

One to Rule Them All: Towards Joint Indic Language Hate Speech Detection.

Mehar Bhatia^{*1}, Tenzin Singhay Bhotia^{*1}, Akshat Agarwal¹, Prakash Ramesh¹, Shubham Gupta¹, Kumar Shridhar¹, Felix Laumann¹ and Ayushman Dash¹

¹NeuralSpace, London, United Kingdom

Abstract

This paper is a contribution to the *Hate Speech and Offensive Content Identification in Indo-European Languages* (HASOC) 2021 shared task. Social media today is a hotbed of toxic and hateful conversations, in various languages. Recent news reports have shown that current models struggle to automatically identify hate posted in minority languages. Therefore, efficiently curbing hate speech is a critical challenge and problem of interest. Our team, ‘*NeuralSpace*’ presents a multilingual architecture using state-of-the-art transformer language models to jointly learn hate and offensive speech detection across three languages namely, English, Hindi, and Marathi. On the provided testing corpora, we achieve Macro F1 scores of 0.7996, 0.7748, 0.8651 for sub-task 1A and 0.6268, 0.5603 during the fine-grained classification of sub-task 1B. These results show the efficacy of exploiting a multilingual training scheme.

Keywords

Hate Speech, Social Media, Indic Languages, Low Resource, Multilingual Language Models

1. Introduction

Since the proliferation of social media users worldwide, platforms like Facebook, Twitter, or Instagram have suffered from a rise of hate speech by individuals and groups. A large-scale study on Twitter, and Whisper, [1] empirically shows the prevalence of abusive comments and toxic languages in these platforms, targeting users mostly based on race, physical features, and gender.


A Bloomberg article[2] reports that users have even found new ways of bullying others online using euphemistic emojis. Widespread use of such abusive language on social media platforms often causes public embarrassment to victims leading to major repercussions. Recently, Twitch filed a lawsuit against two users who targeted LGBTQ+ and Black streamers with hate speech [3]. One week later, content creators boycotted the game-streaming platform due to the inability to control the hateful content. Observing the growing usage of online hate speech often anonymous, overwhelming and unmanageable by human moderators. It is essential for social media platforms to control the abuse of users’ freedom of expression and maintaining an inclusive and respectful society. To enforce such supervision, online platforms must be able to

FIRE 2021: Forum for Information Retrieval Evaluation, December 13-27 December, 2021

✉ mehar@neuralspace.ai (M. Bhatia*); tenzin@neuralspace.ai (T. S. Bhotia*); akshat@neuralspace.ai (A. Agarwal); prakash@neuralspace.ai (P. Ramesh); shubham@neuralspace.ai (S. Gupta); kumar@neuralspace.ai (K. Shridhar); felix@neuralspace.ai (F. Laumann); ayushman@neuralspace.ai (A. Dash)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

develop monitoring systems that can identify hate speech amongst billions of text comments posted by users.



Figure 1: Overview of HASOC 2021 Problem Statement

There have been research contributions in solving the problem of identifying abusive comments or other forms of toxic language [4, 5, 6, 7]. However, most of them have majorly focused on high-resource languages, predominantly English. As social media connects people from all over the world, communicating in different languages, much of the potentially hateful content is present in a multilingual setting. The failure to pay attention to non-English languages has allowed such offensive speech to flourish. The lack of datasets and models for various low-resource languages has made the task of hate speech identification extremely difficult. In this paper, we present our findings on a subset of Indic low-resource languages.

The HASOC (Hate Speech and Offensive Content) 2021 challenge has been organized as a step towards this direction in three languages - English, Hindi, and Marathi. Figure 1 demonstrates HASOC 2021 problem statement. We focus on sub-task 1A and 1B of this competition, which we describe in the following paragraph.

Sub-task 1A focuses on hate speech and offensive language identification in English Hindi, and Marathi. It is a simple binary classification task in which participating systems are required to classify tweets into one of the two classes, namely:

- **(HOF) Hate and Offensive:** Posts of this category contain either hate, offense, profanity, or a combination of them.
- **(NOT) Non-Hate and offensive:** Posts of this category do not contain any hate speech, profane or offensive content.

