Transliterate or translate? Sentiment analysis of code-mixed text in Dravidian languages

Karthik Puranik[a], Bharathi B[b] and Senthil Kumar B[b]

[a]*Indian Institute of Information Technology Tiruchirappalli*
[b]*Computer Science and Engineering, SSN College of Engineering, Chennai*

Abstract

Sentiment analysis of social media posts and comments for various marketing and emotional purposes is gaining recognition. With the increasing presence of code-mixed content in various native languages, there is a need for ardent research to produce promising results. This research paper bestows a tiny contribution to this research in the form of sentiment analysis of code-mixed social media comments in the popular Dravidian languages Kannada, Tamil and Malayalam. It describes the work for the shared task conducted by Dravidian-CodeMix at FIRE 2021 by employing pre-trained models like ULMFiT and multilingual BERT fine-tuned on the code-mixed dataset, transliteration (TRAI) of the same, English translations (TRAA) of the TRAI data and the combination of all the three. The results are recorded in this research paper where the best models stood 4th, 5th and 10th ranks in the Tamil, Kannada and Malayalam tasks respectively.

1. Introduction

Sentiment analysis is a popular technique for analysing and evaluating textual content to learn the attitude and thoughts expressed in it [1]. The term "Sentiment analysis" was first introduced in Nasukawa and Yi. This method is largely employed in the marketing sector to realize the opinion of the customers on a certain product without reading all the feedbacks. Natural language processing (NLP) truly automates the wearisome tasks like analysing feedbacks. Several other tasks like sentiment classification, sentiment extraction, opinion summary, and subjectivity detection can also be performed [3] for various applications like spam email detection[4], fake news detection [5], hate and hope speech detection [6], finding inappropriate texts in social media [7, 8] and many others [9, 10]. This paper concentrates on the sentiment analysis of code-mixed social media comments for Dravidian languages.

Social media is known to us as a virtual space to share our opinions, and communicate. However, social media is the largest hub for marketing. They provide spaces for brands to advertise products and target interested customers, which is the prime source of income for these platforms [11]. In order to market the right product which appeals to its user, the social media platforms monitor their activities and comments[12, 13]. This enables them to know the

user's sentiment towards a product and company [14]. Another crucial application of sentiment analysis is to automatically spot comments or posts which are offensive, abusive or spreads hatred in the social media platforms [15]. Social media is a free space and no restrictions can be imposed on the comments or posts being circulated. However, the comments can certainly be detected and overseen to protect underage and the users who are vulnerable to get offended [16, 17].

Social media features multilingual speakers from all over the world, and people tend to use a lot of variations while expressing their thoughts[18]. Native speakers writing in Roman script is the most common scene due to the easy accessibility and customary usage of Roman script keyboards in mobile phones and desktop keyboards[19]. However, some users tend to write in the native script too. Finally, there is a case of code-mixing where two or more languages are merged in respect to the script or the usage [20]. Sentiment analysis becomes difficult for such texts. In this paper, the method of transliterating the text is applied. Transliterating is the process of converting a text from one script to another while maintaining the pronunciation [21, 22]. This brings about a uniformity in the text and helps the model learn better. However, due to the presence of English text in the code-mixed dataset, there has also been a slight effort to translate [23] the text transliterated in the native language to English and train the model with it.

This research paper depicts our work for the shared task Dravidian-CodeMix[1] at FIRE 2021 [24, 25]. The task was to detect the sentiment in the sentences for three of the major Dravidian languages [26] Kannada, Tamil, and Malayalam. Our system models stood 5th, 4th and 10th respectively in the shared task. The codes for the model and the transliterated and translated datasets are provided in the link[2].

2. Dataset

The dataset provided by the organizers of the shared task has been used to train the models [12, 13, 27, 28]. It contains annotated sentences obtained by cleaning YouTube comments[3]. The sentences are highly code-mixed and contains inter-sentimental, intra-sentimental and tag switching which are prevalent in code-mixed data to be classified into five classes namely, positive, negative, unknown state, mixed feelings and not the intended language. The train, development and test distribution can be viewed in Table 1.

