*CSE556 Natural Language Processing*
*Md. Shad Akhtar*
*Indraprastha Institute of Information and Technology, Delhi*

# MULTILINGUAL TOXIC COMMENT CLASSIFICATION USING SIAMESE NEURAL NETWORK - [SNN]

**Samridh Girdhar**
2021282

**Manan Chugh**
2021335

**Ankit Gautam**
2021518

**Abhijeet Anand**
2021509

**ABSTRACT**
The online toxic comments are extremely destructive to the society, with toxicity that describes anything that is unkind, insulting, or likely to drive someone out of a conversation. To have a safer, more collaborative internet, grateful contributions are made by a main area of focus on NLP technologies to identify toxicity across multiple languages, whereas part of misinformation disseminates in other languages. This work explains and implements an approach to toxicity models educate with the Jigsaw Multilingual Toxic Comment Classification dataset. We implement XLM-Roberta a transformer based model for feature extraction. The embeddings are then used in Siamese networks-using the tokens created for multilingual toxic comments and thereby removing the need to have a translation to English before processing, and Post Processing to improve the classification accuracy indispensably. Our final model achieved an AUC of 0.7469 for the training set and 0.7485 for the validation set, demonstrating the effectiveness of performance under cross-lingual toxicity detectors.

**KEYWORDS:** *Siamese network, XLM-Roberta, Toxic comments, Multilingual classification, NLP, Jigsaw dataset.*

## 1. INTRODUCTION

Whether we like it or not, the internet is now an integral part of our live. Major part of our daily activities uses the internet, especially *communication*. A big part of this networking is through the internet with networking platform. Even though this situation is extremely advantageous by optimizing the rate of information exchange for us, unfortunately, this fast and free flow of information exchange throughout the world also has its adverse effect, which is caused by toxic comments. People create and share content on social media platforms almost every day. However, toxic comments on the Internet severely disrupt the online experience of users and even lead to tension and controversies on the platform. Building models that can automatically classify toxic comments has become a focus of researchers and industry. Earlier toxic comment classification models were generally based on manual *rule building and simple classifiers*. Gitari et al[1] had built a toxic vocabulary corpus to record and classify toxic comments, while Rdulesuc et al[2] distinguished toxic comments from normal comments by calculating the similarity between sentences. Ravi et al[3] implemented ML models such as Naive Bayes, Logistic Regression and Support Vector Machines for toxic comment classification. Recently, since deep learning/ Neural networks-based models help avoid the limitations of manual feature engineering, it results in better toxic comment classification, models represented by Convolutional Neural Networks and Recurrent Neural Networks have proven to work great. However, since CNNs and RNNs rely on static word embedding vectors and cannot dynamically adjust the word vector representation based on different contexts. Research has been done with data to fine-tune BERT and then used the model for toxic comment classification, obtaining better results than the previous models. However, along with the exchange and integration of cultures around the world, people from different countries communicating on international social platforms resulting in multilingual comments. Though BERT is effective in toxic comment classification in a single language, but it lacks some generality when facing multilingual comments.

**Table 1.** Online comment examples.

| | |
|---|---|
| S1 | "Esta canción es tan sentida!" |
| S2 | "Estoy muy emocionado por dentro, So easy!" |
| S3 | "Hi, guys. Eres basura" |
| S4 | "Me decepciono tanto, you are son of a b**ch." |
| S5 | "Put up or shut up" |

## 2. LITERATURE REVIEW

### *2.1. Monolingual Toxic Text Detection*

Monolingual Toxicity detection has been extensively studied. Most studies are conducted on English datasets while some studies have been done on other languages as well, such as Korean, Hindi, Spanish, and Russian. The problem can be either formulated as binary or multi-class classification. For example, a widely studied dataset, "Toxic Comment Classification Challenge" (https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge (accessed on 10 March 2021)) contains 6 classes which includes toxic, severe toxic, obscene, threat, insult, and identity hate. Additionally, Waseem and Hovy [6] has developed a custom dataset with three classes, including racist, sexist, and neutral, for offensive language detection.

