KBCNMUJAL@HASOC-Dravidian-CodeMix-FIRE2020: Using Machine Learning for Detection of Hate Speech and Offensive Code-Mixed Social Media text

Varsha Pathak[a], Manish Joshi[b], Prasad Joshi[c], Monica Mundada[d] and Tanmay Joshi[e]

[a] *Institute of Management and Research, Jalgaon, Affil- KBC North Maharashtra University, Jalgaon MS India*
[b] *School of Computer Sciences, KBC North Maharashtra University, Jalgaon, MS India*
[c] *JET's Z. B. College,DhuleAffil- KBC North Maharashtra University, Jalgaon, MS India*
[d] *4Department of Dizitilization, Copenhagen Business School, Denmark*
[e] *Brihan Maharashtra College of commerce, Pune, MS India*

Abstract

This paper describes the system submitted by our team, KBCNMUJAL, for Task 2 of the shared task "Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC)" at Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India. The datasets of two Dravidian languages viz Malayalam and Tamil of size 4000 observations, each were shared by the HASOC organizers. These datasets are used to train the machine using different machine learning algorithms, based on classification and regression models. The datasets consist of tweets or YouTube comments with two class labels "offensive" and "not offensive". The machine is trained to classify such social media messages in these two categories. Appropriate n-gram feature sets are extracted to learn the specific characteristics of the Hate Speech text messages. These feature models are based on TFIDF weights of n-gram. The referred work and respective experiments show that the features such as word, character and combined model of word and character n-grams could be used to identify the term patterns of offensive text contents. As a part of the HASOC shared task, the test data sets are made available by the HASOC track organizers. The best performing classification models developed for both languages are applied on test datasets. The model which gives the highest accuracy result on training dataset for Malayalam language was experimented to predict the categories of respective test data. This system has obtained an F1 score of 0.77. Similarly the best performing model for Tamil language has obtained an F1 score of 0.87. This work has received 2nd and 3rd rank in this shared Task 2 for Malayalam and Tamil language respectively. The proposed system is named HASOC_kbcnmujal.

Keywords

Support Vector Classifier, Multinomial Bayes, LR, Random Forest Classifier, n-gram model, Text Classification

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India
varsha.pathak@imr.ac.in (V. Pathak); joshmanish@gmail.com (M. Joshi); sayprasaadajoshi@gmail.com (P. Joshi); monicamundada5@gmail.com (M. Mundada); joshi.tanmay09@gmail.com (T. Joshi)

1. Introduction

Social media has become a modern channel of public expression for the people irrespective of the socio-economic boundaries. Due to the pandemic situations, even the common people have started looking at social media as a formal medium to remain connected with masses. Though such mediums are available majorly for constructive and creative expressions, these days it is found to be in many negative and offensive expressions. Some people take disadvantage of the language based social boundaries. They use absurd and hateful speech using their native languages to hurt other communities. Some people, using hate speech or offensive language intentionally or unintentionally, can hurt some popular person, specific community or even innocent people. If not detected in time, such messages could damage social health. Ignoring such messages could push certain unhealthy issues which could turn into disastrous events at a certain point of time in future. There are cases that such offensive comments have brought serious threats to the community disturbances. To identify such content is today's need and significant work has been done for the English language.

1.1. Transliterated Text

India is a multi-state, multilingual nation. Each state has its own official spoken language and respective script.
The transliterated, Romanized text of the native language with English as the binding language is termed as Code-mixed Text. Many people use code-mixed text to create their social media contents. Most of the southern Indian languages have their origin in Dravidian language. Due to this the languages such as Malayalam of Kerala state, Tamil of Tamilnadu, Telugu of Andhra Pradesh are commonly called as Dravidian Languages. It's a challenge for the research community to trace and restrict such offensive content from the native code-mixed text messages of Dravidian languages.
The **"Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC)"**, at Forum for Information Retrieval Evaluation, 2019 [1] is the first such initiatives as a shared task on offensive language. The HASOC track has further introduced the shared task on Dravidian Code-Mixed text in "Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India" [2], including Malayalam and Tamil languages. The goal of Task 2 is to classify the tweets, into offensive or not-offensive categories.
This paper presents development and implementation of our model for Task 2. The Task 2 challenge provides the respective datasets in Romanized transliterated code forms. The challenges and related study of transliterated search in the context of Indian languages, with different native scripts, could be found in literature like [3]. The HASOC problem could be found as an extension to the transliterated search. Similarly, on social media, people use Short Messages Systems (SMS). Thus "SMS based transliterated code", is the upcoming challenge of Natural Language Processing (NLP) [4]. With the interest to work on different extensions and applications of this recent area of NLP, our team has participated in this HASOC shared Task 2.
In this section we have discussed on the need and relevance of HASOC problem. The related work on the concept of Hate Speech and Offensive Language detection is discussed in the following subsection. The problem statement, a brief information of Hate Speech Dataset and

