UMUTeam@TamilNLP-ACL2022: Abusive Detection in Tamil using Linguistic Features and Transformers

José Antonio García-Díaz and **Manuel Valencia-García** and **Rafael Valencia-García***
Facultad de Informática, Universidad de Murcia, Campus de Espinardo, 30100, Spain
{joseantonio.garcia8,manuelv,valencia}@um.es

Abstract

Social media has become a dangerous place as bullies take advantage of the anonymity the Internet provides to target and intimidate vulnerable individuals and groups. In the past few years, the research community has focused on developing automatic classification tools for detecting hate-speech, its variants, and other types of abusive behaviour. However, these methods are still at an early stage in low-resource languages. With the aim of reducing this barrier, the TamilNLP shared task has proposed a multi-classification challenge for Tamil written in Tamil script and code-mixed to detect abusive comments and hope-speech. Our participation consists of a knowledge integration strategy that combines sentence embeddings from BERT, RoBERTa, FastText and a subset of language-independent linguistic features. We achieved our best result in code-mixed, reaching 3rd position with a macro-average f1-score of 35%.

1 Introduction

Some users make use of social networks to attack others. Bullies target vulnerable individuals groups with the goal of putting them down. This harassment is done on basis of traits such as sexual orientation, religious affiliation, gender, or ethnicity. This speech is known as hate-speech and its automatic detection has recently been explored because the number of daily posts on social networks make it impossible to review all of them manually. The biggest challenges of automatic hate classification are the use of figurative language and that it is not enough just to use offensive language to consider a document as hate speech. Besides, although the performance of hate-speech detectors is not bad (at least in controlled environments), they are language and cultural dependent. This makes it difficult to automatically detect hope and hate speech in low-resource languages like Tamil, where some of the state-of-the-art techniques have yet to be explored.

In these working-notes, the participation of the UMUTeam in the TamilNLP shared task (Priyadharshini et al., 2022) (ACL-2022) is described. In this shared task, the organisers want the participants to detect abusive comments in comments posted in YouTube (Chakravarthi, 2020; Chakravarthi and Muralidaran, 2021; Hande et al., 2021). This is a multi-classification task. The labels are *misandry*, *counter-speech*, *misogyny*, *xenophobia*, *hope-speech*, *homophobia*, *transphobia*, and *none-of-the-above*. The overall performance of each submission is measured using the macro average precision, recall and f1-score.

Two datasets are published. One in Tamil script and another in Tamil using Latin characters (code-mixed). The comments from YouTube are mostly composed by only one sentence. The dataset annotators rate each comment individually (that is, the annotators did not know if the comment is response to another comment or which is the context of the video). The task organisers published the datasets divided into training and development. Table 1 depicts the number of labels per dataset. It can be seen that, on the one hand, there is a strong imbalance between the labels and, on the other, that the code-mixed dataset is much larger.

Label	Tamil-script	Code-mixed
none-of-the-above	1642	4639
misandry	550	1048
counter-speech	185	443
misogyny	149	367
xenophobia	124	266
hope-speech	97	261
homophobia	43	215
transphobic	8	197

Table 1: Dataset statistics per label

* Corresponding author

Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages, pages 45 - 50

2 Related work

Automatic abusive comment detection has gained academic relevance. In fact, it is a trending topic in international workshops on Natural Language Processing. For instance, the MEX-A3T shared-task (IberLEF-2019), Germ-Eval 2018 (Wiegand et al., 2018), or EvalIta 2018 (Bosco et al., 2018) among others.

The common approaches for the development of automatic abusive comment detectors are based on automatic document classification. Therefore, the most common way to do it is by building an automatic classifier based on supervised learning. To do this, some approaches rely on extracting statistical features, such as Bag-of-words, TF–IDF, word or sentence embeddings, and use them to train an automatic classifier based on traditional machine-learning models or neural networks with a convolutional, recurrent or based on transformers architecture.

