The Impact of Deep Learning on Multilingual Toxic Comments

MSc Research Project
Data Analytics

Gulbahar Erol
Student ID: 23136235

School of Computing
National College of Ireland

Supervisor: Naushad Alam

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Gulbahar Erol
Student ID:	23136235
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Naushad Alam
Submission Due Date:	12/08/2024
Project Title:	The Impact of Deep Learning on Multilingual Toxic Comments
Word Count:	XXX
Page Count:	14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Gulbahar Erol
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

The Impact of Deep Learning on Multilingual Toxic Comments

Gulbahar Erol
23136235

Abstract

This study covers the effectiveness of deep learning models in detecting multilingual toxic comments. With the rise of social media platforms, there has been an increase in the number of cyberbullying, hate speech, and toxic content. This situation can negatively affect the mental health of individuals. In the study, deep learning methods are used to detect toxic comments and reduce their effects. Previous studies by Singh and Chand (2022) were taken as a reference and expanded, and better deep learning methods were applied. In addition to the studies, F1 score values over 80% were obtained in different languages using multilingual datasets. Most of the previous studies were limited to English datasets, and limited research has been done on multilingual datasets. In this study, a multilingual dataset containing 6 different languages was examined and experiments were conducted using three deep learning methods, namely Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN). The results of the study showed that these models were successful in detecting toxic comments.

1 Introduction

With today's technology and accessibility, and especially with the rise of social media platforms, there is an increased number of instances of cyber bullying, hateful and toxic content and comments. In addition to people expressing their opinions freely, expressing our opinions without thinking and judging people easily and harshly has negative and hurtful effects on the other person. Singh and Chand (2022) describes how easy it is for people to communicate with each other while remaining anonymous when using social media, but also how they can be toxic, aggressive, threatening, degrading or abusive towards other people. It underlines that this situation can have such severe effects that some individuals become dispirited or depressed, or even think of ending their lives. It is important for public health to filter and, if possible, prevent this chaos and cruelty seen on online platforms.

Husnain et al. (2021) highlight the increase in online bullying and the harm this increase creates in society. They argue that identifying these harmful comments and grouping them correctly and implementing various policies are effective in increasing security.

Understanding the severity and target of bullying in the comments is important to find the purpose of the study. Therefore, determining the starting point correctly and drawing the project boundaries will be effective in the execution process of the project.

Machine learning and deep learning methods can help in the process of detecting harmful and toxic contents online. In this work, we aim to address and mitigate the impact of toxic comments using deep learning methods and attempt to address to following research question,

Are helpful deep learning models at detecting and reducing toxic multilingual comments in online communities?

The main purpose and motivation of the study is to extend the work of Singh and Chand (2022) with better deep learning methods. While the existing work primarily has explored traditional machine learning methods, there was a white space to explore and apply deep learning methods to this problem. On the other hand, expanding the dataset to multi-language is another insight of the project. Our experiments show superior performance were obtained with F1-score values over 80% in the models we created using different languages over existing approaches .

Previously works, although very substantial have not explored multilingual datasets to solve this problem. The studies were mostly trained with English language datasets. Some of them tested with multilingual data after trained English dataset. Some of them used single language to train and test.Our work builds on top of the preceding works done in this domain. We investigate a multilingual dataset including 6 different languages demonstrating the novelty of this study. Based on the approaches in the paper presented by Androcec (2020), we experimented with 3 deep learning approaches: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and Recurrent neural network (RNN).

The report is structured as follows:

In the section 2, we discuss a comprehensive literature review of the problem and discuss the state of the art in domain of toxic comment classification. In the section 3, discusses the methodology where the data set and the general outlines of the study are explained. In the section 4, the design specification where the implementation processes, models, requirements and general frameworks are explained. In the section 5 explains the implemented outputs and converted data. In the section 6, evaluation with the evaluation and comparison of the results. In the section 7 we conclude the work and discuss areas of future studies in this domain.

2 Related Work

In today's world, any post shared on social media that has the potential to lead to cybercrimes is often considered toxic or harmful. Such content has become a focal point of numerous studies due to its damaging effects and in this content various studies have been conducted with machine learning and deep learning such as distinguishing meanings, connections, and patterns in textual data and sentiment analysis. Machine learning and deep learning have proven to be effective in these studies.

Although the studies conducted differ in the models and data transformation stages, the operations applied to the data are fundamentally similar and each aims to obtain maximum efficiency from the model.