Split	Kannada	Tamil	Malayalam
Training	6,213	35,657	15,889
Development	692	3,963	1,767
Test	768	4,403	1,963
Total	7,673	44,023	19,619

Table 1
Train-Development-Test Data Distribution

---

[1]https://dravidian-codemix.github.io/2021/index.html
[2]https://github.com/karthikpuranik11/FIRE2021
[3]https://www.youtube.com/

Further, the transliterations (TRAI) of the code-mixed training (TRA) dataset in the respective Dravidian languages were used. Small preprocessing steps like removing the language tag and brackets, removing all the sentences which belong to "not-language" were removed. The English translations (TRAA) of these transliterations was also used as a part of this research. It was evident that English was the most widely used language after the Dravidian language. A few comments represented in the TRA, TRAI and TRAA datasets belonging to the five classes are tabulated in Table 2.

3. Methodology

Based on previous researches, two of the most promising pre-trained models, ULMFiT [29] and BERT [30] with bidirectional LSTM layers [31], were used to determine the sentiment of the sentences. These models were fine-tuned separately on the training data provided by the organizers, the transliterated data combined with the training data, the translated data combined with the training data and the combination of all the three datasets.

3.1. BERT

Bidirectional Encoder Representation from Transformers (BERT) is one of the most popular transformer based models, trained extensively on the entire Wikipedia and 0.11 million Word-Piece sentences [32] for over 104 languages in the world. The unprecedented methods like Next Sentence Prediction (NSP) and Masked Language Modelling (MLM) successfully catch a deeper context of the languages. For the particular task, **bert-base-multilingual-cased** [33] from HuggingFace[4] [34] has been used. It comprises 12 layers and attention heads and about 110M parameters.

This model was further concatenated with bidirectional LSTM layers, which are known to improve the information being fed. The bidirectional layers read the embeddings from both the directions, hence, boosts the context and the F1 scores drastically. Further, the training was done with an Adam optimizer [35], a learning rate of 2e-5 with the *cross-entropy* loss function [36, 37] for a total of 5 epochs. The various parameters employed in the BERT+ BiLSTM model van be viewed in Table 3.

3.2. ULMFiT

Universal Language Model Fine-tuning, or ULMFiT was one of the initial transfer learning method to produce state-of-the-art results for NLP tasks. It was trained on very huge datasets like Wikitext-103[5] with around 103M sentences. It employs three novel techniques for fine-tuning the language models for various NLP tasks, which are discriminative fine-tuning, slanted triangular learning rates (STLR) and gradual unfreezing. AWD-LSTM language model [38, 39], a standard LSTM consisting 3 layers and 1150 hidden activation per layer and an embedding size of 400 and without any attentions and just well tune dropouts, is generally used. Adam

---

[4]https://huggingface.co/
[5]https://blog.einstein.ai/the-wikitext-long-term-dependency-language-modeling-dataset/

Dataset	Sentence	Language	Class
TRA	Prethekichu onnumillankilum... Ithil entho undd kelkkaathirikkaan pattanilla supr song	Malayalam	Positive
TRAI	പ്രതിലിച്ച ഒന്നുമില്ലാങ്കിലും... ഇതിൽ എന്തോ ഉണ്ട് കേൾക്കാതിരിക്കാൻ പറ്റാനില്ല സൂപ്പ് സോംഗ്		
TRAA	Even if there's nothing against it, I can't hear anything about it		
TRA	Promotion chennagi nadtilla	Kannada	Negative
TRAI	ಪ್ರೊಮೋಷನ್ ಚೆನ್ನಾಗಿ ನಟ್ಟಿಲ್ಲಾ		
TRAA	Promotions not good		
TRA	Ithu puthusa irukune puthusa irukku.	Tamil	Unknown state
TRAI	இது புதுச இருகுணி புதுச இருக்கு.		
TRAA	This is a new thing.		
TRA	Game yavadu bro adu	Kannada	Mixed feelings
TRAI	ಗೇಮ್ ಯಾವದು ಬ್ರೋ ಅದು		
TRAA	What game is this?		
TRA	SUPER FILM SUPER SONG.2019	Malayalam	Not Malayalam
TRAI	സൂപ്പർ ഫിലിം സൂപ്പർ സോംഗ്.2019		
TRAA	Super movie super Soze 2019.		
TRA	Amazing trailer jayam and kajal gonna rock now!!!!	Tamil	not-Tamil
TRAI	அமேஜிங் டிரைலர் ஜெயம் அண்ட் கஜால் கோன்னா ராக் நோ!!!!		
TRAA	Amazing trailer Jayam and Gajal Konna Rock No!!!!		

