Taming Toxicity: Learning Models for Hate Speech and Offensive Language Detection in Social Media Text

Prajnashree M, Rachana K, Asha Hegde, Kavya G, Sharal Coelho and H L Shashirekha

Department of Computer Science, Mangalore University, Mangalore, Karnataka, India

Abstract

User-friendly social media platforms like Twitter, Facebook, etc., provide opportunities for their users' to voice their opinions against anything and everything. People, irrespective of the age group, use these social media platforms to share every moment of their life making these sites flooded with user-generated text. However, the anonymity of users on these platforms is misused by some culprits to spread hate speech (hatred and unrealistic comments) and/or abusive/offensive content, against anybody with an ulterior motive to tarnish one's image and status in the society. Identifying such messages and filtering them out to stop spreading further has become very crucial in maintaining a healthy social media ecosystem. With the increase in user-generated text in low-resourced Indo-Aryan languages, identifying Hate Speech and Offensive Content (HASOC) in these languages is increasing gradually. To address the challenges of identifying HASOC in Indo-Aryan languages, in this paper, we - team MUCS, describe the learning models submitted to "Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages (HASOC) 2023" at Forum for Information Retrieval Evaluation (FIRE) 2023. This shared task has four subtasks and we participated in Task 1A and 1B (to identify hate speech, offensive language, and profanity, in Sinhala and Gujarati respectively) and Task 4 (to detect hate speech in Assamese, Bengali and Bodo). Several experiments are carried out with hand crafted features (syllable n-grams extracted from the given text and character (char) n-grams extracted from romanized text) and fastText word embeddings, to train various Machine Learning (ML) classifiers to identify HASOC in Task 1A and Task 4. Due to very small training data in Task 1B, this task is modeled as Few-Shot Learning (FSL) problem and experimented with Siamese Network using Long Short-Term Memory (LSTM) (trained with Gujarati fastText word embeddings) and Ensemble of ML classifers with hard voting (trained with Sentence Transformer (ST)), to identify HASOC in Gujarati. Among the proposed models, Support Vector Machine (SVM) trained with char n-grams features obtained a better macro F1 score of 0.78 for Sinhala language in Task 1A, and Siamese-LSTM model obtained a better macro F1 score of 0.72 for Gujarati language in Task 1B. Further, SVM trained with syllable n-grams and char n-grams of romanized text obtained better macro F1 scores of 0.688, 0.668, and 0.836 for Assamese, Bengali, and Bodo languages respectively in Task 4.

Keywords

Machine learning, Few-shot learning, Ensemble, fastText word embeddings, Syllable n-grams, character n-grams

Forum for Information Retrieval Evaluation, December 15-18, 2023, India

prajnapushparaj27@gmail.com (P. M); rachanak749@gmail.com (R. K); hegdekasha@gmail.com (A. Hegde); kavyamujk@gmail.com (K. G); sharalmucs@gmail.com (S. Coelho); hlsrekha@mangaloreuniversity.ac.in (H. L. Shashirekha)

1. Introduction

In this digital era, to a larger extent, social media has become a power of expression enabling the individuals to connect with each other, share their thoughts, and engage themselves in new ways. At the same time, the anonymity of users on social media is being misused by many culprits to spread HASOC [1]. The values of equality, diversity, and tolerance that support democratic societies are exceedingly threatened by hate speech, which is characterized by expressions of prejudice, discrimination, and hatred, toward individuals or groups based on their race, religion, ethnicity, gender, or other protected characteristics [2].

Hate speech is any verbal or written expression that advocates damage or prejudice towards an individual or group of individuals, based on the characteristics including race, religion, ethnicity, gender, sexual orientation, or disabilities. Offensive content on the other hand is defined as the content having the potential to incite hatred or discrimination but is more likely to make individuals or groups feel uncomfortable, disgusted, or distressed, as the content may be contentious, vulgar or obscene, expressed in disrespectful language or gestures [3]. HASOC encourages animosity and intolerance and has the potential to provoke violence as they offend accepted community norms and can suffocate online discourse. Hence, identifying HASOC on online platforms and filtering them out to avoid further spread has become a crucial aspect to maintain a good ecosystem on social media platforms.

