# JudithJeyafreedaAndrew@DravidianLangTech-EACL2021:Offensive language detection for Dravidian Code-mixed YouTube comments

**Judith Jeyafreeda Andrew**
Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France
judithjeyafreeda@gmail.com

## Abstract

Messaging online has become one of the major ways of communication. At this level, there are cases of online/digital bullying. These include rants, taunts, and offensive phrases. Thus the identification of offensive language on the internet is a very essential task. In this paper, the task of offensive language detection on YouTube comments from the Dravidian languages of Tamil, Malayalam and Kannada are seen upon as a mutliclass classification problem. After being subjected to language specific pre-processing, several Machine Learning algorithms have been trained for the task at hand. The paper presents the accuracy results on the development datasets for all Machine Learning models that have been used and finally presents the weighted average scores for the test set when using the best performing Machine Learning model.

## 1 Introduction

With the growing freedom on the internet, facing digital bullying has become a daily phenomenon. Offensive languages can be found everywhere including comments on social media. These text are more often targeted to an individual or to a group. The task presented in this paper is to identify offensive language in YouTube comments and classify them in different categories. The comments are in code-mixed Dravidian languages of Tamil, Malayalam and Kannada. Tamil is a Dravidian language natively spoken by South Asia's Tamil people (Chakravarthi, 2020b). For over 2600 years, Tamil literature has been recorded. Sangam literature, the oldest period of Tamil literature, is dated from ca. 600 BC-300 AD. Among the Dravidian languages, Tamil has the oldest existing literature. Over 55 percent of the epigraphic inscriptions discovered by the Archaeological Survey of India (about 55,000) are in the Tamil language. Malayalam is Tamil's nearest major relative; the two started diverging around the 16th century AD.

Code-Mixing is mixing of two or more language in the same utterance. Many user generated in India are code-mixed (Chakravarthi et al., 2018; Chakravarthi, 2020a). Most of the comments contain multiple types of offensive contents. For this purpose a Multiclass classification method has been adapted to the task. Multiclass text classification is a process of classifying an instance into one of the multiple classes possible. In a multi class classification problem, an instance can belong only to one class. Several Machine Learning algorithms have been well established for Multiclass classification (Thavareesan and Mahesan, 2019, 2020a,b). However, not all of them suit the task at hand. It also has to be noted that Machine Learning algorithms has to be tuned to fit the Dravidian Languages under consideration (Tamil, Malayalam and Kannada). Thus several pre-processing techniques have been proposed for Dravidian Languages and Machine Learning algorithms have been fine tuned to suit the task (Ghanghor et al., 2021b,a; Puranik et al., 2021; Hegde et al., 2021; Yasaswini et al., 2021).

## 2 Related Work

Offensive Language Detection is one of the interesting topics where a lot of research has already been done. However, they have all been language specific. (Yin et al., 2009) has used a supervised learning approach with the context, sentiment and contextual features of the document for identifying harassment on the web. (Dadvar et al., 2013) also use supervised learning techniques with three important features: content-based features, cyberbullying features and user-based features. The content-based are based on the content of the text, the cyberbullying features aim to identify frequently bullied groups such as minority races, religion or physi-



*Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 169–174

cal features, the user-based features exploits the user identity. (Razavi et al., 2010) uses pattern recognition and machine learning methods for offensive language detection. This method extracts features in different conceptual levels and applies a multilevel classification on them. (Spertus, 1997) presents Smokey, which is a system built for automatic recognition of hostile messages. (ming Xu et al.) uses a combination of Text categorization, Role labelling, sentiment analysis and topic modelling for identifying bullying on social media data. (Dinakar et al., 2012) uses common sense reasoning to identify and mitigate cyberbullying. . For this purpose, it uses a common sense knowledge base is used, which permits recognition over a broad spectrum of topics in everyday life. (Dinakar et al., 2012) concentrates on a more narrow range of subject matter associated with bullying like appearance,intelligence, racial and ethnic slurs, social acceptance, and rejection and constructs BullySpace, which is a commonsense knowledge base that encodes particular knowledge about bullying situations.

In this paper, the task of offensive language is approached as a Multiclass classification problem. (Pranckevičius and Marcinkevičius, 2017) compares various machine learning algorithms - Naïve Bayes, Random Forest, Decision Tree, Logistic Regression and support vector machines for the classification of text reviews. The findings indicate that the Logistic Regression for multi-class classification for product reviews is the best method in terms of accuracy. It should also be noted that the overall classification accuracy in combination with uni/bi/tri-gram models increases the average of classification accuracy.