Table 2
Examples of the code-mixed sentences, its transliteration and translation in Kannada, Tamil and Malayalam

optimizer with starting learning rate of 1e-8 and an end learning rate of 1e-2 and a dropout of 0.5 is used.

Parameter	Value
Number of LSTM units	256
Dropout	0.4
Activation Function	ReLU
Max Len	128
Batch Size	32
Optimizer	AdamW
Learning Rate	2e-5
Loss Function	cross-entropy
Number of epochs	5

Table 3
Parameters for the BERT+BiLSTM model.

3.3. Transliteration

The IndianNLP-Transliteration[6] tool from AI4Bharat was used to get the transliterations of the training dataset. This deep transliteration tool can transliterate from Roman script to any low resourced Indian language. The architecture majorly consists of Recurrent Neural Networks (RNN) [40] with encoders and decoders [41]. The decoder employs top 'k' predictions and then re-ranked to get the most probable word [42]. It is observed that most of the sentences in the Dravidian language present in the code-mixed dataset is the languages written in Roman script. The multilingual pre-trained models, largely trained on these Dravidian languages in their original scripts, might find it hard to comprehend such sentences. Transliterating them back to the original script could possibly improve the accuracy [43].

3.4. Translation

The transliterated data in the Dravidian language is translated to English using IndicTrans [44] from AI4Bharat[7]. This PyTorch Fairseq[8] [45, 46] based Transformer NMT model, is trained on a large parallel corpus containing 46.9 million sentences of Samanantar dataset. The model is known to produce state-of-the-art BLEU [47] scores for 11 Indian languages. The translations given by the IndicTrans baseline model on the transliterated dataset was used. The reason for using the translated data was due to the presence of excessive English in the code-mixed dataset, and most of the pre-trained models are trained on large number of English sentences.

4. Results

In this section, the F1 scores of the BERT and ULMFiT models for the sentiment analysis of Kannada, Tamil and Malayalam datasets are compared, and suitable analysis are recorded. The weighted F1 scores are tabulated in Table 4. The models are fine-tuned on training dataset

---

[6] https://github.com/AI4Bharat/IndianNLP-Transliteration
[7] https://github.com/AI4Bharat/indicTrans
[8] https://github.com/pytorch/fairseq

(TRA), the combination of transliterated dataset and TRA (TRAI), translated (TRAA) dataset and TRA and all 3 merged.

Table 4
Weighted F1-scores of sentiment analysis on the test datasets, where P: Precision, R: Recall and F1: F1 score.

Dataset	Kannada					
	BERT			ULMFiT		
	P	**R**	**F1**	**P**	**R**	**F1**
Train (TRA)	0.5952	0.6185	0.6040	**0.6547**	**0.6276**	**0.6389**
Transliterate + TRA (TRAI)	0.5587	0.6133	0.5831	0.6239	0.6081	0.6150
Translate + TRA (TRAA)	0.6176	0.6367	0.6231	0.6078	0.5990	0.6031
Merged (TRA+TRAI+TRAA)	0.6079	0.6172	0.6113	0.6172	0.5885	0.5993
	Tamil					
	BERT			ULMFiT		
	P	**R**	**F1**	**P**	**R**	**F1**
Train (TRA)	0.5291	0.5572	0.5308	0.6544	0.6229	0.6362
Transliterate + TRA (TRAI)	0.5334	0.5502	0.5366	**0.6889**	**0.6372**	**0.6583**
Translate + TRA (TRAA)	0.5284	0.5427	0.5310	0.6694	0.6379	0.6514
Merged (TRA+TRAI+TRAA)	0.5298	0.5570	0.5367	0.6629	0.6306	0.6432
	Malayalam					
	BERT			ULMFiT		
	P	**R**	**F1**	**P**	**R**	**F1**
Train (TRA)	0.6238	0.6733	0.6457	0.7084	0.6937	0.6990
Transliterate + TRA (TRAI)	0.6874	0.7018	0.6933	**0.7139**	**0.7013**	**0.7062**
Translate + TRA (TRAA)	0.5976	0.7142	0.6467	0.7086	0.6901	0.6970
Merged (TRA+TRAI+TRAA)	0.6822	0.6927	0.6863	0.7041	0.6952	0.6984