For example, Husnain et al. (2021) and Kumar et al. (2023) used different machine learning models in their studies, and cleaned the data from special characters, numbers, capital letters, and stopwords before applying the models. Tokenization, stemming, or lemmatization steps were also applied. Husnain et al. (2021) presented two different approaches to determine toxicity levels in his study. He performed both binary classification and multi-label classification and interpreted the results. Experimental results show that the logistic regression model has better performance than other machine learning models such as Naive Bayes and Decision Tree Classifiers. The authors also demonstrated the potential of the logistic regression model in toxicity classification problems through experiments. Kumar et al. (2023) compared the results using Logistic Regression, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, and Gaussian Naïve Bayes models. They included a comparison of vectorization methods such as TF-IDF, InferSent, BERT, and T5 in the study and presented us how these methods performed with the models. According to the study results, the T5 approach outperformed other models with the highest F1 Score along with the Random Forest Classifier model.

In similar study, Singh and Chand (2022) highlights the need for new approaches to automatically detect anti-social behavior through NLP, machine learning and artificial intelligence techniques in their study. They used Multinomial Naive Bayes, Logistic Regression and Support Vector Machine (SVM) models with TF-IDF vectorization to detect the level of negativity in the comments. They argue that Logistic Regression outperforms other models with the highest F1-Score and the lowest Hamming loss. This shows its effectiveness in multi-label comment classification.

Deep learning models have generally demonstrated superior performance over traditional machine learning methods in cases such as distinguishing meanings, connections and patterns in textual data, and sentiment analysis. Li et al. (2022) has considered a deep learning model based on the XLM-RoBERTa model for the classification of multilingual toxic comments. LSTM and RNN models were also been used to classify toxic comments in various languages, but XLM-RoBERT provided more successful results.

Haralabopoulos et al. (2020) used 5 different Deep neural networks (DNN) architectures for multilabel. The first one represents a basic Convolutional Neural Network (CNN). Another method is a hybrid model that combines pre-trained embedding layers and recurrent layers (GRU). The third one represents a fully connected (dense) deep neural network using TF-IDF embedding layers. Another one uses an LSTM-based model for text classification. Finally, it presents a CNN model designed to capture different n-grams (unigram, bigram, trigram). He also used ensemble learning methods with another dataset. The author, who uses DNN outputs as input in the ensemble learning method, provides more effective results with ensemble learning.

Despite the efficacies of deep learning and machine leaning techniques in addressing this problem, there are still a few challenges. Every languages have different word patterns in the common language that have different meanings but do not appear toxic from the outside. It may cause problem in labelling. Incorrect labeling may cause modeling to produce false positive and false negative results. Experiencing comments blocked by incorrect classification may restrict freedom of expression and therefore may lead to ethically incorrect results. It can lead to difficulties in classifying misspellings, abbreviations and slang(van Aken et al. (2018)) The context of the word is also related to the neighboring words((Georgakopoulos et al. (2018))) . The word embedding method performs the process of correctly interpreting and classifying the input by generating a dense vector with a defined direction and fixed values for each word.

The studies using different languages as train data will also shed light on new studies. Due to the complexity of the data, it is quite common to prefer complex models in text classification studies. Because of the libraries used work more efficiently on English words, datasets containing predominantly English data are preferred in studies. Ağduk et al. (2023) also mentions the difficulty of classifying texts in different languages, in their study.From this perspective, this study will contribute to filling the gap in the literature. The basis of the necessity of the study stems from this gap.

However, Kapse et al. (2023) examined toxic comments containing multiple languages with deep learning models and provided effective results. In text classification studies, whether the data set is monolingual or multilingual causes differences in data preparation and model application processes. Texts have been tokenized and padding applied. In addition, they aimed to reduce the randomness of the words and make the machine's job easier by applying lemmatization. LSTM, QRNN, GRU and BERT models were compared in the study and according to the results, the LSTM model gave more effective results than the others. On the other hand, also Naseeba et al. (2023) used deep learning model such as LSTM, CNN in toxic comment classification work and they proved LSTM got better results. In Naidu et al. (2023) 's study, LSTM is used as the base model for the classification of toxic comments, and also CNN models are trained at the Character level and Word level. It creates a hybrid model by combining LSTM and CNN layers. The models classify toxic and non-toxic comments received from online platforms.

