KaustubhSharedTask@LT-EDI 2023: Homophobia-Transphobia Detection in Social Media Comments with NLPAUG-driven Data Augmentation

Kaustubh Lande[1], **Rahul Ponnusamy**[2], **Prasanna Kumar Kumaresan**[2], **Bharathi Raja Chakravarthi**[3]
[1] Indian Institute of Technology Kharagpur, India
[2] Insight SFI Research Centre for Data Analytics, University of Galway, Ireland
[3] School of Computer Science, University of Galway, Ireland
kaustubhlande2002@gmail.com
{rahul.ponnusamy, prasanna.kumaresan}@insight-centre.org
bharathi.raja@universityofgalway.ie

Abstract

Our research in Natural Language Processing (NLP) aims to detect hate speech comments specifically targeted at the LGBTQ+ community within the YouTube platform shared task conducted by the LT-EDI workshop[1]. The dataset provided by the organizers exhibited a high degree of class imbalance, and to mitigate this, we employed NLPAUG, a data augmentation library. We employed several classification methods and reported the results using recall, precision, and F1-score metrics. The classification models discussed in this paper include a Bidirectional Long Short-Term Memory (BiLSTM) model trained with Word2Vec embeddings, a BiLSTM model trained with Twitter GloVe embeddings, transformer models such as BERT, DistilBERT, RoBERTa, and XLM-RoBERTa, all of which were trained and fine-tuned. We achieved a weighted F1-score of 0.699 on the test data and secured fifth place in task B with 7 classes for the English language.

1 Introduction

The term "hate speech" refers to a specific style of offensive language that uses generalizations and stereotypes to convey an ideology of hatred (Warner and Hirschberg, 2012; Subramanian et al., 2022). Many people agree on the definition of "hate speech" as any kind of expression that targets an individual or group because of their race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic (Schmidt and Wiegand, 2017; Priyadharshini et al., 2022; Swaminathan et al., 2022b; Hariprasad et al., 2022). The development of user-generated content online, particularly on social media platforms, has contributed to an increase in the amount of hate speech that is being distributed (Karim et al., 2022; Chakravarthi et al., 2023a; B and Varsha, 2022).

[1]https://codalab.lisn.upsaclay.fr/competitions/11077

Over the past several years, there has been a rise in the amount of hate speech that can be found online, which has led to an increase in interest in the process of automating its detection (Santhiya et al., 2022). One such form of hate speech is homophobic or transphobic comments, which is hate targeted towards LGBTQ+ peoples (Chakravarthi, 2023). The procedure of Transphobia and Homophobia Detection entails discerning and isolating anti-LGBTQ+ content within a given corpus. Hate speech includes both homophobic and transphobic words, both of which are harmful to the LGBTQ+ community (Chakravarthi et al., 2022b; Shanmugavadivel et al., 2022).

In our research, we addressed the issue of class imbalance in our dataset by employing data augmentation techniques using NLPAUG, a Python library specifically designed for augmenting text data. This approach effectively mitigated the degree of class imbalance. Subsequently, we trained our models using various architectures including BiLSTM (Graves et al., 2005) with pre-trained Word2Vec (Church, 2017) embeddings and Twitter GLoVe (Pennington et al., 2014) embeddings, as well as BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), Roberta (Liu et al., 2019), and XLM-Roberta (Conneau et al., 2019), while fine-tuning them. Among these models, the best performance was achieved by the BiLSTM + Word2Vec model, which yielded a weighted F1-score of 0.56 on the validation dataset and 0.699 on the test dataset, placing it in the fifth rank in English.

2 Related Work

Chakravarthi et al. (2022a) created the homophobic/transphobic dataset and first to release the for public research. Chakravarthi et al. (2022b) and Chinnaudayar Navaneethakrishnan et al. (2023) organized shared first shared tasks to detect the

LT-EDI 2023 – Third Workshop on Language Technology for Equality, Diversity and Inclusion, pages 71–77, Varna, Bulgaria, Sep 7, 2023.
https://doi.org/10.26615/978-954-452-084-7_010

homophobia and transphobia comments from social media in English, Tamil, Tamil-English, and Malayalam language settings. There were many participants who competed in the shared tasks and produced system description papers.

