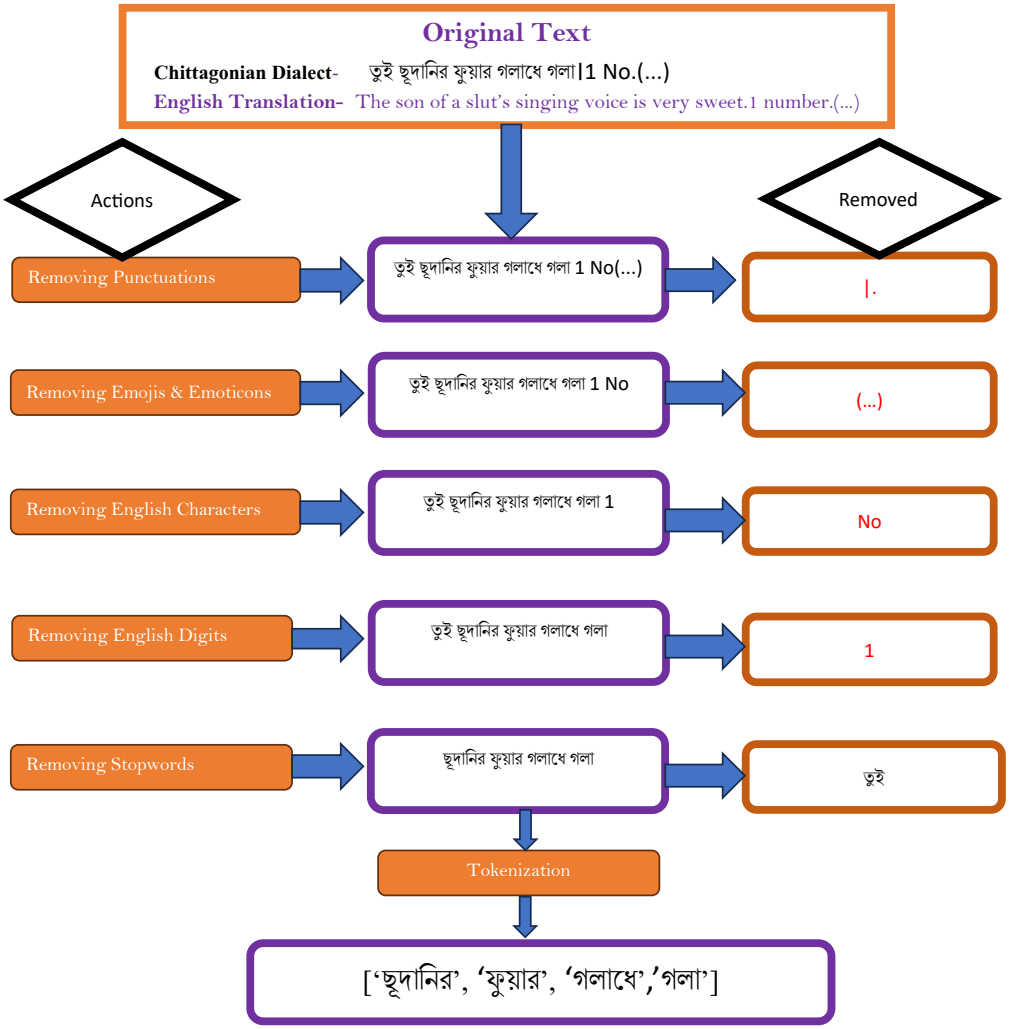

Figure 8. Step-by-step data preprocessing.

3.6. Classification

We investigated two deep learning (DL) algorithms and five machine learning (ML) algorithms in total during our experimentation. We used logistic regression (LR), support vector machine (SVM), decision tree (DT), random forest (RF), and multinomial naive Bayes (MNB) for the traditional ML approaches. We also used the long short-term memory network (LSTM) and the simple recurrent neural network (simpleRNN) in the context of deep learning.

3.6.1. Logistic Regression

A supervised machine learning algorithm called logistic regression (LR) is employed for binary classification tasks [63]. By fitting a logistic function, it models the association between a dependent binary variable and one or more independent variables. The logistic function converts the input features into probabilities, which are then used to categorize instances into one of the two classes. LR is widely used for a variety of applications, including cyberbullying detection [59], text classification [73,74], and sentiment analysis [75] because of its simplicity, interpretability, and efficiency.

3.6.2. Support Vector Machines

Generally used for classification tasks [59], a support vector machine (SVM) is a supervised machine learning algorithm. It looks for the best hyperplane to divide data

points into distinct classes in a high-dimensional space. Due to SVM's robustness and effectiveness in handling complex decision boundaries, the margin (distance) between the closest data points of various classes is maximized. When the data cannot be linearly separated in the original feature space, it can also use kernel functions to transform the data into higher dimensions, allowing for effective classification [76].

3.6.3. Decision Tree

A popular supervised machine learning algorithm called a decision tree (DT) is often used for classification [59] and regression tasks. In order to produce a tree-like structure of choices, it operates by recursively partitioning the data into subsets according to the feature values. In classification or regression, each leaf node stands in for a class label while each internal node represents a feature test. Decision trees are used in many different applications because they can handle both categorical and numerical data [50].

3.6.4. Random Forest

An algorithm for collective learning called random forest (RF) is used for both classification [59] and regression tasks. In order to create more precise and reliable predictions, it builds multiple decision trees during training. In order to lessen overfitting and improve generalization, each tree is constructed using a random subset of features and data samples. Random forest creates a strong and adaptable machine learning model by combining the predictions of individual trees, which is widely used for numerous real-world applications [77].

3.6.5. Multinomial Naive Bayes

The probabilistic machine learning algorithm known as multinomial naive Bayes (MNB) is frequently employed for text classification tasks [59]. Based on the Bayes theorem [78,79], it makes the assumption that features are conditionally independent given the class label. MNB performs well when using features that represent word counts or term frequencies in the context of text classification. Despite its simplicity, MNB frequently performs surprisingly well in tasks like sentiment analysis and document categorization [80].

3.6.6. Simple Recurrent Neural Network

To process sequential data, such as time series or text, a simple recurrent neural network (RNN) is a type of neural network architecture [81]. It has a feedback loop that enables information to endure over time and can deal with inputs of varying length. A basic drawback of a simple RNN is the vanishing gradient problem [82], which makes it difficult for it to recognize long-term dependencies in sequences. Due to this problem, more sophisticated RNN variants, such as long short-term memory (LSTM) [83] and gated recurrent unit (GRU) [84], were created in order to solve the vanishing gradient issue and enhance performance on sequential tasks.

3.6.7. Long Short-Term Memory Network

Recurrent neural network (RNN) architectures with long short-term memory (LSTM) are made to handle sequential data [85]. It fixes the vanishing gradient issue [82] that prevents conventional RNNs from detecting long-range dependencies in sequences. For tasks like natural language processing, LSTMs use specialized memory cells with input, output, and forget gates that allow them to retain and forget information over time.

3.7. Performance Evaluation Metrics

Model evaluation involves assessing the performance of a model on test data. In this study, the following evaluation metrics were used: *precision* (*PRE*), *recall* (*REC*), *F1-score* (*F1*), *and accuracy* (*ACC*). These metrics are computed based on the counts of *true positives*

(*TPv*), *false positives* (*FPv*), *true negatives* (*TNv*), and *false negatives* (*FNv*). The calculations for these metrics are given by Equations (5)–(8).

$$Accuracy(ACC) = \frac{TPv + TNv}{TPv + TNv + FPv + FNv} \tag{5}$$

$$Precision(PRE) = \frac{TPv}{TPv + FPv} \tag{6}$$

$$Recall(REC) = \frac{TPv}{TPv + FNv} \tag{7}$$

$$F1 - score(F1) = 2 \times \left(\frac{Precision(PRE) \times Recall(REC)}{Precision(PRE) + Recall(REC)} \right) \tag{8}$$

In these equations:

True positives (*TPv*) are positive instances that were accurately predicted as positive. *False positives* (*FPv*) are instances that were predicted as positive when the actual label is negative. Observations that were accurately classified as negative are known as*true negative* (*TNv*) observations. Observations that were incorrectly classified as negative when they actually belonged to the positive class are known as *false negatives* (*FNv*).

