Sentiment Classification of Code-Mixed Tweets using Bi-Directional RNN and Language Tags

Sainik Kumar Mahata[1], Dipankar Das[2], Sivaji Bandyopadhyay[3]
[1]Institute of Engineering and Management, Kolkata, India
[2,3]Jadavpur University, Kolkata, India
[1]sainik.mahata@gmail.com, [2]dipankar.dipnil2005@gmail.com
[3]sivaji.cse.ju@gmail.com

Abstract

Sentiment analysis tools and models have been developed extensively throughout the years, for European languages. In contrast, similar tools for Indian Languages are scarce. This is because, state-of-the-art pre-processing tools like POS tagger, shallow parsers, etc., are not readily available for Indian languages. Although, such working tools for Indian languages, like Hindi and Bengali, that are spoken by the majority of the population, are available, finding the same for less spoken languages like, Tamil, Telugu, and Malayalam, is difficult. Moreover, due to the advent of social media, the multi-lingual population of India, who are comfortable with both English ad their regional language, prefer to communicate by mixing both languages. This gives rise to massive code-mixed content and automatically annotating them with their respective sentiment labels becomes a challenging task. In this work, we take up a similar challenge of developing a sentiment analysis model that can work with English-Tamil code-mixed data. The proposed work tries to solve this by using bi-directional LSTMs along with language tagging. Other traditional methods, based on classical machine learning algorithms have also been discussed in the literature, and they also act as the baseline systems to which we will compare our Neural Network based model. The performance of the developed algorithm, based on Neural Network architecture, garnered precision, recall, and F1 scores of 0.59, 0.66, and 0.58 respectively.

1 Introduction

Sentiment analysis is the interpretation and classification of emotions (positive, negative, and neutral) within text data using text analysis techniques. It is one of the most important research areas in the domain of Natural Language Processing (NLP) and has garnered much attention in the recent past. Throughout the years, multiple state-of-the-Art sentiment analysis models have been developed for the well known European languages, using classical Machine Learning (ML) algorithms as well as the recently developed Neural Network (NN) models. In contrast, very few such models have been developed for Indian languages, due to their lower digital footprint, which results in the lack of annotated data. Also, various pre-processing tools like Parts-of-Speech (POS) taggers, tokenizers, parsers, etc., for Indian languages, are not readily available or are not of competitive quality. Although, recent advances have been made for the Indian languages that are spoken by the majority of the population, like Hindi and Bengali, the same cannot be said for under-resourced languages such as, Tamil, Telugu, and Malayalam. For over 2600 years, recorded Tamil literature has been documented. Sangam literature, the earliest period of Tamil literature, is dated from around 600 BC- 300 AD. Among the Dravidian languages, Tamil has the oldest existing literature. Tamil is the oldest living language in India.

Moreover, with the advent of social media, sentiment analysis research has become even more wide-spread (Mahata et al., 2020; Garain et al., 2020) as it takes into account conversations of customers around the social space and puts them into context. But, in the context of the Indian subcontinent, social media texts are not in one language and are largely code-mixed in nature. This is because India, has had much foreign acquaintance historically, and this has led the diaspora to adopt English as one of their official languages. Due to this, much of the Indian population are familiar with English as well as one or more regional languages (Mahata et al., 2019). This leads to communication in sentences, which contain more than one language in the same phrase (Soumil Mandal and Das, 2018).

Furthermore, in a code-mixed communication,

Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages, pages 28–35
April 20, 2021

words of different languages are generally written in Roman script, which leads to the formation of complex syntactical structures that are difficult to parse with traditional NLP tools. While traditional sentiment analysis models can model themselves on social media texts in one language, the same cannot be said for texts that are code-mixed in nature and also comprised of Indian low-resourced languages.