In one study, Georgakopoulos et al. (2018) adapted the CNN model for text classification. Although CNN is mainly used in image processing, the author also used it for text classification using the word embedding method. When compared with traditional data mining models, the results obtained with CNN are clearly ahead. In the literature review, CNN has been included in many studies on text classification. As we mentioned above, Naseeba et al. (2023) also used CNN in his study, but achieved more successful results with LSTM. Another study, Neog and Baruah (2024) developed a hybrid model by combining CNN with BiLSTM. It aimed to produce a more powerful solution by overcoming the limitations of both languages.

In contrast to the studies mentioned up to this point, Morzhov (2020) emphasized that every piece of information has a contribution to the model and because of that he defended to skip the preprocessing steps such as stemming, lowercasing or stopword removal. He also added that in NLP tasks, in order to represent words in a way that the computer can understand, it is necessary to use pre-trained word embeddings using a fixed dictionary and vectors reflecting semantic similarity and to perform text preprocessing accordingly. The author only removed things like IP addresses, links, numbers and replaced the signs at the end of sentences with short codes. Although the working logic of the models used is consistent with the author's attitude, the preprocessing stage was insufficient. Although he also stated that the data was unbalanced, he did not apply any balancing steps for this. Although the results show high values, the accuracy of the data is questionable. Because of the imbalancing on the dataset, results can be effected overfitting. Maslej-Krešňáková et al. (2020) oppose the view that preprocessing techniques may cause the text to lose its original features and therefore performance may decrease, and they mention the importance of preprocessing steps in the studies. Their main motivation in the research is to prove the contribution of success to the model by comparing different techniques. The models' structure which are used in this study are required well organised dataset. All datas should be understandable and trainable. Since this is achieved through the preprocessing phase, these stages should not be skipped. Hasanin and Khoshgoftaar (2018) also mentions the random sampling method, which is another pre-processing stage in his study. He demonstrates the effectiveness of the random undersampling method for large data and the success of the method by presenting that the results are close to the results of the study conducted with all data.

Although machine learning models show remarkable results, it will be a possible outcome that stronger and complex models will provide more effective values for this study. This research will propose analysis and experiments based on deep learning approaches. While the purpose of the studies, the data sets used, and the preprocessing stages vary, the results also vary according to these conditions. These all works based on english comments. Based on this, we will see how successful the same models will be by applying similar studies to the multilingual toxic comment dataset in the following sections.

3 Methodology

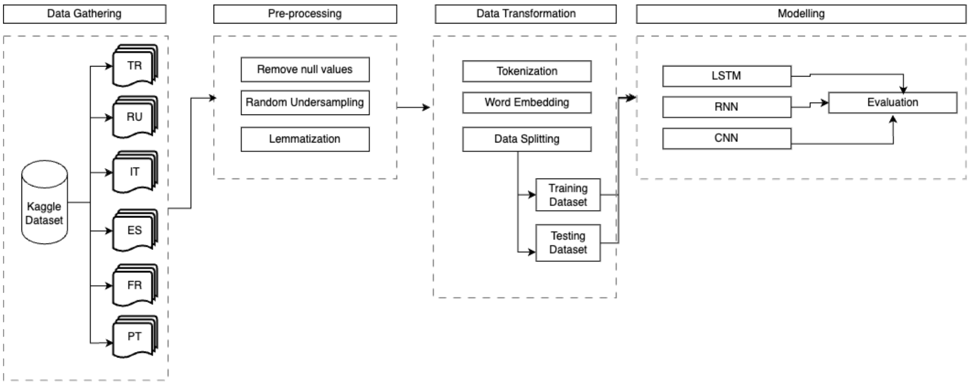

Figure 1: Methodology

The methodology consist of data gathering, preprocessing, transferin,modelling and evaluation. The general flow, the steps of the pre-processing phase, the applied models and metrics are as in the Figure 1.

3.1 Understanding Data

Table 1: Overview Dataset

Attributes	**Rows**	**Description**
Id	1341294	Identifier
comment text	1340101	Text
toxic	1341294	binary label
severe toxic	1341294	binary label
obscene	1341294	binary label
threat	1341294	binary label
insult	1341294	binary label
identity hate	1341294	binary label

The data was originally taken from the Toxic Comment Classification Challenge competition, translated with Google API, and then shared under the Kaggle-Jigsaw Train Multilingual Comments (Google API) title. Data sets from 6 different languages were combined in a single frame. The data set contains a total of 1341294 rows of data and 1340101 comments.

