Abusive and Threatening Language Detection in Urdu using Machine Learning Techniques

Karthik Raja Anandan[1], Aarthi Suresh Kumar[1], B Bharathi [1], Bhuvana Jayaraman[1] and Mirnalinee T.T[1]

[1]*Department of CSE*
Sri Sivasubramaniya Nadar College of Engineering,
Chennai, Tamil Nadu, India

Abstract

With the growing number of groups for native people using native scripts the need for an automated model to classify abusive and threatening messages to maintain decorum has become more and more important in social media platforms. The more prevalent models are mostly trained for English scripts and may not work as good as a native classifier trained for this specific purpose. Here in this work we have used classic models from Sklearn library to classify the data given in task HASOC 2021 - Abusive and Threatening language detection in Urdu. It has been observed that the best model for abusive classification was MLP with paraphrase multilang v1 encoding and for threatening language dataset, the best model observed was an *nu*-SVM.

Keywords

Abusive language identification, Threatening language detection, Tf-Idf Vectorization, Transformers, Sentiment analysis, Tokenizer,

1. Introduction

The boom in social media platforms has led to the creation of communities based on various sects and classes and lingual basis. In addition, the rise of native language communities raised the need for detecting threatening and abusive language in their native scripts. The myriad of variations in the meaning for the same scripts in a different language removes the possibility for one model classifier for all languages. This creates a need for classifiers in each language, English being a more widely used language has a lot of good existing classifiers. Roman Urdu, where Urdu is written in English script has also seen a lot of input. Here we have created a model to classify the threatening and abusive nature of a sentence in native Urdu script. The perceptron model, Logistic Regression, SVM, KNN were used to perform this classification task.

FIRE 2021: Forum for Information Retrieval Evaluation, December 13-17, 2021, India
karthikraja19048@cse.ssn.edu.in (K. R. Anandan); aarthi19003@cse.ssn.edu.in (A. S. Kumar);
bharathib@ssn.edu.in (B. B.); bhuvanaj@ssn.edu.in (B. Jayaraman); mirnalineett@ssn.edu.in (M. T.T)
https://www.ssn.edu.in/staff-members/dr-b-bharathi/ (B. B.);
https://www.ssn.edu.in/staff-members/dr-j-bhuvana/ (B. Jayaraman);
https://www.ssn.edu.in/staff-members/dr-t-t-mirnalinee/ (M. T.T)
0000-0001-7279-5357 (B. B.); 0000-0002-9328-6989 (B. Jayaraman); 0000-0001-6403-3520 (M. T.T)

2. Literature Survey

Identifying hate speech in social media is one of the most essential tasks to prevent spreading hatred nowadays. Detection of such hate speech is tedious and the organisations hosting social media platforms are taking steps to prevent the hate speech spreading through their platforms. Several works have been carried out to identify abusive and hate speech in different languages. This section gives the background and state of the current approaches to identify hate speech in Urdu. Generally the structure of the sentence, Bag of words, N-gram features, linguistic features [1] like parts of speech, are used to identify the hate speech. [2], [3], [4] are few of the works where both machine learning techniques and deep learning algorithms have been used for text classification.

Variety of machine learning algorithms such as, Linear regression, SVM, Random Forest, Naive Bayes and SGD classifier are applied on custom Roman Urdu [5] dataset with a 10 fold cross-validation. Among all the listed approaches SVM has reported to give 77.45% of accuracy.

Different models with n-gram pre-processing have been used for Offensive classification in Urdu sentences[6]. In their experiment character trigram-gram preprocessing and simple logistic model proved to be the best model

RUT (Roman Urdu Toxic) corpus has been developed to identify toxic comments in Urdu [1]. 72000 comments are labelled manually with two classes as toxic and non toxic. This work has applied the conventional machine learning approaches along with 3 deep learning techniques namely, modified Convolutional Neural Network, Bidirectional LSTM and Bidirectional GRU. Apart from the individual application of these classifiers, ensemble of all machine learning, all deep learning, best of both learning approaches and ensemble of models have also been reported. Ensemble of all models have observed to attain a F1 score of 86.35%.

Classification of hate speech, target group and aggressiveness are observed [7] in English and Spanish languages. Multinomial Naive Bayes and Logistic Regression are the two classifiers helped in identifying the hate speech in these two languages. The input text are pre processed using stemming and removing stop words and then bag of words with TF and TF-IDF have been used as features.

Propaganda Spotting in Online Urdu Language (ProSOUL) [8] is designed to identify the sources of propaganda in Urdu language. Psycho-linguistic features were extracted using Linguistic Inquiry and Word Count. NEws LAndscape (NELA) along with TF-IDF, N-grams, Word2Vec and BERT features are fed to CNN and Logistic regression classifiers. It is reported that out of all the listed features Word2Vec has outperformed BERT.