Sub-task 1B is a multi-class classification task in English and Hindi. In this task, hate speech and offensive posts from sub-task A are further classified into the following three categories.

- **(HATE)** *Hate speech*: Posts of this category contain hate speech content.
- **(OFFN)** *Offensive*: Posts of this category contain offensive content.
- **(PRFN)** *Profane*: Posts of this category contain profane words.

In this paper, we make the following contributions:

- A pre-processing pipeline for modeling hate speech in the text of tweet domain.
- A joint fine-tuning procedure that empirically proves to outperform other approaches in hate speech detection.
- A summary of different approaches that did not work as expected.
- The implementation and idea behind our winning approach for one of HASOC 2021 sub-tasks.

In the forthcoming sections, we give a brief overview of past approaches as related work in section 2. Then, we present a detailed description of the statistics of datasets used in section 3. We present our approach in section 4, delineating upon our pre-processing steps and model architecture. We highlight our model hyperparameters and other experimental details in section 5. Later, in section 6, we display our final results and elaborate on various other approaches that did not work well in section 7. We end with our conclusion and point to future work in section 8.

2. Related Work

In the past, there have been many approaches to tackle the problem of hate speech identification. Kwok and Wang [8] have experimented with a simple bag of words (BOW) approach to identify hate speech. While being light-weight, these models performed poorly with high false positive rates. Including various core natural language processing (NLP) features like part of speech tags [9] and N-gram graphs [10] have helped in improving the performance. Lexical methods using TF-IDF and SVM as a classification model have achieved surprisingly good performance [11].

With the rise of embedding words in distributed representations, researchers have leveraged word embeddings like Glove [12], and FastText [13] for embedding discrete text into a latent space and have improved the performance over standard BOW and lexical approaches.

Recurrent Neural Networks (RNNs) for many years were the de-facto approach for tackling any natural language problem. The winning approach at the 2020 HASOC competition for Hindi [14] used a one-layer BiLSTM with FastText embeddings to identify hate speech. Similarly for English, the most accurate model [15] used an LSTM with Glove embeddings to represent text inputs. Mohtaj et al. [16] also used a character-based LSTM following a similar trend.

In recent times, self-attention-based transformer [17] models and language models derived from its huge corpus trained encoders like BERT [18] have shown more promise than standard RNNs for most of the NLP tasks. Many researchers have found BERT-like models to perform

much better than other approaches majorly due to their high transfer learning prowess [19]. While there has been a lot of research on hate speech in general, experiments especially focusing on low-resource languages are less popular. Simple logistic regression using LASER embeddings has been shown to perform better than BERT-based models [20] indicating the need for more accurate multilingual base language models. Since then, we have witnessed the rise of multilingual language models like XLM-Roberta [21]. In the following sections, we will delineate our approach of building a solution using XLM-Roberta for identifying hate speech along with an exhaustive comparison to other approaches.

3. Dataset Description

Datasets for HASOC 2021 [22] for English [23], Hindi [23], and Marathi [24] languages were collected from social media platforms and comprises of two sub-tasks. We focus only on the first task (named as subtask1 as per HASOC website) on Hindi, English, and Marathi datasets which is further divided into sub-tasks A and B. As shown in Table 1, each dataset instance consists of a unique *hasoc_id*, a *tweet_id*, full *text* of the tweet, and target variables *task_1* and *task_2* for the sub-task 1A and 1B respectively. sub-task 1A is a binary classification problem with two target classes namely, **HOF** (*Hate and Offensive*) and **NOT** (*Non-Hate-offensive*), whereas sub-task 1B is a further fine-grained classification. The data is further classified into four classes, namely **OFFN** (*Offensive*), **PRFN** (*Profane*), **HATE**, and **NONE** class. sub-task 1A requires us to work with datasets in English, Hindi, and Marathi languages, whereas only English and Hindi datasets are available for sub-task 1B. The statistics of both the train and test data are shown in Table 2 and Table 3.