With respect to research works done on Dravidian languages, particularly Dravidian code mixed text, a shared task has been proposed for the task of sentiment analysis of YouTube comments in Dravidian code-mixed text (Mandl et al., 2020). (Chakravarthi, 2020b) presents an improvement of word sense translation for under-resourced languages. It focuses on cleaning the noisy corpus in the form of code-mixed content at word-level based on orthographic information which results in improvement of Dravidian languages. It also proposes to alleviate the problem of different scripts by transcribing the native script into a common representation such as the Latin script or the International Phonetic Alphabet (IPA). (Jeyafreeda, 2020) proposed a Multiclass Classification method, where several Machine Learning algorithms have been adapted to the task of sentiment analysis and based on the accuracy of the algorithms on the development set the best suited technique is chosen for the language and the task. The languages involved are the Dravidian languages of Tamil and Malayalam. The Tamil language performed well with the Naive Bayes algorithm while the Malayalam language performed well with the Logistic Regression technique.

## 3 Data

The data for the shared task of offensive language detection in Dravidian Code-mixed languages (Chakravarthi et al., 2021) is a collection of YouTube comments. The languages used are Tamil, Malayalam and Kannada. The Tamil code-mixed YouTube comments are obtained from (Chakravarthi et al., 2020b). The Malayalam code-mixed YouTube comments are obtained from (Chakravarthi et al., 2020a). The Kannada code-mixed YouTube comments are obtained from (Hande et al., 2020). The classes are "Not-offensive", "offensive-untargeted", "offensive-targeted-individual", "offensive-targeted-group", "offensive-targeted-other", or "Not-in-indented-language". The data set for the Tamil code-mixed YouTube comments has 35,139 instances in the train set, 4,388 instances in the dev set and 4,392 instances in the test set. The data set for the Malayalam code-mixed YouTube comments has 16,010 instances in the train set, 1,999 instances in the dev set and 2,001 instances in the test set. The data set for the Kannada code-mixed YouTube comments has 6216 instances in the train set, 776 instances in the dev set and 778 instances in the test set.

## 4 Pre-processing

The Dravidian languages used needs some pre-processing in order to be able to adapt to machine learning algorithms. The pre-processing used in this paper are as follows:

- Firstly, the words in the script of the Dravidian languages of Tamil, Malayalam and Kannada are replaced by latin text (International Phonetic Alphabet (IPA)).
- Secondly, the emojis are replaced by the words that the emoji represents like happy, sad etc.

- Thirdly, removing stop words and punctuation. For this purpose, python packages for language specific stop words. The $advertools$ and $stopwordsiso$ are used for language specific stopwords.

For the purpose of training a supervised classifier, each YouTube comment in the dataset is represented by a numerical feature vector. One common approach for extracting features from text is to use the bag of words model. In this model, the frequency of the words is taken into consideration, but the order in which they occur is ignored. The Term Frequency, Inverse Document Frequency (tf-idf) measure is calculated for each term in the dataset (individually for Tamil, Malayalam and Kannada).

## 5 Machine Learning

### 5.1 Naïve Bayes

Naïve Bayes is a fairly simple yet powerful for classification. The Naïve Bayes uses conditional probabilities given by the equation 1.

$$P(h|d) = (P(d|h) * P(h))/P(d) \quad (1)$$

Where,

- P(h/d) is the probability of hypothesis h given the data d. This is called the posterior probability.
- P(d/h) is the probability of data d given that the hypothesis h was true.
- P(h) is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.
- P(d) is the probability of the data (regardless of the hypothesis).

### 5.2 Support Vector Machines (SVM)

SVMs are very good classification algorithm. The idea is to identify hyper-planes that will separate the various features. A linear SVM is used in this paper. The classification decision is thus performed as follows:

$$f(x) = sign(W^*.x + b^*) \quad (2)$$

where x represents the input feature, W represents the model weight and b represents the bias. For the multi-class classification problem, a one-vs-rest (also known as one-vs-all) approach is used. It involves splitting the dataset into multiple binary classification problems. Thus a binary classification boundary is constructed to train each binary SVMs and the one with the highest confidence is used to solve the multi-class classification problem.

