

# NLP-CUET@DravidianLangTech-EACL2021: Offensive Language Detection from Multilingual Code-Mixed Text using Transformers

Omar Sharif<sup>†</sup>, Eftekhar Hossain\* and Mohammed Moshikul Hoque<sup>†</sup>

<sup>†</sup>Department of Computer Science and Engineering

\*Department of Electronics and Telecommunication Engineering

Chittagong University of Engineering and Technology, Bangladesh

{omar.sharif, eftekhar.hossain, moshiul\_240}@cuuet.ac.bd

## Abstract

The increasing accessibility of the internet facilitated social media usage and encouraged individuals to express their opinions liberally. Nevertheless, it also creates a place for content polluters to disseminate offensive posts or contents. Most of such offensive posts are written in a cross-lingual manner and can easily evade the online surveillance systems. This paper presents an automated system that can identify offensive text from multilingual code-mixed data. In the task, datasets provided in three languages including Tamil, Malayalam and Kannada code-mixed with English where participants are asked to implement separate models for each language. To accomplish the tasks, we employed two machine learning techniques (LR, SVM), three deep learning (LSTM, LSTM+Attention) techniques and three transformers (m-BERT, Indic-BERT, XLM-R) based methods. Results show that XLM-R outperforms other techniques in Tamil and Malayalam languages while m-BERT achieves the highest score in the Kannada language. The proposed models gained weighted  $f_1$  score of 0.76 (for Tamil), 0.93 (for Malayalam), and 0.71 (for Kannada) with a rank of 3<sup>rd</sup>, 5<sup>th</sup> and 4<sup>th</sup> respectively.

## 1 Introduction

The exponential increase of offensive contents in social media has become major concern to government organizations and tech companies. It is impossible to identify offensive texts manually from enormous amounts of online contents generated every moment in social media and other online platforms. Therefore, academicians, policymakers and stakeholders are trying to develop robust computational systems to limit the spread of offensive contents using state of the art NLP techniques (Kumar et al., 2020; Akiwowo

et al., 2020). In the last few years, several studies have been conducted regarding offensive languages such as hate speech, aggression, toxicity and abusive language. Although most social media users used their regional languages to carry out communication; however, plenty of resources developed in English, Arabic (Mubarak et al., 2017) and other high-resources languages. Lately, people started using code-mixed texts on social media (Jose et al., 2020; Priyadharshini et al., 2020). Linguistic level understanding of the text, complex morphological structure, and lack of training corpora are crucial barriers to classifying code-mixed data. Moreover, the system trained on monolingual data alone may fail to classify code-mixed data because of the texts linguistic level code-switching complexity. This work aims to address the above mentioned problems by contributing the following:

- Prepared transformer-based methods to identify the offensive texts from multilingual (Tamil, Malayalam, Kannada) code-mixed data.
- Perform experiments on the dataset with detail performance and error analysis, thus setting an important baseline to compare in future.

## 2 Related Work

Offensive social media content might trigger objectionable consequences to its user like mental health problem, and suicide attempts (Bonanno and Hymel, 2013). To keep the social media ecosystem, coherent researchers and stakeholders should try to develop computational models to identify and classify offensive contents within a short period. Zampieri et al. (2019a) develop an offensive language identification dataset using hierarchical annotation schema. Three layers of annotation are

used in the dataset: offensive language detection, categorization of offensive language, and offensive language target identification. SemEval task 6 is organized based on this dataset which had opened interesting research directions (Zampieri et al., 2019b). In early stages, computational models created by using support vector machine, naive Bayes and other traditional machine learning approach (Dadvar et al., 2013). These models performance is not up to the mark as they could not capture the semantic and contextual information in texts. This problem was mitigated with the arrival of word embeddings and recurrent neural networks (Aroyehun and Gelbukh, 2018). Networks like bidirectional LSTMs and GRUs can hold contextual information from both past and future, creating more robust classification systems (Mandl et al., 2019; Chakravarthi et al., 2020c; Mandl et al., 2020). In recent years, transformer-based model such as BERT (Sharif et al., 2021), XLM-R gained more attention to identify and classify offensive texts. These large pre-trained model can classify code-mixed texts of different languages with astonishing accuracy (Hande et al., 2020; Chakravarthi et al., 2020b,a).