Modern approaches for detecting abusive comments are based on ensemble learning. For instance, the authors of (Molina-González et al., 2019), which participated in the MEX-A3T, proposed an ensemble learning model based on a soft-voting strategy. To the best of our knowledge, nevertheless, little research has evaluated knowledge integration strategies for abusive comment detection. In (Ahuja et al., 2021), the authors combined four traditional machine-learning models based Bag-of-Words features, and two deep-learning architectures (a convolutional and a recurrent neural network) based on pretrained word embeddings from FastText and GloVe. In (García-Díaz et al., 2022), the authors compared ensemble learning strategies with knowledge integration with four datasets of hate-speech datasets in Spanish. Their evaluation suggest that knowledge integration outperforms ensemble learning slightly.

There is also some work focused on specific types of hate-speech. Our research group, for example, compiled the Spanish MisoCorpus 2020 (García-Díaz et al., 2021a), concerning different types of misogynistic behaviour in Spanish.

3 Methodology

Our methodology is depicted in Figure 1. In a nutshell, it can be described as follows. For both datasets, we extract four feature sets: LF, SE, BF, and RF. The details of each feature set are described in more detail in these working notes. Next, we train a neural network model for each feature set. We use these neural networks to build a new model based on ensemble learning. This new model combines the predictions of each model. Besides, we also evaluate a knowledge integration strategy. With the knowledge integration strategy, a new neural network is trained with all the feature sets at once. For this, we connect each feature set to a input layer and combine their weights in a new hidden layer. Finally, we select the best strategy and obtain the predictions of the official test split.

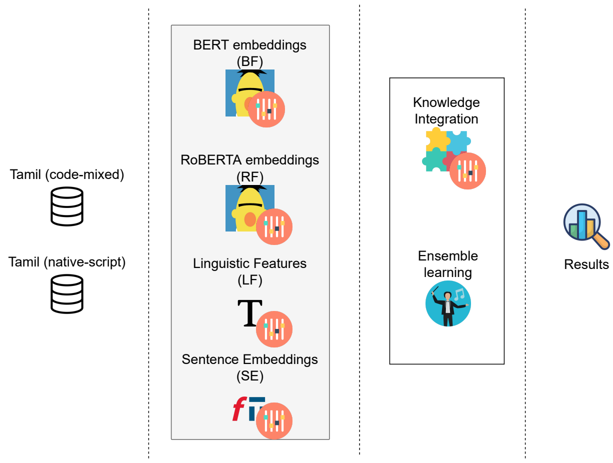

Figure 1: System architecture

Next, the feature sets are explained in detail. The first feature set (LF) is a subset of language-independent linguistic features from the UMU-TextStats tool[1] (García-Díaz et al., 2021b; García-Díaz and Valencia-García, 2022). These features include stylometric features (for instance, word and sentence average and Type-Token Ratio), emojis, and Part-of-Speech features. The second feature set (SE) are non-contextual sentence embeddings from FastText (Mikolov et al., 2018). It is worth noting that FastText has a model for Tamil (Grave et al., 2018). FastText provides a tool to extract sentence embeddings. These embeddings are made up of the average of all the words in each document. The embeddings obtained from FastText are non contextual (they ignore word order). The third and forth feature sets are sentence embeddings from BERT (BF) (Devlin et al., 2018) and RoBERTa (RF) (Liu et al., 2019). In case of Tamil, we use multilingual BERT (Devlin et al., 2018) and XLM RoBERTa (Conneau et al., 2019).

To extract the sentence embeddings from BERT and RoBERTa we conduct a hyperparameter se-

[1]https://umuteam.inf.um.es/umutextstats

lection stage that consisted in the evaluation of 10 models with Tree of Parzen Estimators (TPE) (Bergstra et al., 2013). We evaluate a weight decay between 0 and .3, 2 batch sizes (8 and 16[2]), four warm-up speeds (between 0 and 1000 with steps of 250), from 1 to 5 epochs, and a learning rate between 1e–5 and 5e–5. Once we obtained the best configuration for BERT and for RoBERTa, we extract their sentence embeddings extracting the `[CLS]` token (Reimers and Gurevych, 2019).