### 5.3 $K$ Nearest Neighbor(KNN)

As the name suggests, the ”neighbor” plays a very important role. This algorithm calculates the distance between the new data point and the other data points. The data points with the shortest distances are selected and the new data variable is then assigned to the class with the most number of close neighbors. $K$ refers to a the number of data points with which the comparisons of distance is performed.

### 5.4 Decision Trees and Random Forests

A decision tree is the diagrammatic representation of classification. Decision trees are made through a flow-chart like structure whose:

- Internal node symbolizes an attribute
- Each branch symbolizes the outcome of the test
- Each leaf node symbolizes a class label
- The paths from the root to leaf symbolizes classification rules

Random Forest is a collection of large number of individual decision trees. Every decision tree predicts a class. Following this, each decision tree predicts a class. A vote is performed on all predicted results. The class with the maximum vote is decided on to be the output class. For the training process, the random subspace method is used (i.e) if one or a few features are very strong predictors for the target output, these features will be selected in many of the decision trees. This makes the features more correlated.

### 5.5 Logistic Regression

The well established multi-class logistic regression model is implemented for the task at hand (LR, 2017). The model of logistic regression for a multi-class classification problem forces the output layer to have discrete probability distributions over the possible $k$ classes. This is accomplished by using the softmax function. Given the input vector(z), the

| Model | Parameters |
|---|---|
| SVM | classifier=SVC; C=[0.001, 0.01, 0.1, 1, 10]; cv=3; n_jobs=4 |
| KNN | n_neighbors= [3,5,7,9];weights=['uniform', 'distance'] |
| Logistic Regression | multi_class='auto'; solver='newton-cg' |
| Naïve Bayes | alpha=0.7 |
| Decision Trees | max_depth=800; min_samples_split=5 |
| Random Forest Classifier | max_depth=800; min_samples_split=5 |

Table 1: Implementation details for the various machine learning models

| Model | Accuracy | Language |
|---|---|---|
| SVM | 0.67 | Tamil |
| KNN | 0.60 | Tamil |
| Logistic Regression | 0.67 | Tamil |
| Naïve Bayes | 0.01 | Tamil |
| Decision Tree | 0.51 | Tamil |
| Random Forest Classifier | 0.66 | Tamil |
| SVM | 0.94 | Malayalam |
| KNN | 0.92 | Malayalam |
| Logistic Regression | 0.93 | Malayalam |
| Naïve Bayes | 0.89 | Malayalam |
| Decision Tree | 0.92 | Malayalam |
| Random Forest Classifier | 0.93 | Malayalam |
| SVM | 0.64 | Kannada |
| KNN | 0.61 | Kannada |
| Logistic Regression | 0.62 | Kannada |
| Naïve Bayes | 0.59 | Kannada |
| Decision Tree | 0.60 | Kannada |
| Random Forest Classifier | 0.63 | Kannada |

Table 2: Accuracy of the different models.

softmax function works as follows:

$$softmax(z) = \frac{e^z}{\sum_{i=1}^{k} e^{z_i}} \quad (3)$$

At this point, there are $k$ outputs and thus there is a necessity to impose weights connecting each input to each output. The model thus is as follows:

$$\hat{y} = softmax(xW + b) \quad (4)$$

where, W is the weight matrix between the input and output, x being the input and b is the bias.

## 6 Implementation

The sklearn[1] package in Python is used for the feature extraction and model training. The models of logistic regression, linear support vector classification, multinomialNB, KNN, Decision Trees and random Forest provided by the sklearn toolkit are used for the training of the machine learning models discussed in section 5. The implementation details for these models are shown in table 1.

Every model described in section 5 is trained using the training sets for the three Dravidian Languages of Tamil, Malayalam and Kannada. The accuracy of each model for the three languages are calculated. The accuracy of the model is calculated using the development set. The accuracy of each model for the different languages are presented in table 2.

As seen from table 2, all models are quite closer to each other in terms of accuracy. However, the highest accurate model to use for the task of offensive language detection of YouTube comments in all three Dravidian languages(Tamil, Malayalam and Kannada) is the Support Vector Machine(SVM) classifier. From table 2, the Logistic Regression model for the Dravidian language of Tamil has the same accuracy as the SVM classifier. As the SVM classifier has the highest accuracy for the Malayalam and Kannada language, SVM classifier is still used for the Tamil language as well.