### *2.2. Multilingual Toxic Text Detection*

Language gap: the monolingual detection model cannot generalize to other languages [8]. Most studies ignore the language gap between languages. To handle texts in various languages, one study straightforwardly translates multiple source languages into a single target language, extracts grammatical and semantic features from the translated corpus, and eventually classifies the text [11]. For this method, the drawback is that the translation of multiple source languages will lead to excessive data noise, and these translation errors are transmitted to the underlying learning module. Ethem et al. trained a model on an English corpus and translated text comments in different languages into English to classify them. Wang et al. enrich the machine translation with bilingual affective features and use a text label propagation algorithm to attain text classification. This approach has improved the F1 value in each class compared with the classification model, which only considers monolingual texts. Multilingual toxic text detection also faces the challenge of lack of extensive training data, which is more available in a monolingual setting, primarily in English, which is still the focus of most studies in toxic language analysis [1,15,33]. Data augmentation [16] and transfer learning are two common approaches to address the low resource challenge. The former aims to enhance the training set, while the latter employs unsupervised learning on large unlabeled dataset to pre-train a LM, which is then fine-tuned on a smaller labeled training dataset.

### *2.3. Toxicity Detection Models*

#### *2.3.1. Conventional Learning Models*

while deep learning models have been widely adopted, traditional models have not disappeared. As presented in , studies have shown that, in the low-resource condition, LR significantly outperforms deep learning models. The actual potential of deep learning can only be harnessed with enough annotated data. Moreover, the simple feature-based methods maintain the model's general interpretability, a phenomenon that most deep learning models do not. Thus, the natural concern has been with the LSTMs' generalization to other baseline models.

#### *2.3.2. Deep Learning Models*

Typically, toxic comment detection is framed as a many-to-one sequence modeling problem. It is commonly solved with RNN [20] , particularly two of its differentiated variants: LSTM and GRU [21] since they have mechanisms to combat the gradient vanishing and explosion weakness issue that RNN generally possesses. In addition, Bi-LSTM and Bi-GRU [22] can encode backward and forward contextual information, which has been reported to produce some ability. Bert is grounded in the Transformer model, which is an architecture based on the multi-headed implementation of self-attention. These studies enable the model to explore how and to what degree each word in a sequence is heedful of every other word, producing a deeper sense of the contextual meaning of a term in a chain. BERT has demonstrated SOTA performance in numerous NLP tasks [23], including toxic comment detection [24]. Another way to handle the tokens in a sentence is to stack the token embeddings to form a matrix, which can then be processed by a Convolutional Neural Network (CNN) [15,45] for feature extraction and detection. Embedding can be done at the character [26], word [15], or even sentence-level [27]. The character-level encoding enriches the textual representation and enables the mining of multiple features to represent textual information at a finer granularity. Kim et al. proposed the word embedding vector-based model char-CNN [26] to make character-level representations reduce the number of lexicons for each language from hundreds of thousands and millions to tens of thousands, and even thousands, in a multilingual text message task. Blunsom et al. [27] propose a CNN-based model which can learn not only the word-level contextual information, but also the feature information at the global sentence-level granularity. This model has achieved a better model perplexity compared to the word-level encoding model on English and other language datasets. In addition, using external knowledge in detection models has been explored. It enables a detection model to combine a detection model with handcrafted domain keywords while training. Pamungkas et al. proposed a joint model based on Facebook's Multilingual Unsupervised and Supervised Embeddings and leveraged Hurtlex , a multilingual lexicon of toxic words, to help detect Hate speech. The results showed that the domain knowledge injection could help the performance, especially for the positive sample detection.