The proposed approach aims to mitigate this research problem for English-Tamil code-mixed texts and uses Bi-Directional Long-Short-term-Memory (LSTM)s (Hochreiter and Schmidhuber, 1997) to tag the texts with their respective sentiment. Language tagging of individual words was used as additional features while training the classification model. Moreover, the training corpus was passed through FastText (Bojanowski et al., 2016) embedding, to map the semantically similar words in a common 3D space. This mapping was also used to build the classification system. The designed model, when evaluated on test data, garnered an F1 score of 0.58.

Other baseline sentiment analysis models were also developed using classical ML algorithms and were used to compare the quality of the proposed algorithm developed by using NNs.

The rest of the paper is organized as follows. Section 2 describes some of the previous research work conducted on the domain of language identification and sentiment analysis of code-mixed texts. Section 3 describes the training and the test data used to develop and analyze our model. Section 4 introduces the model developed for identifying languages of individual words in a code-mixed sentence. Also, it describes our developed model and all the baseline models that were developed using traditional ML algorithms.Finally, section 5 and 6 deals with the evaluation of our model and the concluding remarks.

2 Related Work

Social media has become the voice of many people over the decades and it has special relations with real-time events. With its rise, a lot of data is being generated every day and information extraction from such data has become an important research area. Also, the multi-lingual speaker, who prefers to communicate in more than one language, when expressing their opinions, generates a new kind of language, known as code-mixed language. Since, these kinds of data are more or less always written in the Roman script, analyzing these kinds of data, with help of NLP tools, becomes even more difficult.

Over the years, many experiments have been performed on code-mixed data. These include language identification, sentiment analysis, etc., to name a few. Language identification tasks have been earlier performed on various language pairs, such as Spanish-English (Negrón Goldbarg, 2009), French-English (Voss et al., 2014), Hindi-English (Vyas et al., 2014; Das and Gambäck, 2014) and Bengali-English (Das and Gambäck, 2014). While these experiments were conducted with the help of dictionary word matching and ML-based algorithms such as Support Vector Machines (SVM), word-based logistic regression classifiers, and Latent Direchlet Allocation (LDA) (Blei et al., 2003), we use more state-of-the-art deep learning approaches to achieve the same.

Also, sentiment analysis or opinion mining from code-mixed data is a trivial task because

- Generally, code-mixed data is noisy in nature and requires cleaning and normalization.
- It needs several steps such as language identification and POS tagging.
- There is no sentiment annotated code-mixed lexicon available for any language pairs.
- The available code-mixed datasets are small in size to perform any unsupervised classification.

Sentiment analysis of Hindi-English code-mixed was performed by Joshi et al. (2016) which used sub-word level representations in LSTM architecture to perform it. Shalini et. al. (Shalini et al., 2018), attempted a case-study on sentiment analysis of English-Kannada, English-Hindi, and English-Bengali texts using various machine and deep learning methods settings, like i. Doc2Vec+SVM, ii. FastText+Softmax, iii. Bi-LSTM+SoftMax and iv. CNN+SoftMax. Their reported results showed better accuracy when using deep learning methods as compared to traditional machine learning methods.

Our work, on the other hand, is an amalgamation of all the methods pointed out earlier and incorporates language identification modules, the usage of FastText embeddings, and Bi-LSTM cells to develop the deep learning model.

3 Data

The data for building the sentiment analysis model for English-Tamil code-mixed data was collected from the ”Dravidian-CodeMix - FIRE 2020”[1] shared task. The organizers of the task provided us with Tamil-English and Malayalam-English code-mixed text data, derived from YouTube video comments. The dataset contained all the three types of code-mixed sentences – Inter-Sentential switch, Intra-Sentential switch, and Tag switching and had five output labels; Positive, Negative, Mixed Feelings, Not Tamil, and Unknown State. Most comments were written in Roman script with either Tamil / Malayalam grammar with English lexicon or English grammar with Tamil / Malayalam lexicon. Some comments were written in Tamil / Malayalam script with English expressions in between. Further, the English-Tamil dataset was divided into training, validation, and test data which had 11,335, 1,260, and 3,149 code-mixed sentence instances respectively.

