information

Article

A Study of Multilingual Toxic Text Detection Approaches under Imbalanced Sample Distribution

Guizhe Song [1], Degen Huang [1] and Zhifeng Xiao [2,*]

[1] School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China; guizhesong@mail.dlut.edu.cn (G.S.); huangdg@dlut.edu.cn (D.H.)
[2] School of Engineering, Penn State Erie, The Behrend College, Erie, PA 16563, USA
* Correspondence: zux2@psu.edu

Abstract: Multilingual characteristics, lack of annotated data, and imbalanced sample distribution are the three main challenges for toxic comment analysis in a multilingual setting. This paper proposes a multilingual toxic text classifier which adopts a novel fusion strategy that combines different loss functions and multiple pre-training models. Specifically, the proposed learning pipeline starts with a series of pre-processing steps, including translation, word segmentation, purification, text digitization, and vectorization, to convert word tokens to a vectorized form suitable for the downstream tasks. Two models, multilingual bidirectional encoder representation from transformers (MBERT) and XLM-RoBERTa (XLM-R), are employed for pre-training through Masking Language Modeling (MLM) and Translation Language Modeling (TLM), which incorporate semantic and contextual information into the models. We train six base models and fuse them to obtain three fusion models using the F1 scores as the weights. The models are evaluated on the Jigsaw Multilingual Toxic Comment dataset. Experimental results show that the best fusion model outperforms the two state-of-the-art models, MBERT and XLM-R, in F1 score by 5.05% and 0.76%, respectively, verifying the effectiveness and robustness of the proposed fusion strategy.

Keywords: toxic comment; imbalanced positive and negative samples; pre-training models; multilingual classification; XLM-RoBERTa; MBERT

Citation: Song, G.; Huang, D.; Xiao, Z. A Study of Multilingual Toxic Text Detection Approaches under Imbalanced Sample Distribution. *Information* **2021**, *12*, 205. https://doi.org/10.3390/info12050205

Academic Editor: Arkaitz Zubiaga

Received: 8 April 2021
Accepted: 7 May 2021
Published: 12 May 2021

1. Introduction

With the rapid development of online media platforms, an increasing number of people participate in the sharing and dissemination of social data. Recent years have witnessed a surge in the number of toxic messages on various social platforms [1]. The emergence of buzzwords also makes online media an ideal place to post toxic comments, which generally refer to hate speech, insults, threats, vulgar advertisements, and misconceptions about political and religious tendencies. These spam messages have severely affected users' browsing experience on the platform and hindered the healthy development of social platforms. Therefore, it is crucial to conduct research on the identification of toxic comments to filter and clean the Internet environment.

Prior studies have treated toxic comment detection as either a binary [2–5] or multi-class [6–8] classification problem. A variety of methods have been explored and can be categorized into three types: dictionary-based approach [9], machine-learning-based approach [10–14], and deep-learning-based approach [15–18]. The dictionary-based approach employs a keyword list for precise matching. Whether this approach is advantageous mostly depends on the dictionary quality, which requires domain knowledge, resulting in poor generalization capability. Therefore, the approach is often combined with other text features [19,20] rather than serving as a standalone approach. Conventional machine learning models require manual feature engineering. Commonly used features in natural language processing (NLP) include N-gram features, part-of-speech (POS) features, syntactic dependency features, keyword importance features (e.g., term frequency-inverse document frequency (TF-IDF), TextRank), etc. The chosen features are then used to represent an input text sample. To train a classifer, a learning algorithm is adopted to fit the data in a training set to minimize the prediction error in an iterative fashion until convergence. These feature-based learning models have demonstrated satisfying performance in various text classification tasks. However, the quality of feature engineering largely depends on the model developer's expertise in a domain, which is generally hard to obtain. Deep neural networks, on the other hand, can address this challenge by capturing the text semantic information from raw text data, without manual feature engineering and also boost the detection performance [21]. To this end, deep learning algorithms have recently appeared in numerous studies on text classification.