#### *2.3.3. Transfer Learning via Masked Language*

*Models*

By using a pre-trained model to encode contextual information embedded in the raw datum, the model can have a better understanding of what the given character/word/sentence means in its context. BERT is a leading language model breakthrough that applies MLM since it trained by self-supervised based on large text's corpora . MBERT [43] is BERT's pre-trained model on corpora in many languages. While BERT and MBERT are strong enough, Liu et al. [31] prove that BERT is still under-tuned, and they present an optimized version of BERT, RU. XLM [32] strengthens RoBERTa by adding TLM into the pre-training. RU 's XLM's same authors also presented XLM-R, a pre-trained model on large corpora in 100 languages. XLM-R obtains SOTA performance in cross-lingual detection, sequence labeling, and question answering [32]. Several of the latest studies have also employed XLM-R for toxic text analysis [22,32] and obtained SOTA performance. Most studies ignore the language gap between languages. To handle texts in various languages, one study straightforwardly translates multiple source languages into a single target language, extracts grammatical and semantic features from the translated corpus, and eventually classifies the text [11]. For this method, the drawback is that the translation of multiple source languages will lead to excessive data noise, and these translation errors are transmitted to the underlying learning module. Ethem et al. trained a model on an English corpus and translated text comments in different languages into English to classify them. Wang et al. enrich the machine translation with bilingual affective features and use a text label propagation algorithm to attain text classification. This approach has improved the F1 value in each class compared with the classification model, which only considers monolingual texts. Multilingual toxic text detection also faces the challenge of lack of extensive training data, which is more available in a monolingual setting, primarily in English, which is still the focus of most studies in toxic language analysis [1,15,33]. Data augmentation [16] and transfer learning are two common approaches to address the low resource challenge. The former aims to enhance the training set, while the latter employs unsupervised learning on large unlabeled dataset to pre-train a LM, which is then fine-tuned on a smaller labeled training dataset.

*2.3.4. Siamese Networks*

It is a commonly used trick to improve prediction performance by aggregating several existing classifiers [53]. In this study, we apply Siamese networks on custom loss functions, which, to our best knowledge, has not been investigated in prior studies. Table 2 compares the prior studies on the multilingual toxic text detection problem from four aspects, including the task, used model, language setting, and used dataset.

Another work was Multilingual HASOC [33] which was a shared task targeting hate speech and offensive language classification in English, German, and Hindi . Precisely, they had around 7K tweets and Facebook posts with manually annotated labels. The first sub-task distinguishes posts having content that is either hate or offensive from other types of posts. The second sub-task distinguishes the different categories of posts having hate and offensive content . Deep learning-based approaches were the best in classifying data in all three languages. OffensEval 2020 had more training data for English, around 9M tweets, albeit the annotations were semi-automatic. Additionally, there exist datasets for Arabic, Danish, Greek, Turkish, with manual annotations. In all languages, the performance of the BERT models was superior.

Our work is different from previous approaches because:

(i) Uses a large-scale dataset for a language other than English, that was created with the aim to reduce demographic biases;

(ii) experiment with multilingual approaches,

**Table 2.** A comparative table of prior studies on multilingual text classification tasks.

| Work | Task | Model | # Languages | Dataset |
|---|---|---|---|---|
| Roy et al. [32] | Hate speech detection | Transformer | Three | HASOC 2020 |
| Ranasinghe et al. [23] | Offensive language detection | Transformer | Five | OffensEval 2020 |
| Becker et al. [24] | Emotion detection | Stacking of meta learners | Four | SemEvalNews and BRNews |
| Ousidhoum et al. [25] | Hate speech detection | BiLSTM and LR | Three | Collected from Twitter |
| Huang et al. [39] | Demographic bias analysis | LR, CNN, RNN, and BERT | Five | Collected from Twitter |
| Corazza et al. [26] | Hate speech detection | LSTM, BiLSTM, and GRU | Three | From three sources |
| Aluru et al. [40] | Hate speech detection | LR and mBERT | Nine | from 16 sources |
| Pamungkas et al. [27] | Misogyny Detection | LSTM, GRU, and BERT | Three | AMI IberEval 2018 |
| Rasooli et al. [28] | Sentiment analysis | LSTM | Sixteen | Collected from Twitter |
| Dong et al. [29] | Sentiment analysis | dual-channel CNN | Nine | From five sources |
| Zhang et al. [56] | Sentiment analysis | attention network | Two | Emotion corpus |
| Kalouli et al. [34] | Question classification | Heuristics | Four | KRoQ |
| Can et al. [30] | Sentiment analysis | RNN | Five | Amazon and Yelp reviews |
| Our work | Toxic text detection | MBERT and XLM-R | Seven | Jigsaw 2020 |

including transfer learning and zero-shot-learning so that the embedding can be utilized through transferable learning methods;

(iii) perform an analysis of the amount of data needed to train reliable Siamese models which to the best of our knowledge have not been yet used for such task and,

(iv) experiment with multilabel classification, providing first insights into this challenge task.