It is firstly clear from Table 4 that ULMFiT manages to get better F1 scores than BERT concatenated with biLSTM layers for the majority of the datasets. The unique transfer learning techniques used by ULMFiT like the discriminative fine-tuning, slanted triangular learning rates and gradual unfreezing seem to successfully produce exceptional F1 scores. Discriminative fine-tuning allows us to fine-tune each layer separately with different learning rates. Gradual unfreezing improves it further by keeping the last layer frozen in the first epoch and unfreezing layer by layer for the further epochs. Except for the Tamil data, BERT manages to give results comparable to ULMFiT for the other languages.

ULMFiT fine-tuned on the TRA dataset gives the best F1-score of 0.639 for the Kannada task. It is followed by BERT fine-tuned on the TRAA dataset with 0.623. Other models gave similar results. It is surprising how the models managed to give F1 scores akin to other languages, considering the limited size of the dataset. ULMFiT manages to surpass BERT by a huge difference for the Tamil task. The presence of class imbalances in the Tamil dataset could be a

reason for this issue. The "positive" comments are 2,830 in number out of the 4,402 sentences in the test dataset, while "not-Tamil" which is just 210. This imbalance causes a variation in the results. ULMFiT on TRAI and TRAA gave nearly similar F1 scores of 0.658 and 0.651 respectively. ULMFiT trained on all the four datasets gave equivalent results for the Malayalam task, with TRAI giving the best score of 0.706. BERT trained on TRAI gave a competitive score of 0.6933 for the same task.

The basic observation derived while comparing the various datasets is the equal contention between the four datasets used. But, the most common scenario is that the TRAI dataset manages to have the upper hand in the majority of the cases. The most plausible explanation to this is due to the fact that the dataset is code-mixed and the Dravidian text written in Roman script. When that is converted to the native script, the model manages to fine-tune well. With the original data also present, the model manages to fine-tune on the English text too. However, we can't be entirely sure of the accuracy of transliterations from the IndianNLP-Transliteration tool. Another drawback of transliterating the code-mixed sentences is that the English and other language also get transliterated to the Kannada/Tamil/Malayalam script. Such words might not be able to recognized by the model at all. Dravidian languages can be complex and there might be several ways in which the comments in the Roman script can be transliterated, and a slight variation can change the meaning entirely[48]. However, to tackle these, we merge the transliterated dataset with the TRA data so that the model manages to learn the other languages in the code-mixed data too.

The TRAA and the merged dataset proves to be inefficient due to its low F1 scores. The TRAA dataset is not significantly behind the TRAI data, which proves that there is a scope to increase the accuracy with further research. Though IndicTrans one of the best models for machine translation of Indian languages has been employed, we can surely not rely entirely on the translations of the transliterated data. Further, cleaning of the TRAA data by removing sentences which fail to make any sense and fine-tuning the IndicTrans model on a suitable parallel corpus before translating it can be done to obtain good F1 scores for the TRAA dataset. The combination of the three datasets however fails miserably in most of the cases due to the repetition of the sentences in different forms, which seems to make the model not learning anything productively, and the inaccuracies in the TRAA and TRAI datasets add up to reduce the F1 scores even further.

5. Conclusion

Sentiment analysis of social media comments emerges as one of the most notable tasks of natural language processing (NLP). In order to obtain good F1 scores for the sentiment analysis of social media comments in code-mixed Dravidian languages Kannada, Tamil and Malayalam, after careful experimentation with Transformer based ULMFiT and mBERT fine-tuned on TRA, TRAI, TRAA and merged dataset, ULMFiT proved to give the best F1 scores for all the three languages. For Kannada, it was on the TRA dataset, while TRAI proved effective for Tamil and Malayalam. This paper introduces the use of TRAA dataset which can be worked upon in the future.

References

[1] M. Rambocas, J. Gama, Marketing Research: The Role Of Sentiment Analysis, FEP Working Papers 489, Universidade do Porto, Faculdade de Economia do Porto, 2013. URL: https://ideas.repec.org/p/por/fepwps/489.html.

