Nepali Transformers@NLU of Devanagari Script Languages 2025: Detection of Language, Hate Speech and Targets

Pilot Khadka[1], **Ankit B.K.**[1], **Ashish Acharya**[2], **Bikram K.C.**[3], **Sandesh Shrestha**[3], **Rabin Thapa**[3]
[1]Thapathali Campus, Tribhuvan University, Kathmandu, Nepal
[2]Kathmandu University, Dhulikhel, Nepal
[3]Institute of International Management Science, Kathmandu, Nepal
{khdpilot, ankitbk75, ashishacharya048, kcvikram44, sandeshrestha115} @gmail.com
rabin@iimscollege.edu.np

Abstract

The Devanagari script, an Indic script used by a diverse range of South Asian languages, presents a significant challenge in Natural Language Processing (NLP) research. The dialect and language variation, complex script features, and limited language-specific tools make development difficult. This shared task aims to address this challenge by bringing together researchers and practitioners to solve three key problems: Language identification, Hate speech detection, and Targets of Hate speech identification. The selected languages-Hindi, Nepali, Marathi, Sanskrit, and Bhojpuri-are widely used in South Asia and represent distinct linguistic structures. In this work, we explore the effectiveness of both machine-learning models and transformer-based models on all three sub-tasks. Our results demonstrate strong performance of the multilingual transformer model, particularly one pre-trained on domain-specific social media data, across all three tasks. The multilingual RoBERTa model, trained on the Twitter dataset, achieved a remarkable accuracy and F1-score of 99.5% on language identification (Task A), 88.3% and 72.5% on Hate Speech detection (Task B), and 68.6% and 61.8% on Hate Speech Target Classification (Task C).

1 Introduction

With the advent of the internet and its application in recent years, user-generated content has increased exponentially, with much of it in different regional languages. Devanagari, one of South Asia's most extensively used scripts, is adopted by languages like Hindi, Marathi, Nepali, Bhojpuri, and Sanskrit (Ajmire et al., 2015). The rising presence of Devanagari content online has called for hate speech detection and content moderation.

The challenges in detecting hate speech are due to phonetically similar text across scripts and the complex evolution of Indo-Aryan languages which makes it difficult(Sharma et al., 2018; Kumar et al., 2018). As languages like Hindi, Nepali, Marathi, Sanskrit, Bhojpuri, etc. uses the Devanagari script, better Language identification is important for any downstream application such as machine translation and hate speech detection.

This issue highlights the need for accurate Devanagari language recognition to combat hate speech and support online diversity. While significant studies have been done towards the automatic detection of hate speech in resource-rich languages like English (Gitari et al., 2015; Burnap and Williams, 2016; Davidson et al., 2017; Gambäck and Sikdar, 2017) and Germany (Schneider et al., 2018; Wiedemann et al., 2018; Corazza et al., 2018), there is limited research on hate detection in Devanagari scripts. So, there is an increasing necessity for more robust cross-linguistic models that can better generalize hate speech even when the language changes to provide a safer online environment for these communities.

For hate speech analysis, identifying the specific target is essential for addressing and mitigating harm (Parihar et al., 2021). This shared task (Thapa et al., 2025; Sarveswaran et al., 2025) aims to identify the different Davanagari languages, detect hate speech, and classify it by target type (individual, organization, or community).

Our work makes the following key contributions:

- We evaluate a range of transformer-based models, including general-purpose baselines, language-specific, and domain-adapted approaches.

- We demonstrate the importance of using multilingual and domain-specific pertaining by showing the superior performance of the Twitter-trained multilingual RoBERTa model across all subtasks.

Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL 2025), pages 314–319
January 19, 2025. ©2025 International Committee on Computational Linguistics

2 Literature Review

Significant research has been conducted in the field of script identification.(Indhuja et al., 2014) used the character and word n-grams model to identify languages: Hindi, Sanskrit, Marathi, Nepali, and Bhojpuri. (Kumar et al., 2018) utilized a Linear SVM classifier for identifying five closely related Indo-Aryan languages of India. It used 5-fold cross-validation, with the C hyper-parameter tuned via Grid Search to optimize the model. Character 5-grams achieved the best result with an impressive 96% accuracy over 13,744 sentences.

Hate speech identification plays a pivotal role in providing an inclusive environment by identifying and moderating the use of harmful language A notable study on Sanskrit and Bhojpuri utilized a dataset of 7,248 records and employed a Random Forest classifier, yielding F1 scores of 0.87 for Non-Offensive, 0.71 for Other Offensive, 0.45 for Racist, and a low 0.01 for Sexist content (Niraula et al., 2021). In Hindi and Marathi, the RoBERTa Hindi base model outperformed other models on the HASOC 2021 dataset, achieving the best results in identifying offensive content (Velankar et al., 2021).

