SSNCSE_NLP@HASOC-Dravidian-CodeMix-FIRE2020: Offensive Language Identification on Multilingual Code Mixing Text

Nitin Nikamanth Appiah Balaji, B. Bharathi

Department of CSE, Sri Siva Subramaniya Nadar College of Engineering,Tamil Nadu, India

Abstract

The number of social media users is increasing rapidly. A myriad of people have started using native languages in Roman alphabets. Therefore, it has becomes a big concern to regulate the quality of the text content and messages that are being shared to the internet. In this paper we study the task of offensive message identification for Tamil-English and Malayalam-English code-mixed content. The char n-gram, TFIDF and fine-tuned BERT are compared in combination with machine learning models such as MLP, Random Forest and Naive Bayes. This work explains the submissions made by SSNCSE_NLP in HASOC Code-mix tasks for Hate Speech and Offensive language detection. We achieve F1 scores of 0.94 for task1-Malayalam, 0.75 for task2-Malayalam and 0.88 for task2-Tamil on the test-set.

Keywords

Machine Learning, Hate Speech Detection, NLP, Offensive language identification, BERT embedding

1. Introduction

With recent development, the technology space has rapidly reduced the cost of devices. The exponential growth of connectivity to the internet even in the most remote parts of the world has encouraged people to use communication services and social media. Social media has even proven to change the status of a country's elections. So it becomes necessary to automate the process of censorship of comments and the messages that are being posted.

As the code-mixed comments and messages consist of native languages mixed with roman alphabets, it becomes difficult to model a generalized solution that interchangeably works with different alphabet set variations and mixes [1, 2]. With recent development in the multilingual unsupervised training have elicited models that could be fine-tuned for code-mixed classification tasks. Also the n-gram based models could learn much easily with a limited amount of data-set, in a short span of time.

In this paper, we present automation models for the identification of hate speech or content in two different Dravidian languages Malayalam and Tamil mixed with English [3, 4]. Tamil and Malayalam belong to Dravidian language family spoken mainly in south of India, Sri Lanka, and Singapore [5]. We have analyzed various techniques such as n-gram and multilingual BERT [6] fine-tuned with added neural network layers. The char n-gram model produced the best

FIRE 2020: Forum for Information Retrieval Evaluation, December 16-20, 2020, Hyderabad, India

EMAIL: nitinnikamanth17099@cse.ssn.edu.in (N. N. A. Balaji); bharathib@ssn.edu.in (B. Bharathi)
ORCID: 0000-0002-6105-0998 (N. N. A. Balaji); 0000-0001-7279-5357 (B. Bharathi)

CEUR Workshop Proceedings (CEUR-WS.org)

Table 1
Data Distribution

Task Description	Data Set	Number of comments		
		Offensive	*Not-Offensive*	*Total*
Ml-En task1	Train-set	567	2,633	3,200
	Dev-set	72	328	400
	Test-set	-	-	400
Ml-En task2	Train-set	1,953	2,047	4,000
	Test-set	-	-	951
Ta-En task2	Train-set	1,980	2,020	4,000
	Test-set	-	-	940

results, with a comparable performance by the BERT model. We achieved an F1 score of 0.94 for the task1 - YouTube comments in Code-mix (a mixture of Native and Roman) Tamil and Malayalam. An F1 score of 0.75 for Manglish and 0.88 for Tanglish is achieved.

The paper expounds on the experiments and the submissions made for the Offensive Language Detection task [7, 8]. The remainder of the paper is organized as follows. Section 2 discusses data-set distribution and techniques implemented to balance classes. Section 3 outlines the features used for the experiments for both task 1 and task 2. Results are discussed in Section 4. Section 5 concludes the paper.

2. Data-set Analysis and Preprocessing

The HASOC data-set is a collection of message-level labeled comments for offensive language detection. It consist of Tamil-English [9] and Malayalam-English [10] YouTube comments. The comments contain writings in Roman lexicons with Tamil/Malayalam grammar or English grammar with Tamil/Malayalam lexicons. Task 1 contains Malayalam-English comments and corresponding **Not Offensive, Offensive** class labels. Task 2 contains two sub-tasks, one for Malayalam-English comments and the other for Tamil-English comments. The train-set, dev-set and test-set distribution with class-wise distribution is shown in 1.

There is a clear imbalance in the data-set distribution. This could cause a bias towards a particular class and the model trained on this data-set would be more inclined towards the dominant class. The comments from the class containing a lesser number of instances are randomly duplicated to get an equal distribution in the training data-set. The final train-set consists of 5,266 comments (2,633 Off and 2,633 Not-Off) for task1. Similar duplication strategy is applied to Task 2 data-set. For task2 Malayalam-English, the re-sampled train-set consists of 4,094 comments (2,047 Off and 2,047 Not-Off) and for task2 Tamil-English, the re-sampled train-set consist of 4,040 comments (2,020 Off and 2,020 Not-Off).