4. Results and Discussion

4.1. Discussion on Performance of Keyword-Based Vulgarity Extraction and Classification Baselines

In this section, we present the results and analysis of three different methods for vulgar word extraction using keyword matching described in Section 3.3 and Figure 7. These methods involve various approaches to filtering non-vulgar words and determining the relevance of extracted words based on *TF-IDF* scores and probability of occurrences. The goal of this proposed method was to evaluate the effectiveness of these methods in identifying and extracting vulgar words from a given text.

4.1.1. Discussion on Ratio of Vulgar Words Extracted with TF-IDF Weighting

To estimate the potential of the method for automatic extraction of vulgar words, we first calculated the *TF-IDF* scores for each word in both the vulgar and non-vulgar parts of the dataset. Then, we manually determined the ratio of words that were actually vulgar within the top 10, 20, 30, etc., word spans from that list. As TF-IDF has the well-recognized potential to place words that are the most relevant for each compared group (here vulgar vs. non-vulgar) at the top of the list, this would show the accuracy of using only TF-IDF to extract vulgar words.

The results show that the method achieved the extraction ratios of 0.8, 0.75, 0.667, 0.6, 0.58, 0.583, 0.557, 0.525, 0.489, and 0.45 within the top 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 extracted words, respectively. The high accuracy at the beginning of the list indicates that the majority of extracted words with high *TF-IDF* scores were indeed vulgar. However, the lower accuracy in the later word spans suggests roughly half of the extracted words, although being statistically relevant to the vulgar group, were indeed not vulgar. Table 4 and Figure 9 represent the extraction accuracy for a wider span up to the first thousand words extracted with TF-IDF. As one can see, at the end of the list, only about sixteen percent of the extracted words (i.e., 161 in 1000 words precisely) were indeed vulgar. This suggests that using only TF-IDF without any additional filtering of non-vulgar words will not yield high scores and will not be practical in the long run. Therefore, we needed to improve the extraction method with an additional algorithm for filtering out the words that were most certainly non-vulgar from the group extracted with TF-IDF.

4.1.2. Baseline 1: Keyword-Matching Method Based on TF-IDF Term Extraction with No Additional Filtering

To verify the practical usability of the automatic vulgar term extraction method, we applied the automatically extracted terms as a lexicon (word list) in a simple keyword matching-based method for vulgar sentence detection.

In this method, we performed keyword matching without any additional filtering of the terms extracted with the TF-IDF scores. This means that in the list of the extracted terms there might be some non-vulgar words. As vulgar remarks can be expressed without specifically vulgar terms, this could either improve or impair vulgar sentence detection.

The results indicated an accuracy of the detection of vulgar sentences at levels of 0.197, 0.236, 0.255, 0.269, 0.280, 0.293, 032, 0.343, 0.352, and 0.36 for detection when only the words from the top 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 of the list of words extracted with TF-IDF (See Table 4 and Figure 9 for reference) were used, respectively. The accuracy suggests that the method successfully identified a significant number of vulgar sentences from the dataset. Interestingly, even using only the top 10 words allowed for a close to 20% accuracy. This is even more impressive if we acknowledge that some of those words were not specifically vulgar.

However, the overall low accuracy, reaching only 62%, indicates that (1) some non-vulgar sentences were incorrectly matched to the list of automatically extracted vulgar term keyword candidates (false positives), and (2) many vulgar sentences were not matched due to the limitations of purely keyword-matching-based method. This shows that there is a wide range of vulgar sentences where the vulgar meaning is not expressed with any specifically vulgar terms. Moreover, although the keyword-matching-based methods are advantageous in terms of processing speed, they have limited applicability, which confirms similar findings from previous research [86–88].

4.1.3. Baseline 2: Keyword-Matching Method Based on TF-IDF Term Extraction with Only Manual Filtering

In this method, after extracting the vulgar keyword candidates automatically using the TF-IDF, we filtered out the non-vulgar words manually. This was not a difficult task for the first several spans (top 10, 20, up to around 100 words), but as the extraction list became longer, it became apparent that continuing this task in the future would be time-consuming and unpractical, especially in the future when novel vulgar words are added to the everyday Internet vocabulary.

Despite the impracticality, the keyword-matching method based on this manually filtered list achieved somewhat satisfying results. For the lower spans of the top 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100, the accuracy values were 0.211, 0.284, 0.327, 0.362, 0.381, 0.41, 0.432, 0.467, 0.489, and 0.523, respectively (see Table 4 and Figure 9). Even so, much higher accuracy for the longer word list was achieved, reaching even 73% for the first 1000 automatically extracted words, with 161 actually vulgar words. This can already be considered applicable in practice, especially since the number of vulgar words is very low. Achieving over 70% accuracy suggests that a large majority of vulgar sentences can in fact be detected with simple keyword-matching-based methods. The only problem remaining for this method thus would be the automation of the additional filtering of non-vulgar words to decrease the necessity of human effort in updating this method in the future, which refers to the final baseline method proposed in the following section.

Table 4. Ratios of extracted vulgar terms, automatic filtering, and performance of the keyword-matching baselines (no. of non-vulgar words deleted manually separately for each span (A); no. of non-vulgar words deleted by humans cumulatively (B); no. of vulgar words (C); ratio of vulgar words in extracted words (D); keyword-matching accuracy with no additional filtering of non-vulgar words (E); keyword-matching accuracy with manual filtering of non-vulgar words (F); keyword-matching accuracy with automatic filtering of non-vulgar words (G); no. of non-vulgar words deleted by automatic filtering separately for each span (H); no. of non-vulgar words deleted by automatic filtering cumulatively (I); accuracy of the automatic filtering method in the automatic filtering of non-vulgar words (J).)

Top # Extracted Words	A	B	C	D	E	F	G	H	I	J
10	2	2	8	0.800	0.197	0.211	0.200	1	1	0.50
20	3	5	15	0.750	0.236	0.284	0.245	3	4	0.80
30	5	10	20	0.667	0.255	0.327	0.300	0	4	0.40
40	6	16	24	0.600	0.269	0.362	0.324	2	6	0.38
50	5	21	29	0.580	0.280	0.381	0.363	2	8	0.38
60	4	25	35	0.583	0.293	0.410	0.385	0	8	0.32
70	6	31	39	0.557	0.320	0.432	0.427	2	10	0.32
80	7	38	42	0.525	0.343	0.467	0.449	1	11	0.29
90	8	46	44	0.489	0.352	0.489	0.467	1	12	0.26
100	9	55	45	0.450	0.360	0.523	0.475	4	16	0.29
200	71	126	74	0.370	0.44	0.66	0.553	30	46	0.37
300	77	203	97	0.323	0.48	0.665	0.571	33	79	0.39
400	79	282	118	0.295	0.510	0.681	0.59	28	107	0.38
500	85	367	133	0.266	0.541	0.685	0.626	42	149	0.41
600	91	458	142	0.237	0.576	0.689	0.637	46	195	0.43
700	96	554	146	0.209	0.597	0.691	0.649	52	247	0.45
800	97	651	149	0.186	0.606	0.698	0.681	50	297	0.46
900	90	741	159	0.177	0.618	0.71	0.689	41	338	0.46
1000	98	839	161	0.161	0.620	0.73	0.695	68	406	0.48

4.1.4. Baseline 3: Keyword-Matching Method Based on TF-IDF Term Extraction with Automatic Filtering of Non-Vulgar Words

Finally, to check to what extent the TF-IDF term extraction can be improved automatically as well as to decrease the human effort, we applied an additional automatic filtering of non-vulgar words.