### 3 Dataset

In order to detect offensive text from social media, task organizers developed a gold standard corpus. As many social media texts are code-mixed, so system trained on monolingual data fails to classify code-mixed data due to the complexity of code-switching at different linguistic levels in the texts. To address this phenomenon, Chakravarthi et al. (2020b) developed a code-mixed text corpus in Dravidian languages. Corpus has three types

of code-mixing texts (Tamil-English, Kannada-English and Malayalam-English). The task aims to implement a system that can successfully detect code-mixed offensive texts. To implement such a system, we utilize the corpus provided by the workshop organizers<sup>1</sup>(Chakravarthi et al., 2021). The corpus consists of the text of three different languages, i.e. Tamil, Kannada and Malayalam.

System developed with the different number of train, validation and test examples for each language. The assigned task is a multi-class classification problem where the model has to classify a potential text into one of six predefined classes. These classes are Not offensive (NF), Offensive-Targeted-Insult-Other (OTIO), Offensive-Targeted-Insult-Individual (OTII), Offensive-Targeted-Insult-Group (OTIG), not-Tamil (NT)/not-Malayalam (NM)/not-Kannada (NK), and Offensive-Untargetede (OU). Table 1 shows the number of instances for each class in train, validation and test sets. Datasets are highly imbalanced where a number of instances in NF class is much higher compare to other classes. Before system development, a cleaning process is applied to every dataset. In this phase, unwanted characters, symbols, punctuation, emojis, and numbers are discarded from the texts and thus, a cleaned dataset prepared for each language.

### 4 Methodology

In this part, we briefly describe the methods and techniques employed to address the previous section’s problem. First, features are extracted with different feature extraction techniques and

<sup>1</sup><https://dravidianlangtech.github.io/2021/index.html>

	Tamil			Malayalam			Kannada		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
NF	25425	3193	3190	14153	1779	1765	3544	426	427
OTIO	454	65	71	-	-	-	123	16	14
OTII	2343	307	315	239	24	27	487	66	75
OTIG	2557	295	288	140	13	23	329	45	44
OU	2906	356	368	191	20	29	212	33	33
NT	1454	172	160	-	-	-	-	-	-
NM	-	-	-	1287	163	157	-	-	-
NK	-	-	-	-	-	-	1522	191	185
<b>Total</b>	23962	4850	2500	58500	5842	1923	36009	5724	2682

Table 1: Class wise distribution of train, validation and test set for each language. The acronym used for classes abbreviated in Section 3.

various machine learning (ML) and deep learning algorithms are used for the baseline evaluation. Furthermore, different transformer models, i.e. XLM-R, m-BERT and Indic-BERT, are exploited as well. Figure 1 shows the schematic process of the developed system.

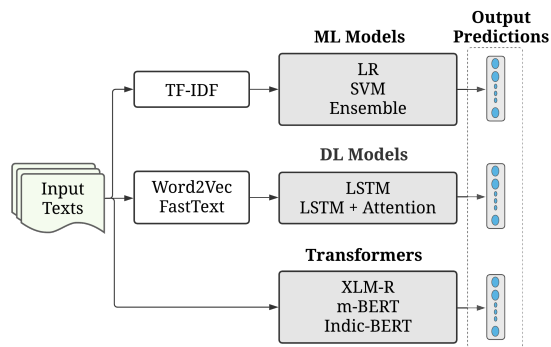


Figure 1: Abstract process of offensive language detection

**Feature Extraction:** ML and DL algorithms are unable to learn from raw texts. Therefore, feature extraction is required to train the classifier models. For ML methods, tf-idf technique (Tokunaga and Makoto, 1994) is applied to extract the unigram features. On the other hand, Word2Vec (Mikolov et al., 2013) and FastText (Grave et al., 2018) embeddings are used as feature extraction techniques for DL models. Word2Vec implemented using Keras embedding layer with embedding dimension 100 for all languages while pre-trained embedding matrix of each language is utilized for FastText embedding.