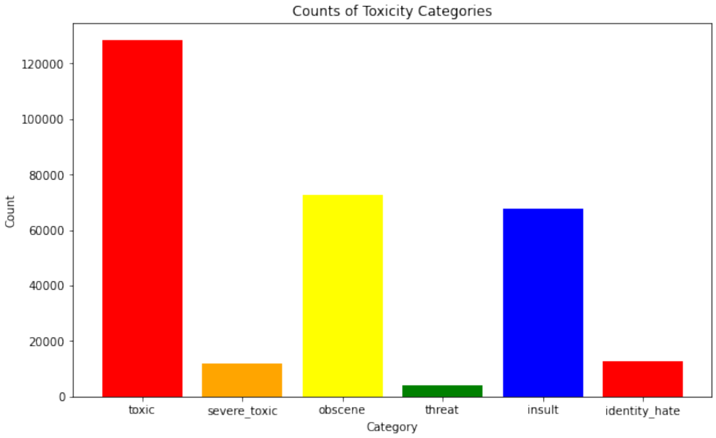

Figure 2: Distribution of Categories

As seen in Table 1, there are 6 binary label variables. Each comment can contain more than one label. The dataset has a balanced distribution for each language.

Figure 2 shows the distribution of labels. The unbalanced distribution in the data is clearly visible in Figure 5. Since it was not suitable for this model, it will be balanced with the Random Undersampling technique in next step .

3.2 Preprocessing

The preprocessing phase of the data involves bringing the comment texts into the correct format for classification modelling. The steps of removing punctuation marks, special characters and numbers, converting them into words and removing blocked words were performed respectively. Because of imbalance in dataset Random Undersampling is applied. Lemmatization is then performed to reduce the words to their basic forms.

3.2.1 Missing Data

In the dataset, there are only 193 null data in the comment text column. This number is negligible compared to the size of the dataset, so these rows were removed.

3.2.2 Normalization

Figure 3 shows the distribution of things such as extra spaces, special characters, capital letters in comment texts. Punctuation, special characters, HTML tags, numbers, digits have been removed. Special symbols are defined and all of them removed. The images of the most frequently used words in each language are displayed in Figure 4.

3.2.3 Stopword Removal

It is part of the Natural Language Toolkit (NLTK) library and is a widely used library in NLP. Tokenization, parse tree visualization, and other activities can be carried out by giving numerous text processing libraries access to test datasets (Vidhya (2021)). It was

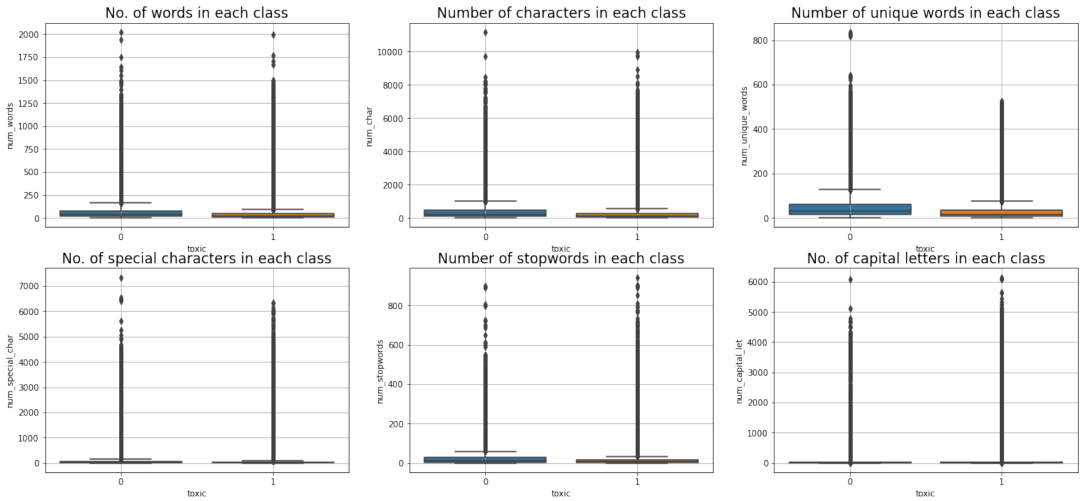

Figure 3: Comment Text Analysis

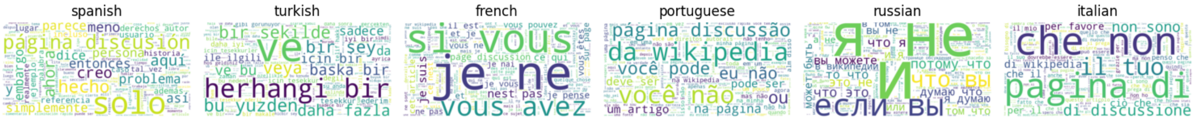

Figure 4: Most Common Words

used for removing from the text with the necessary plugins for each language, stopwords in the study.