4 Framework

After we collected the English-Tamil code-mixed labeled dataset, the initial pre-processing steps included the removal of extra characters to clean the data. The extra characters that were removed/cleaned included

- Removing mentions
- Removing punctuation
- Removing URLs
- Contracting extra white space
- Extracting words from hashtags

After the pre-processing step, we proceeded with tokenizing the cleaned sentences using the NLTK[2] library. Subsequently, we used this data to train FastText embedding. This was done, to map the words with similar meaning and context, close to each other in a 3D space. The skip-gram model was used instead of the continuous-bag-of-words (CBOW) model as skip-gram works best for low data sizes. The model took into account character n-grams from 3 to 6 characters. Using the trained model, we were able to extract word vectors of size 100. These word vectors were preserved to be used as input for our sentiment analysis model.

4.1 Language Identification

Apart from providing our model, with the sequential word vectors of sentences, we also decided on providing an extra input in the form of language tags of every word of the sentences. For this, we developed a language identification system, that was trained to classify individual words, written in Roman script, as either English or Tamil. To achieve this, we used the character-level LSTM architecture put forward by Mandal et al. (2018). This is a model having stacked LSTM of sizes 193-128-128-1, in order where 193 is the input dimension while 1 is the output dimension.

The training data was acquired by concatenating different datasets for both English and Tamil. For the English data, we used the words from the NLTK corpus, that contained 2,34,377 unique English words. For the Tamil data, we used the data from Google Dakshina Dataset[3]. This dataset contained 48,998 Tamil words, transliterated in Roman script. After adding up both the datasets, we were able to gather 2,83,332 words. Of this, 3,35,792 words were used for the training data and the rest 5,000 words were used as the test data. Also, since the data labels were imbalanced, we used the class_weight feature of sklearn[4] package to assign class weights.

The schematic of the developed language identification model is shown in Figure 1. After testing the model with 5,000 words, the model returned an accuracy of 96.89%. The other metrics for the model are shown in Table 1.

Metrics	Value
Accuracy	96.89%
Precision	0.94
Recall	0.96
F1-Score	0.95

Table 1: Accuracy metrics of the Language Identification model.

[1] https://dravidian-codemix.github.io/2020/

[2] https://www.nltk.org/

[3] https://github.com/google-research-datasets/dakshina

[4] https://scikit-learn.org/stable/modules/generated/sklearn.utils.\class_weight.compute_class_weight.html

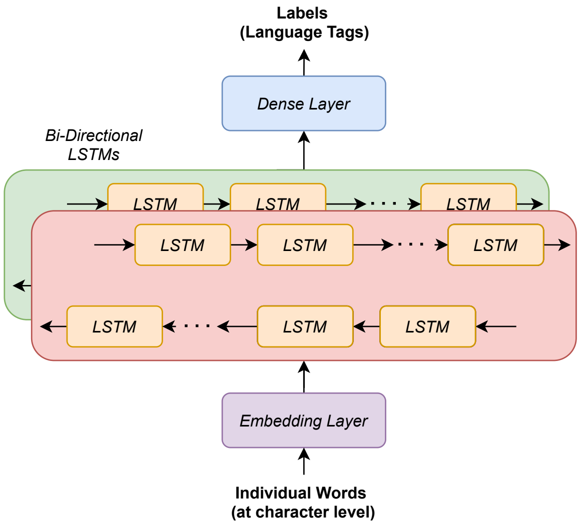

Figure 1: Classification model for language identification.