García-Díaz et al. (2022) used a knowledge integration technique to train a neural network that effectively merges multiple feature sets. These include sentence embeddings in context and out of context, as well as linguistic features retrieved using a technique created by their research group. They got seventh, third, and second rank in English, Tamil, and Tamil-English respectively.

Following the implementation of sampling techniques to correct the data imbalance, feature extraction was conducted using a count vectorizer, TF-IDF, and a variety of classifiers. Among other techniques, SVM Classifiers, word embeddings, and BERT-based transformers were utilized by Swaminathan et al. (2022a). For the vectorization of remarks, TF-IDF has been combined with various bigram models, and Support Vector Machines was used to create the model by Ashraf et al. (2022). Upadhyay et al. (2022) utilized a collection of transformer-based models to construct a classifier and their system placed second for English, eighth for Tamil, and tenth for Tamil-English. Nozza (2022) used data augmentation and ensemble modeling along with different large language models (BERT, RoBERTa, and HateBERT) to fine-tune, and the weighted majority vote was applied to their predictions. Her proposed model received scores of 0.48 and 0.94 for the macro and weighted F1 scores, placing it in third place in English.

3 Dataset

The training dataset comprised 3,164 data, while the validation dataset contained 999 data, with both datasets featuring a single column for text and another column for associated labels. The test dataset encompassed 990 data, with the objective to predict the corresponding labels. During preprocessing, the validation dataset underwent cleaning, resulting in a reduced size of 792 data. The validation dataset underwent cleaning because some data points did not have the output so we tried to remove those text data which didn't have its output labels (Chakravarthi et al., 2023b).

The number of labels for each class in the training dataset and validation dataset are given in Table 1 and Table 2 respectively. To develop our classification models, we trained them on the training dataset and assessed their performance on the validation dataset. Ultimately, our final predictions for the labels were generated using the test dataset. Notably, the training dataset exhibited a substantial class imbalance, where certain classes were significantly underrepresented compared to others. Class imbalance occurs when the distribution of instances across different classes is skewed in this manner.

Type of labels	Training labels size
None-of-the-above	2240
Hope-Speech	436
Counter-speech	302
Homophobic-derogation	167
Homophobic-Threatening	12
Transphobic-derogation	6
Transphobic-Threatening	1

Table 1: Labels sizes in training dataset.

Type of labels	Validation labels size
None-of-the-above	553
Hope-Speech	111
Counter-speech	84
Homophobic-derogation	41
Transphobic-derogation	2
Homophobic-Threatening	1

Table 2: Labels sizes in validation dataset.

4 Methodology

In order to address the issue of high-class imbalance present in the dataset, we applied data augmentation techniques using the NLPAUG (Ma, 2019) library in Python. By augmenting the dataset, we aimed to ensure that the text inputs used for training the model would be diverse and representative,

Type of labels	Augmented labels size
None-of-the-above	2240
Hope-Speech	2114
Counter-speech	1746
Homophobic-derogation	1210
Homophobic-Threatening	786
Transphobic-derogation	109
Transphobic-Threatening	12

Table 3: Labels sizes in augmented dataset.

Figure 1: Text preprocessing steps

minimizing the risk of creating biases towards any specific label during prediction. The NLPAUG used for augmentation, augmented each text around 7-10 times similarly the label size of the Transphobic derogation in the original dataset consisted of only 1 label so we can augment it to only 12 sentences. Further experimental analysis and the detailed methodology for conducting these experiments are elaborated in the subsequent subsections of this paper.