[2] T. Nasukawa, J. Yi, Sentiment analysis: Capturing favorability using natural language processing, 2003, pp. 70–77. doi:10.1145/945645.945658.

[3] B. Keith, E. Fuentes, C. Meneses, A hybrid approach for sentiment analysis applied to paper, in: Proceedings of ACM SIGKDD Conference, Halifax, Nova Scotia, Canada, 2017, p. 10.

[4] A. F. Anees, A. Shaikh, A. Shaikh, S. Shaikh, Survey paper on sentiment analysis: Techniques and challenges, EasyChair2516-2314 (2020).

[5] A. Hande, K. Puranik, R. Priyadharshini, S. Thavareesan, B. R. Chakravarthi, Evaluating pretrained transformer-based models for covid-19 fake news detection, in: 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 766–772. doi:10.1109/ICCMC51019.2021.9418446.

[6] K. Puranik, A. Hande, R. Priyadharshini, S. Thavareesan, B. R. Chakravarthi, Iiitt@lt-edi-eacl2021-hope speech detection: There is always hope in transformers, 2021. arXiv:2104.09066.

[7] K. Yasaswini, K. Puranik, A. Hande, R. Priyadharshini, S. Thavareesan, B. R. Chakravarthi, 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, Kyiv, 2021, pp. 187–194. URL: https://aclanthology.org/2021.dravidianlangtech-1.25.

[8] P. K. Jada, D. S. Reddy, K. Yasaswini, C. Prabakaran, A. Sampath, S. Thangasamy, IIIT@Dravidian-CodeMix-FIRE2021: Transformer Model based Sentiment Analysis in Dravidian Languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[9] A. Hande, K. Puranik, R. Priyadharshini, B. R. Chakravarthi, Domain identification of scientific articles using transfer learning and ensembles, in: Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2021 Workshops, WSPA, MLMEIN, SDPRA, DARAI, and AI4EPT, Delhi, India, May 11, 2021 Proceedings 25, Springer International Publishing, 2021, pp. 88–97.

[10] A. Hande, S. U. Hegde, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, S. Thavareesan, B. R. Chakravarthi, Benchmarking multi-task learning for sentiment analysis and offensive language identification in under-resourced dravidian languages, 2021. arXiv:2108.03867.

[11] T. Oikonomidis, K. Fouskas, Is Social Media Paying Its Money?, 2019, pp. 999–1006. doi:10.1007/978-3-030-12453-3_115.

[12] 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.

[13] 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.

[14] F. Neri, C. Aliprandi, F. Capeci, M. Cuadros, T. By, Sentiment analysis on social media, in: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2012, pp. 919–926. doi:10.1109/ASONAM.2012.164.

[15] B. R. Chakravarthi, 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, 2020, pp. 41–53.

[16] Y. Chen, Y. Zhou, S. Zhu, H. Xu, Detecting offensive language in social media to protect adolescent online safety, in: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, IEEE, 2012, pp. 71–80.

[17] S. U. Hegde, A. Hande, R. Priyadharshini, S. Thavareesan, R. Sakuntharaj, S. Thangasamy, B. Bharathi, B. R. Chakravarthi, Do images really do the talking? analysing the significance of images in tamil troll meme classification, 2021. arXiv:2108.03886.

[18] U. Barman, A. Das, J. Wagner, J. Foster, Code mixing: A challenge for language identification in the language of social media, in: Proceedings of the first workshop on computational approaches to code switching, 2014, pp. 13–23.

[19] A. Hande, R. Priyadharshini, A. Sampath, K. P. Thamburaj, P. Chandran, B. R. Chakravarthi, Hope speech detection in under-resourced kannada language, 2021. arXiv:2108.04616.

[20] S. Thara, P. Poornachandran, Code-mixing: A brief survey, in: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 2382–2388. doi:10.1109/ICACCI.2018.8554413.

[21] K. Regmi, J. Naidoo, P. Pilkington, Understanding the processes of translation and transliteration in qualitative research, International Journal of Qualitative Methods 9 (2010) 16–26.