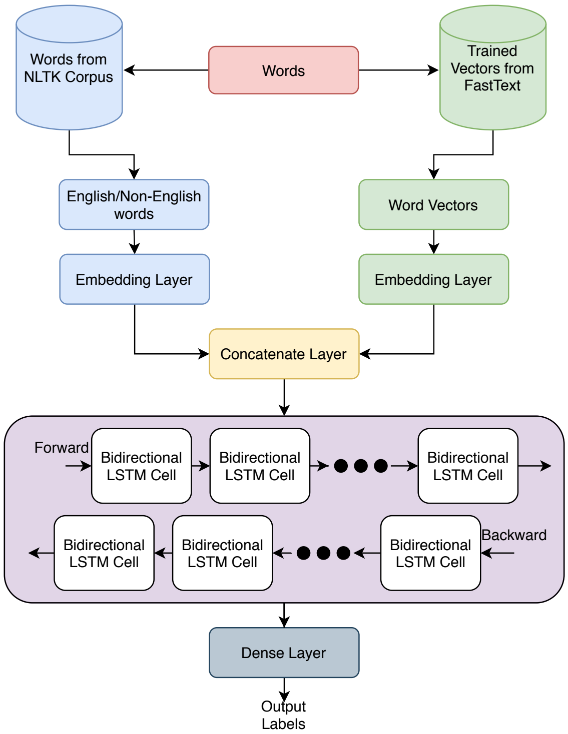

Figure 2: Code-Mixed Sentiment Analysis model.

4.2 Sentiment Classification

Using the language identification model, we were able to classify the words of the validation and the training data into either English or Tamil. Now, the next step was to develop the sentiment classification model which was to be designed for taking two inputs; i. the individual words of the code-mixed tweets and ii. the language tags of the individual words in the code-mixed tweets. The vectors of the individual words of the training data, as discussed earlier, were extracted from the already trained FastText embedding file. Thereafter, vectors of sentences of the train and validation dataset were extracted from the trained embedding. The language tags and the word vectors were merged using a Concatenation layer and were given as input to a Bi-Directional LSTM cell. The context vector was then mapped to the output labels with the help of a Dense layer.
The schematic of the model is shown in Figure 2.
Other parameters of the model are as follows.

- batch size: 32
- epochs: 50
- optimizer: adam
- loss: sparse categorical cross-entropy
- validation split: 0.1

On validating the developed model using a validation split of 0.1 (1,260 sentences), it garnered accuracy and F1-Score of 70.42% and 0.63 respectively. We also trained three other models, where the basic architecture was the same, the difference being the usage of LSTM/Bi-Directional LSTM and language tag features. The models were

- Bidirectional LSTM without the language tag feature.
- LSTM with the language tag feature.
- LSTM without the language tag feature.

The accuracy and F1-Score of every model are shown in Table 2.

4.3 Baseline Models

For developing the baseline models, we decided on using traditional ML algorithms. The algorithms chosen were,

- Naive Bayes algorithm
- Logistic Regression algorithm
- Support Vector Machine algorithm
- Random Forest algorithm

Four types of models with different features were selected to develop the models. Count Vectorizer, which converts a collection of text documents to a

Model	Bi-LSTM+ln tag	Bi-LSTM	LSTM+ln tag	LSTM
Accuracy	70.42%	70.82%	70.62%	70.22%
F1-Score	0.63	0.61	0.62	0.62
Precision	0.62	0.59	0.63	0.62
Recall	0.70	0.71	0.71	0.70

Table 2: Comparison of accuracy scores of the developed models built using NN architecture.

Model	Algorithm	Features	Accuracy	Precision	Recall	F1-Score
NB	**CV**	**Word**	65.23%	0.52	0.65	0.58
		Word+Ln Tag	69.20%	0.55	0.69	0.61
	TF-IDF	**Word**	64.96%	0.51	0.65	0.57
		Word+Ln Tag	69.68%	0.56	0.70	0.62
LR	**CV**	**Word**	65.22%	0.51	0.65	0.57
		Word+Ln Tag	68.65%	0.53	0.69	0.60
	TF-IDF	**Word**	66.54%	0.53	0.67	0.59
		Word+Ln Tag	70.23%	0.56	0.70	0.62
SVM	**CV**	**Word**	65.34%	0.52	0.65	0.58
		Word+Ln Tag	68.88%	0.53	0.69	0.60
	TF-IDF	**Word**	65.89%	0.52	0.66	0.58
		Word+Ln Tag	69.44%	0.53	0.69	0.60
RF	**CV**	**Word**	65.12%	0.49	0.65	0.56
		Word+Ln Tag	69.76%	0.54	0.70	0.61
	TF-IDF	**Word**	64.27%	0.51	0.64	0.57
		Word+Ln Tag	69.60%	0.53	0.70	0.60