Target classification in hate speech has been the new emerging interest for many researchers. (Surendrabikram Thapa, 2023) utilized a large-scale dataset of 13,505 Nepali tweets related to Nepal's local elections for hate speech and its target identification. In their experiment, they explored classical machine learning (ML) algorithms and transformer-based models like NepBERTa (Timilsina et al., 2022) and RoBERTa (Liu et al., 2019a) in which NepBERTa secured the highest F1-score of 0.68. Similarly, (Sharma et al., 2024) used 11,549 Hindi comments to classify the target in hate speech where the classes were Islam, Hinduism, Christianity, and None. They benchmarked with deep learning (DL) models, including CNN (Dai, 2021), BERT (Devlin, 2018), and MulRIL (Khanuja et al., 2021). Among all models, MulRIL performed the best with an F1 score of 0.72.

3 Dataset and Task

The shared task includes three different subtasks: Sub-Task A, intent on Devanagari Script Language Identification, Sub-Task B concentrates on hate speech detection, and Sub-Task C focuses on target identification of hate speech. For the shared task, the dataset was collected from different sources: Hindi (Jafri et al., 2024, 2023), Nepali (Surendrabikram Thapa, 2023; Rauniyar et al., 2023), Bhojpuri (Ojha, 2019), Marathi (Kulkarni et al., 2021), and Sanskrit(Aralikatte et al., 2021).

3.1 Sub-Task A

This Subtask involves multi-class classification for identifying the particular languages (Nepali, Marathi, Sanskrit, Bhojpuri, and Hindi). The dataset includes 52,422 training samples, 11,233 evaluation samples, and 11,234 test samples.

3.2 Sub-Task B

Sub-task B includes binary classification with two annotated labels: "hate" or "non-hate". The associated dataset comprises 19,019 training samples, 4,076 evaluation samples, and 4,076 test samples for this task.

3.3 Sub-Task C

The last Sub-task focuses on identifying the targets of hate speech among "individual", "organization", and "community". For this task, 2,214 training samples, 474 evaluation samples, and 475 test samples of datasets were provided.

4 Methodology

4.1 Dataset preparation

Our pre-processing pipeline consisted of three key steps. First, we replaced the Twitter username with a generic "@" token to maintain structural information. All hyperlinks were removed to focus on textual content. We also removed emojis using unicode ranges including emoticons, symbols, and special characters. Before these steps, entries with missing values were removed.

4.2 Feature engineering and Embeddings

For text representation, we experimented with multiple embedding approaches. We used the TF-IDF vectorization as our baseline representation for ML models. However, given the shared tasks's focus on Devanagari languages, we recognized the need for embedding that better captures the semantic relationship in these languages. Word2Vec and GloVe embeddings that were specifically trained on the Nepali corpus were included (Koirala and Niraula, 2021).

Sub-Task	Classes	Train	Eval	Test	Train (Augmented)
Detection of Devanagari Script	Nepali	12,544	2,688	2,688	-
	Marathi	11,034	2,364	2,365	
	Sanskrit	10,996	2,356	2,356	
	Bhojpuri	10,184	2,182	2,183	
	Hindi	7,664	1,643	1,642	
Hate Speech Identification	Non-hate	16,805	3,602	3,601	16,805
	Hate	2,214	474	475	10,000
Hate Speech Targets Identification	Individual	1,074	230	230	2,185
	Organization	856	183	184	2,228
	Community	284	61	61	1,010

Table 1: Original and Augmented Dataset distribution

4.3 Dataset Oversampling and Synthesis

No augmentation was performed on Sub-Task A as the original distribution had slightly underrepresented Hindi sampled. However, in Sub-Task B we address the significant disparity between hate and non-hate speech instances (2,214 vs 16,805 samples) by applying random oversampling to increase the minority class to 10,000 instances.

For Sub-Task C, due to the limited sample size, we used the multilingual Aya Expanse 8-B model(Cohere For AI, 2024) to generate target classifications using the hate speech instances from Sub-Task B. The augmented dataset distribution is shown in Table 1.

4.4 Hyperparameter Search

We use Bayesian optimization to find the optimal hyperparameters for the transformer models. The search space was defined based on the model architecture requirements and computational constraints. The number of epochs was task-specific, considering the dataset characteristics, computational efficiency, and early results. Each model then underwent 20 Bayesian optimization runs.