The next step in our pipeline is the training of the neural network models. For this, we conduct several hyperparameter optimisation stages with Tensorflow and RayTune (Liaw et al., 2018). This stage is used for each feature set (LF, SE, BF, RF) and for the knowledge integration strategy (LF + SE + BF + RF). Each hyperparameter optimisation stage evaluated 20 shallow neural networks and 5 deep neural networks. The shallow neural networks contains one or two hidden layers max with the same number of neurons per layer. For these, we evaluate `linear`, `ReLU`, `sigmoid`, and `tanh` as activation functions. The deep-learning networks can be from 3 to 8 layers. Besides, each hidden layer can have different number of neurons. These hidden layers and their neurons are arranged in shapes, namely brick, triangle, diamond, rhombus, and funnel. For the deep neural networks we evaluated `sigmoid`, `tanh`, `SELU` and `ELU` as activation functions. In these experiments, we test two learning rates: `10e-03` and `10e-04`. We also evaluate large batch sizes (128, 256, 512) due to class imbalance. Our objective is that every batch has sufficient number of instances of all classes. Besides, we also include a regularisation mechanism based on dropout, testing different ratios between .1 and .3.

Due to page length restrictions, we only report the results achieved with the knowledge integration strategy, as it is the neural network that we use for our official participation. The results achieved with the validation split are depicted in Table 2. We report a macro f1-score of 49.834% for Code-mixed and 46.167% for Tamil-script. Concerning the individual labels, the best results are obtained with the *none-of-the-above* label (the majority class). We observed that documents labelled as *transphobic* label in Tamil-script (66.667%) achieved promising results whereas its counter-part in Code-mixed achieved limited results (24.561%). This behaviour is explained due to the limited number of examples of this label in Code-mixed. In fact, the results are usually better for Tamil except with documents labelled as *xenophobia*, in which our model achieved very good precision in Code-mixed (80.357%) but limited in Tamil (48.936%).

Besides, we include the confusion matrix for Code-mixed (top) and Tamil-script (bottom) in Figure 2. With the confusion matrix, we can observe what are the wrong classifications made by each model. As expected, the *none-of-the-above-label* (that is, the neutral label) is the label that has the larger number of wrong classifications. In case of Tamil-script, we can observe that documents labelled as *hope-speech* are commonly misclassified.