3.2.4 Handling Imbalance Data

Dataset is not balanced and some steps should be taken to balance it out. Since the size of the dataset is large enough, it is more appropriate to balance the dataset by reducing it instead of generating random data. Using a random selection process, Random Undersampling removes samples from the majority class from the training dataset. Hasanin and Khoshgoftaar (2018) created five new unbalanced large datasets with positive class targets of 10%, 1%, 0.1%, 0.01%, and 0.001% from the original full datasets and examined these unbalanced datasets to see if random information loss would occur . Random undersampling was applied to balance the binary class in each created unbalanced dataset and using Models Random Forest classifier with 50:50% class ratios proved that the majority class found good ratios without losing much data. Adequate performance is achieved on the 0.1% to 1.0% minority class even when compared to a 10% or even a 100% fully balanced dataset. Additionally, applying random decay to a 50:50 negative class ratio showed similarities to the performance obtained using the entire large dataset.

3.2.5 Lemmatization

Shambharkar et al. (2023) defined lemmatization as the process of combining various differences of a word to detect it as a single object by a machine learning algorithm. This reduces the size of the data and the number of different words in the data. Lemmatization

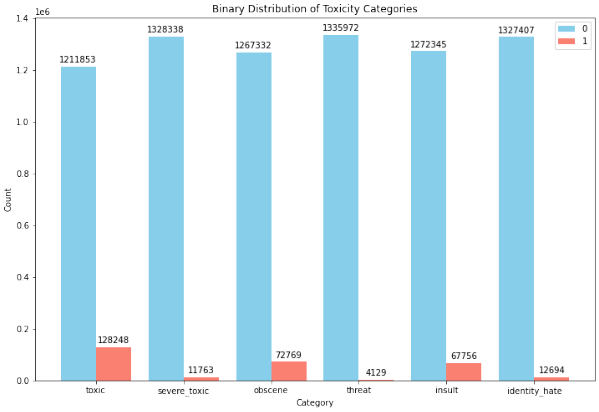

Figure 5: Data Balance

takes into account the entire vocabulary of a language to apply a morphological analysis to words. This steps help models to understand text.

.

3.3 Modelling

The dataset was divided into 20%-80% as test and training sets. In order to bring the texts into a format that the models can understand, they are divided into smaller pieces with tokenization and converted into binary format using vectorization. In his research, Androcec (2020) conducted an in-depth literature research and examined toxic comment analysis studies. It provides a general resource that includes the results obtained, the models used, and the datasets. In line with the project objectives, Recurrent Neural Network (RNN), Long Short-Term Memory(LSTM), Convolutional Neural Network (CNN) were selected, taking Androcec (2020)'s work as a reference. Accuracy is the metric that will be used for the success of the models and their suitability for work.

3.3.1 Recurrent Neural Network (RNN)

Since it uses past information as input in the next step, RNN gives more successful results in NLP compared to other deep learning models. **?** explains the logic of RNN as follows: While reading a text, our ability to understand the words we read better is due to the fact that we remember the previous words of the sentence. In RNN, it takes the previous output and uses it as input and continues the model. In other words, input and output values are not independent from each other.

3.3.2 Long Short-Term Memory (LSTM)

LSTM works like RNN but with random intervals. It is successful with subjects such as sentiment analysis, text generation and time series, and is also very successful in text classification.Memory is an advantage of LSTM (Dubey et al. (2020)). Because memory is available, LSTM networks, in contrast to RNNs, may also store and recall lengthy sequences. As mentioned in the second section, it has been used in many studies and has given good results. In this study, we will see the effect of LSTM in multilingual text classification.

3.3.3 Convolutional Neural Network (CNN)

CNN is a deep learning algorithm that is mostly used in image processing and takes images as input, but it is also widely used in text classification. The algorithm consists of different layers. The image that passes through these layers, which are Convolutional Layer, Pooling and Fully Connected, is subjected to different processes and becomes ready for the deep learning model. The components of an image are pixels represented by integer values in a certain range. Therefore, it is difficult to process the original raw data for text classification problems. In the Georgakopoulos et al. (2018) study, before feeding the CNN, it encodes the components, namely words, in text classification. Then, it converts them into fixed-size matrices. After converting the words into low-dimensional vectors for the embedding layer, it processes them with CNN.