The search space is presented in the Table 4:

4.5 Machine Learning Models

We experimented with a diverse set of traditional machine learning models for the three sub-tasks. Logistic Regression was used as our baseline linear model, Decision Tree as a baseline for tree-based methods, and Support Vector Machines (SVM) for handling high dimensional feature space. Our ensemble method included Random Forest, XGBoost, and AdaBoost classifiers. For comparison, each model was trained on the same feature sets (TF-IDF, Word2Vec, and GloVe embeddings). The hyperparameters used for each model are presented in Table 5.

4.6 Deep Learning Models

Our selection of models was motivated by the need to establish a strong baseline with general-purpose models like BERT (Devlin et al., 2018), DistilBERT(Sanh et al., 2019), and RoBERTa(Liu et al., 2019b). The Devanagari-specific models (Nepali DistilBERT (Shrestha, 2021), and Nepali RoBERTa(Chaudhary, 2021))were chosen for their potential to better capture the linguistic nuances in Devanagari text. And, the Twitter-dataset trained XLM-RoBERTa (Barbieri et al., 2020) was included to evaluate the impact of domain adaptation on hate speech and Target identification of hate speech (Sub-Task B and C).

5 Result and Discussion

This section presents the results of the three subtasks along with an in-depth analysis and interpretations of the findings.

5.1 Machine Learning Models

We trained our models using various embeddings, including TF-IDF, GloVe, and Word2Vec. In subtask A, SVM with Word2Vec achieved the highest accuracy and f1 score (97.9% and 97.7%). Logistic Regression with TF-IDF achieved the highest accuracy of 88.6% and XGBoost with Word2Vec has the highest f1 score of 53.7% on Task B. Random forest performed better on sub-task C, achieving 62.9% accuracy and 50.4% F1-score. On the augmented dataset on sub-task B, the f1 score obtained by XGBoost with Word2Vec was 63.9%, which was a 10% increase, and accuracy reached 88.8% by Random Forest with Word2vec, .2% increase

Embedding	**Model**	**Original Dataset**						**Augmented Dataset**			
		Task A		**Task B**		**Task C**		**Task B**		**Task C**	
		acc	f1	acc	f1	acc	f1	acc	f1	acc	f1
TF-IDF	LR	0.957	0.954	**0.886**	0.537	0.606	0.437	0.859	0.637	0.612	**0.517**
	RF	0.942	0.938	0.883	0.498	**0.629**	0.456	0.882	0.505	0.610	0.445
	DT	0.618	0.647	0.877	0.567	0.532	0.383	0.856	0.560	0.505	0.421
	SVM	0.959	0.956	0.808	0.571	0.610	0.427	0.799	0.578	**0.614**	0.430
	XGBoost	0.935	0.931	0.881	0.558	0.576	0.453	0.873	0.619	0.587	0.460
	AdaBoost	0.798	0.787	0.881	0.549	0.553	0.445	0.846	0.604	0.562	0.470
GloVe	LR	0.960	0.958	0.879	0.520	0.578	**0.504**	0.788	0.620	0.572	0.501
	RF	0.938	0.935	0.883	0.483	0.593	0.414	0.886	0.549	0.591	0.441
	DT	0.830	0.823	0.811	0.548	0.486	0.409	0.754	0.577	0.448	0.374
	SVM	0.971	0.969	0.696	0.568	0.623	0.449	0.710	0.570	0.578	0.445
	XGBoost	0.963	0.962	0.881	0.567	0.574	0.431	0.871	0.609	0.587	0.476
	AdaBoost	0.788	0.749	0.878	0.516	0.534	0.446	0.774	0.618	0.530	0.450
Word2Vec	LR	0.969	0.967	0.877	0.534	0.576	0.497	0.796	0.636	0.578	0.513
	RF	0.953	0.950	0.883	0.481	0.597	0.420	**0.888**	0.548	0.587	0.447
	DT	0.827	0.820	0.816	0.549	0.440	0.377	0.777	0.572	0.480	0.421
	SVM	**0.979**	**0.977**	0.713	0.568	0.610	0.441	0.661	0.544	0.602	0.472
	XGBoost	0.973	0.971	0.885	**0.593**	0.587	0.442	0.882	**0.639**	0.602	0.483
	AdaBoost	0.808	0.782	0.881	0.569	0.501	0.369	0.772	0.612	0.543	0.432