4.1 Data Augmentation with NLPAUG

Data augmentation is a technique of artificially increasing the training set by creating modified copies of a dataset using existing data. This simply means we want to generate more data and more examples from our current dataset. So if there is a data (X, Y), where X is a sentence and Y is its corresponding label. So, we can imagine it to be like X is a comment and Y is the label associated with that comment. As a part of data augmentation, we transform this X and create X' out of it, while still preserving the label Y.

$$(X, Y) - - - T - - - > (X', Y) \qquad (1)$$

So, as we can see since Y is still preserved, which means the transformation that we want to apply, say, T, has to be semantically invariant which means it doesn't change the meaning of the original sentence. So, X' could be syntactically a little different compared to X, but semantically it should mean the same thing. To deal with the Data augmentation technique NLPAUG (a Python library) was used for textual augmentation. The goal was to improve deep learning model performance by generating textual data. Using NLPAUG reduced the degree of class imbalance which would make the model train better and generalize the labels better.

NLPAUG provides three different types of augmentation:

- Character level augmentation
- Word level augmentation
- Flow/Sentence level augmentation

As the class imbalance was very high we tried to augment each label in many different ways by using insert, substitute, swap, delete, and split actions on the words of the text so that they can augment sentences in many ways so that sentences generated should not repeat. As there was a need for the generation of many sentences we tried them to augment in many different ways. We tried with Word level augmentation.

Word-level augmentation uses trained word embeddings like GloVe, Word2Vec, and fastText to replace words with similar word embeddings. It helps to identify the closest word vector from latent space to replace the original sentence. Thus it helps to substitute and insert words with similar meanings and generate more sentences. We also tried Back Translation which comes with the NLPAUG package and generated a few sentences. The basic idea behind back-translation is to translate a sentence into another language and then back into the original language, with few word changes. So that it can be used to generate more training data to improve the model performance. We tried to give the parameter to the sentence to limit the words which can be changed or can be inserted or can be deleted so that they cannot change the whole meaning of the whole sentence. After augmentation, the labels generated with their sizes are shown in Table 3.

4.2 Preprocessing

To facilitate the hate speech detection task, necessary transformations were applied to the collected comments in the dataset, specifically targeting Homophobic and Transphobic data. This involved a series of preprocessing steps shown in Figure 1 to ensure the data was in a suitable format for analysis.

To improve the text understanding and minimize noise interference in algorithms, special characters, as well as numbers, were eliminated from the dataset. This preprocessing step was accomplished by utilizing the regular expressions (regex) library in Python. By removing these non-essential elements, the dataset was streamlined for further analysis and algorithmic processing.

In order to enhance the processing of meaningful data and account for potential gender biases when analyzing hate speech related to the LGBTQ+ community, we utilized the Natural Language Toolkit (NLTK)[2] library to create a list of stopwords. Stopwords are commonly used words in a language that contribute little information to the text. However, to ensure the preservation of gender-specific context and avoid potential bias, we made modifications to the stopwords class by removing certain words {"he," "him," "his," "himself," "she," "she's," "her," "hers," and "herself"}. By excluding these words from the stopwords class, we aimed to retain their impact and relevance in our analysis of hate speech targeting the LGBTQ+ community.

Lemmatization is an advanced form of stemming. Stemming might not result in an actual word, whereas lemmatization does conversion properly with the use of vocabulary, normally aiming to return the base form of a word, which is known as the lemma. To achieve this, we utilized the WordNetLemmatizer package from the NLTK library, ensuring the proper transformation of words in our analysis.

4.3 Training with BiLSTM using Word Embeddings

Word embeddings refer to a technique that converts individual words into numerical representations, commonly known as vectors. In this approach, each word is associated with a unique vector, and these vectors are learned in a manner resembling a neural network. The objective is to capture the diverse characteristics of each word within the context of the entire text. By leveraging word embeddings, we can effectively represent and analyze the semantic relationships between words in a text corpus.

We utilized pre trained Word2Vec and Twitter Glove embeddings to generate word representations, which were then employed to train a Bidirectional LSTM (BiLSTM) model. BiLSTM is a variation of the LSTM architecture, that enables the processing of data in both the forward and backward directions, effectively capturing contextual information from both past and future contexts.

We implemented a BiLSTM model by fine-tuning it using Grid Search CV. The maximum sequence length was set to 64, and each word was represented by a 128-dimensional vector. Our BiLSTM model consisted of a BiLSTM layer with 32 LSTM units. The output from the BiLSTM layer was then passed to a flattened layer to reshape the data. Finally, we added a dense layer with 7 units and used the softmax activation function to obtain the probabilities for each of the 7 labels. This allowed us to predict the label for a given text based on the label with the highest probability.