methodology applied in the work has been discussed in the 2nd section. This is followed by two sections related to the experimental work and result analysis respectively.

1.2. Related Work

Many researchers have published their work on automated detection of hate speech and offensive content. Malmasi and Zampieri[5] has adopted a linear support vector classifier on word skip-grams, brown cluster and surface n-grams. Arup Baruah et al.[6] works on support vector machine, BiLSTM and neural network models on the TF-IDF features of character and word n-grams, embeddings from language models. Glove and fastText embeddings. Saroj et al. [7] applies traditional machine learning classifiers: Support Vector Machine, XGBOOST. Nemanja Djuric et al.[8] works on paragraph2vec and Continuous Bag of Words to approach the neural language model.
The related study shows that a significant work has been done on detecting hate speech in other than the English language. The approach of system developed by Mubarak et. al.[9] is based on Seed Words, word unigrams, word bigrams. This system detects abusive language in Arabic social media. Another work done by Su et.al.[10] detects and rephrases profanity in Chinese. Bharathi et.al.[11] uses many machine learning classifiers to determine the sentiments from Malayalam-English, code-mixed data.

2. Problem Definition

We propose a model that uses best performing classifier on the given dataset. Best resulting features are used by extracting language specific and language independent characteristics of the tweeter dataset. The approach applied for the development of this model is explained in this section. The statistical details of Hate Speech Dataset is described in the next subsection. This is followed by the methodology and experimental work explained in respective subsections.

2.1. Hate Speech Dataset

The Task 2 dataset which has been released for the HASOC shared task of as discussed above, is consisting of two CSV files of tweets. First is of Manglish (Malayalam + English) and other one is of Tamglish (Tamil+ English) [12]. This training dataset has 3 columns with column names as Tweet ID, Tweet text or YouTube comments and the Label respectively. The Label column has values either **OFF**, indicating offensive text or **NOT**, indicating non-offensive tweet. The number of tweets in each file is around 4000. This is a good number to carry the experimental work of training the machine by applying appropriate machine learning algorithms. The Test dataset is then released in later part of **HASOC Task 2** shared task. The test data set is consisting of only first two columns. The third column i.e. Label column is missing. The machine after the training in first phase has to predict the labels of the respective tweets. Approximately, 1000 tweets are available in this dataset for both the languages. Table 1, presents the statistical data about this Training and Test Data Set for both the Malayalam and Tamil languages. The details of how these datasets are constructed and the overview of the shared HASOC tasks is available in [13].

Table 1
Training and Test Set Statistics for Malayalam & Tamil languages

Indian Language	Dataset	Type of tweet	%	Total
Malayalam	Training	Not Offensive	2047 (51.18%)	4000
Malayalam	Training	Offensive	1953 (48.82%)	
Malayalam	Test	Not Offensive	Not Known	1000
Malayalam	Test	Offensive	Not Known	
Tamil	Training	Not Offensive	2020 (50.5%)	4000
Tamil	Training	Offensive	1980 (49.5%)	
Tamil	Test	Not Offensive	Not Known	940
Tamil	Test	Offensive	Not Known	

2.2. Methodology

In the experimental work, a supervised machine learning approach is applied. The tweets' from the labeled dataset are preprocessed to remove noisy elements from its text contents. Appropriate feature extraction model is developed that enables the machine to learn offensive/non-offensive terms and respective patterns in the text.
Finally, the performance of different classifiers based on extracted features are compared using standard measures to develop our best performing proposed model. We named this proposed system as HASOC_kbcnmujal system. Figure 1, shows the architectural view of the machine learning approach of this system.

