Leveraging Dynamic Meta Embedding for Sentiment Analysis and Detection of Homophobic/Transphobic Content in Code-mixed Dravidian Languages

Asha Hegde, Hosahalli Lakshmaiah Shashirekha

Department of Computer Science, Mangalore University, Mangalore, India

Abstract

Sentiment Analysis (SA) examines people's feelings, opinions, sentiments, views, and attitudes towards entities such as products, movies, services, organizations, and so on, whereas Homophobic/Transphobic (H/T) content identification aims to detect abusive behaviors, such as hate speech, sexism, racism specifically toward Lesbian, Gay, Bisexual, and Transgender (LGBT) people in any text. In parallel with the growth of social media, the code-mixed content for SA and H/T detection is also increasing creating a demand for the tools which efficiently analyze such content. However, SA and H/T content detection tasks in social media text are challenging due to the complex nature of the code-mixed text. To tackle this issue, in this paper, we - team MUCS, describe a learning model submitted to "Sentiment Analysis and Homophobia Detection of YouTube Comments in Code-Mixed Dravidian Languages" shared task at Forum for Information Retrieval Evaluation (FIRE) 2022. The proposed methodology makes use of Dynamic Meta Embedding (DME) to train the Deep Learning (DL) based Long Short Term Memory (LSTM) model to perform SA and detect H/T content in code-mixed Dravidian languages viz. Kannada, Malayalam, and Tamil. Models submitted to the shared tasks, obtained 6th, 4th, and 9th rank for Tamil, Malayalam, and Kannada in Task A and 1st, 4th, 1st, and 5th rank for Tamil, English, Tamil-English, and Malayalam in Task B respectively.

Keywords

Dravidian Languages, Code-mixed, Sentiment Analysis, Homophobia, Transphobia, Dynamic Meta Embedding

1. Introduction

The increasing number of social media platforms and the anonymity of users on these platforms have enabled more people to share their freedom of expression than ever before. This is increasing the user-generated content such as opinions, sentiments, reviews about products and movies, likes and dislikes about an event or news, objectionable content such as threats and remarks directed at individuals, groups or organizations: fake news, abusive language, hope and motivational words, and so on. SA aims to identify the sentiments of the given text and categorize them into predefined classes such as positive, negative, neutral, etc., and has

Forum for Information Retrieval Evaluation, December 9-13, 2022, India
hegdekasha@gmail.com (A. Hegde); hlsrekha@gmail.com (H. L. Shashirekha)
https://mangaloreuniversity.ac.in/dr-h-l-shashirekha (H. L. Shashirekha)

received considerable attention in industries as a means of determining customer fulfillment with services and products [1]. H/T content identification deals with detecting abusive speech toward LGBT people only because of who they love, how they appear, or who they are. Across the globe, LGBT people are subjected to violence, inequity, torture, and even execution. Due to this, LGBT people who seek online support are being targeted, threatened, and abused, resulting in severe mental health problems. Hence, automatic identification and removal of such content from social media is the need of the day towards promoting equality, diversity, and inclusion in society [2].

SA and identifying H/T content in social media text is challenging because of the complex nature of code-mixed text available on social media platforms. Usually, social media text is written by mixing one or more local or regional languages, for instance, Kannada, Malayalam, Tamil, etc., with English, either at word and/or sentence level [3] [4]. Additionally, the usage of short forms for words, (ex. 'g8' for 'good night'), internet slangs (ex. 'plz' for 'please'), words/phrases from other languages, emojis, hashtags, text consisting of recurrent characters (ex. 'soooooo sad' for 'so sad'), etc., escalates the complexities in processing code-mixed text [5]. Further, the rapid growth of social media users intensifies the problem further necessitating efficient tools or learning models for SA and H/T content identification. The sample text from the dataset provided by the organizers of the shared task is given in Table 1.