## 3. METHODOLOGY

We built a comprehensive framework for a multilingual toxic comment classification system using a Siamese neural network architecture based on the transformer model XLM-Roberta.

*(1) Dataset and preprocessing:*

The dataset used is JIGSAW dataset available on the platform Kaggle. Each comment is labeled with a binary indicator (0 or 1) where '1' indicates toxicity. The **load_train_set** and **load_test_set** functions parse CSV files, extracting relevant fields like text content, language, and toxicity labels. This allows the system to handle data specific to each language separately.

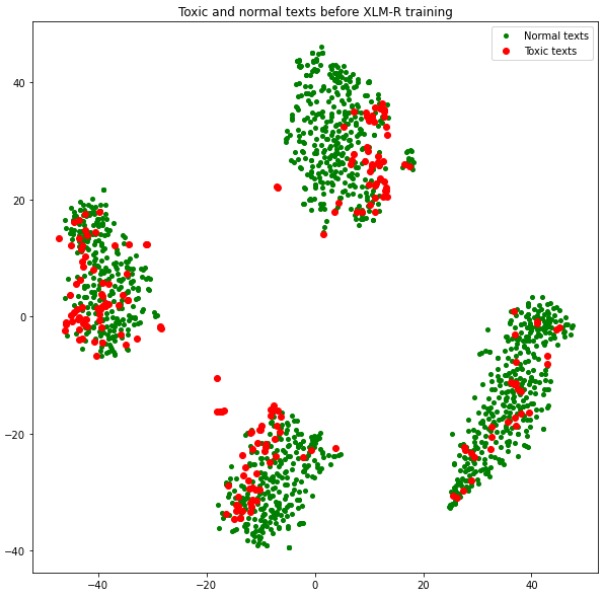
*(2) Tokenization:*

Text data are tokenized using the **AutoTokenizer** which is obtained from the Hugging Face **transformers** library. Specifically using the XLM-Roberta model tokenizer. This tokenizer converts text into a format that's suitable for input into the transformer model, handling vario us languages by recognizing language-specific tokens.

*(3) Siamese Network Setup:*

The Siamese network architecture is central in ensuring effective learning of how to identify toxic and non-toxic comments. The network has two branches of a neural network with the same weights:

*Feature extraction[1]:* Each branch uses the XLM-Roberta model to transform two texts input into two embeddings. These embeddings represent the text's semantic feature, which is essential in toxicity labeling.




*Distance calculation[2]:* Following the input embeddings, the network computes the Euclidean distance between the two resulting embeddings of a text. The idea is that pairs of texts are alike based on toxicity will have close embeddings points, but different pairs , toxic, and non-toxic will have far embeddings points.

**MODEL ARCHITECTURE:**

- Input Layer:

  Text Input: The model takes pairs of text inputs. Each text is tokenized using XLM-Roberta's tokenizer, converting raw text into a sequence of tokens that can be processed by the model.

- Embedding and Feature Extraction:

  *XLM-Roberta Model:* Utilizes the MRobertaModel, a transformer-based model pre-trained on multiple languages. This model converts tokenized text into dense vector representations (embeddings). Each text input is independently fed into this shared transformer model to obtain embeddings.

  *Attention Mask:* An AttentionMaskLayer is applied to handle padding and focus the model on meaningful tokens only.

- Build Siamese Dataset:

  This function is constructing a Siamese dataset for training models on the paired text data. It is preparing text pairs from various languages, with both similar and dissimilar labels, it also tokenizes them using a multilingual tokenizer, and batches them for model input, with the option to shuffle the pairs.

- Pooling Layer:

  Global Average Pooling: After obtaining sequence outputs from XLM-Roberta, a global average pooling layer is applied to reduce the sequence of embeddings into a single embedding vector per input text. This pooled output represents the overall semantic content of the input text.

- Normalization:

  G Layer Normalization: The pooled embeddings are normalized using a layer normalization step to stabilize the learning process and improve the training dynamics.

- Distance Calculation:

  Euclidean Distance Layer: A custom Lambda layer calculates the Euclidean distance between the normalized embeddings of the paired texts. This distance metric is crucial for the Siamese network to learn the similarity between text pairs.

- Optimizer: Adam optimizer with a triangular cyclic

learning rate. This setup helps in managing learning rate adjustments throughout training, facilitating better convergence.