In this method, for keyword matching we used a predefined list of vulgar words with automatically filtered-out words that had no probability of occurrence in vulgar context, as explained in Section 3.3. The method achieved accuracies of 0.2, 0.245, 0.3, 0.324, 0.363, 0.385, 0.427, 0.449, 0.467, and 0.475 within the top 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 extracted words, respectively (see Table 4 and Figure 9). Moreover, for the longer word lists, the method, despite filtering out on average only half of the actually non-vulgar words, achieved accuracies close to purely human-based filtering. This suggests the following: (1) the automatic filtering method can reduce human effort by half, and at the same time, (2) when no additional human effort is applied, the method still achieves near-human-level accuracy all of the time. For example, for the longest checked word list, namely, 1000 automatically extracted vulgar word candidates with 406 non-vulgar words additionally filtered out automatically, the method achieved 95% of the human-level accuracy (0.695/0.73 accuracy). Even after normalizing this by considering Baseline

1 as a starting point, the automatic filtering method still covered 68% of human level. The normalized human level is calculated as follows: (Baseline 3 accuracy − Baseline 1 accuracy)/(Baseline 2 accuracy − Baseline 1 accuracy). However, for a slightly shorter word list, namely the top 800 words, with 149 vulgar words and 297 automatically filtered out non-vulgar words overall, the method achieved a performance at 98% of the human level and 82% of the normalized human level.

The accuracy score, in general, suggests sufficiently accurate identification of vulgar sentences after automatic filtering based on the probability of occurrence.

These results indicate that a combination of keyword matching with additional filtering techniques, such as threshold-based or probability-based filtering, can improve the performance of vulgar word extraction. Although the choice of extraction and filtering methods could depend on the specific requirements and trade-offs between the accuracy and efficiency of the given application, the usefulness of such a simple yet effective keyword-based method can be of value both for vulgar sentence detection and especially for the extraction of new vulgar words in the future.

Further research could explore more advanced techniques, such as machine learning- or deep learning-based approaches for keyword extraction [35] to optimize the filtering process and improve the accuracy of vulgar word extraction. Additionally, the evaluation of these methods on larger and more diverse datasets would provide a better understanding of their generalizability and robustness in real-world scenarios.

Consequently, in the following Sections 4.2 and 4.3 we introduce a detailed description of such machine learning and deep learning frameworks for vulgar word detection.

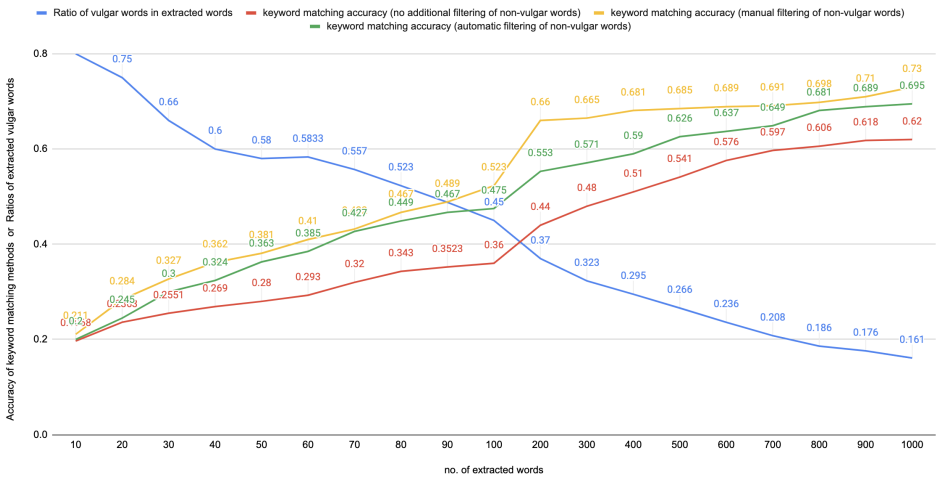

Figure 9. Comparison between different types of keyword-matching methods with the respective ratio of vulgar words in the lexicon.

4.2. Machine Learning Models for Vulgarity Detection

Detecting vulgar remarks in text data typically involves the application of machine learning (ML) and deep learning (DL) techniques. Figure 10 shows the main steps for detecting vulgar remarks.

In this study, we focused on detecting vulgarity in the Chittagonian language using classic ML algorithms. We used two different feature extraction techniques: CountVectorizer and *TF-IDF* Vectorizer. The dataset was divided into an 80–20 ratio for training and testing the models. In this data partitioning, a total of 1988 data points were designated for the training set, encompassing 80% of the entire dataset. The testing set, on the other hand, comprises 20% of the data, consisting of 497 data points.

Table 5 and Figure 11 presents the overall performance of the machine learning models using CountVectorizer. The logistic regression (LR) model performed the best, achieving the highest accuracy of 0.91. It also outperformed other models in this study and current

research [35] in terms of recall for both classes. The random forest (RF) model was the second-best performer with an accuracy of 0.87, while multinomial naive Bayes (MNB) demonstrated good precision and recall compared to LR.

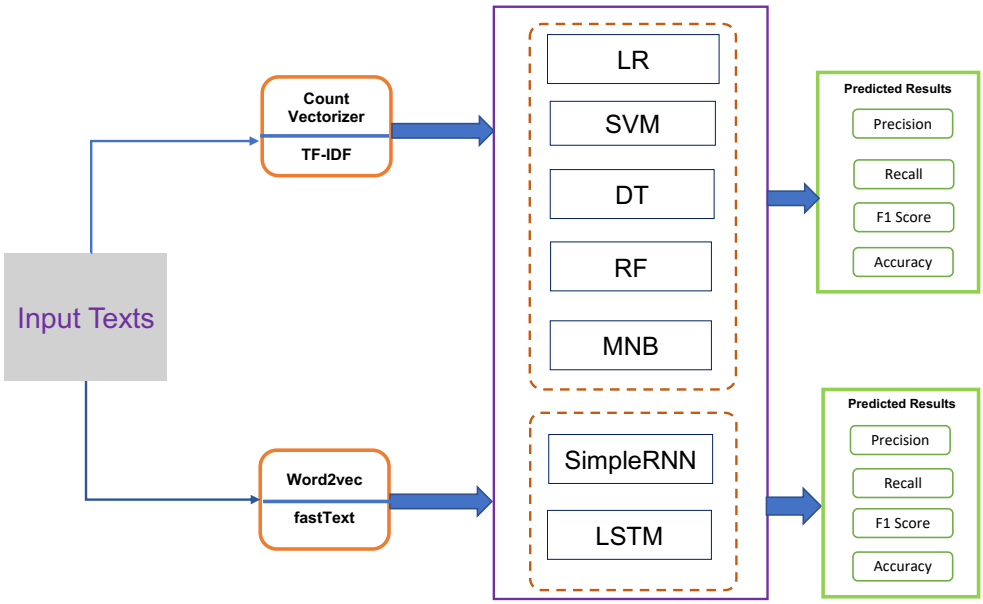

Figure 10. Layout of the experimental procedure for selecting optimal ML/DL model for vulgarity detection.