Sentiment Analysis			
Language	**Sample text**	**Label**	**English translation**
Tamil	திருவண்ணாமலை யாதவர்கள் சார்பாக படம் வெற்றி பெற வாழ்த்துக்கள்	Postive	On behalf of the Thiruvannamalai Yadavs, I wish the film success
Malayalam	എവിടെയോ രാക്ഷസൻ bgm പോലെ	Mixed_feelings	Somewhere like monster bgm
Kannada	ನಿಮ್ಮ ಈ ಸಾಮಾನ್ಯ ಜೀವನ ನೋಡಿ ಕುಷಿ ಆಯಿತು	Positive	I was happy to see your normal life

Homophobic/Transphobic Content Detection			
Language	**Sample text**	**Label**	**English translation**
Tamil	விபச்சாரத்தை விட கொடியது ஓரின சேர்க்கை	Homophobic	Homosexuality is worse than adultery
Malayalam	സത്യത്തിൽ ഇത് സ്ത്രീ പുരുഷ അവഹേളനമാണ് ട്രാൻസ്ജനറേഷൻ	Transphobic	In fact, it is a transgenerational insult to men and women

Table 1
Sample text from the given dataset for SA and H/T content detection

To address the challenges of processing social media text particularly in code-mixed Dravidian Languages for SA and H/T content identification, in this paper we - team MUCS describe the models submitted to "Sentiment Analysis and Homophobia detection of YouTube comments in Code-Mixed Dravidian Languages" shared task[1] at FIRE 2022. The shared task consists of two subtasks: i) Task A - is a message-level polarity classification task for SA in code-mixed Dravidian languages viz. Kannada, Tamil, and Malayalam, and ii) Task B - is to identify H/T content in code-mixed Tamil, Malayalam, and English texts written in their native script and Tamil-English text written in Latin script [6]. The proposed methodology makes use of DME to

[1]https://codalab.lisn.upsaclay.fr/competitions/5310#learn_the_details

train DL based LSTM models to perform SA and detect H/T content in code-mixed text.

The rest of the paper is structured as follows: Section 2 contains related works and Section 3 explains the methodology. Section 4 describes the experiments, as well as the outcomes, and the paper concludes in Section 5 with future work.

2. Related work

Several researchers have explored SA in Dravidian languages and few of the relevant ones are described below:

Chakravarthi et al. [7] created a Tamil-English code mixed corpus of 15,744 YouTube comments for sentiment classification. Their study uses Machine Learning (ML) models (Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), Naive Bayes (NB), k-Nearest Neighbor (kNN)) and DL based 1D Convolutional-Long Short Term Memory (1D-convLSTM) classifier and transformer-based classifier with multilingual Bidirectional Encoder Representations from Transformers (mBERT) to classify YouTube comments. Term Frequency-Inverse Document Frequency (TF-IDF) of n-grams in the range n = (1, 3) is used to train ML classifiers and Keras embeddings to train 1D-convLSTM classifier. Among all the models, RF classifier obtained a maximum weighted F1 score of 0.65. Kusampudi et al. [8] presents code-mixed Telugu-English corpus extracted from Twitter and blogs of size 9,657 and 24,404 sentences respectively to perform SA. The authors developed ML models (SVM, NB, LR, kNN, and RF) for SA with TF-IDF of character and word n-grams both in the range n = (1, 3) as features. They also implemented DL based Bidirectional LSTM (BiLSTM) and a hybrid model combining BiLSTM and Conditional Random Field (BiLSTM+CRF) to perform SA with Keras embeddings as features. BiLSTM model obtained a better accuracy of 0.98 on the blog dataset and BiLSTM+CRF model exhibited an accuracy of 0.99 on the Twitter dataset. Chakravarthi et al. [9] created a Malayalam-English code-mixed dataset of 6,738 sentences extracted from YouTube comments using YouTube comment scraper[2] for SA. The authors implemented ML models (LR, SVM, DT, RF, MNB, and kNN), DL models (1DConvLSTM and LSTM), and a transformer-based classifier with mBERT to perform SA. They used TF-IDF of word tri-grams and Keras embeddings as features to train ML and DL models respectively. Among all the models, mBERT outperformed the other models with an F1 score of 0.75.

Several workshops and shared tasks are focusing on H/T content identification in social media text and prominent among them is the Homophobia/Transphobia Detection shared task at Language Technology for Equality, Diversity and Inclusion (LT-EDI) - Association for Computational Linguistics (ACL) 2022 which focuses on detecting H/T content in English and in code-mixed Dravidian languages viz. Tamil text in the native script and Tamil text in Latin script[3] [10]. The following are few of the recent works related to the detection of H/T content in Dravidian languages:

Swaminathan et al. [11] developed two SVM classifiers with TF-IDF and GloVe embeddings as features and a transformer-based classifier with mBERT to detect H/T content. Transformer-based classifier with mBERT outperformed the SVM classifier with weighted F1 scores of

[2]https://github.com/philbot9/

[3]https://competitions.codalab.org/competitions/36394

0.93, 0.75, and 0.87 securing 11th, 9th, and 9th rank for English, Tamil, and Tamil-English respectively. Transformer-based classifiers proposed by Bhandari and Goyal [12] to detect H/T content makes use of IndicBERT, cross-lingual language models with Robustly Optimized BERT (XLM-RoBERTa), and mBERT as features to train transformer-based classifiers. Among all the models, the transformer-based classifier with mBERT exhibited maximum weighted F1 scores of 0.42, 0.64, and 0.58 placing 9th, 6th, and 3rd ranks in the shared task for English, Tamil, and Tamil-English respectively.