**Machine Learning Models** To develop the offensive text detection system, we begin our investigation with traditional ML approaches, including LR and SVM. The ensemble of multiple ML classifiers is also employed to achieve better performance. We use ‘lbfgs’ optimizer with  $C$  value of 0.4, 0.7, and 5 for Tamil, Malayalam and Kannada languages to implement LR. We selected ‘linear’ SVM and settled  $C$  value to 3, 10, and 7. Besides, Decision Tree (DT) and Random Forest (RF) classifiers are incorporated with LR and SVM to construct the ensemble approach. For LR and SVM, the same parameters were retained while ‘n\_estimators = 100’ is used for DT and RF. Majority voting technique is applied to get the prediction from the ensemble method.

**Deep Learning Models** DL algorithms are proven superior over many traditional ML approaches in various classification tasks (Sharif et al., 2020). We employ LSTM and combination of LSTM with Attention-based approach to classify the offensive text to continue the investigations. LSTM is well known for its ability to capture the semantic information as well as long term dependencies. Bidirectional LSTM (BiLSTM) with 100 cells are used to exploit the information from both past and future states. To mitigate the chance of overfitting, a dropout technique utilized with a rate of 0.1. Finally, the output of the BiLSTM layer transferred to a softmax layer for the prediction. However, sometimes all the words in a text do not contribute equally for the classification. To emphasize the words that have a noticeable impact on the input text, we use attention (Vaswani et al., 2017) mechanism. An attention layer of 20 neurons is constructed on the top of a BiLSTM layer. The attention operation applied upon the output of BiLSTM layer and the attention vector passed to a softmax layer for the prediction. We employ the same architectures for all the languages and use ‘sparse\_categorical\_crossentropy’ as the loss function. ‘Adam’ optimizer with learning rate  $1e^{-3}$  and batch size 32 is used to train the models for 20 epochs. The best intermediate model is stored by using Keras callbacks.

**Transformer Models** In recent years, transformers have gained popularity for its tremendous performance in almost every aspect of NLP. As our given datasets consist of cross-lingual texts of different languages, we choose three transformers such as XLM-R (Conneau et al., 2020), m-BERT (Devlin et al., 2019), Indic-BERT (Kakwani et al., 2020) to develop our models. XLM-R is a self-supervised training technique for cross-lingual understanding particularly well for low resource languages. On the other hand, m-BERT is a transformer model pre-trained over 104 languages, and Indic-BERT is specifically pre-trained on Indian languages such as Kannada, Tamil, Telugu and Malayalam. These models are culled from Pytorch Huggingface<sup>2</sup> transformers library and fine-tuned on our dataset using ktrain (Maiya, 2020) package. To fine-tune these models, we use ktrain ‘fit\_onecycle’ method with learning rate  $2e^{-5}$  for all the languages. We observed that each input text’s average length is less than 50

<sup>2</sup><https://huggingface.co/transformers/>

words for Kannada and 70 words for Tamil and Malayalam languages. Therefore, to reduce the computational cost, the input texts’ maximum size settle to 50 for Kannada and 70 for Tamil and Malayalam. All the models have trained up to 20 epochs with batch size 4 for XLM-R and 12 for m-BERT and Indic-BERT.

## 5 Results and Analysis

This section presents a performance comparison of various ML, DL, and transformer-based offensive text detection models for all the mentioned languages. The superiority of the models is determined based on the weighted  $f_1$  score. However, the precision and recall metrics also considered. Table 2 reports the evaluation results of models. The results showed that the ML models ensemble achieved the highest  $f_1$  score of 0.73 for the Tamil language. Both SVM and ensemble methods obtained a similar  $f_1$  score for Malayalam (0.88) and Kannada (0.48) languages. In contrast, for all the languages, the LR method was poorly performed where a small difference ( $< 0.04$ ) in  $f_1$  score value was observed with other ML models. In case of DL techniques, LSTM (with Word2Vec) model obtained comparatively higher  $f_1$  score of 0.72, 0.87 and 0.47 than that of provided by LSTM (FastText) modes which are 0.68, 0.86, and 0.45 for Tamil, Malayalam and Kannada languages. However, a combination of LSTM and Attention model shows 0.01 rise amounting to 0.87 and 0.46 respectively for Malayalam and Kannada languages.