On the other hand, TF-IDF Vectorizer assesses the relevance of words within the dataset. LR also achieved the highest accuracy of 0.91 with good recall, and performed better than state-of-the-art methods [35]. MNB had an accuracy of 0.83, lower than RF, but it showed a balanced performance across other metrics, as depicted in Table 6 and the corresponding Figure 12.

Overall, the study suggests that using CountVectorizer with the LR model yields the best results for vulgarity detection in Chittagonian, while TF-IDF Vectorizer also performed well with LR and MNB models showing competitive performance.

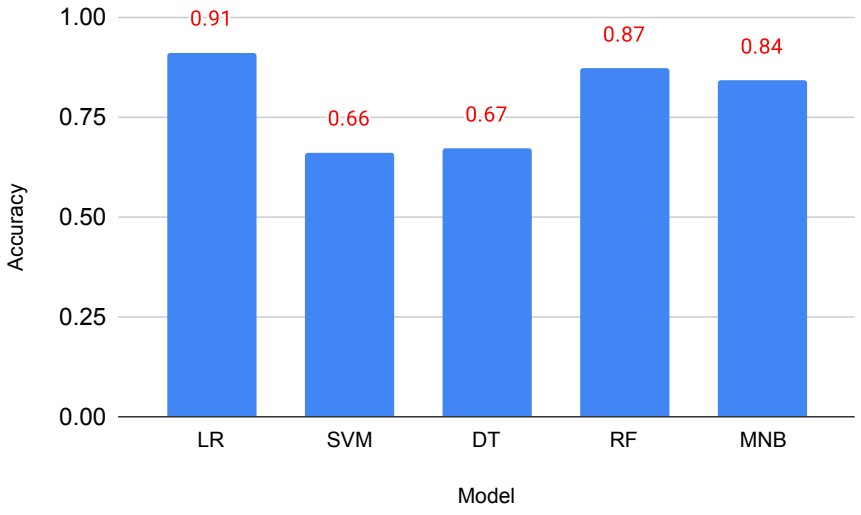

Figure 11. Model accuracy using CountVectorizer.

Table 5. Model performance utilizing CountVectorizer.

Model	Vulgar			Non Vulgar			*ACC*
	PRE	***REC***	***F1***	***PRE***	***REC***	***F1***	
Logistic Regression (LR)	0.800	0.921	0.860	0.910	0.761	0.833	0.910
Support Vector Machines (SVM)	0.654	0.721	0.682	0.680	0.600	0.631	0.660
Decision Tree (DT)	0.623	0.861	0.722	0.771	0.470	0.583	0.671
Random Forest (RF)	0.670	0.942	0.791	0.900	0.534	0.673	0.871
Multinomial Naive Bayes (MNB)	0.811	0.910	0.863	0.902	0.791	0.842	0.842

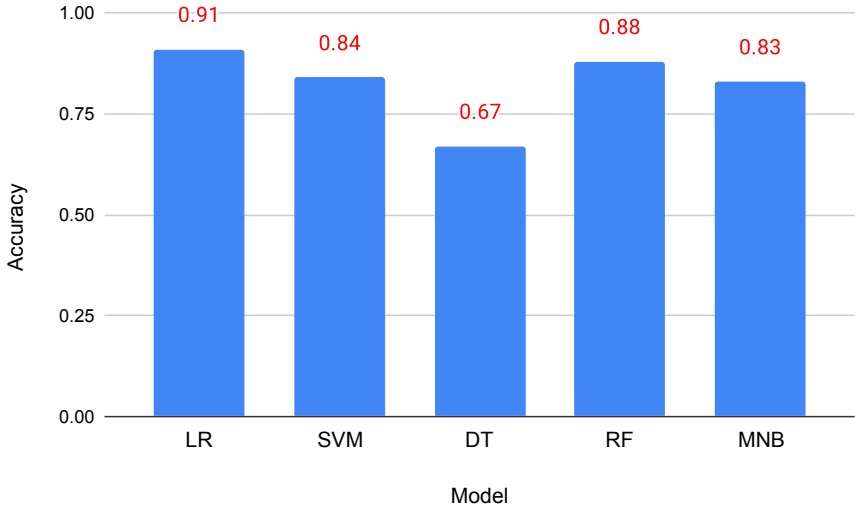

Figure 12. Model accuracy using TF-IDF Vectorizer.

Table 6. Model performance utilizing *TF-IDF* Vectorizer.

Model	Vulgar			Non Vulgar			*ACC*
	PRE	***REC***	***F1***	***PRE***	***REC***	***F1***	
Logistic Regression (LR)	0.820	0.921	0.870	0.901	0.802	0.853	0.911
Support Vector Machines (SVM)	0.810	0.890	0.853	0.881	0.792	0.832	0.843
Decision Tree (DT)	0.561	0.963	0.712	0.852	0.211	0.341	0.671
Random Forest (RF)	0.643	0.971	0.770	0.942	0.453	0.612	0.881
Multinomial Naive Bayes (MNB)	0.801	0.913	0.854	0.891	0.770	0.832	0.832

4.3. Deep Learning Models for Vulgarity Detection

The hyperparameter tuning section for the paper outlines the key choices made in configuring the neural network model, particularly the SimpleRNN and LSTM architectures. In this study, the hyperparameter tuning process was designed to optimize the performance of the text classification models. The first critical hyperparameter is the input dimension of the embedding layer, which directly corresponds to the size of the vocabulary in the corpus. The choice of this parameter is pivotal, as it determines the richness of the word representations. In our experiments, the input dimension was set based on the vocabulary size extracted from the dataset, ensuring that the model could effectively capture the lexical diversity present in the text data. Additionally, the output dimensions of the embedding layer were set to 64, determining the length of the word vectors. This value was selected through experimentation to strike a balance between model expressiveness and computational efficiency. The maximum length of a sequence was set to 100, aligning

with the nature of the text data, ensuring that sequences of text were standardized to this length during preprocessing. Another crucial set of hyperparameters pertained to the optimization process. We employed the Adam optimizer, a popular choice for gradient-based optimization, known for its adaptive learning rate capabilities. The learning rate, set at 0.001, is a hyperparameter that can significantly impact training dynamics. The choice of batch size was set to 32, balancing the trade-off between computational efficiency and model convergence speed. The training duration spanned ten epochs, allowing the model to iteratively update its parameters while monitoring convergence. Lastly, it is noteworthy that the dataset comprised 2500 data samples with balanced labels, ensuring that the model's performance was evaluated on a representative and unbiased dataset.

In this study, we explored the effectiveness of deep learning-based models, specifically RNN and LSTM, using Word2vec and fastText word embeddings for various NLP tasks. To ensure robust evaluation, we divided the dataset into three parts: the training set, the validation set (to check for overfitting), and the test set (to evaluate the model's performance). In this data split, a total of 1741 data points have been allocated to the training set, which constitutes 70% of the entire dataset. The validation set comprises 15% of the data, containing 372 data points, while the testing set also consists of 15% of the data, encompassing another 372 data points.

When employing Word2vec word embeddings, we observed that the SimpleRNN model outperformed the LSTM model and performed better than state-of-the-art methods [35], achieving an accuracy of 0.84 compared to LSTM's 0.63. Furthermore, SimpleRNN exhibited superior precision and recall for both classes, as illustrated in Table 7.

Moreover, we explored the application of fastText word embedding technique for both deep learning models. In this scenario, SimpleRNN achieved an impressive accuracy of 0.90, demonstrating its proficiency at the task. Notably, both models displayed excellent performance in detecting both classes, as shown in Table 7 and Figure 13.