Most of the HASOC identification works focus on high-resource languages such as English, Spanish etc., giving less importance for low-resource languages such as Assamese, Bengali, Kannada, Tulu, etc. Identifying HASOC in low-resource languages poses significant challenges due to limitations in data, annotated data, pre-trained models and other computational tools. Collecting/creating suitable annotated corpora by consulting the right annotators is a major issue in low-resource languages [4]. As most of the datasets available for low-resource languages are small in size, they fail to capture the nuances and variations in the data and this affects the performance of the learning models. Further, the datasets may be imbalanced due to the non-availability of data representing all the categories equally in the dataset.

To address the challenges of identifying HASOC on online platforms, in this paper, we - team MUCS, describe the learning models submitted to HASOC 2023[1] shared task at FIRE 2023[2]. This shared task has four subtasks and we participated in Task 1A and 1B (to identify hate speech, offensive language, and profanity in Sinhala and Gujarati respectively) and Task 4 (to detect hate speech in Assamese, Bengali and Bodo). Information about these subtasks and the statistics of the datasets of these subtasks are shown in Table 1. The proposed methodologies include: i) ML classifiers trained with hand-crafted features (for Sinhala in Task 1A) [5], ii) FSL with Siamese Network using LSTM model trained with Gujarati fastText embeddings and Ensemble of ML classifiers with hard voting trained with ST (for Gujarati in Task 1B) [6], and iii) ML classifiers trained with hand-crafted features and fastText word embeddings (for Bengali, Assamese, and Bodo in Task 4) [7, 8], for identifying HASOC.

The rest of the paper is structured as follows: Section 2 contains related works and Section 3 explains the methodology. Section 4 describes the experiments and results and the paper

[1] https://hasocfire.github.io/hasoc/2023/

[2] http://fire.irsi.res.in/fire/2023/home

Table 1
Information about the subtasks of the shared tasks in which we participated

Subtasks	Descriptions		Train Set	Test Set
Task 1A	Identifying Hate, Offensive and Profane Content in Sinhala		7,500	2,500
Task 1B	Identifying Hate, Offensive and Profane Content in Gujarati		200	1,196
Task 4	Annihilate Hates in	Assamese	1,679	420
		Bodo	4,036	1,009
		Bengali	1,281	320

concludes in Section 5 with future work.

2. Related Work

The negative effects of HASOC on social media users' well-being and social cohesion have been significantly studied by researchers. This has sparked a growing interest in developing efficient techniques for the detection of HASOC on online platforms. Some of the relevant works are described below:
Banerjee et al. [9] fine-tuned mBERT-base, Cross-lingual Language Model Robustly Optimized BERT Approach (XLMR)-large and XLMR-base models, to obtain the contextualized embeddings by the attention mechanism to identify HASOC in English and code-mixed Hindi texts and obtained macro F1 scores in the range 0.6447 to 0.8006. Kumari and Singh [10] trained ML classifiers (Logistic Regression (LR), SVM, and Random Forest (RF)) with Term Frequency-Inverse Document Frequency (TF-IDF) of word unigrams for the Identification of Conversational Hate-Speech in Code-Mixed Languages (ICHCL) in two categories: binary classification (ICHCL-binary) and multiclass classification (ICHCL-multiclass) and offensive language identification in Marathi (Marathi-binary, Marathi-multiclass with 3 labels (3B-Marathi), and Marathi-multiclass with 4 labels (3C-Marathi)), at HASOC 2022 shared task. Among the proposed models, RF model outperformed other models with a macro F1 score of 0.60 for ICHCL-binary task and SVM model obtained macro F1 score of 0.416 for ICHCL-multiclass task. Further, for Marathi-binary task, LR model achieved a macro F1 score of 0.92 and SVM models obtained macro F1 scores of 0.44 and 0.74 for 3B-Marathi and 3C-Marathi subtasks respectively.

Kui [11] proposed hybrid models for identifying HASOC in: i) English, Hindi, and Marathi languages (Subtask A) and ii) English and Hindi languages (Subtask B), at HASOC 2021 shared task. They designed hybrid models by fine-tuning various Language Models (LM) (Bidirectional Encoder Representations from Transformers (BERT), A Lite BERT (ALBERT), mBERT, Decoding-enhanced BERT with disentangled attention (DeBERTa), XLNet, SqueezeBERT) and integrating them with various Neural Network (NN) models (Recurrent Neural Network (RNN), Bidirectional Long Short-Term Memory (BiLSTM) and Convolutional Neural Network (CNN)). Among these models, DeBERTa+BiLSTM model performed best on English with a macro F1 score of 0.8030 and mBERT+CNN model achieved the highest macro F1 scores of 0.7725 and 0.8611 for Hindi

and Marathi datasets respectively, in Subtask A. Further, DeBERTa+BiLSTM model obtained a macro F1 score of 0.6116 for English and mBERT+CNN model obtained a macro F1 score of 0.5509 for Hindi, in Subtask B.