4.4 Modelling with Transformers

We train our models using the Huggingface Transformers [3] library using the TensorFlow backend for implementation. We fine-tune our four pre-trained language models BERT, RoBERTa, DistilBERT, and XLM-RoBERTa. All the above models follow similar architecture related to BERT. We used Hugging face Huggingface's AutoNLP to tokenize the texts and we generated 768 dimensional embeddings for each token through it. We set the learning rate to $2e{-}5$ and used AdamW optimizer and trained the model.

BERT (Bidirectional Encoder Representations from Transformers)[4] is an innovative technique in natural language processing (NLP) that has been developed by Google. It leverages transformer-based models to generate contextualized word embeddings, setting it apart from traditional unidirectional models. By employing bidirectional training, BERT processes the complete input sentence or paragraph concurrently, enabling it to capture the contextual dependencies and subtleties of each word by considering both its preceding and succeeding words. This bidirectional approach empowers BERT to comprehensively understand the intricate interplay of words and their context, thereby enhancing its ability to represent the nuanced semantics of the language. We used Bert base uncased model for training.

RoBERTa [5] is a modified and optimized version of BERT, trained on a larger dataset for an extended period. It outperforms BERT by 4% - 5%in natural

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

[3]https://huggingface.co/models

[4]https://huggingface.co/bert-base-uncased

[5]https://huggingface.co/roberta-base

Model	P_m	R_m	$F1_m$	P_w	R_w	$F1_w$	Acc
Word2Vec+BiLSTM	0.15	0.17	0.18	0.42	0.44	**0.43**	0.49
TwitterGloVe+BiLSTM	0.12	0.13	0.14	0.39	0.41	0.40	0.43
BERT	0.04	0.16	0.10	0.06	0.15	0.07	0.13
DistilBERT	0.11	0.15	0.10	0.05	0.18	0.08	0.15
RoBERTa	0.03	0.11	0.08	0.05	0.14	0.07	0.14
XLM-RoBERTa	0.06	0.17	0.09	0.06	0.17	0.08	0.14

Table 4: Classification report on the original dataset where P_m : Macro-average Precision, R_m : Macro-average Recall, $F1_m$: Macro-average F1-score, P_w : Weighted-average Precision, R_w : Weighted-average Recall, $F1_w$: Weighted-average F1-score, Acc : Accuracy.

Model	P_m	R_m	$F1_m$	P_w	R_w	$F1_w$	Acc
Word2Vec+BiLSTM	0.14	0.18	0.15	0.50	0.64	**0.56**	0.64
TwitterGloVe+BiLSTM	0.15	0.15	0.15	0.51	0.53	0.52	0.53
BERT	0.09	0.28	0.13	0.05	0.18	0.08	0.18
DistilBERT	0.08	0.27	0.12	0.05	0.16	0.08	0.16
RoBERTa	0.09	0.28	0.13	0.06	0.16	0.09	0.16
XLM-RoBERTa	0.08	0.29	0.13	0.05	0.17	0.08	0.17

Table 5: Classification report on the augmented dataset where P_m : Macro-average Precision, R_m : Macro-average Recall, $F1_m$: Macro-average F1-score, P_w : Weighted-average Precision, R_w : Weighted-average Recall, $F1_w$: Weighted-average F1-score, Acc : Accuracy.

language inference tasks and employs a byte-level BPE tokenizer, which leverages a universal encoding scheme for improved performance. We used roberta base model for training.

XLM-RoBERTa[6] represents a multilingual adaptation of the RoBERTa model that has undergone pre-training on a vast corpus of filtered CommonCrawl data, encompassing 2.5 TB and comprising content from 100 diverse languages. We used xlm roberta base model for training.