Table 2: Performance of Machine Learning models

Model	**Original Dataset**						**Augmented Dataset**			
	Task A		**Task B**		**Task C**		**Task B**		**Task C**	
	acc	f1	acc	f1	acc	f1	acc	f1	acc	f1
BERT-base	0.991	0.990	0.763	0.613	0.597	0.425	0.781	0.623	0.602	0.520
RoBERTa	0.991	0.990	0.872	0.593	0.595	0.516	0.806	0.626	0.595	0.499
Distil-BERT	0.990	0.989	0.867	0.620	0.574	0.483	0.763	0.612	0.576	0.506
Nepali RoBERTa	0.994	0.993	0.830	0.675	0.656	0.544	0.874	0.672	0.629	0.548
Nepali DistilBERT	0.994	0.994	0.851	0.700	0.658	0.561	0.840	0.677	0.642	0.548
Twitter XLM-RoBERTa	**0.995**	**0.995**	**0.883**	**0.725**	**0.686**	**0.618**	0.872	0.720	0.612	0.545

Table 3: Performance of Deep Learning Models

compared to the original dataset. In sub-task C, Logistics regression with TF-IDF achieved an F1-score of 51.7%, which was 1% higher than the original dataset. The accuracy achieved was similarly higher, at 61.2%.

5.2 Deep Learning Models

Transformer-based models showed a superior performance across three tasks. Furthermore, the performance of the language-specific and domain-adapted model was higher over the general-purpose baseline. The multilingual RoBERTa model, which was specifically trained on the Twitter dataset, consistently outperformed other architectures across all three tasks.

For task A, the Twitter dataset trained multilingual RoBERTa achieved superior performance with both accuracy and F1-score reaching 99.5%. Task B and Task C were both best handled by the Twitter-trained multilingual RoBERTa, achieving scores of 88.3% accuracy, 72.5% F1-score, and 68.6% accuracy, 61.8% F1-score respectively.

6 Conclusion

In this research, we used a variety of machine learning and deep learning models for collaborative activities. Deep learning models outperformed machine learning models on all tasks. Twitter XLM-RoBERTa achieved greater F1 scores across all challenges. The highest f1-scores for Sub-Tasks A, B, and C are 99.5%, 72.5%, and 61.8%, respectively. We also investigated data augmentation for sub-tasks B and C because the dataset contained fewer instances, which allowed us to improve performance.

7 Limitations

Our study demonstrated strong results across all tasks, particularly with Twitter-trained multilingual RoBERTa. However, some limitations exist.

Our search space could be considered constrained due to limited optimization runs. Which, while computationally practical, may not have been sufficient to properly explore the search space.

Our work tested Nepali-based transformer models, future work could expand by exploring other Devanagari language models, like those trained in Hindi language.

References

PE Ajmire, RV Dharaskar, and VM Thakare. 2015. Handwritten devanagari (marathi) compound character recognition using seventh central moment. *International Journal of Innovative Research in Computer and Communication Engineering*, 3(6):5312–5319.

Rahul Aralikatte, Miryam De Lhoneux, Anoop Kunchukuttan, and Anders Søgaard. 2021. Itihasa: A large-scale corpus for sanskrit to english translation. In *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*, pages 191–197.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Pete Burnap and Matthew L Williams. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data science*, 5:1–15.

Amit Chaudhary. 2021. https://huggingface.co/amitness/roberta-base-ne.

Cohere For AI. 2024. Aya-expanse: An open-source language model for research. `https://huggingface.co/CohereForAI/aya-expanse-8b`.

Michele Corazza, Stefano Menini, Pinar Arslan, Rachele Sprugnoli, Elena Cabrio, Sara Tonelli, and Serena Villata. 2018. Inriafbk at germeval 2018: Identifying offensive tweets using recurrent neural networks. In *Proceedings of the GermEval 2018 Workshop*, pages 80–84.

Dengyuhan Dai. 2021. An introduction of cnn: Models and training on neural network models. In *2021 International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)*, pages 135–138.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230.

K Indhuja, M Indu, C Sreejith, Palakkad Sreekrishnapuram, and PR Raj. 2014. Text based language identification system for indian languages following devanagiri script. *International Journal of Engineering*, 3(4).

Farhan Ahmad Jafri, Kritesh Rauniyar, Surendrabikram Thapa, Mohammad Aman Siddiqui, Matloob Khushi, and Usman Naseem. 2024. Chunav: Analyzing hindi hate speech and targeted groups in indian election discourse. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Farhan Ahmad Jafri, Mohammad Aman Siddiqui, Surendrabikram Thapa, Kritesh Rauniyar, Usman Naseem, and Imran Razzak. 2023. Uncovering political hate speech during indian election campaign: A new low-resource dataset and baselines.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Pravesh Koirala and Nobal B. Niraula. 2021. NPVec1: Word embeddings for Nepali - construction and evaluation. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 174–184, Online. Association for Computational Linguistics.

Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi. 2021. L3cubemahasent: A marathi tweet-based sentiment analysis dataset. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 213–220.