Caparrós-Laiz et al. [12] proposed three distinct models: i) BERT model with transformers classifier (includes fine-tuning of BERT), ii) Hybrid model in which BERT tokens are used to train a NN model, and iii) Ensemble of 110 NN models trained with different features (linguistic features, sentence embeddings, fastText word embeddings for English (for multiclass classification), Hindi and Marathi, GloVe word embeddings and BERT), for identifying offensive content in English, Hindi and Marathi languages respectively. Their proposed ensemble models obtained the macro F1 scores of 0.6289, 0.7520, 0.5167, and 0.8423 for English (for multiclass classification), Hindi (binary classification), Hindi (multiclass classification), and Marathi (binary classification) languages respectively.

Kumar et al. [13] proposed ensemble models (SVM, LR, RF, gradient boosting, and Adaboost classifiers) for binary classification of hate speech and multiclass classification of offensive content in English and Hindi social media posts. The ensemble models trained with word n-grams in the range (1, 3) for English dataset obtained macro F1 scores of 0.79 (for binary) and 0.59 (for multiclass), while the model with character n-grams in the range (1, 6) for Hindi language obtained macro F1 scores of 0.75 (for binary) and 0.47 (for multiclass classification).

Nayel, Hamada and Shashirekha, H. [14] presented the ML models (SVM, Linear Classifier, and Multilayer Perceptron (MLP)) to identify offensive content in three languages (English, German, and Hindi) considering the tasks as binary and multiclass classification problems. Their proposed SVM classifier trained with TF-IDF of word n-grams in the range (1, 2) outperformed other models with the macro F1 scores of 0.66, 0.75, and 0.46 for binary classification and 0.42, 0.47, and 0.23 for multiclass classification, for English, Hindi, and German languages respectively.

Dowlagar and Mamidi [15] have explored BERT and mBERT models for identifying HASOC in English, German, and Hindi languages. Using fine-tuned BERT model, they got accuracies of 88.33% and 81.57% for binary and multiclass classification tasks respectively, for English language. For German language, using fine-tuned mBERT, they got accuracies of 82.51% and 80.42% for binary and multiclass classification tasks respectively. Also, for Hindi language using mBERT they got accuracies of 74.96% and 73.15% for binary and multiclass classification tasks respectively.

Identification of HASOC in Indo-Aryan languages is gradually gaining popularity because of the increase in the amount of user-generated text, availability of domain specific datasets though in small size and pre-trained models. The results of the learning models in the related work reveals that there is ample room for further research and innovation in this topic.

3. Methodology

The methodologies include Pre-processing to clean the text data which is common to all the learning models followed by building learning models for the identification of HASOC in Sinhala, Gujarati, Assamese, Bengali, and Bodo. Pre-processing and the three proposed learning models: i) ML classifiers trained with a combination of hand crafted features and fastText word embeddings for Task 1A and Task 4 and ii) FSL approaches with Siamese Network using LSTM

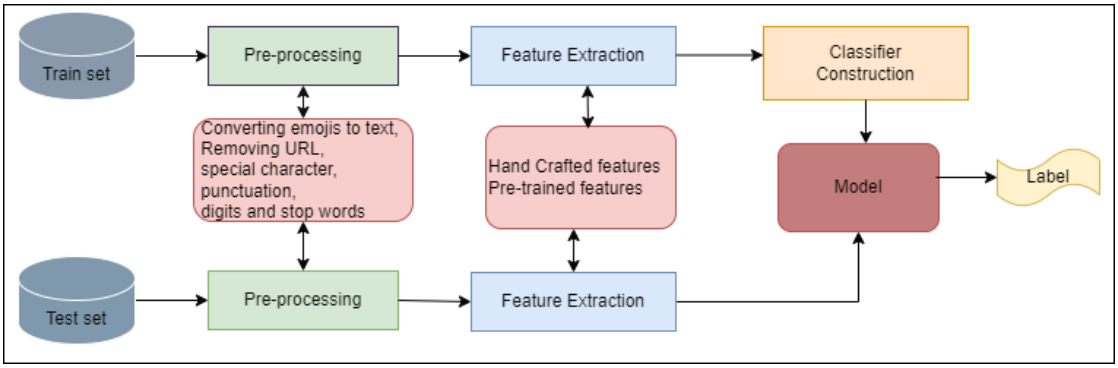

Figure 1: Framework of the proposed ML model

model trained with Gujarati fastText embeddings and Ensemble of ML classifiers with hard voting trained with ST, for Task 1B, are explained below:

3.1. Pre-processing

Pre-processing encompasses various techniques to remove noise from the text data with the aim of improving the performance of the learning models. As emojis depict user's intention, they are converted to text using demoji[3] library. URLs, user mentions, hash tags, special characters, punctuation, and numeric information, present in the text data do not contribute to the classification task and hence are removed. Stopwords are a set of commonly used words in any language and they do not contribute significantly to the classification task and hence are removed. Assamese[4], Sinhala[5], Gujarati[6], and Bengali[7]stopwords, available at GitHub repositories are used as references to remove the stopwords from the respective languages. The remaining words are the content bearing words which goes as input to feature extraction.

3.2. Machine Learning Models

The proposed models include individual ML classifiers and ensemble of ML classifiers trained using hand-crafted features and fastText word embeddings, to identify HASOC. Framework of ML classifiers is shown in Figure 1 and the steps involved in building ML models are described below:

3.2.1. Feature Extraction

The objective of feature extraction is to extract distinguishable features from the text for the identification of HASOC effectively. The features extraction steps are described below:

[3]https://pypi.org/project/demoji/
[4]https://noixobdo.blogspot.com/2011/03/assamese-stop-word-list.html
[5]https://github.com/nlpcuom/Sinhala-Stopword-list/blob/master/stop%words.txt
[6]https://github.com/gujarati-ir/Gujarati-Stop-Words/blob/master/gujarati_stop_words.zip
[7]https://github.com/stopwords-iso/stopwords-bn/blob/master/stopwords-bn.txt

- **Handcrafted Features** - are the language-independent features such as character, syllable, character (char) n-grams and syllable n-grams, which play a significant role in text classification. While char represents a single character in romanized/English text, syllable is a unit of pronunciation having one vowel sound and for languages with non-romanized script, syllable representation gives meaningful tokens. Char/syllable n-grams are 'n' contiguous sequences of characters/syllables in a word. As the given dataset contains the text in native script, the text is romanized using libindic[8] library to get the character representations of the dataset. Char and syllables n-grams in the range (1, 3) are obtained from the given dataset. The sample words with their syllable and char; unigrams, bigrams, and trigrams, are shown in Table 2.
  TF-IDF vectors enhance the normalized representation of text documents by mitigating the influence of excessively repeated words. These vectors indicate the significance of a word within a document relative to the entire corpus. Char n-grams and syllable n-grams are vectorized using the TFIDFVectorizer[9].
- **Pre-trained Word Embeddings** - are vector representations for words pre-computed by considering a large amount of text data in any natural language. These embeddings capture semantic and syntactic information about words, allowing them to encode the relationships between words based on their context. fastText pre-trained word embeddings are used in this work.
  fastText - developed by Facebook AI Research, is a powerful open-source library designed for both learning word embeddings and performing classification. It includes a wide range of pre-trained models, which have been trained on diverse textual data sources, including Wikipedia, across more than 157 languages. For Gujarati and Bengali words in their native scripts, word vectors are extracted from Gujarati and Bengali fastText word embeddings respectively and for English words, word vectors are extracted from English fastText word embeddings. The vocabulary sizes of Gujarati, Bengali and English fastText word embeddings are 5,54,518, 14,68,579 and 20,00,001, respectively, with word embeddings having a vector of dimension 300 each.

The resultant feature vectors which capture the essential information from the datasets and used in training and evaluating the respective learning models.