Hate Speech Roman Urdu 2020 (HS-RU-20) [9] corpus has been created in order to classify the Roman Urdu tweets into three levels namely, Neutral - Hostile, Simple - Complex, and Offensive classes. Both conventional machine learning classifiers and one deep learning approach, CNN have been applied to detect the offensive ones, where 90% of F1 score has been achieved by Logistic Regression.

Task	Abusive	Not Abusive
Task1(Abusive classification)	1187	1213

Table 1
Training set split for Abusive Task

Task	Threatening	Not threatening
Task2 (Threat classification)	1071	4929

Table 2
Training set split for Threatening Task

3. Datasets

The training dataset for abusive language classification task contains 2400 sentences, out of which 1213 are non-abusive and 1187 sentences are abusive. The test data contains 1100 sentences, out of which 537 are non-abusive and 563 are abusive comments which are given in Table 1. The training dataset for threat detection has 4929 non threatening tweets and 1071 threatening tweets which are given in Table 2. The test dataset for threat contained 3231 non threatening tweets and 719 were threatening tweets.

4. Implementation and Experiments

4.1. MLP Classifier

Multilayer perceptron (MLP) classifier is an multilayer neural network. It uses back propagation the error to tune its weights and learns from the loss function. MLP works for even linearly-unseparable problems. The model used for both the tasks contain 2 hidden layers with 256, 128 neurons respectively, and the neural weights were adjusted through 300 epochs with learning rate of 0.001. The default relu activation function was used for all layers.

4.2. Logistic Regression

Logistic Regression is a classical statistical analysis approach that relies on prior observation. Logistic regression is usually used for classification sort of problems. It uses the sigmoid on the given parameters to perform binomial classification.

$$h_\theta(x) = \frac{1}{1 + e^{-}\theta^T x} \tag{1}$$

4.3. Support Vector Machine

Support Vector Machine (SVM) classification algorithms like C-SVC and its re-parametrized version *nu*-SVC algorithms were trained and tested for different parameters for Kernel, *nu*, C, degree and gamma where C is the regularization parameter used to differentiate the objective of correct classification vs broader gap between hyper planes to account for unseen data; kernel is Linear (a straight hyperplane to divide the classes in 2d plane), Poly (the classes can be marked on higher dimensions that take any shape), RBf (where the classes marked on higher dimensions are governed by a Gaussian radial basis function); *nu* is the Regularization parameter similar to C, is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors relative to the total number of training samples. In this work the best model came for RBF kernel for *nu* value in the range 0.3 to 0.5.

4.4. KNN

K Nearest Neighbors (KNN) classification is a clustering algorithm that labels a point with the class of majority of its neighbours. Our model analyses 50 neighbours for every node to predict the class of the node. Since the training set was not biased towards any one class, this algorithm gave reasonable results for this classification task. The KNN model requires high dimensional vectors as input which was derived using Tf-Idf vectorization method available in the sklearn library. The parameters passed were n_neighbors=50, weights='uniform', algorithm='auto'. The model gave an accuracy of 82% on the validation dataset.

4.5. Feature Extraction

4.5.1. Embeddings for MLP

Two embeddings from the SentenceTransformers, a Python framework for state-of-the-art sentence, text and image embeddings were used to tokenize the sentences. The train data was lemmatized using the lemmatizer in the urduhack, an NLP library for Urdu language. The lemmatized sentences are tokenized using the above mentioned transformers: 'distiluse-base-multilingual-cased-v2' and 'paraphrase-xlm-r-multilingual-v1'. A maximum epochs of 300 and the size of (256,128) were passed as parameters. The results of training the model on encodings of the lemmatized versions of the sentences was better than just training on raw data. This might be because of an internal working of the pretrained models used for tokenizing the sentences.

4.5.2. Tf Idf for KNN, LR, SVM

Tf-Idf Vectorization was used to vectorize the sentences along with a character 10-gram with a max features of 50000. While vectorizing, the sentences are converted to lower case in order to avoid the confusion caused by the case of words in learning the context of the sentence. Lemmatizing the words did not help in improving the accuracy. It might be because the root word of any Urdu word does not necessarily have the same meaning in the context or the

derived word has a more precise meaning which helps the model understand the context of the sentence better. So only the Tf-Idf vector of raw sentences were fed as input to these models.

Computation is done in Python using Google Colab notebook and its GPU was used to train the model. A general purpose RAM size of 8GB was allotted with a 2.3GHz Intel Xenon CPU. Python note books associated with the abusive task is given in the link [1]. Python note books associated with the threatening task is given in the link [2]

The above algorithms with the aforementioned extracted features are tested and the best models and their parameters are tabulated for each task below.

4.6. Performance analysis

The performance of the proposed system for abusive language detection using training data are tabulated in Table 3 and training data performance of threatening language detection is shown in Table 4.