2.2.1. Data Pre-processing

In general, the social media users enjoy flexibility in forming their tweets or comments. They would not worry about applying any specific grammar of respective languages. Generally, they use their native language with casual expressions. Due to this flexibility in code-mixed text[13], an appropriate pre-processing method is the important concern.
In our system unnecessary and stop words are removed. Similarly, white spaces, digits, special characters [@,#,%,$,;(,)], extra spaces etc. are eliminated to simplify the text messages. In addition to this the special tags like @USER, @RT(retweet) and TAG are also removed.
Thus the tweet's text data is cleaned by applying appropriate Pre-processing as above. As a result both the Manglish and Tamglish datasets are ready for the feature extraction phase which is the next phase of the HASOC_kbcnmujal system.

2.2.2. Features Extraction

For feature extraction we applied two major methods viz. TF-IDF and custom word embedding methods. The details of these methods are given below.

- **Using TF-IDF**: The Term Frequency (TF) and Inverse Document Term Frequency (IDF) are the two important measures that reflect the specificity and relevance of terms with the information carried by the documents. These n-grams are useful to capture the small

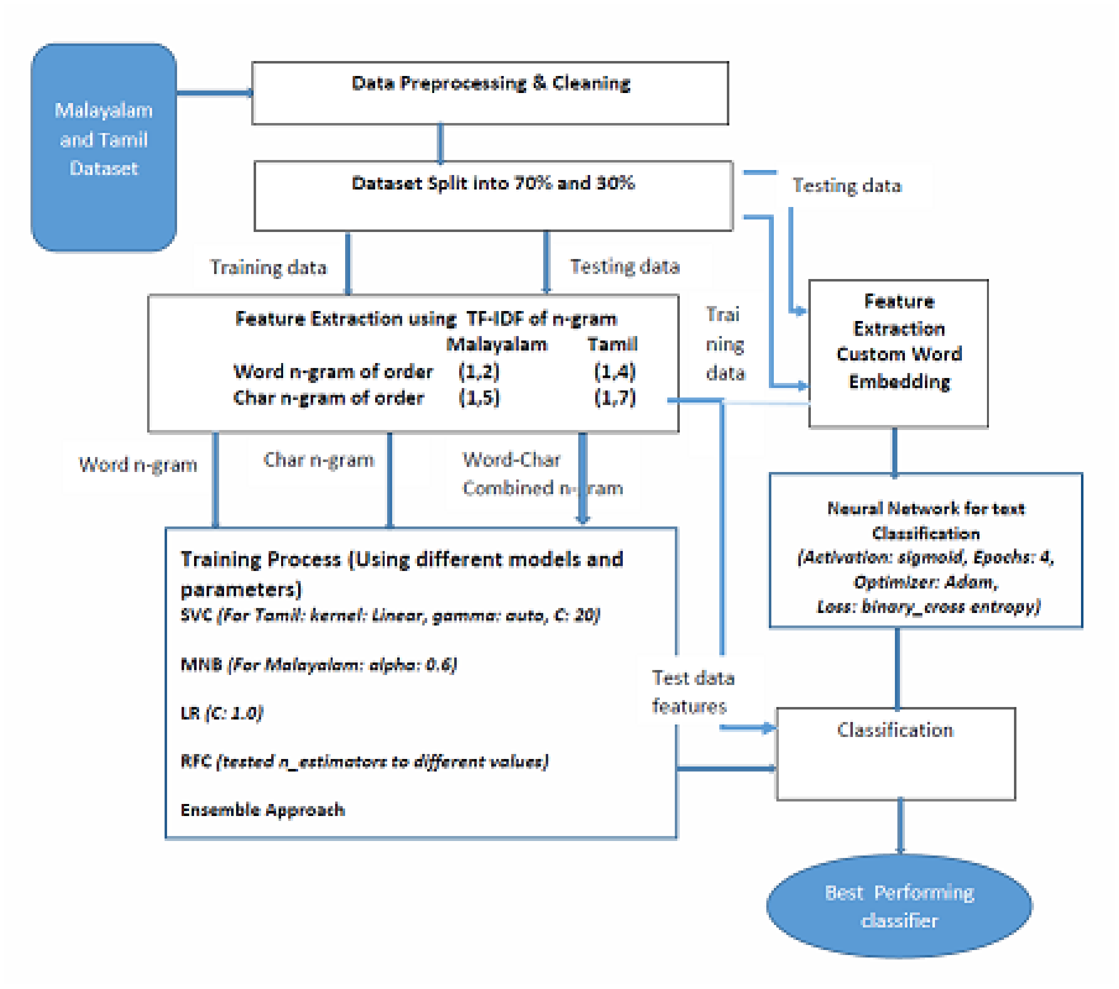
Malayalam and Tamil Dataset
Data Preprocessing & Cleaning
Dataset Split into 70% and 30%
Testing data
Training data
Testing data
Feature Extraction using TF-IDF of n-gram
Malayalam Tamil
Word n-gram of order (1,2) (1,4)
Char n-gram of order (1,5) (1,7)
Trai ning data
Feature Extraction Custom Word Embedding
Word n-gram
Char n-gram
Word-Char Combined n-gram
Neural Network for text Classification (Activation: sigmoid, Epochs: 4, Optimizer: Adam, Loss: binary_cross entropy)
Training Process (Using different models and parameters)
SVC (For Tamil: kernel: Linear, gamma: auto, C: 20)
MNB (For Malayalam: alpha: 0.6)
LR (C: 1.0)
RFC (tested n_estimators to different values)
Ensemble Approach
Test data features
Classification
Best Performing classifier