4 Design Specification

After cleaning the data, the first step is to balance the data. Random undersampling was applied to balance the data. Using the lemmatization technique, text data was divided into word roots. The lemmatization processes for turkish language data were applied separately. The spacy library was used for Lemmatization. An open-source Python package called spaCy was created especially for natural language processing (NLP) applications like dependency parsing, named entity recognition, and part-of-speech tagging. It is very powerful for NLP and supports many languages except Turkish language. The dataset is grouped by language. For each language, lemmatization is applied with the relevant plugin downloaded from spacy. A separate function has been prepared for Turkish with trnlp. Due to the abundance of data and the complexity of lemmatization, data created separately were combined into a single data frame.

With Keras tokenizer, texts are divided into smaller pieces that LTSM model can understand. The model was limited to the 5000 most frequently used words and the input length to 100 words. The dropout rate, that is, the rate at which neurons will be randomly disabled, was determined as 20%. The model was trained 15 epoches. After the tokenization step, words were converted to vectors with word embedding. This step was completed with Word2vec. Word2vec is an algorithm based on the assumption that words tend to be found in similar contexts. It finds semantic similarities in the vector space of words. It consists of two main models, Continuous Bag of Words(CBOW) and skip-gram. CBOW uses the surrounding words to find the word. Skip-gram tries to find the surrounding words using the word.

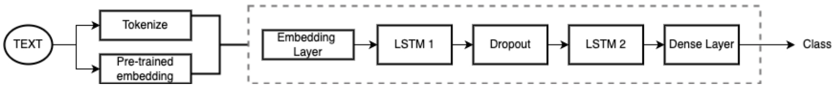

Figure 6: LTSM Architecture

Severe toxic, obscene, threat,toxic,insult,identity hate columns are not imbalanced and manipulated models negatively. So, the study is based solely on toxicity classification.

The parameters for the LTSM model were set as follows: the most frequent 10000 words and the input length were limited to 100 words. The 10,000 word embedding layer was converted into a 100 word vector. LSTM layer added with 64 units. The third layer is the 50% Dropout layer to prevent overfitting and the last layer is the Dense layer and produces predictions for toxicity.

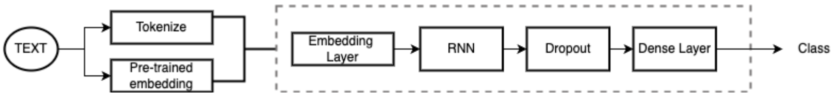

Figure 7: RNN Architecture

For RNN, text data was serialized with Tokenizer. The model was limited to the 10000 most frequently used words and the input length to 100 words. The 10,000 word embedding layer was converted into a 100-word-sized vector. The next layer is a SimpleRNN layer with 64 neurons. The third layer is the 50% Dropout layer to prevent overlearning. The last layer is the Dense layer and produces predictions for toxicity.

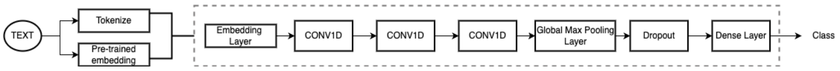

Figure 8: CNN Architecture

And CNN, after converting words to numerical vectors in the Embedding Layer, 3 1D convolutional layers are added, each using 128 filters and 5-dimensional kernel. GlobalMaxPooling applies maximum pooling in the 1D layer. It applies 50% dropout to prevent overfitting in the dropout. It performs binary classification with sigmoid activation function in the Dense Layer.