- Callbacks: Include early stopping to prevent overfitting and a model checkpoint to save the best model based on validation loss.

*(4) Loss Function:*

The Loss Function DBLLogLoss is utilized during the training process. The customized loss function is utilized to model the output of the Siamese network: the output, which is the distance between pairs of inputs similar to an influencial distance matrix . The loss gets the siamese network to put the distance between the embeddings of the similar class tuples to each other at 0. On the other hand it will make the distance between the embeddings of the non-similar class at large distances. The loss is used to accomplish the above goals.

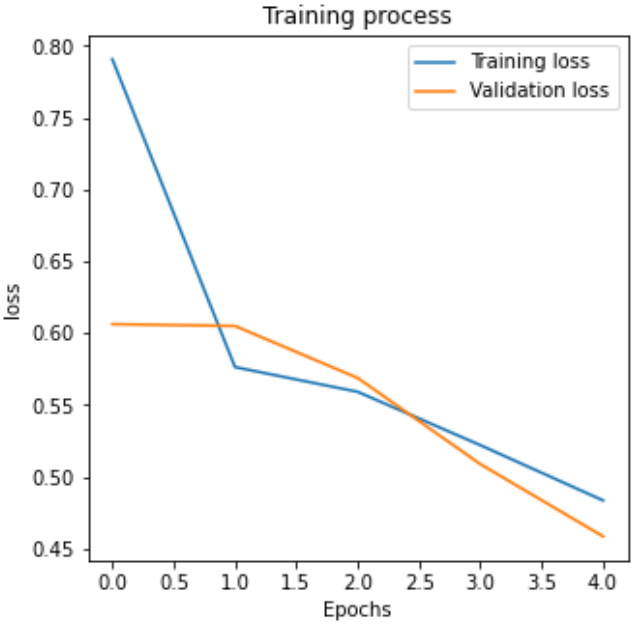
*(5) Training the Model:*

The model is trained with text pairs. In the training phase, pairs of text samples are assigned to the Siamese network. The present text pair embeddings distance is calculated. Next, the custom loss function is employed to describe the siamese network as a geometric layer. This method is based on a variety of models and entails various training times. The process is controlled based upon a numerical value.

The final train_loss achieved was 0.4755

The final val_loss achieved was 0.5523

*(6) Evaluation:*



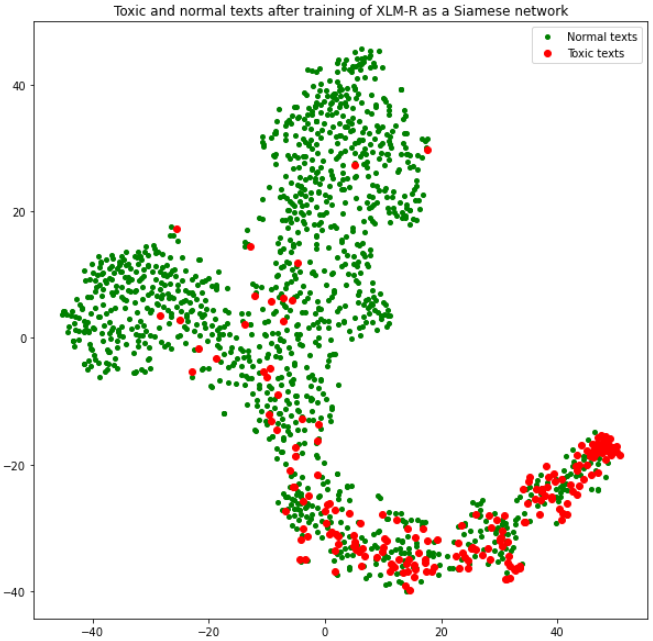

Post training, the model can predict the toxicity of new comments. It does this by calculating the distance of a new comment's embedding to known toxic and non-toxic embeddings, classifying the comment based on which it is closer to.

*(7) Feature Extraction and visualization:*

After-training, features (embeddings) of the texts are extracted and can be visualized using techniques like t-SNE, which helps to demonstrate how well the model has learned to separate toxic from non-toxic comments in a low-dimensional space.