Figure 1: HASOC_kbcnmujal system: An architectural view

and localized syntactic patterns within text in flexible language.
We have applied three variations of TF-IDF weights n-gram model. "Word n-grams of order(1, n)", "Character n-grams of order(1, n)" and the "Combined both these Word-Char n-grams". The value of n could be 2 or more. These three TF-IDF n-gram models produced 38536, 81191 and 119727 features respectively in case of Malayalam. Similarly in case of the Tamglish, all the above three feature models with minor variation has produced 117173, 325902 and 443075 features respectively. Figure 1, presents this feature data of both the languages.

- **Using Custom Word Embedding**: We apply another feature model, that is termed as "Custom Word Embedding". For this, we have extracted 15430, 15292 unique words from Malayalam and Tamil dataset respectively. The length of longest sentence i.e. MaxLen is considered for the custom embedding of all the sentences. For the given dataset, MaxLen("Malayalam")= 65 and MaxLen("Tamil")=64. In the next step the length of all the sentences is modified to MaxLen. To achieve this, zeros are appended at the end of each sentence using pad_sequences method from *Keras*[1] . This has resulted in 13,000 dimensional vector for Malayalam and 12800 dimensional vector for Tamil.

2.2.3. Classifier Models

We have adopted various classifiers like SVC, MNB, LR, AdaBoost, DTC and RF. The above mentioned features are extracted from the training data. For Neural Network model, we have used the "custom word embedding" feature set [14]. The 70% of the given training datasets is used to train the Classifier. We have evaluated the performance of these classifiers by measuring the accuracy of the result of classification. For this the remaining 30% of the given training dataset has been considered as the test data. The parameters of each classifier are varied to find the best performing parametric values. For training and evaluation of these classifiers *sklearn*[2] is implemented. From this experimental work, a few well performing classifiers are described in next section.

3. Experimental work

To develop a best performing model for the HASOC Task 2 problem, the supervised machine learning approach is applied. The designed machine is trained by using different set of extracted features. Various classifier algorithms are experimented by using these feature sets. The hyper parameters of each classifier is tuned to found the best performing parameters of respective classifier. For both languages similar method is applied. For evaluation purpose appropriate measures viz. accuracy, precision, recall and F1-score are applied. We now discuss the experimental set up of some selected classifiers in this section, as below.