Meanwhile, transformer-based models showed outstanding performance for all the languages. For Tamil, Indic-BERT obtained  $f_1$  score of 0.74

while both m-BERT and XLM-R exceeds all the previously mentioned models by achieving maximum  $f_1$  score of 0.76. Though two models give an identical  $f_1$  score, XLM-R is opted as the best model by considering both the precision, recall scores. In Malayalam language, m-BERT and Indic-BERT obtained  $f_1$  score of 0.90 and 0.92 respectively. While XLM-R shows a rise of 0.01 and thus beats all the other models. For the Kannada language, both XLM-R and m-BERT outperformed all the previous models obtaining  $f_1$  score of 0.71. However, m-BERT is selected as the best model as it outdoes XLM-R concerning precision and recall values.

The results revealed that the transformer-based models performed exceptionally well to accomplish the assigned task for all the languages. Compared to other techniques, the ML models performed moderately while the deep learning models lag than the other models. The possible reason for this weaker performance is the extensive appearance of cross-lingual words in the text. As a result, Word2Vec and FastText embeddings have failed to create the appropriate feature mapping among the words. Thus, LSTM and attention-based models may not find sufficient relational dependencies among the features and performed below the expectation.

### 5.1 Error Analysis

Table 2 showed that XLM-R is the best performing for Tamil and Malayalam languages model, whereas, for Kannada, m-BERT is the best model. To get more insights, we present a detail error analysis of these models. The error analysis is carried out by using the confusion matrix (Figure 2).

Method	Classifiers	Tamil			Malaylam			Kannada		
		P	R	F	P	R	F	P	R	F
ML models	LR	0.76	0.65	0.69	0.88	0.84	0.86	0.52	0.44	0.47
	SVM	0.74	0.68	0.70	0.88	0.87	0.88	0.48	0.48	0.48
	Ensemble	0.72	0.76	0.73	0.88	0.89	0.88	0.46	0.51	0.48
DL models	LSTM (Word2vec)	0.73	0.72	0.72	0.86	0.87	0.86	0.48	0.44	0.46
	LSTM (Fasttext)	0.70	0.67	0.68	0.87	0.85	0.86	0.50	0.45	0.45
	LSTM + Attention	0.71	0.73	0.72	0.86	0.87	0.87	0.49	0.46	0.47
Transformers	m-BERT	0.74	0.78	0.76	0.93	0.88	0.90	0.70	0.74	<b>0.71</b>
	Indic-BERT	0.74	0.78	0.74	0.95	0.91	0.92	0.69	0.74	0.70
	XLM-R	<b>0.75</b>	0.78	<b>0.76</b>	0.92	0.94	<b>0.93</b>	0.71	0.70	0.71

Table 2: Evaluation results of ML, DL and transformer-based models on the test set. Here P, R, F denotes the precision, recall and weighted  $f_1$  score.