5 Implementation

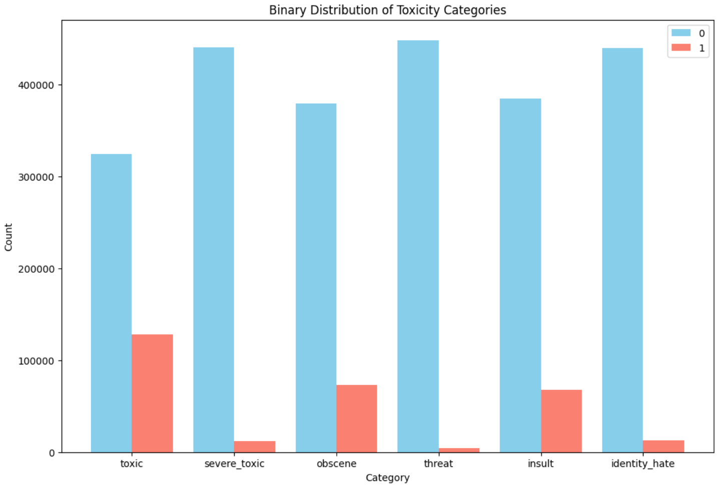

Figure 9: Balance after Random Undersampling

The data was first examined in the local environment. Necessary graphs and tables were created to understand the data. Language column was added and separate preprocessing processes were created according to languages. The stopword removal step was performed using components from the Spacy library specific to each language. In this step, the trnlp library was additionally used for Turkish data. Due to the large data size and the complex structure of the models used, the rest of the study was carried out using Google Colab Pro+.

After the undersampling step, the dataset was reduced to 303998 rows. This is not a problem as the data set is still large enough. After this stage, the status of the dataset is shown in Figure 9. Although there are still imbalances, it is better than before. Data size also decreased with random under sampling. This provided a more suitable dataset for the model.

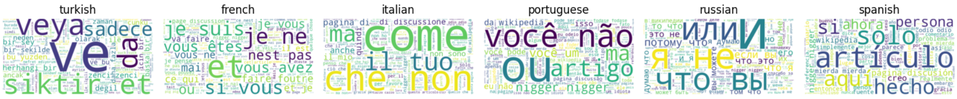

Figure 10: Word Cloud after preprocessing

After the data is edited and lemmatization operations are applied, the image in the word cloud is shown in the Figure 10.

Models were applied in Table 2 Table 4. Models

Table 2: CNN

Layer (type)	**Param #**
`embedding_1 (Embedding)`	61,586,500
`simple_rnn (SimpleRNN)`	0 (unbuilt)
`dropout_1 (Dropout)`	0 (unbuilt)
`dense_1 (Dense)`	0 (unbuilt)

Table 3: Layer Details of the Model

Table 4: CNN

Layer (type)	**Param #**
`embedding_2 (Embedding)`	61,586,500
`conv1d (Conv1D)`	0 (unbuilt)
`conv1d_1 (Conv1D)`	0 (unbuilt)
`conv1d_2 (Conv1D)`	0 (unbuilt)
`global_max_pooling1d`	0 (unbuilt)
`dropout_2 (Dropout)`	0 (unbuilt)
`dense_2 (Dense)`	0 (unbuilt)

Table 5: Layer Details of the Model

6 Evaluation

The dataset contains 6 different languages and 6 different labels. Since the main goal of the study is to classify the toxicity in comments, the main toxic column was considered.

Table 6: Results

Model	Accuracy	F1-score	Recall	Precision
LSTM	0.88	0.84	0.82	0.88
RNN	0.84	0.78	0.76	0.81
CNN	0.88	0.85	0.83	0.87

The results are presented in Table 6 with different metrics. Since the dataset is unbalanced, it may be misleading to consider only accuracy here. Also evaluating the F1-score metric will give the most reliable results. However, other metrics are also worth considering. The aim of this study is to work on a multi-language dataset for toxic comment classification in light of previous studies, to train successful models with multiple languages and obtain successful metrics. In direct proportion to the success of the models, deep learning will detect toxic comments and then apply the necessary procedures, which will reduce the negative effects of these comments on people. The dataset organization, normalization, and lemmatization stages have directly affected the model metrics. Although the working logic of deep learning models does not require much preprocessing, it is essential for the models to work correctly that the data is in a format that the models can understand and that it is as simple and basic as possible.

In this context, as seen in Table 6, the values reveal the success of the models. Although the accuracy values are seen as quite successful in metrics, they are not the most accurate criteria. Therefore, we can see that the results are successful mainly in the F1-score values.

According to the results, the CNN model generally exhibited the best performance. It obtained the highest values in terms of both F1 score and recall. The LSTM model attracts attention with its high accuracy rate and balanced F1 score. The RNN model exhibited lower performance compared to the other models, falling behind in terms of F1 score and sensitivity. This evaluation shows that CNN and LSTM models are more suitable for this task.