It can be seen that the datasets are highly imbalanced. For sub-task 1A, we notice that the number of hate and offensive tweets is almost double than that of non-hate or offensive tweets for English and Marathi. On the other hand, the number of non-hate-offensive tweets is 55% higher than that of hate and offensive tweets for the Hindi dataset. Similarly sub-task 1B, which deals with English and Hindi language, also have highly imbalanced datasets.

Table 1
Dataset Description

Columns	Description
hasoc_id	unique hasoc ID for each tweet
tweet_id	unique value for each tweet
text	full text of the tweets
task_1	target value for sub-task 1A (HOF or NOT)
task_2	target value for sub-task 1B (OFFN, PRFN, HATE or NONE)

4. Approach

In this section, we demonstrate our approach of solving HASOC 2021 sub-task1A and sub-task1B tasks.

Table 2

Class division of both sub-tasks for Train Dataset

Language	Sub-Task1A		Sub-Task1B				TOTAL
	HOF	NOT	HATE	OFFN	PRFN	NONE	
English	2501	1342	683	622	1196	1342	3843
Hindi	1433	3161	566	654	213	3161	4594
Marathi	1205	669	—	—	—	—	1874

Table 3

Class division of both sub-tasks for Test Dataset

Language	Sub-Task1A		Sub-Task1B				TOTAL
	HOF	NOT	HATE	OFFN	PRFN	NONE	
English	798	483	224	195	379	483	1281
Hindi	505	1027	215	215	44	1027	1532
Marathi	483	418	—	—	—	—	901

4.1. Preprocessing

For preprocessing the tweet data and hashtags, we use two python libraries, tweet-preprocessor¹ and ekphrasis², a segmenter built on Twitter corpus. For English data, the tweet-preprocessor’s clean functionality extracts, clean, parses and tokenizes the tweet texts. For Hindi and Marathi data, first the tweets are tokenized on whitespaces and symbols including colons, commas, semicolons, dashes, and underscores. Secondly, we use the tweet-preprocessor python library for the removal of URLs, hashtags, mentions, emojis, smileys, numbers, and reserved words (such as @RT which stands for Retweets). We also notice the usage of words in English and Arabic in the Hindi and Marathi datasets. We first transliterate this text to the desired language using NeuralSpace’s transliteration tool³. Later, if English or Arabic occurrences remain, we used python library langdetect⁴ (a re-implementation of Google’s language-detection library⁵ from Java to Python) to extract the pure Hindi and Marathi text within the tweet.

4.2. Feature Extraction

To extract features for our classifier, we use tweet-preprocessor to supply various information fields, in addition to the cleaned content. The first feature is obtained from the hashtag text which is segmented into constituent and meaningful tokens using the ekphrasis segmenter. *Ekphrasis* tokenizes the text based on a list of regular expressions. For example, the hashtags ‘##JitegaModiJitegaBharat’, ‘#IPL2019Final’, ‘#hogicongresskijeet’ is tokenized to ‘Jitega Modi Jitega Bharat’, ‘IPL 2019 Final’, ‘hogi congress ki jeet’. Other features are acquired from URLs

¹<https://github.com/s/preprocessor>

²<https://github.com/cbaziotis/ekphrasis>

³<https://docs.neuralspace.ai/transliteration/overview>

⁴<https://pypi.org/project/langdetect/>

⁵<https://github.com/shuyo/language-detection>

within the text, name mentions such as ‘BJP4Punjab’, ‘aajtak’, ‘PMOIndia’, and ‘narendramodi’, and smileys and emojis. The extracted emojis were processed in two ways.