- **Support Vector Classifier (SVC)**: We used Linear SVC. SVC fits the data, and returns a best fitting hyper-plane that divides the data points in two categories. It scales to a large number of samples and has more flexibility in the choice of penalties and loss function [15]. For Malayalam as well as for Tamil, we trained by applying all the three TF-IDF n-gram feature extraction models as discussed in previous section. For both the languages, we tuned the parameters: kernel with ‘linear’, ‘rbf’ and gamma with ‘auto’, ‘scale’. The L2 regularization is used and the hyper-parameter ’C’ is also tuned with these features. For Tamil language, SVC is trained using character n-gram, which gives best result. The Hyper-parameters kernel is set to ‘linear’, gamma is set to ‘auto’ and ’C’ is set to 20. For both languages, we found that the "character n-grams" feature model, has increased the accuracy of the classifier as compared to that of "word n-gram" and "Combined Word-char n-gram" feature models.

[1]https://keras.io/
[2]http://scikit-learn.org/

- **Multinomial Naive Bayes (MNB)**: This is a probabilistic model and specialized version of Naive Bayes. Simple Naive Bayes represents a document with the presence or absence of a particular word, whereas Multinomial Naive Bayes explicitly represents the document with the word counts and adjusts the underlying calculations to identify them from the document set [16]. It works very well on small amounts of training data and gets trained relatively fast compared to other models. We have trained MNB by using the same set of TF-IDF features on both languages tweets datasets as mentioned above. The hyper-parameter "alpha" has been set to 0.6. The "combined word-character n-gram" feature model gives best accuracy in case of both the Manglish and Tanglish datasets.

- **Logistic Regression (LR)**:This is a statistical model. It transforms its output into a probability values which can be mapped to two or more discrete classes. LR is used to conduct regression when the dependent variable is binary. We have trained LR in the same way as that of SVC and MNB. L2 regularization is used with the hyper-parameter "C" is set to default value="1.0". The respective experimental work shows higher accuracy for the "Combined word-character n-gram feature model", for both languages.

- **Ensemble Approach**: We used a hard voting approach. Hard voting sums the predicates for each class label from multiple models and predict the class label with maximum votes. We combined the predictions of our top three models: SVM, MNB and LR. The "Combined word-char n-gram feature", gives best accuracy for datasets of both languages.

- **Random Forest Classifiers (RFC)**: It is an ensemble approach combining multiple decision trees and producing them randomly without defining the rules [10]. We keep all the hyper-parameters to their default setting. Only we have tested n-estimators to different values. We have found that "character n-gram" feature and "combined word-char n-gram" features have near about same accuracy for both the languages.

- **Neural Network for Text Classification**: A Simple text classification neural network model is created using Python's *Keras Library*. Keras is one of the most famous and commonly used deep learning library. It can be used to learn custom words embedding or can be used to load "pre-trained word embedding". We have used Keras Sequential model and have added the embedding layer as its first layer. "Keras embedding layer", takes 3 parameters as arguments as shown below.

  keras.layers.Embedding(size_of_vocabulary, number_of_word_dimensions, length_of_longest_sentence)

  Here, for both datasets, size_of_vocabulary is nothing but the number of unique words which are 15430(Malayalam) & 15292(Tamil). At the time of training the model, we have

Table 2
Results of models

	Results for Malayalam Language			Results for Tamil Language		
Classifier	Features	Acc	F1	Features	Acc	F1
SVC	Word n-gram (1,2)	0.71	0.68	Word n-gram (1,4)	0.84	0.83
SVC	Char n-gram (1,5)	0.75	0.72	Char n-gram (1,7)	0.87	0.86
SVC	Combine Char & Word n-gram	0.74	0.72	Combine Char & Word n-gram	0.86	0.85
MNB	Word n-gram (1,2)	0.71	0.68	Word n-gram (1,4)	0.84	0.83
MNB	Char n-gram (1,5)	0.74	0.72	Char n-gram (1,7)	0.84	0.84
MNB	Combine Char & Word n-gram	0.76	0.74	Combine Char & Word n-gram	0.85	0.85
LR	Word n-gram (1,2)	0.70	0.65	Word n-gram (1,4)	0.84	0.83
LR	Char n-gram (1,5)	0.74	0.71	Char n-gram (1,7)	0.85	0.84
LR	Combine Char & Word n-gram	0.75	0.72	Combine Char & Word n-gram	0.86	0.85
RFC	Word n-gram (1,2)	0.66	0.62	Word n-gram (1,4)	0.80	0.79
RFC	Char n-gram (1,5)	0.69	0.64	Char n-gram (1,7)	0.82	0.81
RFC	Combine Char & Word n-gram	0.70	0.65	Combine Char & Word n-gram	0.81	0.80
Ensemble	Word n-gram (1,2)	0.70	0.65	Word n-gram (1,4)	0.85	0.84
Ensemble	Char n-gram (1,5)	0.74	0.70	Char n-gram (1,7)	0.85	0.84
Ensemble	Combine Char & Word n-gram	0.75	0.72	Combine Char & Word n-gram	0.86	0.85
NN Model	Custom Word Embedding	0.52	0.47	Custom Word Embedding	0.53	0.52