7 Conclusion and Future Work

The main purpose of the study was to show the effect of deep learning models in detecting multilingual toxic comments. The results obtained by passing different languages through the correct preparation stage and bringing them to the correct format for the models demonstrate the success of the study. There are certain key points in this study such as cleaning the texts, lemmatization, and vectorization. The effect of these on the models has been proven in the reference studies and has had an effect on the success of the models in this study.

For future work, the number of languages trained in the model can be increased. However, including languages from different alphabets would be valuable in expanding the study. The lack of support for different languages in the new libraries used in the study has limited the study. Improvements in this section will increase the success of the study. On the other hand, the imbalance of the dataset is another limitation. Making the dataset more balanced and expanding the study as multilabel will also increase the success in direct proportion.

References

Androcec, D. (2020). Machine learning methods for toxic comment classification: a systematic review, *Acta Universitatis Sapientiae, Informatica* **12**: 205–216.

Ağduk, S., Aydemir, E. and Polat, A. (2023). Classification of news texts from different languages with machine learning algorithms, *Journal of Soft Computing and Artificial Intelligence* **4**(1): 29–37.

Dubey, K., Nair, R., Khan, M. U. and Shaikh, P. S. (2020). Toxic comment detection using lstm.

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G. and Plagianakos, V. P. (2018). Convolutional neural networks for toxic comment classification.
URL: *https://doi.org/10.1145/3200947.3208069*

Haralabopoulos, G., Anagnostopoulos, I. and McAuley, D. (2020). Ensemble deep learning for multilabel binary classification of user-generated content, *Algorithms* **13**: 83.

Hasanin, T. and Khoshgoftaar, T. (2018). The Effects of Random Undersampling with Simulated Class Imbalance for Big Data, pp. 70–79.
URL: *https://ieeexplore.ieee.org/document/8424689/?arnumber=8424689*

Husnain, M., Khalid, A. and Shafi, N. (2021). A novel preprocessing technique for toxic comment classification, pp. 1223–1228.

Kapse, A. S., Dubey, A., Bisen, H., Kumar, K. and Tamheed, M. (2023). Multilingual toxic comment classifier, pp. 1223–1228.

Kumar, A. A., Pati, P. B., Sangeetha, S. and Deepa, K. (2023). Toxic comment classification using s-bert vectorization and random forest algorithm, *2023 IEEE International Conference on Contemporary Computing and Communications (InC4)* pp. 1–6.

Li, W., Li, A., Tang, T., Wang, Y. and Fang, Z. (2022). Multilingual Toxic Text Classification Model Based On Deep Learning, pp. 726–729.
URL: *https://ieeexplore.ieee.org/document/9985930*

Maslej-Krešňáková, V., Sarnovský, M., Butka, P. and Machová, K. (2020). Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification, *Applied Sciences* **10**(23).
URL: *https://www.mdpi.com/2076-3417/10/23/8631*

Morzhov, S. (2020). Avoiding unintended bias in toxicity classification with neural networks, *Proceedings of the XXth Conference of Open Innovations Association FRUCT* **26**: 314–320.

Naidu, B. R., Tangudu, N., Sekhar, C. C., Kavitha, K., Ramana, B. V., Reddy, P. V., Sahukaru, J. and Lopinti, R. G. (2023). Toxic comment classification using deep learning, *International Journal on Recent and Innovation Trends in Computing and Communication* **11**(7): 93–104.

Naseeba, B., Sai, P., Karthik, B., Chitteti, C., Sai, K. and Jangaraj, A. (2023). *Toxic Comment Classification*, pp. 872–880.

Neog, M. and Baruah, N. (2024). A hybrid deep learning approach for assamese toxic comment detection in social media, *Procedia Computer Science* **235**: 2297–2306. International Conference on Machine Learning and Data Engineering (ICMLDE 2023).
URL: *https://www.sciencedirect.com/science/article/pii/S1877050924008949*

Shambharkar, P., Singh, H., Raghav, H. and Verma, H. (2023). Exploring the Efficacy of Deep Learning Models for Multiclass Toxic Comment Classification in Social Media Using Natural Language Processing, pp. 1–8.

Singh, N. K. and Chand, S. (2022). Machine learning-based multilabel toxic comment classification, *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)* pp. 435–439.

van Aken, B., Risch, J., Krestel, R. and Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis.

Vidhya, A. (2021). Nltk: A beginner's hands-on guide to natural language processing. Accessed: 2024-08-10.
URL: *https://www.analyticsvidhya.com/blog/2021/07/nltk-a-beginners-hands-on-guide-to-natural-language-processing/*