First, we use emot⁶ python library to obtain the textual description of a particular emoji in the text. Emot uses advanced dynamic pattern generation. For example, ‘rofl’ refers to ‘rolling-on-the-floor-and-laughing face’ and ‘speak-no-evil emoji’ refers to ‘speak-no-evil Monkey’. However, we feel that this mapping is not sufficient as it does not highlight the genuine meaning of what the emoji represents in reality. Given that the usage of such emojis is so prevalent and that most of them inherently have emotions built-in, emojis can give a lot of insights into the sentiment of online text. For this reason, we also consider emoji2vec [25] embeddings for 1661 emoji Unicode symbols learned from a total of 6088 descriptions in the Unicode emoji standard. Previous work has demonstrated the usefulness of this by evaluating various tasks such as Twitter sentiment analysis [25]. For example, consider ‘pray emoji’ and ‘tipping-hand-woman emoji’, which map to ‘the-folded-hands’ symbol and the ‘woman-tipping-hand’ emoji. The textual representation will not showcase the emoji’s association with ‘showing gratitude, expressing an apology, sentiments such as hope or respect or even a high five’ which is its real-world implication. On the other hand, the person-tipping symbol is commonly used to express ‘sassiness’ or sarcasm. We expect emoji2vec to capture these kinds of analogy examples.

4.3. Proposed Architecture

We leverage Transformer-based [17] masked language models to generate semantic embeddings for the cleaned tweet text.

We use the available training corpora and fine-tune the transformer layers in a multilingual fashion for our downstream task. We experimented with various multi-lingual transformer models, i.e XLM-RoBERTa (XLMR), mBERT(multilingual BERT), and DistilmBERT (multilingual-distilBERT). A summary for each model is as follows:

- **XLM-RoBERTa:** The pre-training of XLM-RoBERTa is based on 100 languages, using around 2.5TB of preprocessed CommonCrawl dataset to train cross-language representations in a self-supervised manner. XLM-RoBERTa [21] shows that the use of large-scale multi-language pre-training models can significantly improve the performance of cross-language migration tasks.
- **mBERT:** Multilingual BERT [18] uses Wikipedia data of 102 languages, totaling to 177M parameters, and is trained using two objectives i.e, 1) using a masked language modeling (MLM) when 15% of input is randomly masked, and 2) using next sentence prediction.
- **DistilmBERT:** Distil multilingual BERT [26] is a distilled version of the above mBERT model. It is also trained on the concatenation of Wikipedia in 102 different languages. It has a total of 134M parameters. On average DistilmBERT is twice as fast as mBERT-base.

To solve the sub-task 1A of three languages (English, Hindi, and Marathi), and sub-task 1B of two languages (English and Hindi) at the same time, we adopt these multi-lingual models.

As mentioned in Section 4.2, we generate semantic vector representations for all the emojis and smileys, their respective text, and segmented hashtags within the tweet. We encode the