From the literature, it is clear that though several works are carried out to perform SA and H/T content identification in Dravidian languages, there is still scope for developing tools and models in this direction as the results are considerable.

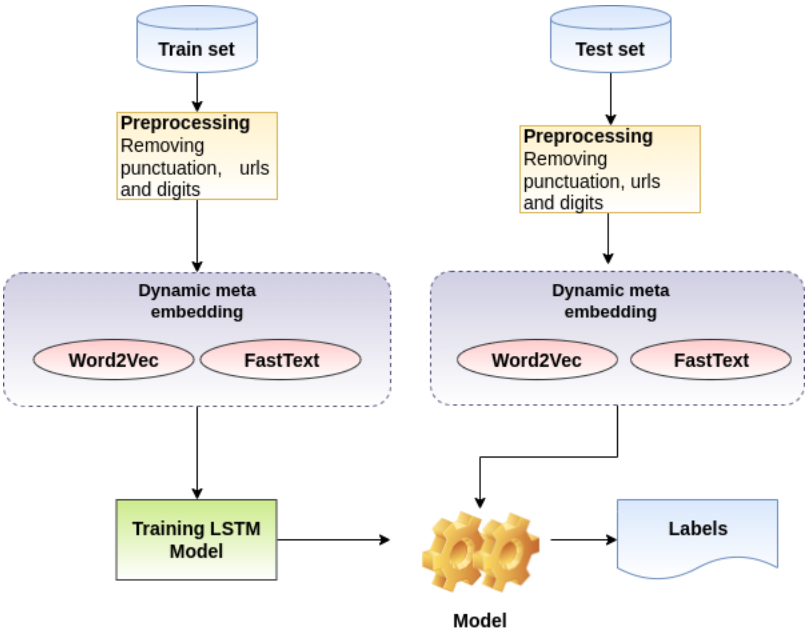

Figure 1: Framework of the proposed method

3. Methodology

The proposed methodology for SA and detection of H/T content in code-mixed Dravidian languages includes three major steps: Preprocessing, Text vectorization, and Classifier construction. The framework of the proposed methodology is shown in Figure 1 and the steps are explained below:

Preprocessing - is the process of cleaning text data with the aim of improving the performance of the classifier. The text is preprocessed by converting emojis into text and removing digits, punctuation, URLs, and stopwords. English stopwords list available in Natural Language

Toolkit (NLTK)[4] library, Kannada stopwords list available at github[5], and Tamil stopwords list available at github[6] are used to remove the stopwords from the respective languages.

Text vectorization - aims to transform the text into vector values which are in turn used to train the learning models. Distributed representation of words, also known as word embeddings, is a popular word representation technique, where each word is represented by a low-dimensional vector such that words having the same meaning will have a similar representation [13]. Word2Vec[7], fastText[8], GloVe[9], etc., are some popular word embedding models with a very large vocabulary available in various dimensions such as 50, 100, 300, etc. However, selecting the correct embeddings out of the available embedding techniques for specific tasks is always challenging. Further, the usefulness of word embeddings for downstream tasks, such as text classification, machine translation, text summarization, natural language understanding, etc., tends to be hard to predict. Therefore, instead of considering any single embeddings it is beneficial to combine the strengths of different word embeddings. This also increases the lexical coverage by allowing systems to take the union of the vocabulary of different embeddings.

DME is a supervised learning of embedding ensembles where the Neural Network (NN) decides which embeddings to use. This is achieved by adding the ensembled embedding layer allowing the network to learn the embeddings it prefers by predicting the weight for each embedding type. Instead of using a single word embedding, the proposed work utilizes DME in which the primary word embeddings are ensembled with additional learnable weights through an LSTM encoder. In this work, Word2Vec[10] and fastText[11] embeddings are built using gensim[12] library considering the training dataset provided by the shared task organizers and these embeddings are then ensembled to create the DME. Both the models are trained with a latent dimension of 100, a window size of 3 followed by a random seed of 33 with 10 epochs. In the proposed method, maximum sequence length is set to 200 followed by the stacking of two LSTM layers with a dropout of 0.3. Eventually, the softmax attention is used as the final layer with adam optimizer.