rounded these values to 15450 & 15300 respectively.
The Second parameter number_of_word_dimensions represents each word as a 200 dimensional vector. And the third parameter length_of_longest_sentence is length of longest sentence from dataset i.e. 65(Malayalam) and 64(Tamil).
At the embedding layer, we got 30,90,000 trainable parameters for Malayalam and 30,60,000 for Tamil. The output of the embedding layer has its sentences with each word represented by a 200 dimensional vector. We flattened this embedding layer to get 13000 and 12800 dimensional vectors for Malayalam & Tamil respectively. The second layer i.e. dense layer has 1 neuron. Since ours is a binary classification problem, for both languages, we use the **Sigmoid** function as loss function at the dense layer, 4 epochs. Adam optimizer and the binary cross-entropy loss function were used for training.

4. Results and Discussion

For evaluating the performance of our model, 70% of the dataset is used for training and 30% of the dataset is used as the test dataset. For both the languages we used the same strategy. Table 2, shows the accuracy F1 score of all classifiers for both the Malayalam and the Tamil language. As shown in Table 2, for Malayalam, MNB has an F1 score of 0.78 for "not offensive" tweets. For "offensive" tweets, it has the highest F1 score of 0.74. **Hence MNB stood out as the best classifier**. The second best F1 score (0.72), for "offensive" tweets, is same for all SVC, LR and Ensemble. Among all the classifiers, performance of SVC & LR are same for "offensive" and "not offensive" tweets. RFC performs very low in predicting both the categories. Simple NN Model has scored 0.52.

Table 3
Confusion matrix for “Not Offensive” “Offensive” tweets in the Malayalam language

	SVC Word n-grams		SVC Char n-grams		SVC Combined Word&Char		MNB Word n-grams		MNB Char n-grams		MNB Combined Word&Char	
NOT	484	140	500	124	493	131	491	133	479	145	499	125
OFF	208	368	180	396	178	398	211	365	171	405	162	414

Table 4
Confusion matrix for “Not Offensive” “Offensive” tweets in the Tamil language

	SVC Word n-grams		SVC Char n-grams		SVC Combined Word&Char		MNB Word n-grams		MNB Char n-grams		MNB Combined Word&Char	
NOT	484	140	500	124	493	131	491	133	479	145	499	125
OFF	208	368	180	396	178	398	211	365	171	405	162	414

In the same Table 2, it could be observed that for the Tamil data,**SVC has the highest F1 score 0.87 for “not offensive” and 0.86 for “offensive” tweets**. The performances of MNB, LR and Ensemble are identical. RFC has also performed well for Tamil language. Simple NN Model does not show any improvement for Tamil and has just scored 0.53.
Table 3, presents the confusion matrices with respect to the best performing classifiers and their different (feature based) variations on Malayalam language' data. The table clearly shows that the "Combined word-char n-grams" features, effectively allow MNB to predict “not offensive” & “offensive” tweets. Using these features SVC can predict high predictions of “not offensive” tweets, but fail to predict “offensive” tweets correctly. Interestingly the "combined Word-char" and "word n-gram" features, have increased the performance for most of the classifiers except SVC for “not offensive” class and for RFC in case of “offensive” class.

Similarly, Table 4 presents the confusion matrix result of Tamil language. Performance of MNB, for predicting “offensive” category tweets is high for all feature models. As we can see, the results of "Char n-grams" and that of "Combined word-char n-gram" features for SVC are marginally the same, the "Char n-gram model" has proved to be best for predictions both classes.