4.4. Comparison with Keyword-Matching Baseline

Each of the four previously mentioned keyword extraction methods had an accuracy rating of roughly up to 70% with varying degrees of success. The results were influenced by the keyword list, the filtering methods, and the presence of vulgar word variations or misspellings. We could locate specific keywords within a text quickly by using these straightforward, effective keyword-matching techniques. However, the results have shown these methods also had problems with keyword variations and the need for frequent keyword list updating remained. Another problem not solvable by simple keyword-matching methods was the use of language that is context-sensitive. This shows that the ability of keyword-matching methods to handle complex linguistic structures or comprehend the context and meaning of the text is limited.

On the other hand, machine learning and deep learning methods leverage algorithms and models trained on labeled datasets to classify text segments as vulgar or non-vulgar. The results are typically measured using metrics such as accuracy, precision, recall, and F1-score. Machine learning and deep learning approaches achieved a high accuracy of approximately 84–91%, or roughly twenty percentage points above the keyword-matching baselines, which shows such algorithms can learn patterns and features from the data more effectively. They are also capable of capturing complex linguistic nuances, identifying context-sensitive vulgar words, and handling variations. A disadvantage of these methods is that they require significant amounts of labeled training data and computational resources for training and inference, which means that unless the training data are constantly updated, their performance will also degrade with time. However, this could be to some extent mitigated by (1) applying the automatic vulgar term extraction method proposed in this paper to update and expand the vulgar term lexicon of keyword-matching baselines and (2) using those baselines to collect potential vulgar and non-vulgar sentence candidates. This would allow for initial information triage [89] and ensure more efficiency in the vulgar remark data collection and annotation process.

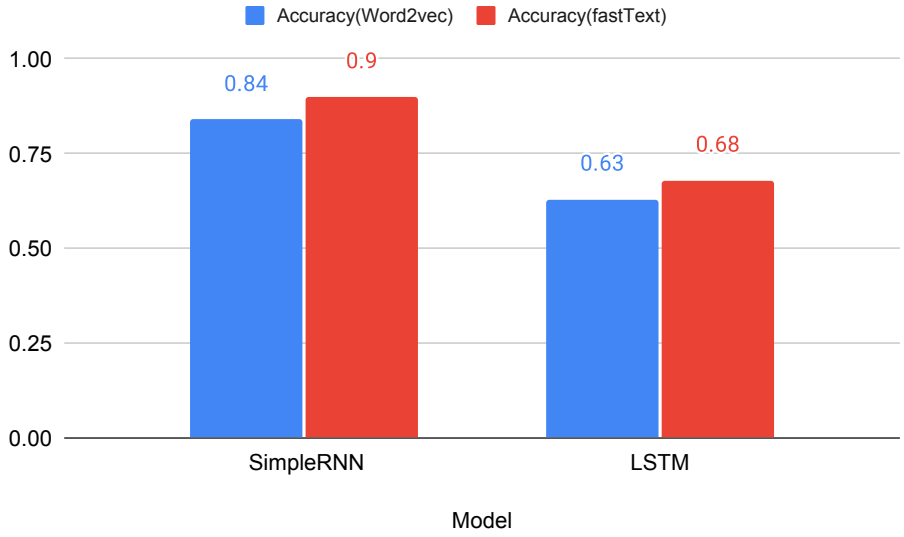

Figure 13. Model accuracy using Word2vec and fastText.

Table 7. Performance of models using Word2Vec and FastText.

Word2vec	**Vulgar**			**Non Vulgar**			***ACC***
	PRE	***REC***	***F1***	***PRE***	***REC***	***F1***	
SimpleRNN	0.784	0.983	0.863	0.972	0.704	0.812	0.842
Long Short-Term Memory(LSTM)	0.612	0.811	0.703	0.682	0.451	0.544	0.631
FastText	**Vulgar**			**Non Vulgar**			***ACC***
	PRE	***REC***	***F1***	***PRE***	***REC***	***F1***	
SimpleRNN	0.943	0.872	0.901	0.872	0.941	0.903	0.902
Long Short-Term Memory(LSTM)	0.632	0.893	0.744	0.792	0.452	0.573	0.681

4.5. Comparison with Previous Studies

To our knowledge, this is the first study to identify vulgar words in the Chittagonian dialect of Bangla. For that, we could not find any previous research to compare with our study. Since there was a study [35] to find vulgar words in the Bengali language, we compared it to our approaches. Below are the main differences and findings compared to the previous study to facilitate our research evaluation.

1. Our research language domain is the Chittagong dialect of Bangla, while previous work focused on Bengali/Bangla. Working with dialects has many challenges such as data collection, data annotation, data processing, dataset validation, model creation, etc. By overcoming all these challenges, we successfully completed the research.
2. We carried out the research in two steps. Firstly, we reported the performance of the three keyword-matching baselines. No previous research has tried this type of method.
3. Then, we built machine learning and deep learning models and compared them with baseline methods. We observed that our models gave comparatively better results then previous studies [35].

4.6. Limitations of Study

There were several limitations of this study: Firstly, the dataset size of 2500 comments, while comprehensive, is relatively small for training deep learning models, potentially limiting their generalizability. Secondly, the focus on the Chittagonian dialect of Bangla narrows the applicability of findings to Bangla and related languages. This means that the

results also need to be confirmed in other widely spoken languages and different language families. Additionally, the initial reliance on a lexicon-based approach for vulgar language detection may not effectively capture nuanced or context-specific variations. Resource constraints in deep learning models, like SimpleRNN, can also impact their competitive performance compared to traditional methods.

4.7. Ethical Considerations

Because this study concerns the human population (social media users), especially the unethical use of language on the Internet, an important part was using an ethical approach to research, beginning with data collection [90]. In order to collect data for this study, only public user posts and comments were collected. Facebook's open access policy permits data collection of public posts. Since we collected only public comments/posts, they were no longer regarded as private, and therefore, no special agreement was necessary for the collection of data and research [91]. Moreover, we adhered to at the following ethical concerns while collecting the data:

1. As the source for our dataset, we primarily used social media groups. Therefore, while gathering the data, we verified and complied with those groups' terms and conditions.
2. We performed anonymization of posts containing such sensitive information as names of private persons, organizations, religious groups, institutions, and states.
3. We deleted personal information such as phone numbers, home addresses, etc.

5. Conclusions and Future Work

5.1. Conclusions

This study primarily centered on identifying vulgar language in social media posts. As the common approach to vulgar remark detection is using simple lists of vulgar keywords, we firstly proposed a method to automatically extract such vulgar keywords from raw data and used those keywords in simple keyword-matching baseline classification methods. The automatic keyword extraction method was able to successfully extract vulgar keywords and additionally successfully filter out half of the non-vulgar words, which allowed the method to reach a satisfying approximately 70% accuracy in detecting vulgar sentences. Next, we also employed machine learning (ML) and deep learning (DL) classifiers, including logistic regression (LR), support vector machine (SVM), decision tree (DT), random forest (RF), multinomial naive Bayes (MNB), simple recurrent neural network (simpleRNN), and long short-term memory (LSTM). These classifiers were coupled with various feature extraction techniques like CountVectorizer, TF-IDF Vectorizer, Word2Vec, and fastText. Our dataset consisted of 2485 comments, balanced between vulgar and non-vulgar remarks. To evaluate the performance of the proposed methods, we ran experiments on this dataset. The results indicated that LR with CountVectorizer or TF-IDF Vectorizer, as well as simpleRNN with Word2Vec and fastText, were particularly effective in detecting vulgar comments.