3.1. Model Construction

The goal of the shared task is to perform SA and detect H/T content in code-mixed Dravidian languages. To address these tasks, DL based LSTM model is implemented using DME features. Though the DL based models, namely Recurrent Neural Network and Convolutional Neural Network produce considerable results, these models suffer from a short-term memory issue during handling longer sentences that lead to vanishing gradient problems. During backpropagation, the gradient grows so small that it approaches zero, rendering the neuron useless for further processing. LSTM which memorizes the important information in the data by assigning

[4]https://www.nltk.org/nltk_data/
[5]https://gist.github.com/MSDarshan91
[6]https://gist.github.com/arulrajnet/
[7]https://code.google.com/archive/p/word2vec/
[8]https://fasttext.cc/docs/en/pretrained-vectors.html
[9]https://nlp.stanford.edu/projects/glove/
[10]https://radimrehurek.com/gensim/models/word2vec.html
[11]https://radimrehurek.com/gensim/models/fasttext.html
[12]https://radimrehurek.com/gensim/

Train set							
Languages	**Positive**	**Negative**	**Mixed feelings**	**Unknown state**	**not Kannada**	**not Tamil**	**not Malayalam**
Kannada	2,823	1,188	574	711	916	-	-
Tamil	20,069	4,271	4,020	5,628	-	1,667	-
Malayalam	6,421	2,105	926	5,279	-	-	1,157
Development set							
Kannada	321	139	69	52	119	-	-
Tamil	2,257	480	611	438	-	176	-
Malayalam	786	237	102	580	-	-	141

Table 2
Classwise distribution of the dataset for Task A

Train set				
Tag	**English**	**Tamil**	**Malayalam**	**Tamil-English**
Non-anti-LGBT+ content	3,001	2,022	2,434	3,438
Homophobic	157	485	491	311
Transphobic	6	155	189	112
Development set				
Non-anti-LGBT+ content	732	526	692	862
Homophobic	58	103	133	66
Transphobic	2	37	41	38

Table 3
Classwise distribution of the dataset for Task B

weights to them can be used to resolve the vanishing gradient problem. Hence, LSTM is helpful when dealing with longer sentences. With appropriate embedding layers and an LSTM encoder, the model will be able to produce good results.

4. Experiments and Results

The statistics of the datasets provided by the shared task organizers for Task A [14] and Task B [15] are given in Table 2 and 3 respectively. It is clear that both the datasets are imbalanced and this may affect the performance of the learning models. The proposed models were used to predict the class labels of the unlabeled Test sets provided by the organizers and the predictions were submitted to the organizers for evaluation. The predictions were evaluated and ranked by the organizers based on the F1 score. As per the results in the leaderboard of the shared task, the proposed DL based LSTM model with DME obtained considerable accuracy. Performance of the proposed method for Task A and B along with the ranks obtained in the shared task are given in Table 4. In Task A, the proposed method exhibited the lowest F1 score of 0.16 for Tamil language, where 56% comments in the Tamil dataset belong to the 'positive' class reflecting the imbalance in the classwise distribution of the dataset. But, the proposed method obtained a better F1 score of 0.61 for Malayalam, as the Malayalam dataset contains better distribution of classes compared to that of Tamil dataset. Similarly, in Task B, Malayalam dataset has fairly

Task A		
Language	**F1 score**	**Rank**
Tamil	0.16	6
Malayalam	0.61	4
Kannada	0.44	9
Task B		
Tamil	0.36	1
Malayalam	0.74	5
English	0.37	4
Tamil-English	0.58	1

Table 4
Performance measure of the proposed method for Task A and B

distributed comments over all the classes compared to the other datasets. Hence, the proposed method obtained better F1 score of 0.74 for Malayalam dataset.