DistilBERT[7] is a compact and efficient transformer-based model, reduces size and computational requirements compared to BERT. It retains over 95% of BERT's performance on the GLUE benchmark, making it ideal for resource-constrained environments. With 40% fewer parameters, DistilBERT achieves faster processing, making it well-suited for real-time NLP applications. Distillation transfers knowledge from BERT, enabling DistilBERT to leverage BERT's language understanding capabilities while addressing computational limitations. We used distilbert base uncased model for training.

[6]https://huggingface.co/xlm-roberta-base
[7]distilbert-base-uncased

5 Results and Conclusions

Table 4 gives the classification report on the original dataset whereas Table 5 gives the report on the augmented dataset. Both tables represented the results of various transformer models and BiLSTM model trained on Word2Vec and Twitter GLoVe embeddings. The models results were based on the validation dataset. We can see that there was significant improved performance on the models after augmentation in the BiLSTM models performance. The BERT models and its variants were showing less performance when compared to BiLSTM. This was because we have not fine tuned these models but after fine tuning it we can achieve much better results. In our model evaluation, we favor the weighted F1 score over accuracy due to the prevalence of imbalanced class distributions in classification problems. The weighted F1 score provides a comprehensive assessment by considering precision, recall, and the imbalances in class distribution. This metric allows us to offer a more accurate and reliable evaluation of the models' performance.In our submission on shared task, we reported the predictions of our BiLSTM + Word2Vec model on the test dataset, achieving a weighted F1-score of 0.699. This performance ranked us fifth in the shared task competition. We conclude that our fine tuned BiLSTM model with Word2Vec exhibit

promising performance and is suitable for future dataset predictions.

6 Future Works

In light of the suboptimal performance exhibited by transformers in this context, our forthcoming research will focus on refining their effectiveness through targeted fine-tuning strategies. Specifically, we intend to explore the efficacy of diverse optimizers such as randomized search and Keras optimizers to enhance the model's capabilities. Additionally, we would aim to incorporate sentence augmentation techniques utilizing established libraries like NLPAUG. Furthermore, the integration of SMOTE (Synthetic Minority Over-sampling Technique) would be explored to introduce text data diversity.

References

Nsrin Ashraf, Mohamed Taha, Ahmed Abd Elfattah, and Hamada Nayel. 2022. NAYEL @LT-EDI-ACL2022: Homophobia/transphobia detection for equality, diversity, and inclusion using SVM. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 287–290, Dublin, Ireland. Association for Computational Linguistics.

Bharathi B and Josephine Varsha. 2022. SSNCSE NLP@TamilNLP-ACL2022: Transformer based approach for detection of abusive comment for Tamil language. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 158–164, Dublin, Ireland. Association for Computational Linguistics.

Bharathi Raja Chakravarthi. 2023. Detection of homophobia and transphobia in youtube comments. *International Journal of Data Science and Analytics*, pages 1–20.

Bharathi Raja Chakravarthi, Adeep Hande, Rahul Ponnusamy, Prasanna Kumar Kumaresan, and Ruba Priyadharshini. 2022a. How can we detect homophobia and transphobia? experiments in a multilingual code-mixed setting for social media governance. *International Journal of Information Management Data Insights*, 2(2):100119.

Bharathi Raja Chakravarthi, Manoj Balaji Jagadeeshan, Vasanth Palanikumar, and Ruba Priyadharshini. 2023a. Offensive language identification in dravidian languages using mpnet and cnn. *International Journal of Information Management Data Insights*, 3(1):100151.

Bharathi Raja Chakravarthi, Rahul Ponnusamy, Malliga Subramanian, Paul Buitelaar, Miguel Ángel García-Cumbreras, Salud María Jiménez-Zafra, José Antonio García-Díaz, Rafael Valencia-García, and Nitesh Jindal. 2023b. Overview of second shared task on homophobia and transphobia detection in english, spanish, hindi, tamil, and malayalam. In *Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion*, Varna, Bulgaria. Recent Advances in Natural Language Processing.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Thenmozhi Durairaj, John McCrae, Paul Buitelaar, Prasanna Kumaresan, and Rahul Ponnusamy. 2022b. 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*, pages 369–377, Dublin, Ireland. Association for Computational Linguistics.