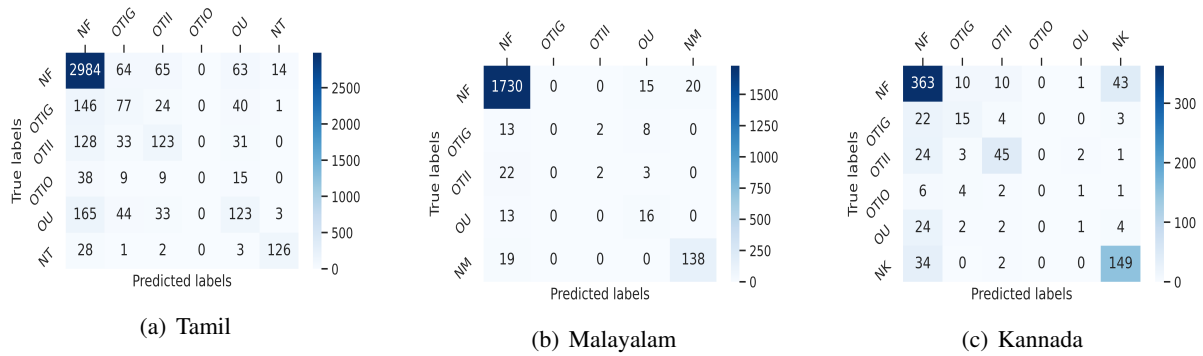


Figure 2: Confusion matrix of the best models for each language task

Only NF and NT classes have a high true-positive rate (TPR) while in other classes, texts have mostly misclassified as non-offensive (Figure 2 (a)). However, among all six classes, OTIO got 0% TPR where among 77 texts more than 50% wrongly classified as NF. The high-class imbalance situation might be the reason for this vulnerable outcome. Figure 2 (b) reveals that the model is biased towards NF class. Due to the insufficient number of instances, the model failed to identify none of the 23 OTIG texts. Among 27 OTII texts, only 2 have correctly classified. Meanwhile, we observe that only NF, OTII and NK classes have a high TPR where most of the cases, the model misclassified as NF class (Figure 2 (c)). In contrast, OTIO and OU classes have shown misclassification rate 100% and 97%.

## 6 Conclusion

In this work, we have described and analyzed the system’s performance implemented as a participation in the offensive language identification shared task at EACL-2021. Initially, SVM, LR, LSTM, LSTM+Attention models have employed with tf-idf and word embedding features. Results indicate that ML ensemble achieved higher accuracy than DL methods. However, the outcomes are not promising for the available datasets. Code-mixing of multilingual texts might be a reason behind this. We applied transformer-based models to overcome this situation, which provides an astonishing rise in accuracy than ML and DL-based methods. Weighted  $f_1$  score increased from 0.73 to 0.76, 0.88 to 0.93 and 0.48 to 0.71 for Tamil, Malayalam and Kannada language respectively. In future, the idea of ensemble technique could be adopted

on transformer-based models to investigate the system’s overall performance.

## References

- Seyi Akiwowo, Bertie Vidgen, Vinodkumar Prabhakaran, and Zeerak Waseem, editors. 2020. *Proceedings of the Fourth Workshop on Online Abuse and Harms*. Association for Computational Linguistics, Online.
- Segun Taofeek Aroyehun and Alexander Gelbukh. 2018. *Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling*. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Rina A Bonanno and Shelley Hymel. 2013. *Cyber bullying and internalizing difficulties: Above and beyond the impact of traditional forms of bullying*. *Journal of youth and adolescence*, 42(5):685–697.
- 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 Priyadarshini, 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.
- Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2020c. [Overview of the Track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text](#). In *Forum for Information Retrieval Evaluation, FIRE 2020*, page 21–24, New York, NY, USA. Association for Computing Machinery.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. [Improving cyberbullying detection with user context](#). ECIR'13, page 693–696, Berlin, Heidelberg. Springer-Verlag.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#).
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. [Learning word vectors for 157 languages](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- 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.
- Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2020. [A Survey of Current Datasets for Code-Switching Research](#). In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141.
- Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. [IndicNLP Suite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.
- Ritesh Kumar, Atul Kr. Ojha, Bornini Lahiri, Marcos Zampieri, Shervin Malmasi, Vanessa Murdock, and Daniel Kadar, editors. 2020. *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. European Language Resources Association (ELRA), Marseille, France.
- Arun S. Maiya. 2020. [ktrain: A low-code library for augmented machine learning](#).
- 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.
- Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. [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*, page 14–17, New York, NY, USA. Association for Computing Machinery.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Distributed representations of words and phrases and their compositionality](#).
- Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. [Abusive language detection on Arabic social media](#). In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56, Vancouver, BC, Canada. Association for Computational Linguistics.
- Ruba Priyadharshini, Bharathi Raja Chakravarthi, Mani Vegupatti, and John P. McCrae. 2020. [Named Entity Recognition for Code-Mixed Indian Corpus using Meta Embedding](#). In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 68–72.
- Omar Sharif, Eftekhar Hossain, and Mohammed Moshui Hoque. 2020. [Techtext: Classification of technical texts using convolution and bidirectional long short term memory network](#).

- Omar Sharif, Eftekhar Hossain, and Mohammed Moshiul Hoque. 2021. [Combating hostility: Covid-19 fake news and hostile post detection in social media.](#)
- Takenobu Tokunaga and Iwayama Makoto. 1994. Text categorization based on weighted inverse document frequency. In *Special Interest Groups and Information Process Society of Japan (SIG-IPSJ)*. Citeseer.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need.](#)
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. [Predicting the type and target of offensive posts in social media.](#) In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. [SemEval-2019 task 6: Identifying and categorizing offensive language in social media \(OffensEval\).](#) In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.