⁶<https://github.com/NeelShah18/emot>

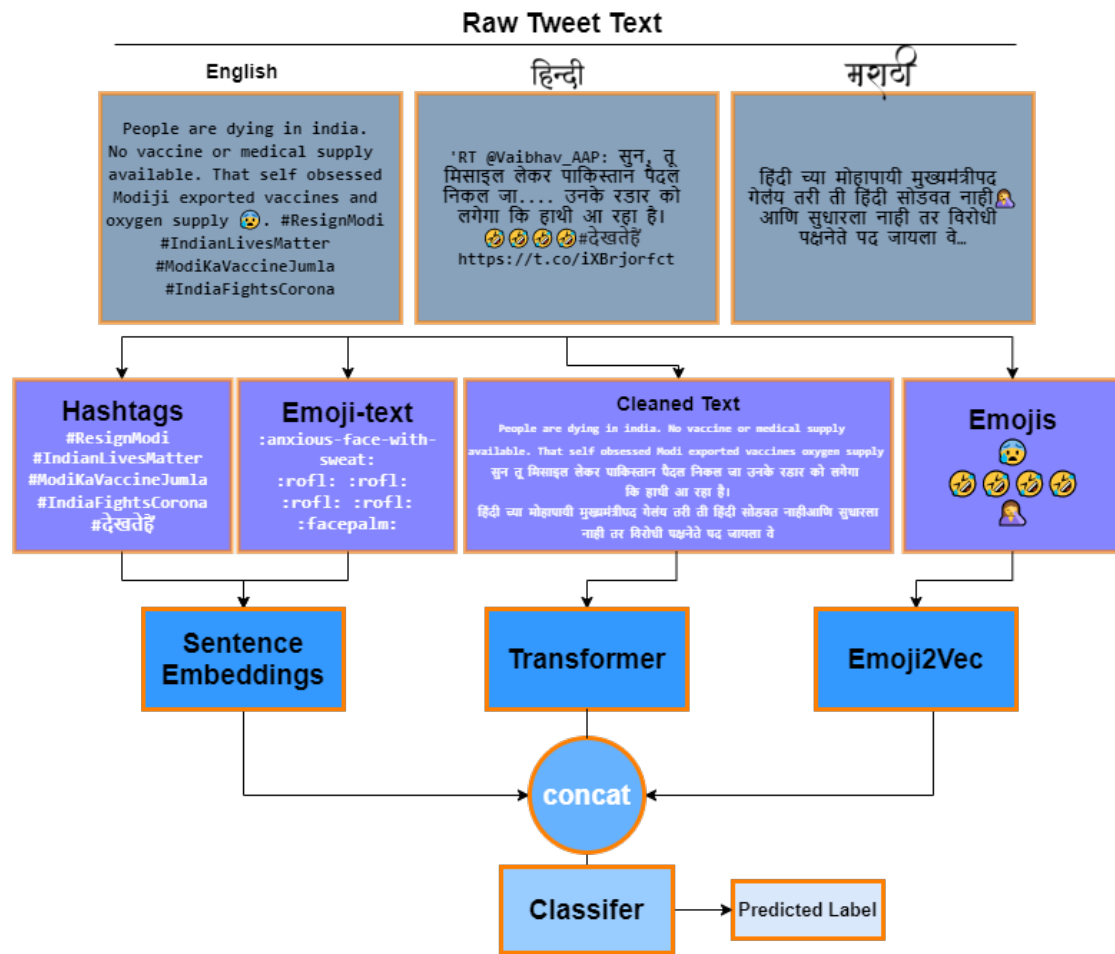


Figure 2: High level overview of our proposed architecture

emoji, smiley text embeddings, and hashtag embeddings in the same latent space. To create the emojis' semantic embeddings, emoji2vec is utilized. An important point to notice is that the segmented hashtags and text descriptions of emojis can be of variable length. Hence, we generate the centralized emoji or hashtag representation by averaging the vector representations. This is a simple approach proposed by [27] to produce a comprehensive vector representation for sentences.

5. Experimental Details

We use Hugging Face’s implementation and corresponding pre-trained models of XLM-RoBERTa⁷, multilingual BERT⁸, and multilingual-distilBERT⁹ in our proposed architecture. Our architectures using Transformer models with custom classification heads are implemented using PyTorch. We use Adam optimizer for training with an initial learning rate of 2e-4, dropout probability of 0.2 with other hyper-parameters set to their default values. We use a cross-entropy loss to update the weights. We also use UKPLab’s sentence-transformers library¹⁰ to encode the hashtags and textual descriptions of the emojis.

All the fine-tuned language models broadly fall into two following categories.

- **Monolingual:** These are a type of models that have been fine-tuned on only the respective target language. For instance, we only use the English train dataset to fine-tune the model and then infer on the English test set only.
- **Multilingual:** These are a type of models that has been fine-tuned on a combination of all available languages irrespective of the target language. For instance, to train a model for the English target language on sub-task 1A, we combine the train datasets for all languages (English, Hindi, and Marathi) and then fine-tune the model once. Such a model may then be used for inference on any given target language. Intuitively, such a training scheme provides three benefits.
 - It enforces joint modeling of the training distribution for all the given languages. Empirically we find this to perform better than individually modeling on respective language.
 - During inference, we only rely on one model to infer instead of a unique model for each language. An approach that can be extremely compute-efficient for production.
 - We combine naturally occurring human-annotated data to form a larger dataset of multiple languages. It becomes a promising approach towards resolving poor model performance due to the data scarcity issue for low-resource languages.