3.2.2. Classifier Construction

The performance of the model relies heavily on the features and the classifier used to carry out the classification. This work utilizes individual ML classifiers (SVM, RF, and Passive Aggressive Classifier (PAC)) and an ensemble of ML classifiers (LR, Bernoulli's Naive Bayes (BNB), and Support Vector Classifier (SVC)) with majority voting, to identify HASOC in the given languages. A brief description of the classifiers is given below:

- **Support Vector Machine** - is a commonly used ML algorithm with an objective of discovering the optimal hyperplane that effectively separates various classes of data

[8]https://github.com/libindic/Transliteration
[9]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Table 2
Sample words with their syllable and character unigrams, bigrams, and trigrams

Languages	Words in Native script	Syllable unigrams, bigrams and trigrams	Words in Roman script	Character unigrams, bigrams and trigrams
Assamese	বানপানী	Unigrams = ['বা', ''ন', 'পা', 'নী'] Bigrams = ['বান', 'নপা', 'পানী'] Trigrams = ['বানপা', 'নপানী']	banyaa	Unigrams = ['b','a','n', 'y','a', 'a'] Bigrams = ['ba','an','ny','ya','aa'] Trigrams = ['ban','any','nya,'yaa']
Gujarathi	ગુજરાત	Unigrams = ['ગુ', ' 'જ', 'રા', 'ત'] Bigrams = ['ગુજ', 'જરા', 'રાત'] Trigrams = ['ગુજરા', 'જરાત']	gujarat	Unigrams = ['g','u','j','a','r','a','a','t'] Bigrams = ['gu','uj','ja','ar','ra','aa','at'] Trigrams = ['guj','uja','jar','ara','raa','aat']
Bodo	सरकार	Unigrams =['स', 'र', 'का', 'र'] Bigrams = ['सर', 'रका', 'कार'] Trigrams = ['सरका', 'रकार']	sarakaar	Unigrams = ['s', 'a', 'r', 'a', 'k', 'a', 'a', 'r'] Bigrams = ['sa','ar','ra','ak','ka','aa','ar'] Trigrams = ['sar','ark','rka','kaa','aar']

within a high-dimensional feature space. SVM aims to identify the most discriminative features capable of effectively distinguishing between different classes.

- **Random Forest** - which contains a number of decision trees on various subsets of the given dataset is a popular ML algorithm used for both Classification and Regression problems. It is based on the concept of ensemble learning, which is a process of combining diverse multiple classifiers to solve a complex problem with the aim of improving the performance of the model.
- **Passive Aggressive Classifier** - is a ML algorithm used for binary classification tasks. It is particularly suited for online learning scenarios where data streams in sequentially, and the model needs to adapt and update itself as new examples arrive. It is unique in its approach as it tries to minimize classification errors while being ”passive” when the predictions are correct and ”aggressive” when they are incorrect.
- **Ensemble model** - is a method of generating a new classifier from heterogeneous multiple base classifiers taking advantage of the strength of one classifier to overcome the weakness of another classifier with the intention of getting better performance for the classification task [16]. The following classifiers are ensembled in this work:
  - **Logistic Regression** - strategically incorporates dependent variables and regularization techniques to safeguard against over-fitting. The features are aggregated through a linear combination followed by transformation using the logistic function - a process that empowers the algorithm to generate predictions and classify instances into one of the predefined classes.
  - **Bernoulli Naïve Bayes** - is a probabilistic ML algorithm that operates on the foundation of Bayes theorem. Assuming feature independence, BNB calculates the probabilities of a sample belonging to the given classes.
  - **Support Vector Classifier** - is highly popular for its effectiveness in high-dimensional feature spaces, making it particularly well-suited for text classification tasks, where data often consists of a large number of features representing words or phrases, It’s

ability to identify intricate and nonlinear relationships between the features allows it to excel in accurately categorizing text documents.

The features, models and the tasks, for which the features and models are applied, are shown in Table 3.

3.3. Few-Shot Learning

The common practice in building ML classifiers is to consider a large labeled data to train the classifiers. But, large labeled data may not be accessible/ available to develop many applications. FSL - a supervised ML approach that involves learning from a small number of labeled data is a solution to such situations [17]. FSL approaches are implemented with Siamese Network using LSTM and Voting classifier using Sentence Transformer (ST) models. The description of the models are given below:

- **Siamese Network** - is a class of NN architectures that contains two or more identical sub-networks and is used to capture the semantic relatedness among documents. The main idea of Siamese network is to learn the vector representation by training a model that discriminates between pairs of examples that are in the same category, and pairs of examples that come from different categories. The given Gujarati dataset which consists of only 200 samples is increased to 19,800 (200 samples from the given dataset + 19,600 synthetic samples) using Siamese Network. Even though the dataset is artificially increased with the samples given in the training set instead of generating new data, this technique is still very powerful.
  LSTM is an RNN architecture used to address the long term dependence and gradient disappearance issues in RNN. It can add and remove information to each unit/cell by carefully regulating its gates (forget gate, input gate, input modulation gate, and output gate). In this work, the Siamese LSTM network is created with Manhattan distance metric to determine the similarity between a pair of vectors (x_left and x_right) and the model is trained with Gujarati fastText embeddings to classify the given text.
- **Ensemble model** - is an ensemble of ML classifiers (LR, BNB, SVC and RF) with hard voting and is trained with 'GroNLP/hateBERT'[10] - a ST. ST is a Python framework that transforms sentences into vector embeddings of size 768, capturing their contextual meaning enabling tasks like semantic search, clustering, and retrieval, by measuring the proximity of similar sentences in a vector space. HateBERT is an English pre-trained BERT model obtained by further training the English BERT base uncased model with more than 1 million posts from banned communites from Reddit.

4. Experiments and Results

Various experiments were carried out with different combinations of features (syllable n-grams, char n-grams, and fastText word embeddings) and different approaches (ML and FSL), to identify the HASOC in the given input.

[10]https://huggingface.co/GroNLP/hateBERT

Table 3
Results of the proposed models

Subtasks	Languages	Features	Models	Test set
Task 1A	Sinhala	char n-grams	RF	0.770
			SVM	**0.78**
			PAC	0.74
Task 1B	Gujarati	fastText vectors	FSL- Siamese-LSTM	**0.727**
		Sentence Transformer	FSL- Ensemble	0.627
Task 4	Bengali	Syllable n-grams	SVM	**0.668**
		char n-grams		0.651
		fastText vectors	Ensemble	0.631
	Assamese	char n-grams	Ensemble	**0.688**
		Syllable n-grams	SVM	0.674
			Ensemble	0.674
	Bodo	Syllable n-grams	SVM	0.830
		char ngrams	SVM	**0.836**
			Ensemble	0.834

The performances of the proposed models for the Test set are shown in Table 3. Among the proposed models, SVM trained with char n-grams obtained the macro F1 score of 0.78 securing 11^{th} rank for Sinhala in Task 1A, Siamese-LSTM trained with fastText embeddings obtained the macro F1 score of 0.72 securing 12^{th} rank for Gujarati in Task 1B. Further, SVM trained with TF-IDF of syllable n-grams and TF-IDF of char n-grams both in the range (1, 3) obtained the macro F1 scores of 0.688, 0.668, and 0.836 securing 11^{th}, 11^{th}, and 14^{th} ranks for Assamese, Bengali, and Bodo languages respectively in Task 4.

5. Conclusion

In this paper, we - team MUCS, describe the models submitted to HASOC 2023 shared task at FIRE 2023, to identfiy HASOC in Indo-Aryan languages, viz., Sinhala, Gujarati, Assamese, Bengali, and Bodo. Experiments are carried out with different hand crafted features (TF-IDF of syllable n-grams and char n-grams of romanized text, both in the range (1, 3)) and fastText word embeddings. While ML classifiers (RF, SVM, and PAC) are trained with TF-IDF of char n-grams to identify HASOC in Sinhala in Task 1A, FSL approaches with Siamese Network using LSTM model is trained with fastText embeddings and Ensemble model (LR, BNB, SVC, and RF, with hard voting) is trained with ST, to identify HASOC in Gujarati. Further, SVM and Ensemble models (LR, BNB, and SVC, with hard voting) are trained with TF-IDF of syllable n-grams and char n-grams, and fastText embeddings, to identify HASOC in Assamese, Bengali, and Bodo languages in Task 4. Among all the models, SVM trained with TF-IDF of char n-grams obtained the macro F1 score of 0.78 for Sinhala in Task 1A, FSL with Siamese Network using LSTM model trained with fastText embeddings obtained the macro F1 score of 0.72 for Gujarati in Task 1B. Further, SVM trained with TF-IDF of syllable n-grams and char n-grams obtained the macro F1 scores of 0.688, 0.668, and 0.836 for Assamese, Bengali, and Bodo languages respectively in Task 4.

References

[1] A. Hegde, M. D. Anusha, H. L. Shashirekha, Ensemble based Machine Learning Models for Hate Speech and Offensive Content Identification, in: Forum for Information Retrieval Evaluation (Working Notes)(FIRE), CEUR-WS. org, 2021.

