Abusive and Threatening Language Detection in Native Urdu Script Tweets Exploring Four Conventional Machine Learning Techniques and MLP

A. Karthikraja [1], Aarthi Suresh Kumar[1], B. Bharathi[1], Jayaraman Bhuvana[1] and T.T Mirnalinee[1]

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

Abstract

The lack of clarity in rules imposed on discussions on social media and the lack of critical eyes on discussions in regional languages, unlike the languages with a greater audience like English, the vulgarity of most of the discussions go unnoticed. This demands an automated model to classify abusive and threatening messages to maintain decorum in social media platforms. 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, Sentence Transformers, Hate Speech Detection, Text Classification, MLP, SVM, Urdu

1. Introduction

The boom in social media platforms has led to a inevitable freedom of self-expression, especially among communities sharing the same native language and script.This rise of access of various native language communities to the means of self-expression via the Internet 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,Roman Urdu, where Urdu is written in English script has seen a lot of input. Here we have created a model to

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

classify the threatening and abusive nature of a sentence in native Urdu script. Independent models based on Multi layer perceptron, Logistic Regression, SVM, KNN were used to perform this classification task.

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[1]. 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.

Variety of machine learning algorithms such as, Linear regression, SVM, Random Forest, Naive Bayes and SGD classifier are applied on custom Roman Urdu [2] dataset with a 10 fold cross-validation. Among all the listed approaches SVM has reported to give 0.774% of accuracy.

Different models with n-gram pre-processing have been used for Offensive classification in Urdu sentences[3]. In their experiment character trigram preprocessing and logistic regression proved to be the best model

Propaganda Spotting in Online Urdu Language (ProSOUL) is designed to identify the sources of propaganda in Urdu language [4] . 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. In [5], describes automatic Abusive Language Detection in Urdu Tweets.

Hate Speech Roman Urdu 2020 (HS-RU-20) corpus has been created in order to classify the Roman Urdu tweets into three classes namely, Neutral - Hostile, Simple - Complex, and Offensive classes [6]. Both conventional machine learning classifiers and CNN have been applied to detect the offensive ones, where an F1-score of 0.90 has been achieved with a Logistic Regression model.

3. Datasets[7]

Classification	Train set	Test set
Abusive	1187	563
Not Abusive	1213	537
Total	2400	1100

Table 1
Categorical data split for for Abusive Task

In [8, 9], the overview of the shared task on threatening and abusive detection in Urdu.

Classification	Train set	Test set
Threatening	1071	719
Non Threatening	4929	3231
Total	6000	3950

Table 2
Categorical data split for Threatening Task

4. Implementation and Experiments

4.1. MLP Classifier

Multilayer perceptron (MLP) classifier is a multilayer neural network. It uses back propagation to tune its weights and learns from the loss function in each iteration. MLP works for even linearly-unseparable problems. The model used for both the tasks contains 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 binary classification.[10]

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

4.3. Support Vector Machine

nu-SVC a Support Vector Machine (SVM) classification algorithms was tested and trained with the given datasets.The best F1-score was achieved for a RBf kernel (where the classes marked on higher dimensions are governed by a Gaussian radial basis function) for regularization parameter 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 both the the classification tasks. The KNN model requires high dimensional vectors as input which was derived using Tf-Idf vectorization method available in the sklearn library version 0.0, default version provided in google colab[11]. The parameters passed were n_neighbors=50, weights='uniform', algorithm='auto'.The same model configuration was used for both the classification tasks.

4.5. Feature Extraction

4.5.1. Embeddings for MLP

Two embeddings from the SentenceTransformers module , a Python framework for state-of-the-art sentence, text and image embeddings were used to fetch corresponding embedding of the training set tweets[12]. The train data was lemmatized using the lemmatizer in urduhack, an NLP library for Urdu language. The lemmatized sentences are transformed to similar embedding using the below mentioned transformers: 'distiluse-base-multilingual-cased-v2' and 'paraphrase-xlm-r-multilingual-v1' and trained separately . 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. Character n-gram gave better results than word n-gram. It might be because of the complex morphology of Urdu that character n-gram works better extracting features from the samples.[13] 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 with 10-word gram was fed as input to these models.

4.6. Hardware specification and link to computation

The training and testing of the models was done in google colab. A general purpose RAM size of 8GB was allotted with a 2.3GHz Intel Xenon CPU was used for training of the above models. Python note books associated with the abusive and Threatening tasks are given in the link. [1]

The above algorithms with the aforementioned extracted features are trained with k fold cross validation and the best models and their parameters are tabulated below for each task.

4.7. 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.

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] https://github.com/ask-1710/Abusive-and-Threatening-Language-Detection-Task-in-Urdu

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, kernel:RBF	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

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-SVC kernel:RBF	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 unintended medium for enabling transfer of 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 explored 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. 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] Z. Laub, Hate speech on social media: Global comparisons (2019). URL: https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons.

[2] 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.

[3] 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.

[4] S. Kausar, B. Tahir, M. A. Mehmood, Prosoul: a framework to identify propaganda from online urdu content, IEEE Access 8 (2020) 186039–186054.

[5] M. Amjad, A. Noman, S. Grigori, Z. Alisa, C.-H. Liliana, G. Alexander, Automatic abusive language detection in urdu tweets, Acta Polytechnica Hungarica (2021).

[6] 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.

[7] M. Amjad, N. Ashraf, A. Zhila, G. Sidorov, A. Zubiaga, A. Gelbukh, Threatening language detection and target identification in urdu tweets, IEEE Access 9 (2021) 128302–128313. doi:10.1109/ACCESS.2021.3112500.

[8] M. Amjad, Z. Alisa, V. Oxana, B. Sabur, A. Hamza Imam, S. Grigori, G. Alexander, Overview of the shared task on threatening and abusive detection in urdu at fire 2021, CEUR Workshop Proceedings (2021).

[9] M. Amjad, Z. Alisa, V. Oxana, B. Sabur, A. Hamza Imam, S. Grigori, G. Alexander, Urduthreat@ fire2021: Shared track on abusive threat identification in urdu, Forum for Information Retrieval Evaluation (2021).

[10] J. Feng, H. Xu, S. Mannor, S. Yan, Robust logistic regression and classification, Advances in neural information processing systems 27 (2014) 253–261.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, Journal of machine learning research 12 (2011) 2825–2830.

[12] N. Reimers, I. Gurevych, Making monolingual sentence embeddings multilingual using knowledge distillation, arXiv preprint arXiv:2004.09813 (2020). URL: http://arxiv.org/abs/2004.09813.

[13] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, M. T. Sadiq, Automatic detection of offensive language for urdu and roman urdu, IEEE Access 8 (2020) 91213–91226. doi:10.1109/ACCESS.2020.2994950.