## 7 Results and Conclusions

The weighted averages for the precision, recall and F-score for the task at hand is shown in table 3. A precision of 0.54, a recall of 0.73 and a F1-score of 0.61 is achieved by the method presented in this paper for the Tamil language. A precision of 0.94, a recall of 0.94 and a F1-score of 0.93 is achieved by the method presented in this paper for the Malayalam language. A precision of 0.66, a recall of 0.67 and a F1-score of 0.63 is achieved by the method presented in this paper for the Kannada language. The same model, SVM, has been used for the offensive language detection for all three Dravidian languages of Tamil, Malayalam and Kannada. The Malayalam language has the highest value of Precision and Recall. On the other hand, Tamil language

[1] https://sklearn.org/

| Language | Precision | Recall | F1-score | Algorithm |
|---|---|---|---|---|
| Kannada | 0.66 | 0.67 | 0.63 | SVM |
| Tamil | 0.54 | 0.73 | 0.61 | SVM |
| Malayalam | 0.94 | 0.94 | 0.93 | SVM |

Table 3: Results.

has the lowest Precision value but the Recall value for the Tamil language is still high. The size of training data for the Tamil language is much higher than that of the other languages (Malayalam and Kannada). The lower precision and recall values for the Tamil language could be the result of over fitting.

Future directions of research would include using deep learning methods for the task at hand.

## References


2017. Multiclass logistic regression from scratch.

Bharathi Raja Chakravarthi. 2020a. HopeEDI: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2020b. *Leveraging orthographic information to improve machine translation of under-resourced languages*. Ph.D. thesis, NUI Galway.

Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae. 2018. Improving wordnets for under-resourced languages using machine translation. In *Proceedings of the 9th Global Wordnet Conference*, pages 77–86, Nanyang Technological University (NTU), Singapore. Global Wordnet Association.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020a. 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)*, pages 177–184, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020b. 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)*, pages 202–210, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan V, Elizabeth Sherly, and John Philip McCrae. 2021. Findings of the shared task on Offensive Language Identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. pages pp 693–696.

Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Trans. Interact. Intell. Syst.*, 2(3).

Nikhil Kumar Ghanghor, Parameswari Krishnamurthy, Sajeetha Thavareesan, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2021a. IIITK@DravidianLangTech-EACL2021: Offensive Language Identification and Meme Classification in Tamil, Malayalam and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, Online. Association for Computational Linguistics.

Nikhil Kumar Ghanghor, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021b. IIITK@LT-EDI-EACL2021: Hope Speech Detection for Equality, Diversity, and Inclusion in Tamil, Malayalam and English. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, Online.

Adeep Hande, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2020. KanCMD: Kannada CodeMixed dataset for sentiment analysis and offensive language detection. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 54–63, Barcelona, Spain (Online). Association for Computational Linguistics.

Siddhanth U Hegde, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. UVCE-IIITT@DravidianLangTech-EACL2021: Tamil Troll Meme Classification: You need to Pay more Attention. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Judith Jeyafreeda. 2020. JudithJeyafreeda@Dravidian-CodeMix-FIRE2020:Sentiment Analysis of YouTube Comments for Dravidian Languages.

Thomas Mandl, Sandip Modha, Anand Kumar M, and Bharathi Raja Chakravarthi. 2020. Overview of the

HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German. In *Forum for Information Retrieval Evaluation*, FIRE 2020, page 29–32, New York, NY, USA. Association for Computing Machinery.

Tomas Pranckevičius and Virginijus Marcinkevičius. 2017. Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing*, 5(2):221.

Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITT@LT-EDI-EACL2021-Hope Speech Detection: There is always hope in Transformers. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Amir H. Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In *Advances in Artificial Intelligence*, pages 16–27, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *Aaai/iaai*, pages 1058–1065.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 320–325.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020a. Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts. In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 272–276.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020b. Word embedding-based Part of Speech tagging in Tamil texts. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 478–482.

Jun ming Xu, Kwang sung Jun, Xiaojin Zhu, and Amy Bellmore. Learning from bullying traces in social media.

Konthala Yasaswini, Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITT@DravidianLangTech-EACL2021: Transfer Learning for Offensive Language Detection in Dravidian Languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Dawei Yin, Zhenzhen Xue, Liangjie Hong, Brian Davison, April Edwards, and Lynne Edwards. 2009. Detection of harassment on web 2.0.