As shown in Table 4 and 5, we empirically observe that a multilingual setting clearly outperforms the monolingual setting across both the tasks in all three languages irrespective of the base model. For English sub-task 1, only mBERT and DistilmBERT score below the monolingual setting, but the difference is not as significant. This experiment suggests that multilingual training can be a preferred approach in obtaining better-performing models, especially as it provides a step towards resolving the data scarcity issue for low-resource languages. It will be interesting to validate the generalizability of this hypothesis on different NLP tasks in the future.

All the experiments were carried out on a workstation with one NVIDIA A100-SXM4-40GB GPU with 12 CPU cores. We use a batch size of 64 throughout. For the initial experiments,

⁷

⁸<https://huggingface.co/bert-base-multilingual-uncased>

⁹<https://huggingface.co/distilbert-base-multilingual-cased>

¹⁰<https://github.com/UKPLab/sentence-transformers>

we divided the released training data into a training set and a validation set and conducted the experiments using accuracy as the performance metric. Finally, we test the performance of the proposed system on the test set released by the organizers. For these experiments, we combine all the training and validation data into a single training set and applied our algorithm. For the multilingual setting, our experiment takes 3.5 hours to train till convergence. For the monolingual setup, our model takes 1.2 hours to train till convergence.

6. Results

Table 4 and 5 presents the performance of the proposed architectures on the provided testing set. The evaluation metric used throughout is the macro F1-score.

Table 4

Results of Proposed Architectures on Test Data (**sub-task 1A**)

Proposed Architecture		English	Hindi	Marathi
Model	Mono/Multi	Sub-task 1A	Sub-task 1A	Sub-task 1A
XLM-R Finetuned	Monolingual	0.7786	0.7585	0.8420
mBERT Finetuned	Monolingual	0.7631	0.7462	0.8269
DistilmBERT Finetuned	Monolingual	0.7637	0.7554	0.8350
XLM-R Finetuned	Multilingual	0.7996	0.7748	0.8651
mBERT Finetuned	Multilingual	0.7626	0.7593	0.8461
DistilmBERT Finetuned	Multilingual	0.7592	0.7589	0.8510

Table 5

Results of Proposed Architectures on Test Data (**sub-task 1B**)

Proposed Architecture		English	Hindi
Model	Mono/Multi	Sub-task 1B	Sub-task 1B
XLM-R Finetuned	Monolingual	0.6148	0.5447
mBERT Finetuned	Monolingual	0.5703	0.5202
DistilmBERT Finetuned	Monolingual	0.5826	0.5267
XLM-R Finetuned	Multilingual	0.6268	0.5603
mBERT Finetuned	Multilingual	0.6121	0.5554
DistilmBERT Finetuned	Multilingual	0.6104	0.5591

It is observed from Table 4 and 5, that for all three languages, XLM-RoBERTa has outperformed similar multilingual Transformer models such as mBERT (multilingual BERT) and distilmBERT (multilingual-distilBERT) on our hatespeech detection task. We observe a minimum absolute gain of 1.63 F1 and 1.20 F1 for sub-task 1A and sub-task 1B respectively via the multilingual approach with XLM-RoBERTa. While a maximum absolute gain of 2.1 F1 and 2.31 F1 have been observed for sub-task 1A and sub-task 1B respectively. Empirically such significant improvements suggest the importance of multilingual training over monolingual training.

Notably, multilingually trained XLM-RoBERTa have secured the 1st position among 24 participants and the 5th position among 34 participants on the HASOC 2021 leaderboard for sub-task 1A and sub-task 1B respectively. Securing such high ranks indicates the importance of the multilingual approach and calls for a detailed investigation of this approach on other tasks as well for future work.