[22] P. Kalyan, D. Reddy, A. Hande, R. Priyadharshini, R. Sakuntharaj, B. R. Chakravarthi, Iiitt at case 2021 task 1: Leveraging pretrained language models for multilingual protest detection, in: CASE, 2021.

[23] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).

[24] B. R. Chakravarthi, R. Priyadharshini, S. Thavareesan, D. Chinnappa, D. Thenmozhi, E. Sherly, J. P. McCrae, A. Hande, R. Ponnusamy, S. Banerjee, C. Vasantharajan, Findings of the Sentiment Analysis of Dravidian Languages in Code-Mixed Text, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021.

[25] R. Priyadharshini, B. R. Chakravarthi, S. Thavareesan, D. Chinnappa, D. Thenmozi, E. Sherly, Overview of the dravidiancodemix 2021 shared task on sentiment detection in tamil, malayalam, and kannada, in: Forum for Information Retrieval Evaluation, FIRE 2021, Association for Computing Machinery, 2021.

[26] B. Krishnamurti, The dravidian languages, Cambridge University Press, 2003.

[27] A. Hande, R. Priyadharshini, B. R. Chakravarthi, 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, Association for Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 54–63. URL: https://www.aclweb.org/anthology/2020.peoples-1.6.

[28] B. R. Chakravarthi, 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, Association for Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 41–53. URL: https://aclanthology.org/2020.peoples-1.5.

[29] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, 2018. arXiv:1801.06146.

[30] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[31] J. P. C. Chiu, E. Nichols, Named entity recognition with bidirectional lstm-cnns, 2016. arXiv:1511.08308.

[32] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. arXiv:1609.08144.

[33] T. Pires, E. Schlinger, D. Garrette, How multilingual is multilingual BERT?, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4996–5001. URL: https://aclanthology.org/P19-1493. doi:10.18653/v1/P19-1493.

[34] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, 2020. arXiv:1910.03771.

[35] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.

[36] Z. Zhang, M. R. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, 2018. arXiv:1805.07836.

[37] A. F. Agarap, Deep learning using rectified linear units (relu), 2019. arXiv:1803.08375.

[38] S. Merity, N. S. Keskar, R. Socher, Regularizing and optimizing lstm language models, 2017. arXiv:1708.02182.

[39] A. Hande, K. Puranik, K. Yasaswini, R. Priyadharshini, S. Thavareesan, A. Sampath, K. Shanmugavadivel, D. Thenmozhi, B. R. Chakravarthi, Offensive language identification in low-resourced code-mixed dravidian languages using pseudo-labeling, 2021. arXiv:2108.12177.

[40] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, Physica D: Nonlinear Phenomena 404 (2020) 132306. URL: http://dx.doi.org/10.1016/j.physd.2019.132306. doi:10.1016/j.physd.2019.132306.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems,

2017, pp. 5998–6008.

[42] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, 2016. arXiv:1409.0473.

[43] M. Thomas, C. Latha, Sentimental analysis of transliterated text in malayalam using recurrent neural networks, Journal of Ambient Intelligence and Humanized Computing (2020) 1–8.

[44] G. Ramesh, S. Doddapaneni, A. Bheemaraj, M. Jobanputra, R. AK, A. Sharma, S. Sahoo, H. Diddee, M. J, D. Kakwani, N. Kumar, A. Pradeep, K. Deepak, V. Raghavan, A. Kunchukuttan, P. Kumar, M. S. Khapra, Samanantar: The largest publicly available parallel corpora collection for 11 indic languages, 2021. arXiv:2104.05596.

[45] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, M. Auli, fairseq: A fast, extensible toolkit for sequence modeling, 2019. arXiv:1904.01038.

[46] K. Puranik, A. Hande, R. Priyadharshini, T. Durairaj, A. Sampath, K. Thamburaj, B. R. Chakravarthi, Attentive fine-tuning of transformers for translation of low-resourced languages @loresmt 2021, 2021.

[47] K. Papineni, S. Roukos, T. Ward, W. J. Zhu, Bleu: a method for automatic evaluation of machine translation (2002). doi:10.3115/1073083.1073135.

[48] A. Kumar, R. Cotterell, L. Padró, A. Oliver, Morphological analysis of the dravidian language family, 2017, pp. 217–222. doi:10.18653/v1/E17-2035.