Subalalitha Chinnaudayar Navaneethakrishnan, Bharathi Raja Chakravarthi, Kogilavani Shanmugavadivel, Malliga Subramanian, Prasanna Kumar Kumaresan, Bharathi, Lavanya Sambath Kumar, and Rahul Ponnusamy. 2023. Findings of shared task on sentiment analysis and homophobia detection of youtube comments in code-mixed dravidian languages. In *Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '22, page 18–21, New York, NY, USA. Association for Computing Machinery.

Kenneth Ward Church. 2017. Word2vec. *Natural Language Engineering*, 23(1):155–162.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

José García-Díaz, Camilo Caparros-Laiz, and Rafael Valencia-García. 2022. UMUTeam@LT-EDI-ACL2022: Detecting homophobic and transphobic comments in Tamil. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 140–144, Dublin, Ireland. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005: 15th International Conference, Warsaw, Poland, September 11-15, 2005. Proceedings, Part II 15*, pages 799–804. Springer.

Shruthi Hariprasad, Sarika Esackimuthu, Saritha Madhavan, Rajalakshmi Sivanaiah, and Angel S. 2022. SSN_MLRG1@DravidianLangTech-ACL2022: Troll meme classification in Tamil using transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 132–137, Dublin, Ireland. Association for Computational Linguistics.

Md Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Md Shajalal, and Bharathi Raja Chakravarthi. 2022. Multimodal hate speech detection from bengali memes and texts. In *International Conference on Speech and Language Technologies for Low-resource Languages*, pages 293–308. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Edward Ma. 2019. Nlp augmentation. https://github.com/makcedward/nlpaug.

Debora Nozza. 2022. Nozza@LT-EDI-ACL2022: Ensemble modeling for homophobia and transphobia detection. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 258–264, Dublin, Ireland. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Subalalitha Cn, Thenmozhi Durairaj, Malliga Subramanian, Kogilavani Shanmugavadivel, Siddhanth U Hegde, and Prasanna Kumaresan. 2022. Overview of abusive comment detection in Tamil-ACL 2022. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 292–298, Dublin, Ireland. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

S Santhiya, P Jayadharshini, and SV Kogilavani. 2022. Transfer learning based youtube toxic comments identification. In *International Conference on Speech and Language Technologies for Low-resource Languages*, pages 220–230. Springer.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Kogilavani Shanmugavadivel, Malliga Subramanian, Prasanna Kumar Kumaresan, Bharathi Raja Chakravarthi, B Bharathi, Subalalitha Chinnaudayar Navaneethakrishnan, Lavanya Sambath Kumar, Thomas Mandl, Rahul Ponnusamy, Vasanth Palanikumar, et al. 2022. Overview of the shared task on sentiment analysis and homophobia detection of youtube comments in code-mixed dravidian languages.

Malliga Subramanian, Rahul Ponnusamy, Sean Benhur, Kogilavani Shanmugavadivel, Adhithiya Ganesan, Deepti Ravi, Gowtham Krishnan Shanmugasundaram, Ruba Priyadharshini, and Bharathi Raja Chakravarthi. 2022. Offensive language detection in tamil youtube comments by adapters and cross-domain knowledge transfer. *Computer Speech & Language*, 76:101404.

Krithika Swaminathan, Bharathi B, Gayathri G L, and Hrishik Sampath. 2022a. 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*, pages 239–244, Dublin, Ireland. Association for Computational Linguistics.

Krithika Swaminathan, Divyasri K, Gayathri G L, Thenmozhi Durairaj, and Bharathi B. 2022b. PANDAS@abusive comment detection in Tamil code-mixed data using custom embeddings with LaBSE. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 112–119, Dublin, Ireland. Association for Computational Linguistics.

Ishan Sanjeev Upadhyay, Kv Aditya Srivatsa, and Radhika Mamidi. 2022. Sammaan@LT-EDI-ACL2022: Ensembled transformers against homophobia and transphobia. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 270–275, Dublin, Ireland. Association for Computational Linguistics.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26, Montréal, Canada. Association for Computational Linguistics.