## 4. Results

The results of the Siamese Neural Network are as follows:




| Parameter/Metric | Result |
|---|---|
| Accuracy | 0.6543 |
| Precision | 0.7478 |
| Recall | 0.7258 |
| F1-Score | 0.71 |

## 5. Findings

*Dataset Scale Impact:*

A larger dataset will better help the model generalize across languages and across biases, as it will have more examples to learn from.

*Transfer Learning & Zero-shot Learning:*

These methods might provide promising results in terms of allowing the model to learn from one language and predict in the other . It would be especially helpful when working with languages that do not have enough training data.

*Siamese Model Relevance:*

As Siamese models are rarely used for toxic comment classification, this project might reveal that Siamese models are especially good at understanding comment similarity and toxicity patterns across languages.

*Multilingual Classification Complexity:*

The results show how complicated it is to use multi-lingual classification and give initial clues on how to solve such issues.

## 6. Discussions:

Does the model exhibit any demographic biases, despite the dataset's aim to reduce them?

- Analysis: Investigate model predictions for potential biases based on demographic indicators inferred from the text.
- Observation: Uncover latent biases in the model predictions, leading to considerations for further bias mitigation techniques.

How well does transfer learning work for multilingual toxicity classification?

- *Analysis:* Measure performance metrics across languages with varying amounts of training data.
- *Observation:* The model may perform exceptionally well on languages closely related to those included in the training set, proving the effectiveness of transfer learning.

## 7. Future Work:

*1.Expanding Contrastive Learning Approaches:*

Further research should explore advanced contrastive learning strategies to fine-tune the embeddings of toxic comments across more diverse languages. This would involve experimenting with various positive and negative sampling techniques to enhance the discriminative power of the embeddings.

*2. Cross-lingual Transfer Enhancement:*

Investigate the impact of leveraging contrastive learning to improve cross-lingual transfer capabilities. By ensuring that the model learns language-agnostic representations, we can improve zero-shot and few-shot learning performances on underrepresented languages.

*3. Domain-Specific Model Adaptation:*

Apply contrastive learning to adapt the model to specific domains or platforms where the nature of toxicity might vary. This includes customizing the model for different online communities, social media platforms, or demographic groups.

*4. Data Augmentation Techniques:*

Explore the use of data augmentation in a contrastive framework to synthetically generate additional toxic and non-toxic comment pairs, thereby addressing data scarcity issues for certain languages or dialects.

*5. Semi-Supervised Learning:*

Implement semi-supervised learning methods to utilize unlabeled data effectively. Contrastive learning can be pivotal in leveraging large amounts of unlabeled text to enhance model robustness.

*6. Contrastive Pre-Training:*

Pre-train language models using a contrastive objective on a vast corpus of multilingual text to learn more universal representations before fine-tuning on the task-specific annotated data.

*7. Robustness to Adversarial Attacks:*

Develop methods to improve the model's robustness against adversarial attacks, where the model is exposed to subtly altered toxic content designed to evade detection. Contrastive learning could be used to recognize these adversarial examples better.

*8. Bias and Fairness Assessment:*

Evaluate and mitigate biases in model predictions using contrastive learning to ensure that the model's understanding of toxicity is equitable across different languages and cultural contexts.

*9. Multimodal Toxicity Detection:*

Extend the model to incorporate multimodal data (e.g., text with images or videos) where contrastive learning can be utilized to understand the nuanced interplay between different data modes in conveying toxicity.

*10. Interactive Learning Environments:*

Create interactive environments where the model can continuously learn from user feedback in a contrastive learning setting, thus improving its performance over time with real-world data.

*Dataset: Kaggle Link*
*Model: Google drive link*
*Code: Github Link*

**REFERENCES**


[l] Gitari, N. D., Zuping, Z., Damien, H., & Long, J. (2015). A lexicon-based approach for hate speech detection. International Journal of Multimedia and Ubiquitous Engineering, 10(4), 215-230.

[2] Rădulescu, C., Dinsoreanu, M., & Potolea, R. (2014). Identification of spam comments using natural language processing techniques. In: 2014 IEEE 10th International Conference on Intelligent Computer Communication

[3] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," 26th Int. World Wide Web Conf. 2017, WWW 2017 Companion, no. 2, pp. 759–760, 2019, doi: 10.1145/3041021.3054223.