3. Experimental setup and features

For feature extraction, the n-gram model and BERT embedding model are experimented upon. As the content of the comments is a mix of Dravidian language grammar in Roman lexicons along with English grammar, it becomes challenging to find pre-trained models for this context. So a simple n-gram approach is considered. Also, the advancements done by the transformer model for pre-training and the availability of multilingual trained models encourage to experiment with BERT pre-trained embeddings.

The extracted features are used to train machine learning models such as Multi-Layer Perceptron, Random Forest, Naive Bayes, and the performance of these models are compared. For task1 the provided dev-set is used and for task2 4 fold cross-validation is used. The metrics such as accuracy and weighted average F1 score are analyzed. The machine learning model implementations and the metrics of comparison is used from Scikit-learn [1]. The implementation with experimented and selected hyper-parameters are available in the link [2].

3.1. Count and TFIDF n-grams

As the data-set consists of mixed Roman and Dravidian lexicons with English and native language grammar used interchangeably, it becomes a new task which doesn't suit any model's training corpus. So the n-gram model trained from scratch is considered for the tasks and this strategy has shown good results with HI-EN and BN-HI-EN datasets [11]. Basic count-based and Term Frequency Inverse Document Frequency-based models are compared for feature extraction. The char TFIDF is calculated as explained in equation 1.

For each char t in a document d from the document set D TD-IDF is calculated as:
N is the total number of documents.

$$tfidf(t,d,D) = tf(t,d).idf(t,D) \tag{1}$$

where

$$tf(t,d) = log(1 + freq(t,d)) \tag{2}$$

$$idf(t,D) = log(\frac{N}{count(d \epsilon D\ :\ t \epsilon d)}) \tag{3}$$

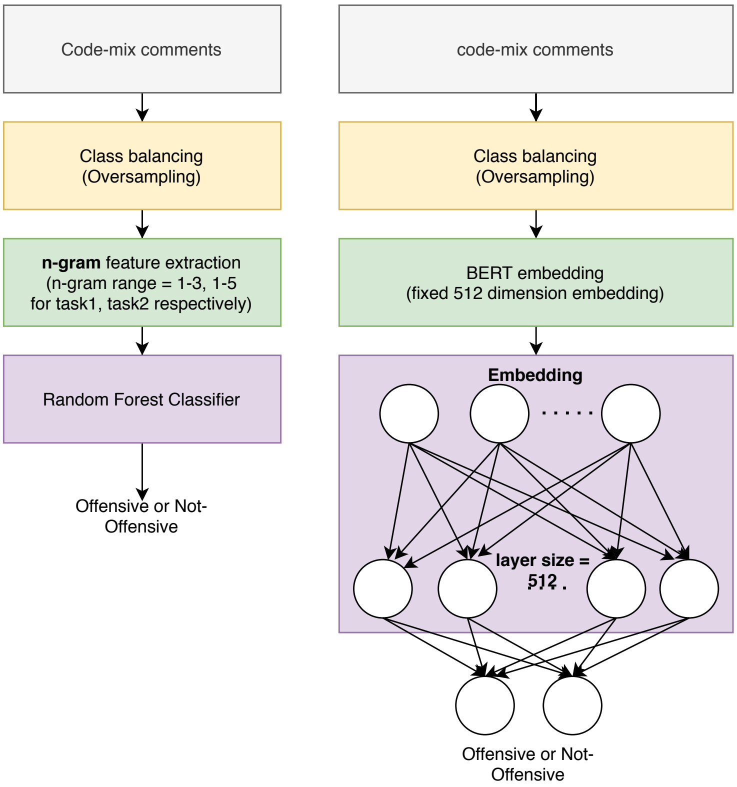
Different character n-gram are constructed with n-gram range varying from 1 to 7. Out of which n-gram range of 1-5 showed good results for task 1 and n-gram range of 1-3 showed good results for task 2. The smaller n-gram range for task2 may be due to the shorter tweeter comments. Out of all the classification models compared, the Random Forest model yielded the best result.

3.2. BERT Embedding

The comments consist of a mix of English and Tamil or English and Malayalam. So BERT multilingual model trained on a large corpus from 104 different languages [6], which includes

[1] https://scikit-learn.org/stable/

[2] https://github.com/nikamanthab/SSN_NLP-FIRE2020/tree/master/HASOC



Code-mix comments
Class balancing (Oversampling)
n-gram feature extraction (n-gram range = 1-3, 1-5 for task1, task2 respectively)
Random Forest Classifier
Offensive or Not-Offensive
code-mix comments
Class balancing (Oversampling)
BERT embedding (fixed 512 dimension embedding)
Embedding
.....
layer size = 512 .
Offensive or Not-Offensive

Figure 1: Flow diagram explaining the pipeline for feature extraction and model training.

Malayalam, Tamil, and English could be considered. It is noticeable that the BERT model has shown excellent results with tasks of sentence classification with Tweet data-sets. So this study can be interpolated to YouTube comments too. A fixed dimension embedding is generated by sentence-transformers, with the pre-trained base-multilingual-cased model implementation as explained in [12].