[2] F. Balouchzahi, H. L. Shashirekha, G. Sidorov, HSSD: Hate Speech Spreader Detection using n-grams and Voting Classifier., in: CLEF (Working Notes), 2021, pp. 1829–1836.

[3] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, Overview of the Hasoc Track at Fire 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German, in: Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation, 2020.

[4] D. Nkemelu, H. Shah, M. Best, I. Essa, Tackling Hate Speech in Low-resource Languages with Context Experts, in: Proceedings of the 2022 International Conference on Information and Communication Technologies and Development, 2022, pp. 1–11.

[5] T. Ranasinghe, I. Anuradha, D. Premasiri, K. Silva, H. Hettiarachchi, L. Uyangodage, M. Zampieri, Sold: Sinhala Offensive Language Dataset, in: arXiv preprint arXiv:2212.00851, 2022.

[6] S. Satapara, H. Madhu, T. Ranasinghe, A. E. Dmonte, M. Zampieri, P. Pandya, N. Shah, M. Sandip, P. Majumder, T. Mandl, Overview of the HASOC Subtrack at FIRE 2023: Hate-Speech Identification in Sinhala and Gujarati, in: K. Ghosh, T. Mandl, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation, Goa, India. December 15-18, 2023, CEUR Workshop Proceedings, CEUR-WS.org, 2023.

[7] K. Ghosh, A. Senapati, A. S. Pal, Annihilate Hates (Task 4, HASOC 2023): Hate Speech Detection in Assamese, Bengali, and Bodo languages, in: Working Notes of FIRE 2023 - Forum for Information Retrieval Evaluation, CEUR, 2023.

[8] T. Ranasinghe, K. Ghosh, A. S. Pal, A. Senapati, A. E. Dmonte, M. Zampieri, S. Modha, S. Satapara, Overview of the HASOC Subtracks at FIRE 2023: Hate Speech and Offensive Content Identification in Assamese, Bengali, Bodo, Gujarati and Sinhala, in: Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE 2023, Goa, India. December 15-18, 2023, ACM, 2023.

[9] S. Banerjee, M. Sarkar, N. Agrawal, P. Saha, M. Das, Exploring Transformer Based Models to Identify Hate Speech and Offensive Content in English and Indo-aryan Languages, in: arXiv preprint arXiv:2111.13974, 2021.

[10] K. Kumari, J. P. Singh, Machine Learning Approach for Hate Speech and Offensive Content Identification in English and Indo-Aryan Code-Mixed Languages, in: Forum for Information Retrieval Evaluation (Working Notes)(FIRE). CEUR-WS. org, 2022.

[11] Y. Kui, Detect Hate and Offensive Content in English and Indo-Aryan Languages based on Transformer, in: Forum for Information Retrieval Evaluation (Working Notes)(FIRE), CEUR-WS. org, 2021.

[12] C. Caparrós-Laiz, J. Antonio, G. Díaz, R. Valencia-Garcia, Detecting Hate Speech on English and Indo-Aryan Languages with BERT and Ensemble learning, in: Forum for Information Retrieval Evaluation (Working Notes)(FIRE), CEUR-WS. org, 2021.

[13] A. Kumar, P. K. Roy, S. Saumya, An Ensemble Approach for Hate and Offensive Language Identification in English and Indo-Aryan Languages, in: Forum for Information Retrieval

Evaluation (Working Notes)(FIRE), CEUR-WS. org, 2021.
[14] Nayel, Hamada and Shashirekha, H., DEEP at HASOC2019 : A Machine Learning Framework for Hate Speech and Offensive Language Detection, 2019, pp. 336–343.
[15] S. Dowlagar, R. Mamidi, Hasocone@ fire-hasoc2020: Using BERT and Multilingual BERT Models for Hate Speech Detection, in: arXiv preprint arXiv:2101.09007, 2021.
[16] A. Hegde, H. L. Shashirekha, Urdu Fake News Detection Using Ensemble of Machine Learning Models, in: Forum for Information Retrieval Evaluation (Working Notes)(FIRE), CEUR-WS. org, 2021.
[17] K. Girish, A. Hegde, F. Balouchzahi, S. Lakshmaiah, Profiling Cryptocurrency Influencers with Sentence Transformers, in: Working Notes of CLEF, 2023.