7. Key Takeaways

In this section, we aim to provide a checklist of various approaches and techniques which we implemented, but failed to secure competitive positions on the leaderboard. We believe that our readers will benefit from this checklist during future work.

To begin with, as the dataset was overall highly imbalanced across all languages, we perform SOUP (Similarity-based Oversampling and Undersampling processing), a technique in which the number of the minority class samples is increased and the number of majority class samples are decreased to obtain a balanced data set. This technique has been suggested by [11] and we use this balanced data for performing the classification task. However, when compared to our best performing model, we see a drop of 5% in accuracy.

Secondly, to add more training samples to our multilingual dataset, we use data augmentation techniques such as back-translation to generate this synthetic data. We adopt ML Translator API, which is Google’s Neural Machine Translation (NMT) system. This translation method has been widely used because of its simplicity and zero-shot translation. With this method, we increase our dataset size by three times, however, we don’t see any performance gains using this augmented dataset for our proposed architecture. Moreover, we observe a reduction of toxicity upon using this back-translation method possibly resulting in false labels for many instances.

Based on winning approaches from [28] and [29], we applied different machine learning algorithms, i.e, random forest, and LightGBM, a gradient boosting framework based on decision trees. These techniques have shown an average drop of 5.3%. We also looked into two different deep neural networks approaches and tested them for all three languages. For the English model, we used GloVe ¹¹ embeddings [12] for both sub-tasks. This embedding layer is fed to a CNN model. The architecture comprises two convolutional, two dropouts, and two max-pooling layers accompanied by a flatten layer and a dense layer. We achieved a macro F1 score of 0.75 and 0.56 respectively on HASOC 2021 sub-task 1A and sub-task 1B test sets. For Hindi and Marathi models, we use fastText ¹² embeddings [13] for both the sub-tasks. Here, the embeddings are passed through a bi-directional LSTM model and a dropout layer, followed by a dense layer. We achieve macro F1 scores of 0.74, 0.54, and 0.84 on Hindi sub-tasks 1A and 1B, and Marathi sub-task 1A, respectively. Overall, we conclude that our final proposed architecture performs the best compared to other approaches for all sub-tasks.

¹¹<https://nlp.stanford.edu/projects/glove/>

¹²<https://fasttext.cc/docs/en/crawl-vectors.html>

8. Conclusion

This work has been submitted to CEUR 2021 Workshop Proceedings for the task, Identification of Hate and Offensive Speech in Indo-European Languages (HASOC 2021). In this research, the problem of identifying hate and offensive content in tweets has been experimentally studied on three different language datasets namely, English, Hindi, and Marathi. We propose a joint language training approach based on recent advances in large-scale transformer-based language models and demonstrate our best results. We plan to further explore other novel methods of capturing social media text semantics as part of future work. We also aim to look at more accurate data augmentation techniques to handle the data imbalance and enhancing hate and offensive speech detection in social media posts.

References

- [1] L. Silva, M. Mondal, D. Correa, F. Benevenuto, I. Weber, Analyzing the targets of hate in online social media, in: Tenth international AAAI conference on web and social media, 2016.
- [2] I. Levingston, Racist Emojis Are the Latest Test for Facebook, Twitter Moderators, 2021. URL: <https://www.bloomberg.com/news/articles/2021-09-13/racist-emojis-are-the-latest-test-for-facebook-twitter-moderators>.
- [3] K. Speakman, Twitch Sues Users Who It Alleges Conducted ‘Hate Raids’, 2021. URL: <https://www.forbes.com/sites/kimberleespeakman/2021/09/10/twitch-sues-users-who-it-alleges-conducted-hate-raids/?sh=36407fe87822>.
- [4] Z. Waseem, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on twitter, in: Proceedings of the NAACL student research workshop, 2016, pp. 88–93.
- [5] H. Watanabe, M. Bouazizi, T. Ohtsuki, Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection, IEEE access 6 (2018) 13825–13835.
- [6] A. Al-Hassan, H. Al-Dossari, Detection of hate speech in social networks: a survey on multilingual corpus, in: 6th International Conference on Computer Science and Information Technology, volume 10, 2019.
- [7] Z. Zhang, L. Luo, Hate speech detection: A solved problem? the challenging case of long tail on twitter, Semantic Web 10 (2019) 925–945.
- [8] I. Kwok, Y. Wang, Locate the hate: Detecting tweets against blacks, in: Twenty-seventh AAAI conference on artificial intelligence, 2013.
- [9] 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 Conference on Social Computing, IEEE, 2012, pp. 71–80.
- [10] C. K. Themeli, Hate Speech Detection using different text representations in online user comments, no. October 2018 (2018).
- [11] Y. Reddy, R. Rajalakshmi, DLRG@ HASOC 2020: A Hybrid Approach for Hate and Offen-