Model	Training Accuracy	F1-Score	ROC_AUC
distiluseMLP	0.82	0.8949	0.8949
KNN	0.82	0.7368	0.7388
Nu-SVCrbf	0.84	0.7983	0.7985
LR	0.83	0.7968	0.7982
MLP-paraphrase	0.81	0.8915	0.8914

Table 3
Performance of Abusive language detection on training data

Model	Training Accuracy	F1-Score	ROC_AUC
distiluseMLP	0.77	0.917	0.842
KNN	0.7991	0.836	0.684
Nu-SVC	0.8038	0.932	0.836
LR	0.8175	0.792	0.567
MLP-paraphrase	0.77	0.926	0.851

Table 4
Performance of threatening language detection on training data

The training performance shows that MLP-paraphrase and Nu-SVC have performed well for Abusive language detection and threatening language detection with 89% and 93.2% respectively. The MLP models trained on the 2 different encodings gave almost similar results on training but paraphrase-xlm-r-multilingual-v1 encoding worked better than the rest on the test data.

The performance of the proposed system for abusive language detection using test data are tabulated in Table 5, threatening language detection performance is tabulated in Table 6.

[1]
[2], https://colab.research.google.com/drive/12Xi55z-mFFO6VbJL1FiZUNeA7Z3q76QE?usp=sharing

Model	private F1	private ROC_AUC	public F1	public ROC_AUC
distiluseMLP	0.722	0.742	0.666	0.709
KNN	0.726	0.723	0.702	0.723
Nu-SVC	0.689	0.687	0.693	0.711
LR	0.723	0.721	0.722	0.734
MLP-paraphrase	0.771	0.757	0.689	0.699

Table 5
Performance of proposed Models on Abusive language detection on test data

Model	private F1	private ROC_AUC	public F1	public ROC_AUC
MLP-distiluse	0.798	0.634	0.817	0.639
KNN	0.738	0.515	0.797	0.539
Nu-SVCrbf	0.800	0.604	0.833	0.611
LR	0.760	0.542	0.815	0.567
MLP-paraphrase	0.805	0.657	0.825	0.661

Table 6
Performance of proposed Models on Threatening language on test data

From Table 5 and Table 6, it has been noted that for both abusive and threatening language detection task, paraphrase-xlm-r-multilingual-v1 embeddings with MLP models produces better results than other approaches.

5. Conclusion

Spreading hatred to the community on the basis of ethnicity, race, religion and gender is a menace to the society. Social media applications nowadays serve as a platform for promoting such abusive and hatred messages. Techniques have to be developed to curtail such abusive messages from spreading. In this work, five machine leaning approaches have been deployed to detect the abusive and threatening Language Task in Urdu Language.

Classical machine learning were able to come close to the MLP for both tasks in terms of F1-scores. SVM model giving better results in threatening model can be used to infer that many Threat words are context independent than compared to the abusive words. This work can be enhanced further by exploring the linguistic features of Urdu and also other deep learning approaches can be employed with fine tuned parameters for this task.

References

[1] H. H. Saeed, M. H. Ashraf, F. Kamiran, A. Karim, T. Calders, Roman urdu toxic comment classification, Language Resources and Evaluation (2021) 1–26.

[2] B. Bharathi, J. Bhuvana, N. N. A. Balaji, Ssncse-nlp@ evalita2020: Textual and contextual stance detection from tweets using machine learning approach, EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020 (2020) 224.
[3] N. N. A. Balaji, B. Bharathi, J. Bhuvana, Ssncse_nlp@ dravidian-codemix-fire2020: Sentiment analysis for dravidian languages in code-mixed text., in: FIRE (Working Notes), 2020, pp. 554–559.
[4] B. Bharathi, M. Anirudh, J. Bhuvana, Bharathi ssn@ inli-fire-2017: Svm based approach for indian native language identification., in: FIRE (Working Notes), 2017, pp. 110–112.
[5] T. Sajid, M. Hassan, M. Ali, R. Gillani, Roman urdu multi-class offensive text detection using hybrid features and svm, in: 2020 IEEE 23rd International Multitopic Conference (INMIC), IEEE, 2020, pp. 1–5.
[6] M. Akhter, Z. Jiangbin, I. Naqvi, M. Abdelmajeed, M. T. Sadiq, Automatic detection of offensive language for urdu and roman urdu, IEEE Access PP (2020) 1–1. doi:10.1109/ACCESS.2020.2994950.
[7] I. Ameer, M. H. F. Siddiqui, G. Sidorov, A. Gelbukh, Cic at semeval-2019 task 5: Simple yet very efficient approach to hate speech detection, aggressive behavior detection, and target classification in twitter, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 382–386.
[8] S. Kausar, B. Tahir, M. A. Mehmood, Prosoul: a framework to identify propaganda from online urdu content, IEEE Access 8 (2020) 186039–186054.
[9] M. M. Khan, K. Shahzad, M. K. Malik, Hate speech detection in roman urdu, ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 20 (2021) 1–19.