Ritesh Kumar, Bornini Lahiri, Deepak Alok, Atul Kr Ojha, Mayank Jain, Abdul Basit, and Yogesh Dawer. 2018. Automatic identification of closely-related indian languages: Resources and experiments. *arXiv preprint arXiv:1803.09405*.

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

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

Nobal B Niraula, Saurab Dulal, and Diwa Koirala. 2021. Offensive language detection in nepali social media. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 67–75.

Atul Kr Ojha. 2019. English-bhojpuri smt system: Insights from the karaka model. *arXiv preprint arXiv:1905.02239*.

Anil Singh Parihar, Surendrabikram Thapa, and Sushruti Mishra. 2021. Hate speech detection using natural language processing: Applications and challenges. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 1302–1308. IEEE.

Kritesh Rauniyar, Sweta Poudel, Shuvam Shiwakoti, Surendrabikram Thapa, Junaid Rashid, Jungeun Kim, Muhammad Imran, and Usman Naseem. 2023. Multi-aspect annotation and analysis of nepali tweets on anti-establishment election discourse. *IEEE Access*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Kengatharaiyer Sarveswaran, Bal Krishna Bal, Surendrabikram Thapa, Ashwini Vaidya, and Sana Shams. 2025. A brief overview of the first workshop on challenges in processing south asian languages (chipsal). In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Julian Moreno Schneider, Roland Roller, Peter Bourgonje, Stefanie Hegele, and Georg Rehm. 2018. Towards the automatic classification of offensive language and related phenomena in german tweets. In *14th Conference on Natural Language Processing KONVENS*, volume 2018, page 95.

Deepak Kumar Sharma, Anurag Singh, and Abhishek Saroha. 2018. Language identification for hindi language transliterated text in roman script using generative adversarial networks. *Towards Extensible and Adaptable Methods in Computing*, pages 267–279.

Deepawali Sharma, Aakash Singh, and Vivek Singh. 2024. Thar- targeted hate speech against religion: A high-quality hindi-english code-mixed dataset with the application of deep learning models for automatic detection. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Dipesh Shrestha. 2021. https://huggingface.co/dexhrestha/nepali-distilbert.

Shuvam Shiwakoti Sweta Poudel Usman Naseem Mehwish Nasim Surendrabikram Thapa, Kritesh Rauniyar. 2023. Nehate: Large-scale annotated data shedding light on hate speech in nepali local election discourse. *26th European Conference on Artificial Intelligence (ECAI), Kraków, Poland (IOS Press) [ECAI'23]*.

Surendrabikram Thapa, Kritesh Rauniyar, Farhan Ahmad Jafri, Surabhi Adhikari, Kengatharaiyer Sarveswaran, Bal Krishna Bal, Hariram Veeramani, and Usman Naseem. 2025. Natural language understanding of devanagari script languages: Language identification, hate speech and its target detection. In *Proceedings of the First Workshop on Challenges in Processing South Asian Languages (CHiPSAL)*.

Sulav Timilsina, Milan Gautam, and Binod Bhattarai. 2022. Nepberta: Nepali language model trained in a large corpus. In *Proceedings of the 2nd conference of the Asia-pacific chapter of the association for computational linguistics and the 12th international joint conference on natural language processing*. Association for Computational Linguistics (ACL).

Abhishek Velankar, Hrushikesh Patil, Amol Gore, Shubham Salunke, and Raviraj Joshi. 2021. Hate and offensive speech detection in hindi and marathi. *arXiv preprint arXiv:2110.12200*.

Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann. 2018. Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. *arXiv preprint arXiv:1811.02906*.

A Appendix

A.1 Hyperparameter Search space

Parameter	Search Space	Distribution
Batch size	[16,32]	Discrete
Learning Rate	[1e-6, 5e-5]	Log-uniform
Weight Decay	[1e-6, 0.1]	Log-uniform
Beta 1	[0.9, 0.99]	Uniform
Beta 2	[0.999, 0.9999]	Uniform
Epochs (Task A)	[2-3]	Discrete
Epochs (Task B)	[2-8]	Discrete
Epochs (Task C)	[2-15]	Discrete

Table 4: Search space for Transformer models

A.2 Hyperparameters of ML models

Model	Hyperparameters
Logistic Regression	max_iter: 1000
Random Forest	n_estimators: 500 min_samples_split: 2
Decision Tree	max_depth: 15 min_samples_split: 2
SVM	max_iter: 1000 kernel: 'rbf'
XGBoost	max_depth: 6 (default) learning_rate: 0.3
AdaBoost	n_estimators: 100 learning_rate: 1.0

Table 5: Hyperparameters for ML models