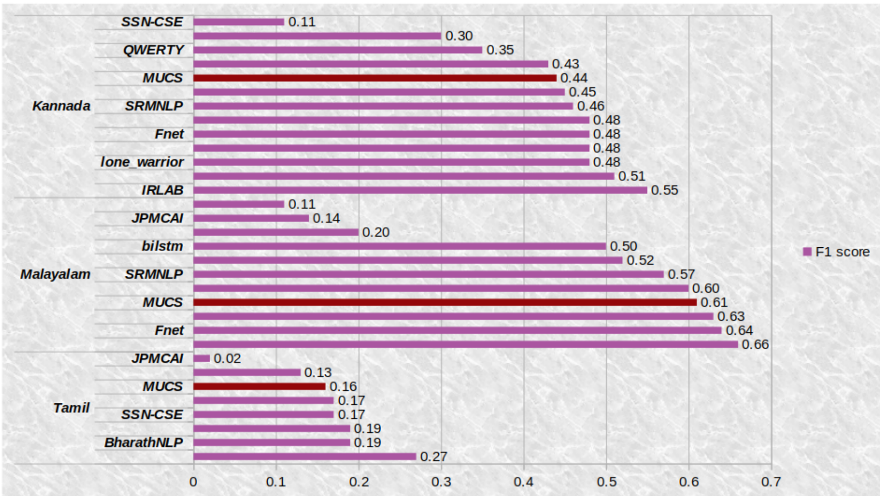

Figure 2: Comparison of F1 scores of the participating teams for Task A

The proposed method exhibited considerable F1 scores of 0.16, 0.61, and 0.44 securing 6^{th}, 4^{th}, and 9^{th} rank for Tamil, Malayalam, and Kannada respectively in Task A. For Task B, the models exhibited F1 scores of 0.36, 0.74, 0.58, and 0.37 securing 1^{st}, 4^{th}, 1^{st}, and 5^{th} rank for Tamil, English, Tamil-English, and Malayalam respectively. Figure 2 and 3 show the comparison of F1 scores of all the participating teams for Task A and B respectively which illustrate that the performance of the proposed DL based LSTM model with DME is considerable.

5. Conclusion and Future work

This paper describes the models proposed by team MUCS for SA and identification of H/T content in the social media text, particularly in code-mixed Dravidian languages submitted to