Table 3: Comparison of accuracy scores of the developed models built using ML algorithms.

matrix of token counts was used as a feature. This implementation produces a sparse representation of the counts. Since we did not provide an a-priori dictionary and did not use an analyzer that does some kind of feature selection, the number of features was equal to the vocabulary size found by analyzing the data.

For the second model, we used the TF-IDF Vectorizer, with maximum features of 5000, where it converts a collection of raw documents to a matrix of TF-IDF features. We used the 2-gram and 3-gram range for this.

Also, for the third and the fourth model, the same features, Count Vectorizer and TF-IDF Vectorizer were used but in this case, we went for data augmentation, where the input was changed from words only to the form of $Word\_LanguageTag$.

On validation, the accuracy metrics garnered by the developed models, are shown in Table 3.

5 Evaluation

From Tables 2 and 3, we can see that though the ML and DL models perform neck-in-neck, but still, we preferred the DL model, developed using Bidirectional LSTM's and language tag feature as it garnered the highest F1-Score. This model was then tested using 3,149 test data, provided by the shared task organizers. The results of the testing phase of the selected model are quantified in Table 4.

Model	Precision	Recall	F1-Score
Bi-LSTM+ ln tag	0.59	0.66	0.58

Table 4: Final evaluation of the model, developed using Bidirectional LSTMs and Language Tag features.

6 Conclusion

In the current work, we attempted to solve the problem of Sentiment Analysis of code-mixed English-Tamil sentences. Our system was based on using Bi-Directional LSTM along with Language Tag features. Also, FastText embedding was used to generate word vectors to train the model. For predicting the language tags, another deep learning system, based on character embedding was also developed. Other models, based on traditional ML algorithms were also developed that was used to compare our developed model. Our system, when evaluated on the test data, garnered an F1 score of 0.58. As future work, we would like to increase this data, as deep learning algorithms tend to work well with higher amount of data and use state-of-the-art Neural Network architectures, like BERT, RoBERTa, etc., on this data, taking into advantage the concept of matrix and embedded language, SentiWordNet, and other NLP features.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Bharathi Raja Chakravarthi. 2020a. 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*, pages 41–53, Barcelona, Spain (Online). Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2020b. *Leveraging orthographic information to improve machine translation of under-resourced languages*. Ph.D. thesis, NUI Galway.

Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae. 2018. Improving wordnets for under-resourced languages using machine translation. In *Proceedings of the 9th Global Wordnet Conference*, pages 77–86, Nanyang Technological University (NTU), Singapore. Global Wordnet Association.

Bharathi Raja Chakravarthi, Mihael Arcan, and John P. McCrae. 2019. WordNet gloss translation for under-resourced languages using multilingual neural machine translation. In *Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation*, pages 1–7, Dublin, Ireland. European Association for Machine Translation.

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, M Anand Kumar, John Philip McCrae, Premjith B, Soman KP, and Thomas Mandl. 2020b. 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*.

Bharathi Raja Chakravarthi and Vigneshwaran Muralidaran. 2021. Findings of the shared task on Hope Speech Detection for Equality, Diversity, and Inclusion. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020c. 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, Shubhanker Banerjee, Richard Saldhana, John Philip McCrae, Anand Kumar M, Parameswari Krishnamurthy, and Melvin Johnson. 2021a. Findings of the shared task on Machine Translation in Dravidian languages. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

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. 2021b. 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. 2020d. 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.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John Philip McCrae. 2020e. Overview of the track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text. In *Proceedings of the 12th Forum for Information Retrieval Evaluation*, FIRE '20.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, and John Philip Sherly, Elizabeth McCrae. 2020f. Overview of the track on Sentiment Analysis for Dravidian Languages in Code-Mixed Text. In *Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2020). CEUR Workshop Proceedings. In: CEUR-WS. org, Hyderabad, India*.