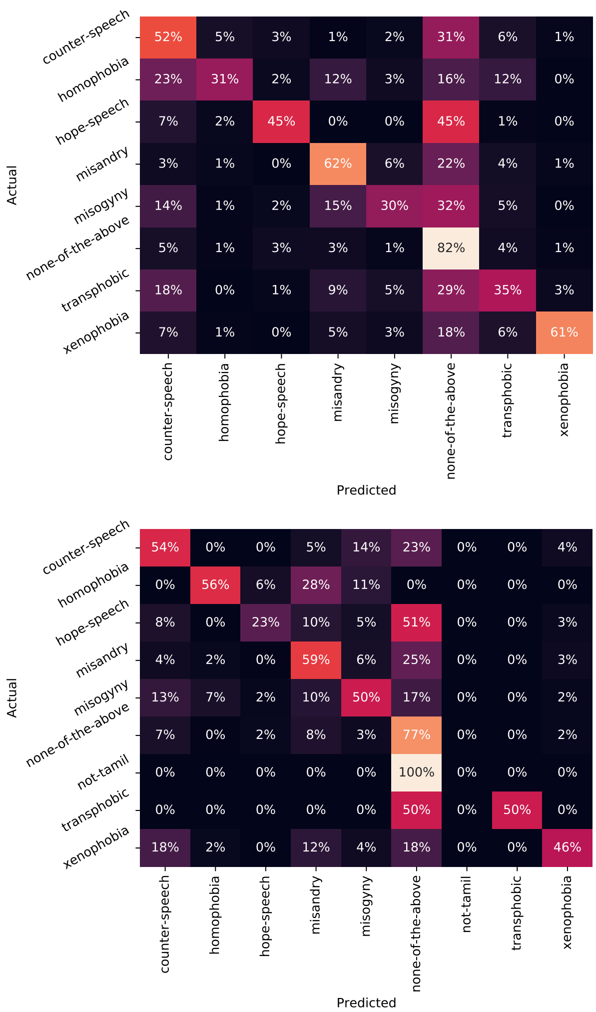

Figure 2: Confusion matrix for report for Code-mixed (top) and Tamil-script (bottom) with the validation split in the neural network that combines all feature sets

[2]In case of Tamil, our GPU does not support batch size of 16, so we only evaluate 8

	precision	recall	f1-score	precision	recall	f1-score
	Code-mixed			Tamil-script		
none-of-the-above	83.93	82.44	83.17	81.64	77.17	79.34
misandry	71.98	62.38	66.84	62.20	59.09	60.61
counter-speech	34.85	51.69	41.63	35.09	54.05	42.55
xenophobia	80.36	61.22	69.50	48.94	46.00	47.42
hope-speech	41.74	44.86	43.24	33.33	23.08	27.27
misogyny	34.78	30.48	32.49	37.50	50.00	42.86
homophobia	45.76	31.40	37.24	43.48	55.56	48.78
transphobic	18.79	35.44	24.56	100.00	50.00	66.67
macro avg	51.52	49.99	49.83	49.13	46.11	46.17
weighted avg	73.05	70.82	71.61	68.65	66.87	67.46

Table 2: Precision, recall, and f1-score for Code-mixed (left) and Tamil-script (right). These results are obtained with the knowledge integration strategy that combined LF, SE, BF, and BF

4 Results and discussion

One of the biggest challenges in this shared task is that the CodaLab leader board is disabled. Therefore, we could not review that the output file is correct.

Table 3 depicts the official leader board for Code-mixed and Table 4 for Tamil-script. Note that these results were provided by the organisers and we can not report more precision. It can be seen that we achieved the 3rd position in the official leader board for code-mixed, with the same f1-score that the second participant (with fewer accuracy and precision but a higher recall). We achieved very limited results in Tamil-script, reaching 9th position in the official ranking. As it can be observed, we obtained very limited precision and recall. In view of these results, it is possible that our neural network model has not learn to classify correctly the labels and it is always predicting the same result.

Team	**Acc**	**m-P**	**m-R**	**m-F1**
abusive-checker	65	46	38	41
GJG_TamilEnglish	60	37	34	35
UMUTeam	59	35	37	35
Optimize_Prime	45	31	38	32
MUCIC	54	40	28	29
CEN-Tamil	56	30	23	25
DLRG	60	18	15	14
BpHigh	15	14	16	10

Table 3: Official results for the code-mixed, showing the accuracy and the macro precision, recall, and F1-score

Team	**Acc**	**m-P**	**m-R**	**m-F1**
CEN-Tamil	63	38	29	32
COMBATANT	53	29	33	30
DE-ABUSE	61	33	29	29
DLRG	60	34	26	27
TROOPER	61	40	23	25
abusive-checker	45	14	14	14
Optimize_Prime	44	13	13	13
GJG_Tamil	43	13	14	13
UMUTeam	39	13	13	13
MUCIC	46	12	13	12
BpHigh_tamil	7	18	12	6

Table 4: Official results for Tamil-script, showing the accuracy and the macro precision, recall, and F1-score

5 Conclusions and promising research lines

This working notes describe the participation of the UMUTeam in the TamilNLP-ACL2022 shared task, concerning abusive detection in Tamil written in Tamil-script and code-mixed. In this work, we have combined four feature sets from linguistic features to three types of sentences embeddings. We have combined these features in a knowledge integration strategy. We reached the 3rd position in Code-mixed and 9th position in Tamil-script.