5.2. Contributions of this Study

In summary, this research makes substantial contributions to the field of vulgar remark detection in the Chittagonian dialect, as follows:

1. Gathered a dataset of 2500 comments and posts from publicly accessible Facebook accounts.
2. Ensured dataset reliability through rigorous manual annotation and validated the annotations using Cohen's Kappa statistics and Krippendorff's alpha.
3. Introduced a keyword-matching-based baseline method using a hand-crafted vulgar word lexicon.
4. Developed an automated method for augmenting the vulgar word lexicon, ensuring adaptability to evolving language.
5. Introduced various sentence-level vulgar remark detection methods, from lexicon matching to advanced techniques.

6. Conducted comprehensive comparisons between keyword-matching and machine learning (ML) and deep learning (DL) models to achieve high detection accuracy.
7. Achieved over 90% accuracy in detecting vulgar remarks in Chittagonian social media posts, demonstrating a performance acceptable for real-world applications.

5.3. Future Work

Building upon the outcomes of this study, our future research directions will focus on devising resource-constrained strategies for vulgar remark recognition with a specific emphasis on the Chittagonian dialect. While the results of our current study are promising for vulgar remark detection in the Chittagonian dialect, there exist several avenues for further investigation and development. One crucial aspect involves expanding the size of our training dataset. A larger, more diversified dataset that encompasses a broader spectrum of vulgar words and contextual nuances will be meticulously gathered and annotated. This expansion aims to enhance the robustness and generalization capabilities of our models, enabling them to effectively identify vulgar language in a wider array of real-world scenarios.

Furthermore, our future research agenda will explore the realm of multi-modal approaches for vulgar remark detection. This entails analyzing not only textual content but also visual elements if available. By combining text and image analysis, we aim to gain a more profound understanding of vulgar remarks, especially in multimedia contexts where visual cues play a significant role. Additionally, we plan to advance our classification methods by incorporating more sophisticated machine learning techniques. Models such as bidirectional LSTM and transformers, renowned for their ability to capture intricate language patterns, will be leveraged to further elevate the accuracy and effectiveness of vulgar remark detection. This holistic approach to future research endeavors aims to refine and expand the capabilities of vulgar remark detection systems, ultimately contributing to a cleaner and safer online environment and more effective content moderation practices.

Author Contributions: Conceptualization, T.M. and M.P.; methodology, T.M.; validation, T.M., M.P. and F.M.; formal analysis, T.M. and M.P.; investigation, T.M. and M.P.; resources, T.M. and M.P.; data curation, T.M. and M.P.; writing—original draft preparation, T.M.; writing—review and editing, T.M. and M.P.; visualization, T.M.; supervision, M.P. and F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: All necessary informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data used to support the findings of this study are available upon reasonable request to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NLP	Natural Language Processing
ML	Machine Learning
DL	Deep Learning
RNN	Recurrent Neural Networks
BTRC	Bangladesh Telecommunication Regulatory Commission
NGO	Non-governmental Organization
TF-IDF	Term Frequency-Inverse Document Frequency
LR	Logistic Regression
SVM	Support Vector Machine
DT	Decision Tree

RF	Random Forest
MNB	Multinomial Naive Bayes
LSTM	Long Short-Term Memory network
BiLSTM	Bidirectional Long Short-Term Memory network
BERT	Bidirectional Encoder Representations from Transformers
ELECTRA	Pre-training Text Encoders as Discriminators Rather Than Generators

References

1. Bangladesh Telecommunication Regulatory Commission. Available online: http://www.btrc.gov.bd/site/page/347df7fe-409f-451e-a415-65b109a207f5/- (accessed on 15 January 2023).
2. United Nations Development Programme. Available online: https://www.undp.org/bangladesh/blog/digital-bangladesh-innovative-bangladesh-road-2041 (accessed on 20 January 2023).
3. Chittagong City in Bangladesh. Available online: https://en.wikipedia.org/wiki/Chittagong (accessed on 1 April 2023).
4. StatCounter Global Stats. Available online: https://gs.statcounter.com/social-media-stats/all/bangladesh/#monthly-202203-202303 (accessed on 24 April 2023).
5. Facebook. Available online: https://www.facebook.com/ (accessed on 28 January 2023).
6. imo. Available online: https://imo.im (accessed on 28 January 2023).
7. WhatsApp. Available online: https://www.whatsapp.com (accessed on 28 January 2023)
8. Addiction Center. Available online: https://www.addictioncenter.com/drugs/social-media-addiction/ (accessed on 28 January 2023).
9. Prothom Alo. Available online: https://en.prothomalo.com/bangladesh/Youth-spend-80-mins-a-day-in-Internet-adda (accessed on 28 January 2023).
10. United Nations. Available online: https://www.un.org/en/chronicle/article/cyberbullying-and-its-implications-human-rights (accessed on 28 January 2023).
11. ACCORD—African Centre for the Constructive Resolution of Disputes. Available online: https://www.accord.org.za/conflict-trends/social-media/ (accessed on 28 January 2023).
12. Cachola, I.; Holgate, E.; Preoţiuc-Pietro, D.; Li, J.J. Expressively vulgar: The socio-dynamics of vulgarity and its effects on sentiment analysis in social media. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 2927–2938.
13. Wang, N. An analysis of the pragmatic functions of "swearing" in interpersonal talk. *Griffith Work. Pap. Pragmat. Intercult. Commun.* **2013**, *6*, 71–79.
14. Mehl, M.R.; Vazire, S.; Ramírez-Esparza, N.; Slatcher, R.B.; Pennebaker, J.W. Are women really more talkative than men? *Science* **2007**, *317*, 82. [CrossRef] [PubMed]
15. Wang, W.; Chen, L.; Thirunarayan, K.; Sheth, A.P. Cursing in English on twitter. In Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, Baltimore, MD, USA, 15–19 February 2014; pp. 415–425.
16. Holgate, E.; Cachola, I.; Preoţiuc-Pietro, D.; Li, J.J. Why swear? Analyzing and inferring the intentions of vulgar expressions. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 4405–4414.
17. Chittagonian Language. Available online: https://en.wikipedia.org/wiki/Chittagonian_language (accessed on 11 February 2023).
18. Lewis, M.P. *Ethnologue: Languages of the World*, 16th ed.; SIL International: Dallax, TX, USA, 2009.
19. Masica, C.P. *The Indo-Aryan Languages*; Cambridge University Press: Cambridge, UK, 1993.
20. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [CrossRef]
21. Krippendorff, K. Measuring the reliability of qualitative text analysis data. *Qual. Quant.* **2004**, *38*, 787–800. [CrossRef]
22. Sazzed, S. A lexicon for profane and obscene text identification in Bengali. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Online, 1–3 September 2021; pp. 1289–1296.
23. Das, S.; Mahmud, T.; Islam, D.; Begum, M.; Barua, A.; Tarek Aziz, M.; Nur Showan, E.; Dey, L.; Chakma, E. Deep Transfer Learning-Based Foot No-Ball Detection in Live Cricket Match. *Comput. Intell. Neurosci.* **2023**, *2023*, 2398121. [CrossRef]
24. Mahmud, T.; Barua, K.; Barua, A.; Das, S.; Basnin, N.; Hossain, M.S.; Andersson, K.; Kaiser, M.S.; Sharmen, N. Exploring Deep Transfer Learning Ensemble for Improved Diagnosis and Classification of Alzheimer's Disease. In Proceedings of the 2023 International Conference on Brain Informatics, Hoboken, NJ, USA, 1–3 August 2023; Springer: Cham, Switzerland, 2023; pp. 1–12.
25. Wu, Z.; Luo, G.; Yang, Z.; Guo, Y.; Li, K.; Xue, Y. A comprehensive review on deep learning approaches in wind forecasting applications. *CAAI Trans. Intell. Technol.* **2022**, *7*, 129–143. [CrossRef]
26. Gasparin, A.; Lukovic, S.; Alippi, C. Deep learning for time series forecasting: The electric load case. *CAAI Trans. Intell. Technol.* **2022**, *7*, 1–25. [CrossRef]
27. Pinker, S. *The Stuff of Thought: Language as a Window into Human Nature*; Penguin: London, UK, 2007.
28. Andersson, L.G.; Trudgill, P. *Bad Language*; Blackwell/Penguin Books: London, UK, 1990.
29. Eshan, S.C.; Hasan, M.S. An application of machine learning to detect abusive bengali text. In Proceedings of the 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 22–24 December 2017; pp. 1–6.