[4] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, "Convolutional neural networks for toxic comment classification," arXiv, 2018. [3] M. Bilewicz and W. Soral, "Hate Speech Epidemic. The Dynamic Effects of Derogatory Language on Intergroup Relations and Political Radicalization," Polit. Psychol., vol. 41, Jun. 2020, doi: 10.1111/pops.12670.

[5] I. M. R. Prawira, Adiwijaya, and M. S. Mubarok, "Klasifikasi Multi-Label Pada Topik Berita Berbahasa Indonesia Menggunakan Multinomial Naïve Bayes," e-Proceeding Eng., vol. 5, no. 3, pp. 7774–7781, 2018.

[6] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," Proc. 11th Int. Conf. Web Soc. Media, ICWSM 2017, no. Icwsm, pp. 512–515, 2017.

[7] I. Tenney, D. Das, and E. Pavlick, "BERT rediscovers the classical NLP pipeline," ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf., pp. 4593–4601, 2020, doi: 10.18653/v1/p19-1452.

[8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf., vol. 1, no. Mlm, pp. 4171–4186, 2019.

[9] Y. Li, L. Xu, F. Tian, L. Jiang, X. Zhong, and E. Chen, "Word Embedding Revisited : A New Representation Learning and Explicit Matrix Factorization Perspective," 2014.

[10] B. van Aken, J. Risch, R. Krestel, and A. Löser, "Challenges for toxic comment classification: An in-depth error analysis," arXiv, 2018, doi: 10.18653/v1/w18-5105.

[11] A. Adhikari, A. Ram, R. Tang, and J. Lin, "DocBERT: BERT for document classification," arXiv, 2019.

[12] I. F. Putra and A. Purwarianti, "Improving Indonesian text classification using multilingual language model," arXiv, 2020.

[13] M. O. Ibrohim and I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter.," ALW3: 3rd Workshop on Abusive Language Online, 2019. https://github.com/okkyibrohim/id-multi-label-hate-speech-andabusive-language-detection (accessed Mar. 05, 2021). and Processing (ICCP). pp. 29-35.

[14] Bhaskaran, J., Kamath, A., & Paul, S. (2017). DISCo: Detecting insults in social commentary

[15] ] Bhaskaran, J., Kamath, A., & Paul, S. (2017). DISCo: Detecting insults in social commentary. [7] Mikolov, T., Kombrink, S., Burget, L., Černocký, J., & Khudanpur, S. (2011). Extensions of recurrent neural network language model. In: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 5528-5531.

[16] Dos Santos, C., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 69-78.

[17] Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., & Plagianakos, V. P. (2018). Convolutional neural networks for toxic comment classification. In: Proceedings of the 10th Hellenic Conference on Artificial Intelligence. pp. 1-6. [10] Li, S. (2018). Application of recurrent neural networks in toxic comment classification (Doctoral dissertation, UCLA).

[18] d'Sa, A. G., Illina, I., & Fohr, D. (2020). Bert and fasttext embeddings for automatic detection of toxic speech. In: 2020 International Multi-Conference on:"Organization of Knowledge and Advanced Technologies"(OCTA). pp. 1-5.

[19] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. [13] Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data, 2(1), 1-21.

[20] Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. Science China Technological Sciences, 1-26.

[21] Erhan, D., Courville, A., Bengio, Y., & Vincent, P. (2010, March). Why does unsupervised pretraining help deep learning?. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 201-208.

[22] Lample, G., & Conneau, A. (2019). Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291.

[23] Shibata, Y., Kida, T., Fukamachi, S., Takeda, M., Shinohara, A., Shinohara, T., & Arikawa, S. (1999). Byte Pair encoding: A text compression scheme that accelerates pattern matching. Technical Report DOI-TR-161, Department of Informatics, Kyushu University. [18] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692. [19] Lison, P. (2015). An introduction to machine learning. Language Technology Group (LTG), 1(35).

[24] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning. pp. 1096-1103. [21] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification?. In: China National Conference on Chinese Computational Linguistics. pp. 194-206.

[25]

[26] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All you Need. In: NIPS.

[27] Browne, M. W. (2000). Cross-validation methods. Journal of mathematical psychology, 44(1), 108-132. [24] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of