5. Conclusion

This paper presents the experimental work and the results of the HASOC Task 2 to detect offensive content in code-mixed dataset of Dravidian languages. We have used different features like word n-gram, character n-gram and combined word-character n-grams and custom word embedding. Custom word embedding is used to train a simple neural network model. Applying a systematic Supervised Machine Learning approach we have developed our HASOC_kbcnmujal system. This system has obtained an F1 score of 0.77 for Malayalam language and has received, the rank of 2nd in HASOC shared task (Task 2) competition. For the Tamil language Model the system has obtained F1 score of 0.87 and has received 3rd rank in the competition.

This work will be further extended to develop a system that could learn offensive terms from the text contents or even from speech irrespective of the language. We are interested in revealing hidden negative messages from the social media comments which may be presented superficially as positive message.
The content of social messages that can damage the social and communal health shall be detected and cured at the right time.

References

[1] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, FIRE '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 14–17. URL: https://doi.org/10.1145/3368567.3368584. doi:10.1145/3368567.3368584.

[2] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text, in: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India, 2020.

[3] V. Pathak, M. Joshi, Itrans encoded marathi literature document relevance ranking for natural language flexible queries, in: Computer Networks & Communications (NetCom), Springer, 2013, pp. 417–424.

[4] V. M. Pathak, M. R. Joshi, Relevance feedback mechanism for resolving transcription ambiguity in sms based literature information system, in: Smart Intelligent Computing and Applications, Springer, 2019, pp. 527–542.

[5] S. Malmasi, M. Zampieri, Detecting hate speech in social media, in: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, INCOMA Ltd., Varna, Bulgaria, 2017, pp. 467–472. URL: https://doi.org/10.26615/978-954-452-049-6_062. doi:10.26615/978-954-452-049-6_062.

[6] A. Baruah, F. A. Barbhuiya, K. Dey, IIITG-ADBU at HASOC 2019: Automated hate speech and offensive content detection in english and code-mixed hindi text, in: P. Mehta, P. Rosso, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019, volume 2517 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 229–236. URL: http://ceur-ws.org/Vol-2517/T3-7.pdf.

[7] A. Saroj, R. K. Mundotiya, S. Pal, Irlab@iitbhu at HASOC 2019: Traditional machine learning for hate speech and offensive content identification, in: P. Mehta, P. Rosso, P. Majumder, M. Mitra (Eds.), Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019, volume 2517 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 308–314. URL: http://ceur-ws.org/Vol-2517/T3-17.pdf.

[8] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, N. Bhamidipati, Hate speech detection with comment embeddings, in: WWW (Companion Volume), 2015, pp. 29–30. URL: https://doi.org/10.1145/2740908.2742760.

[9] H. Mubarak, K. Darwish, W. Magdy, Abusive language detection on Arabic social media, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, Vancouver, BC, Canada, 2017, pp. 52–56. URL: https://www.aclweb.org/anthology/W17-3008. doi:10.18653/v1/W17-3008.

[10] H.-P. Su, Z.-J. Huang, H.-T. Chang, C.-J. Lin, Rephrasing profanity in Chinese text, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, Vancouver, BC, Canada, 2017, pp. 18–24. URL: https://www.aclweb.org/anthology/W17-3003. doi:10.18653/v1/W17-3003.

[11] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, A sentiment analysis dataset for code-mixed Malayalam-English, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 177–184. URL: https://www.aclweb.org/anthology/2020.sltu-1.25.

[12] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, S. Suryawanshi, N. Jose, E. Sherly, J. P. McCrae, Overview of the track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text, in: Proceedings of the 12th Forum for Information Retrieval Evaluation, FIRE '20, 2020.

[13] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus creation for sentiment analysis in code-mixed Tamil-English text, in: Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), European Language Resources association, Marseille, France, 2020, pp. 202–210. URL: https://www.aclweb.org/anthology/2020.sltu-1.28.

[14] B. R. Chakravarthi, Leveraging orthographic information to improve machine translation of under-resourced languages, Ph.D. thesis, NUI Galway, 2020.

[15] scikit-learn developers (BSD License), 2007 - 2020. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html.

[16] A. McCallum, K. Nigam, A comparison of event models for naive bayes text classification, 1998.

A. Online Resources

GitHub Code available via

- GitHub,