30. Akhter, S.; Abdhullah-Al-Mamun. Social media bullying detection using machine learning on Bangla text. In Proceedings of the 2018 10th International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, 20–22 December 2018; pp. 385–388.
31. Emon, E.A.; Rahman, S.; Banarjee, J.; Das, A.K.; Mittra, T. A deep learning approach to detect abusive bengali text. In Proceedings of the 2019 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, 28–30 June 2019; pp. 1–5.
32. Awal, M.A.; Rahman, M.S.; Rabbi, J. Detecting abusive comments in discussion threads using naïve bayes. In Proceedings of the 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh, 27–28 October 2018; pp. 163–167.
33. Hussain, M.G.; Al Mahmud, T. A technique for perceiving abusive bangla comments. *Green Univ. Bangladesh J. Sci. Eng.* **2019**, *4*, 11–18.
34. Das, M.; Banerjee, S.; Saha, P.; Mukherjee, A. Hate Speech and Offensive Language Detection in Bengali. *arXiv* **2022**, arXiv:2210.03479.
35. Sazzed, S. Identifying vulgarity in Bengali social media textual content. *PeerJ Comput. Sci.* **2021**, *7*, e665. [CrossRef]
36. Jahan, M.; Ahamed, I.; Bishwas, M.R.; Shatabda, S. Abusive comments detection in Bangla-English code-mixed and transliterated text. In Proceedings of the 2019 2nd International Conference on Innovation in Engineering and Technology (ICIET), Dhaka, Bangladesh, 23–24 December 2019; pp. 1–6.
37. Ishmam, A.M.; Sharmin, S. Hateful speech detection in public facebook pages for the bengali language. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 555–560.
38. Karim, M.R.; Dey, S.K.; Islam, T.; Sarker, S.; Menon, M.H.; Hossain, K.; Hossain, M.A.; Decker, S. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In Proceedings of the 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), Porto, Portugal, 6–9 October 2021; pp. 1–10.
39. Sazzed, S. Abusive content detection in transliterated Bengali-English social media corpus. In Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching, Online, 11 June 2021; pp. 125–130.
40. Faisal Ahmed, M.; Mahmud, Z.; Biash, Z.T.; Ryen, A.A.N.; Hossain, A.; Ashraf, F.B. Bangla Text Dataset and Exploratory Analysis for Online Harassment Detection. *arXiv* **2021**, arXiv:2102.02478.
41. Romim, N.; Ahmed, M.; Talukder, H.; Islam, S. Hate speech detection in the bengali language: A dataset and its baseline evaluation. In Proceedings of the International Joint Conference on Advances in Computational Intelligence, Dhaka, Bangladesh, 20–21 November 2020; Springer: Singapore, 2021; pp. 457–468.
42. Islam, T.; Ahmed, N.; Latif, S. An evolutionary approach to comparative analysis of detecting Bangla abusive text. *Bull. Electr. Eng. Inform.* **2021**, *10*, 2163–2169. [CrossRef]
43. Aurpa, T.T.; Sadik, R.; Ahmed, M.S. Abusive Bangla comments detection on Facebook using transformer-based deep learning models. *Soc. Netw. Anal. Min.* **2022**, *12*, 24. [CrossRef]
44. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MI, USA, 2–7 June 2019; Association for Computational Linguistics: Minneapolis, MI, USA, 2019; pp. 4171–4186.
45. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. In Proceedings of the International Conference on Learning Representations, ICLR 2020, Virtual, 26 April–1 May 2020.
46. List of Non-Governmental Organisations in Bangladesh. Available online: https://en.wikipedia.org/wiki/List_of_non-governmental_organisations_in_Bangladesh (accessed on 15 February 2023).
47. Pradhan, R.; Chaturvedi, A.; Tripathi, A.; Sharma, D.K. A review on offensive language detection. In *Advances in Data and Information Sciences: Proceedings of ICDIS 2019, Agra, India, 29–30 March 2019*; Springer: Singapore, 2020; pp. 433–439.
48. Khan, M.M.; Shahzad, K.; Malik, M.K. Hate speech detection in roman urdu. *ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP)* **2021**, *20*, 1–19. [CrossRef]
49. Novitasari, S.; Lestari, D.P.; Sakti, S.; Purwarianti, A. Rude-Words Detection for Indonesian Speech Using Support Vector Machine. In Proceedings of the 2018 International Conference on Asian Language Processing (IALP), Bandung, Indonesia, 15–17 November 2018; pp. 19–24. [CrossRef]
50. Kim, S.N.; Medelyan, O.; Kan, M.Y.; Baldwin, T. Automatic keyphrase extraction from scientific articles. *Lang. Resour. Eval.* **2013**, *47*, 723–742. [CrossRef]
51. Li, J.; Jiang, G.; Xu, A.; Wang, Y. The Automatic Extraction of Web Information Based on Regular Expression. *J. Softw.* **2017**, *12*, 180–188.
52. Alqahtani, A.; Alhakami, H.; Alsubait, T.; Baz, A. A survey of text matching techniques. *Eng. Technol. Appl. Sci. Res.* **2021**, *11*, 6656–6661. [CrossRef]
53. Califf, M.E.; Mooney, R.J. Bottom-up relational learning of pattern matching rules for information extraction. *J. Mach. Learn. Res.* **2003**, *4*, 177–210.
54. Ptaszynski, M.; Lempa, P.; Masui, F.; Kimura, Y.; Rzepka, R.; Araki, K.; Wroczynski, M.; Leliwa, G. Brute-force sentence pattern extortion from harmful messages for cyberbullying detection. *J. Assoc. Inf. Syst.* **2019**, *20*, 1075–1127. [CrossRef]