As future work, we will focus on the development of language-independent linguistic features. For example, we have adapted UMUTextStats to use different PoS models from Stanza (Qi et al., 2020), which has allowed to extend the subset of the linguistic features for Tamil. Besides, we will compile idioms and extending the dictionaries to improve the figurative language identification (del

Pilar Salas-Zárate et al., 2020), thus improving the performance of automatic document classification.

Acknowledgements

This work is part of the research project LaTe4PSP (PID2019-107652RB-I00) funded by MCIN/ AEI/10.13039/501100011033. This work is also part of the research project PDC2021-121112-I00 funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. In addition, José Antonio García-Díaz is supported by Banco Santander and the University of Murcia through the Doctorado Industrial programme.

References

Ravinder Ahuja, Alisha Banga, and SC Sharma. 2021. Detecting abusive comments using ensemble deep learning algorithms. In *Malware Analysis Using Artificial Intelligence and Deep Learning*, pages 515–534. Springer.

James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, and Tesconi Maurizio. 2018. Overview of the evalita 2018 hate speech detection task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 2263, pages 1–9. CEUR.

Bharathi Raja Chakravarthi. 2020. 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 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*, pages 61–72, Kyiv. Association for Computational Linguistics.

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. *CoRR*, abs/1911.02116.

María del Pilar Salas-Zárate, Giner Alor-Hernández, José Luis Sánchez-Cervantes, Mario Andrés Paredes-Valverde, Jorge Luis García-Alcaraz, and Rafael Valencia-García. 2020. Review of english literature on figurative language applied to social networks. *Knowledge and Information Systems*, 62(6):2105–2137.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

José Antonio García-Díaz, Mar Cánovas-García, Ricardo Colomo-Palacios, and Rafael Valencia-García. 2021a. Detecting misogyny in spanish tweets. an approach based on linguistics features and word embeddings. *Future Generation Computer Systems*, 114:506–518.

José Antonio García-Díaz, Ricardo Colomo-Palacios, and Rafael Valencia-García. 2021b. Psychographic traits identification based on political ideology: An author analysis study on spanish politicians' tweets posted in 2020. *Future Generation Computer Systems*.

José Antonio García-Díaz, Salud María Jiménez-Zafra, Miguel Angel García-Cumbreras, and Rafael Valencia-García. 2022. Evaluating feature combination strategies for hate-speech detection in spanish using linguistic features and transformers. *Complex & Intelligent Systems*, pages 1–22.

José Antonio García-Díaz and Rafael Valencia-García. 2022. Compilation and evaluation of the spanish saticorpus 2021 for satire identification using linguistic features and transformers. *Complex & Intelligent Systems*, pages 1–14.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Adeep Hande, Ruba Priyadharshini, Anbukkarasi Sampath, Kingston Pal Thamburaj, Prabakaran Chandran, and Bharathi Raja Chakravarthi. 2021. Hope speech detection in under-resourced kannada language.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. 2018. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

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. *CoRR*, abs/1907.11692.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

María Dolores Molina-González, Flor Miriam Plaza del Arco, María Teresa Martín-Valdivia, and Luis Alfonso Ureña López. 2019. Ensemble learning to detect aggressiveness in mexican spanish tweets. In *IberLEF@ SEPLN*, pages 495–501.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Subalalitha Chinnaudayar Navaneethakrishnan, Thenmozhi Durairaj, Malliga Subramanian, Kogilavani Shanmugavadivel, Siddhanth U Hegde, and Prasanna Kumar Kumaresan. 2022. Findings of the shared task on Abusive Comment Detection in Tamil. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language.