Bharathi Raja Chakravarthi, Navaneethan Rajasekaran, Mihael Arcan, Kevin McGuinness, Noel E. O'Connor, and John P. McCrae. 2020g. Bilingual lexicon induction across orthographically-distinct under-resourced Dravidian languages. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 57–69, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text.

Avishek Garain, Sainik Kumar Mahata, and Dipankar Das. 2020. JUNLP at SemEval-2020 task 9:Sentiment Analysis of Hindi-English code mixed data using Grid Search Cross Validation. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain. Association for Computational Linguistics.

Nikhil Kumar Ghanghor, Parameswari Krishnamurthy, Sajeetha Thavareesan, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2021a. IIITK@DravidianLangTech-EACL2021: Offensive Language Identification and Meme Classification in Tamil, Malayalam and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, Online. Association for Computational Linguistics.

Nikhil Kumar Ghanghor, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021b. IIITK@LT-EDI-EACL2021: Hope Speech Detection for Equality, Diversity, and Inclusion in Tamil, Malayalam and English. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, Online.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

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.

Siddhanth U Hegde, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. UVCE-IIITT@DravidianLangTech-EACL2021: Tamil Troll Meme Classification: You need to Pay more Attention. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

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.

Sainik Kumar Mahata, Sushnat Makhija, Ayushi Agnihotri, and Dipankar Das. 2020. Analyzing code-switching rules for english–hindi code-mixed text. In *Emerging Technology in Modelling and Graphics*, pages 137–145, Singapore. Springer Singapore.

Sainik Kumar Mahata, Soumil Mandal, Dipankar Das, and Sivaji Bandyopadhyay. 2019. Code-mixed to monolingual translation framework. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 30–35, New York, NY, USA. Association for Computing Machinery.

Soumil Mandal, Sourya Dipta Das, and Dipankar Das. 2018. Language identification of bengali-english code-mixed data using character & phonetic based lstm models. *arXiv preprint arXiv:1803.03859*.

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.

Rosalyn Negrón Goldbarg. 2009. Spanish-english codeswitching in email communication. *Language@ internet*, 6(3).

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.

Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. IIITT@LT-EDI-EACL2021-Hope Speech Detection: There is always hope in Transformers. In *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*. Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

K. Shalini, H. B. Ganesh, M. A. Kumar, and K. P. Soman. 2018. Sentiment analysis for code-mixed indian social media text with distributed representation. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1126–1131.

Sainik Kumar Mahata Soumil Mandal and Dipankar Das. 2018. Preparing bengali-english code-mixed corpus for sentiment analysis of indian languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Shardul Suryawanshi and Bharathi Raja Chakravarthi. 2021. Findings of the shared task on Troll Meme Classification in Tamil. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (MultiOFF) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation. In *2019 14th Conference on Industrial and Information Systems (ICIIS)*, pages 320–325.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020a. Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts. In *2020 Moratuwa Engineering Research Conference (MERCon)*, pages 272–276.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020b. Word embedding-based Part of Speech tagging in Tamil texts. In *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, pages 478–482.

Clare R Voss, Stephen Tratz, Jamal Laoudi, and Douglas M Briesch. 2014. Finding romanized arabic dialect in code-mixed tweets. In *LREC*, pages 2249–2253.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979.

Konthala Yasaswini, Karthik Puranik, Adeep Hande, Ruba Priyadharshini, Sajeetha Thavareesan, and Bharathi Raja Chakravarthi. 2021. 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.