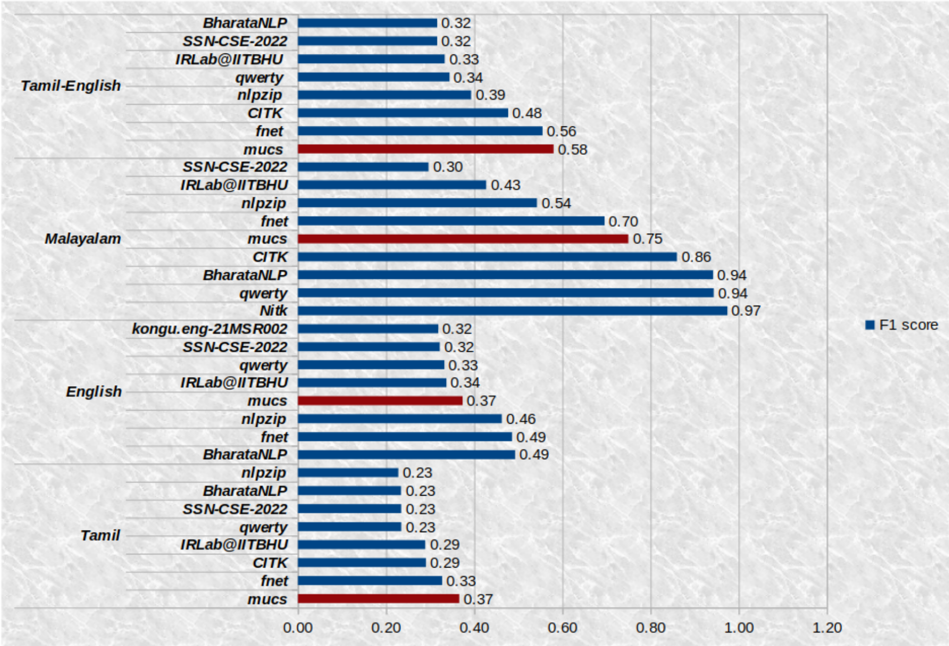

Figure 3: Comparison of F1 scores of the participating teams for Task B

"Sentiment Analysis and Homophobia Detection of YouTube Comments in Code-Mixed Dravidian Languages" - a shared task at FIRE 2022. In the proposed strategy, DME feature is used to train DL based LSTM model for SA and identification of H/T in code-mixed Dravidian languages viz. Kannada, Malayalam, and Tamil. The proposed models have exhibited considerable F1 scores of 0.36, 0.74, and 0.37 for Tamil, English, and Malayalam respectively in Task A and F1 scores of 0.36, 0.74, 0.58, and 0.37 for Tamil, English, Tamil-English, and Malayalam respectively in Task B. These models secured 6th, 4th, and 9th rank for Tamil, Malayalam, and Kannada respectively in Task A and 1st, 4th, 1st, and 5th rank for Tamil, English, Tamil-English, and Malayalam respectively in Task B. Investigation of efficient resampling techniques to handle imbalanced classes with effective feature extraction will be explored in future work.

References

[1] 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, 2020, pp. 54–63.

[2] E. A. McConnell, A. Clifford, A. K. Korpak, G. Phillips II, M. Birkett, Identity, Victimization, and Support: Facebook Experiences and Mental Health among LGBTQ Youth, in: Computers in Human Behavior, Elsevier, 2017, pp. 237–244.

[3] B. B. KACHRU, Toward Structuring Code-Mixing: An Indian Perspective, in: Walter de

Gruyter, 1978, pp. 27–46.
[4] A. Hegde, M. D. Anusha, S. Coelho, H. L. Shashirekha, B. R. Chakravarthi, Corpus Creation for Sentiment Analysis in Code-Mixed Tulu Text, in: Proceedings of SIGUL 2022 @LREC2022, 2022, pp. 33–40.
[5] A. Hegde, M. D. Anusha, H. L. Shashirekha, Ensemble Based Machine Learning Models for Hate Speech and Offensive Content Identification, in: Forum for Information Retrieval Evaluation (Working Notes) (FIRE), 2021, pp. 43–49.
[6] K. Shumugavadivel, M. Subramanian, P. K. Kumaresan, B. R. Chakravarthi, B. B, S. Chinnaudayar Navaneethakrishnan, L. S.K, T. Mandl, R. Ponnusamy, V. Palanikumar, M. Balaji J, Overview of the Shared Task on Sentiment Analysis and Homophobia Detection of YouTube Comments in Code-Mixed Dravidian Languages, in: Working Notes of FIRE 2022 - Forum for Information Retrieval Evaluation, CEUR, 2022.
[7] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, J. P. McCrae, Corpus Creation for Sentiment Analysis in Code-Mixed Tamil-English Text, in: arXiv preprint arXiv:2006.00206, 2020.
[8] S. S. V. Kusampudi, A. Chaluvadi, R. Mamidi, Corpus Creation and Language Identification in Low-Resource Code-Mixed Telugu-English Text, in: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), 2021, pp. 744–752.
[9] 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), 2020, pp. 177–184.
[10] B. R. Chakravarthi, R. Priyadharshini, T. Durairaj, J. P. McCrae, P. Buitelaar, P. Kumaresan, R. Ponnusamy, Overview of The Shared Task on Homophobia and Transphobia Detection in Social Media Comments, in: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, 2022, pp. 369–377.
[11] K. Swaminathan, B. Bharathi, G. Gayathri, H. Sampath, Ssncse_NLP@ LT-EDI-ACL2022: Homophobia/Transphobia Detection in Multiple Languages using SVM Classifiers and Bert-based Transformers, in: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, 2022, pp. 239–244.
[12] V. Bhandari, P. Goyal, bitsa_nlp@LT-EDI-ACL2022: Leveraging Pretrained Language Models for Detecting Homophobia and Transphobia in Social Media Comments, in: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion, Association for Computational Linguistics, 2022, pp. 149–154.
[13] D. J. Chalmers, Syntactic Transformations on Distributed Representations, in: Connectionist natural language processing, Springer, 1992, pp. 46–55.
[14] B. R. Chakravarthi, R. Priyadharshini, V. Muralidaran, N. Jose, S. Suryawanshi, E. Sherly, J. P. McCrae, Dravidiancodemix: Sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text, in: Language Resources and Evaluation, Springer, 2022, pp. 1–42.
[15] B. R. Chakravarthi, R. Priyadharshini, R. Ponnusamy, P. K. Kumaresan, K. Sampath, D. Thenmozhi, S. Thangasamy, R. Nallathambi, J. P. McCrae, Dataset for Identification of Homophobia and Transophobia in Multilingual YouTube Comments, in: arXiv preprint

arXiv:2109.00227, 2021.