55. Beliga, S. *Keyword Extraction: A Review of Methods and Approaches*; University of Rijeka, Department of Informatics: Rijeka, Croatia, 2014; Volume 1.
56. Su, G.y.; Li, J.h.; Ma, Y.h.; Li, S.h. Improving the precision of the keyword-matching pornographic text filtering method using a hybrid model. *J. Zhejiang Univ.-Sci. A* **2004**, *5*, 1106–1113. [CrossRef]
57. Liu, F.; Pennell, D.; Liu, F.; Liu, Y. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In Proceedings of the Human Language Technologies: The 2009 Annual Conference of the North American Chapter of The association for Computational Linguistics, Boulder, CO, USA, 31 May–5 June 2009; pp. 620–628.
58. Ptaszynski, M.; Yagahara, A. Senmon Yogo Chushutsu Sochi, Senmon yogo Chushutsu hoho Oyobi Puroguramu (Technical Term Extraction Device, Technical Term Extraction Method and Program). 16 December 2021. Available online: https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=202103002313491840 (accessed on 29 January 2023). (In Japanese)
59. Mahmud, T.; Ptaszynski, M.; Eronen, J.; Masui, F. Cyberbullying detection for low-resource languages and dialects: Review of the state of the art. *Inf. Process. Manag.* **2023**, *60*, 103454. [CrossRef]
60. Li, D.; Rzepka, R.; Ptaszynski, M.; Araki, K. HEMOS: A novel deep learning-based fine-grained humor detecting method for sentiment analysis of social media. *Inf. Process. Manag.* **2020**, *57*, 102290. [CrossRef]
61. Haque, M.Z.; Zaman, S.; Saurav, J.R.; Haque, S.; Islam, M.S.; Amin, M.R. B-NER: A Novel Bangla Named Entity Recognition Dataset with Largest Entities and Its Baseline Evaluation. *IEEE Access* **2023**, *11*, 45194–45205. [CrossRef]
62. Eronen, J.; Ptaszynski, M.; Masui, F.; Smywiński-Pohl, A.; Leliwa, G.; Wroczynski, M. Improving classifier training efficiency for automatic cyberbullying detection with feature density. *Inf. Process. Manag.* **2021**, *58*, 102616. [CrossRef]
63. Mahmud, T.; Das, S.; Ptaszynski, M.; Hossain, M.S.; Andersson, K.; Barua, K. Reason Based Machine Learning Approach to Detect Bangla Abusive Social Media Comments. In *Intelligent Computing & Optimization, Proceedings of the 5th International Conference on Intelligent Computing and Optimization 2022 (ICO2022), Virtual, 27–28 October 2022*; Springer: Cham, Switzerland, 2022; pp. 489–498.
64. Ahmed, T.; Mukta, S.F.; Al Mahmud, T.; Al Hasan, S.; Hussain, M.G. Bangla Text Emotion Classification using LR, MNB and MLP with TF-IDF & CountVectorizer. In Proceedings of the 2022 26th International Computer Science and Engineering Conference (ICSEC), Sakon Nakhon, Thailand, 21–23 December 2022; pp. 275–280.
65. sklearn.feature_extraction.text.CountVectorizer. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html (accessed on 23 February 2023).
66. Chakraborty, M.; Huda, M.N. Bangla document categorisation using multilayer dense neural network with tf-idf. In Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 3–5 May 2019; pp. 1–4.
67. sklearn.feature_extraction.text.TfidfVectorizer. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (accessed on 23 February 2023).
68. Rahman, R. Robust and consistent estimation of word embedding for bangla language by fine-tuning word2vec model. In Proceedings of the 2020 23rd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 19–21 December 2020; pp. 1–6.
69. Ma, L.; Zhang, Y. Using Word2Vec to process big text data. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015; pp. 2895–2897.
70. facebookresearch/fastText: Library for Fast Text. Available online: https://github.com/facebookresearch/fastText (accessed on 25 February 2023).
71. Research—Meta AI. Available online: https://ai.meta.com/research/ (accessed on 25 February 2023).
72. Mojumder, P.; Hasan, M.; Hossain, M.F.; Hasan, K.A. A study of fasttext word embedding effects in document classification in bangla language. In Proceedings of the Cyber Security and Computer Science: Second EAI International Conference, ICONCS 2020, Dhaka, Bangladesh, 15–16 February 2020; Proceedings 2; Springer: Cham, Switzerland, 2020; pp. 441–453.
73. Shah, K.; Patel, H.; Sanghvi, D.; Shah, M. A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augment. Hum. Res.* **2020**, *5*, 12. [CrossRef]
74. Mahmud, T.; Ptaszynski, M.; Masui, F. Vulgar Remarks Detection in Chittagonian Dialect of Bangla. *arXiv* **2023**, arXiv:2308.15448.
75. Hasanli, H.; Rustamov, S. Sentiment analysis of Azerbaijani twits using logistic regression, Naive Bayes and SVM. In Proceedings of the 2019 IEEE 13th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 23–25 October 2019; pp. 1–7.
76. Hussain, M.G.; Hasan, M.R.; Rahman, M.; Protim, J.; Al Hasan, S. Detection of bangla fake news using mnb and svm classifier. In Proceedings of the 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE), Southend, UK, 17–18 August 2020; pp. 81–85.
77. Alam, M.R.; Akter, A.; Shafin, M.A.; Hasan, M.M.; Mahmud, A. Social Media Content Categorization Using Supervised Based Machine Learning Methods and Natural Language Processing in Bangla Language. In Proceedings of the 2020 11th International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, 17–19 December 2020; pp. 270–273.
78. Joyce, J. Bayes' theorem. In *Stanford Encyclopedia of Philosophy*; Stanford University: Stanford, CA, USA, 2003.
79. Berrar, D. Bayes' theorem and naive Bayes classifier. *Encycl. Bioinform. Comput. Biol. ABC Bioinform.* **2018**, *403*, 412.
80. Islam, T.; Prince, A.I.; Khan, M.M.Z.; Jabiullah, M.I.; Habib, M.T. An in-depth exploration of Bangla blog post classification. *Bull. Electr. Eng. Inform.* **2021**, *10*, 742–749. [CrossRef]

81. Haydar, M.S.; Al Helal, M.; Hossain, S.A. Sentiment extraction from bangla text: A character level supervised recurrent neural network approach. In Proceedings of the 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh, 8–9 February 2018; pp. 1–4.
82. Hu, Z.; Zhang, J.; Ge, Y. Handling vanishing gradient problem using artificial derivative. *IEEE Access* **2021**, *9*, 22371–22377. [CrossRef]
83. Mumu, T.F.; Munni, I.J.; Das, A.K. Depressed people detection from bangla social media status using lstm and cnn approach. *J. Eng. Adv.* **2021**, *2*, 41–47. [CrossRef]
84. Dam, S.K.; Turzo, T.A. Social Movement Prediction from Bangla Social Media Data Using Gated Recurrent Unit Neural Network. In Proceedings of the 2021 5th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 17–19 December 2021; pp. 1–6.
85. Uddin, A.H.; Bapery, D.; Arif, A.S.M. Depression analysis from social media data in Bangla language using long short term memory (LSTM) recurrent neural network technique. In Proceedings of the 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh, 11–12 July 2019; pp. 1–4.
86. Ptaszynski, M.; Dybala, P.; Shi, W.; Rzepka, R.; Araki, K. A system for affect analysis of utterances in Japanese supported with web mining. *J. Jpn. Soc. Fuzzy Theory Intell. Inform.* **2009**, *21*, 194–213. [CrossRef]
87. Ptaszynski, M.; Masui, F.; Dybala, P.; Rzepka, R.; Araki, K. Open source affect analysis system with extensions. In Proceedings of the 1st International Conference on Human–Agent Interaction, iHAI, Sapporo, Japan, 7–9 August 2013.
88. Ptaszynski, M.; Dybala, P.; Rzepka, R.; Araki, K.; Masui, F. ML-Ask: Open source affect analysis software for textual input in Japanese. *J. Open Res. Softw.* **2017**, *5*, 16. [CrossRef]
89. Ptaszynski, M.; Masui, F.; Fukushima, Y.; Oikawa, Y.; Hayakawa, H.; Miyamori, Y.; Takahashi, K.; Kawajiri, S. Deep Learning for Information Triage on Twitter. *Appl. Sci.* **2021**, *11*, 6340. [CrossRef]
90. Gray, D.E. *Doing Research in the Real World*; Sage: Newcastle upon Tyne, UK, 2021; pp. 1–100.
91. Mahoney, J.; Le Louvier, K.; Lawson, S.; Bertel, D.; Ambrosetti, E. Ethical considerations in social media analytics in the context of migration: Lessons learned from a Horizon 2020 project. *Res. Ethics* **2022**, *18*, 226–240. [CrossRef]