These embeddings are then used to classify as Offensive or Not-offensive by training a machine learning model. Different models such as Random Forest, Naive Bayes, and MLP are compared, out of which the MLP with 512 hidden layers, generated the best results. Even though, the models base training corpus is slightly different from the code-mix context with native language grammar in Roman scripts, it was able to learn from the training corpus by fine-tuning.

4. Observations

The output from n-gram embedding is a sparse matrix of high dimension. And as the total number of training samples is limited, it is observed that the Random Forest model gave better results in comparison to the MLP model. Whereas the embedding generated by the BERT model is of 512 dimensions for each comment, so MLP performed relatively well compared to the

Task	Features	n-gram	Classifier	Precision	Recall	F1 score
Task1	TFIDF	1-3	RF	**0.93**	**0.94**	**0.93**
	count vec	1-3	RF	0.93	0.93	0.93
	BERT	-	MLP	0.90	0.90	0.90
Task2-ml	TFIDF	1-5	RF	**0.73**	**0.72**	**0.72**
	count vec	1-5	RF	0.73	0.72	0.72
	BERT	-	MLP	0.62	0.62	0.62
Task2-ta	TFIDF	1-5	RF	**0.83**	**0.82**	**0.82**
	count vec	1-5	RF	0.83	0.82	0.82
	BERT	-	MLP	0.74	0.73	0.72

Table 2
Results of dev-set for task 1 and cross validation for task 2

Task	Precision	Recall	F1 score	Rank
Task1	0.94	0.94	0.94	2
Task2-ml	0.78	0.74	0.75	4
Task2-ta	0.88	0.88	0.88	2

Table 3
Results of test-set for task 1 and task 2

Random Forest model.

The model comparison based on dev-set evaluation for task1 and cross-validation with k = 4 is shown in table 2. As we can see that the overall performance of the TFIDF model is slightly better than the pre-trained multilingual BERT embedding. This maybe is due to the difference in the base corpus the BERT model is trained on, which is conflicting with the given code-mix corpus. Whereas the TFIDF and the Count vectorization yielded similar results because of the short length of the comments and containing varied usage of words by different users. The performance of test-set consisting of 400, 951, 940 instances for task1, task2-Malayalam, task2-Tamil are shown in the table 3. Our model performed well with a rank of 2,4 and 2 in task1, task2-Malayalam, task2-Tamil respectively.

5. Conclusion

The popularity of social media platforms is growing exponentially. Automatic and efficient censorship of hateful and offensive comments is becoming a necessity. In this paper, we have studied the performance of two different feature extraction techniques for Offensive language detection. Pretrained multilingual BERT model with MLP classifier and TFIDF embedding with Random Forest classifier is compared. Code-mixed Malayalam-English and Tamil-English YouTube comment HASOC Code-mix Dravidian data-set is evaluated and a test-set F1 score of 0.94 for task1-Malayalam, 0.75 for task2-Malayalam and 0.88 for task2-Tamil is shown. Our model achieved ranks of 2,4 and 2 for task1, task2-Malayalam, and task2-Tamil respectively.

References

[1] R. Priyadharshini, B. R. Chakravarthi, M. Vegupatti, J. P. McCrae, Named entity recognition for code-mixed Indian corpus using meta embedding, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.

[2] N. Jose, B. R. Chakravarthi, S. Suryawanshi, E. Sherly, J. P. McCrae, A survey of current datasets for code-switching research, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020.

[3] B. R. Chakravarthi, M. Arcan, J. P. McCrae, Comparison of Different Orthographies for Machine Translation of Under-Resourced Dravidian Languages, in: 2nd Conference on Language, Data and Knowledge (LDK 2019), volume 70 of *OpenAccess Series in Informatics (OASIcs)*, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019, pp. 6:1–6:14. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10370. doi:10.4230/OASIcs.LDK.2019.6.

[4] B. R. Chakravarthi, R. Priyadharshini, B. Stearns, A. Jayapal, S. S, M. Arcan, M. Zarrouk, J. P. McCrae, Multilingual multimodal machine translation for Dravidian languages utilizing phonetic transcription, in: Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages, European Association for Machine Translation, Dublin, Ireland, 2019, pp. 56–63. URL: https://www.aclweb.org/anthology/W19-6809.

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

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[7] B. R. Chakravarthi, M. A. Kumar, J. P. McCrae, P. B, S. KP, T. Mandl, Overview of the track on "HASOC-Offensive Language Identification- DravidianCodeMix", in: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India, 2020.

[8] B. R. Chakravarthi, M. A. Kumar, J. P. McCrae, P. B, S. KP, T. Mandl, Overview of the track on "HASOC-Offensive Language Identification- DravidianCodeMix", in: Proceedings of the 12th Forum for Information Retrieval Evaluation, FIRE '20, 2020.

[9] 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.

[10] 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.

[11] P. Mishra, P. Danda, P. Dhakras, Code-mixed sentiment analysis using machine learning and neural network approaches, 2018. arXiv:1808.03299.

[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.