- sive Content Identification in Multilingual Tweets., in: FIRE (Working Notes), 2020, pp. 304–310.
- [12] J. Pennington, R. Socher, C. D. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.
 - [13] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146.
 - [14] R. Raja, S. Srivastava, S. Saumya, NSIT & IIITDWD@ HASOC 2020: Deep learning model for hate-speech identification in Indo-European languages (2021).
 - [15] A. K. Mishra, S. Saumya, A. Kumara, IIIT_DWD@ HASOC 2020: Identifying offensive content in Indo-European languages (2020).
 - [16] S. Mohtaj, V. Woloszyn, S. Möller, TUB at HASOC 2020: Character based LSTM for Hate Speech Detection in Indo-European Languages., in: FIRE (Working Notes), 2020, pp. 298–303.
 - [17] 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.
 - [18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
 - [19] M. Mozafari, R. Farahbakhsh, N. Crespi, A BERT-based transfer learning approach for hate speech detection in online social media, in: International Conference on Complex Networks and Their Applications, Springer, 2019, pp. 928–940.
 - [20] S. S. Aluru, B. Mathew, P. Saha, A. Mukherjee, Deep learning models for multilingual hate speech detection, arXiv preprint arXiv:2004.06465 (2020).
 - [21] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, arXiv preprint arXiv:1911.02116 (2019).
 - [22] S. Modha, T. Mandl, G. K. Shahi, H. Madhu, S. Satapara, T. Ranasinghe, M. Zampieri, Overview of the HASOC Subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages and Conversational Hate Speech, in: FIRE 2021: Forum for Information Retrieval Evaluation, Virtual Event, 13th-17th December 2021, ACM, 2021.
 - [23] T. Mandl, S. Modha, G. K. Shahi, H. Madhu, S. Satapara, P. Majumder, J. Schäfer, T. Ranasinghe, M. Zampieri, D. Nandini, A. K. Jaiswal, Overview of the HASOC subtrack at FIRE 2021: Hate Speech and Offensive Content Identification in English and Indo-Aryan Languages, in: Working Notes of FIRE 2021 - Forum for Information Retrieval Evaluation, CEUR, 2021. URL: <http://ceur-ws.org/>.
 - [24] S. Gaikwad, T. Ranasinghe, M. Zampieri, C. M. Homan, Cross-lingual Offensive Language Identification for Low Resource Languages: The Case of Marathi, in: Proceedings of RANLP, 2021.
 - [25] B. Eisner, T. Rocktäschel, I. Augenstein, M. Bošnjak, S. Riedel, emoji2vec: Learning emoji representations from their description, arXiv preprint arXiv:1609.08359 (2016).
 - [26] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller,

faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).

- [27] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings (2016).
- [28] T. Mandl, S. Modha, A. Kumar M, B. R. Chakravarthi, 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, 2020, pp. 29–32.
- [29] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the HASOC track at FIRE 2019: Hate speech and offensive content identification in Indo-European languages, in: Proceedings of the 11th forum for information retrieval evaluation, 2019, pp. 14–17.