Recent advances have also seen a focus switch from monolingual models [1–5] to multilingual models [22–29]. This need is driven by fast-growing social media users with a diverse linguistic background. For example, one may find comments in multiple languages or even mixed languages under a YouTube video. Table 1 shows five comment examples taken from the Sina Weibo, the largest social media in China, with regular (non-toxic) comments (S1 and S2) and toxic ones (S3–S5). Some comments are in a mix of languages, which limits the predictive capability for models trained on monolingual datasets. Therefore, it is imperative to create a model that can capture critical semantic information from multiple languages for toxic text detection.

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"

Training a model for multilingual toxic text detection is challenging in three aspects. First, the difference between languages creates a gap that is hard to bridge by the methods developed in a monolingual setting [30]. Second, there is usually a lack of annotated training data in less used languages [23,31,32]. The dataset utilized in this study also suffers from the low resource issue. The training set contains a large number of English-only samples that are relatively easy to obtain. In contrast, the validation and test sets contain fewer samples in the other six languages. Third, imbalanced sample distribution is a common challenge, given the fact that the majority of online comments are non-toxic. In a multilingual setting, the positive samples (toxic ones) in less used languages are seldom. Various translation-based methods [30,31,33–35] are employed to tackle the first challenge, and transfer learning [22,32] is the main technique used to overcome the second. Negative sampling and data augmentation [36] can help with the third challenge.

In this paper, we propose a learning pipeline based on model fusion for multilingual toxic text detection. Specifically, the proposed pipeline starts with a series of pre-processing steps to purify and augment training data, build lexicon, and vectorize word tokens to better prepare for the downstream modules. Two models, multilingual BERT (MBERT) [37] and XLM-RoBERTa (XLM-R) [38], regarded as the state-of-the-art (SOTA), are employed for pre-training through Masking Language Modeling (MLM) [37] and Translation Language Modeling (TLM) [38] in multiple languages, incorporating semantic and contextual information into the models. The pre-training phase is self-supervised, greatly alleviating the low resource problem. In addition, the English-only training set is augmented via translation. We fine-tuned the pre-trained models on the augmented training set. Finally, we apply fusion to different loss functions as well as different pre-trained models. The final fused models outperform MBERT and XLM-R in F1 score by 5.05% and 0.76%, respectively, demonstrating the effectiveness and robustness of the proposed fusion strategy.

The rest of the paper is structured as follows: the work related to multilingual text classification model is covered in Section 2; we describe the overall structure and principles of our model in Section 3; we give the experimental data and evaluation indexes, and compare and analyze the results of different detection models on the same dataset in Section 4; Section 5 summarizes the work and proposes a future direction.

2. Related Work

2.1. Monolingual Toxic Text Detection

Monolingual Toxicity detection has been extensively studied in the literature. Most studies are conducted in English datasets [1,15,33], while studies in other languages, such as Korean [3], Hindi [2], Spanish [5], and Russian [4], are also investigated. The problem can be either formulated as binary [2–5] or multi-class [6–8] 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 six classes including toxic, severe toxic, obscene, threat, insult, and identity hate. Additionally, Waseem and Hovy [6] developed a dataset with three classes, including racist, sexist, and neutral, for offensive language detection.

2.2. Multilingual Toxic Text Detection

The language gap makes the monolingual detection model not applicable to other languages [31]. Existing studies strive to bridge the language gap with various methods. One way to deal with texts in multiple languages is to directly translate multiple source languages into a single target language and then extract grammatical and semantic features from the translated corpus before text classification [34]. A shortcoming of this approach is that the translation of multiple source languages introduces excessive data noise, and these translation errors propagate to downward learning modules. Ethem et al. [30] train a model on an English corpus and translate text comments in other languages into English for classification. Wang et al. [35] incorporate bilingual affective features into machine translation and utilize a text label propagation algorithm to attain text classification. This approach improves the F1 value in each class compared with the classification model, which only considers monolingual texts.

Multilingual toxic text detection is also challenged by the lack of abundant 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 [36] and transfer learning [22,32] 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 corpora to pre-train a language model, which is fine-tuned on a smaller labeled training set. In this study, we adopt both methods to minimize the impact of the low resource issue.

2.3. Toxicity Detection Models

2.3.1. Conventional Learning Models

Conventional machine learning models (e.g., Support Vector Machine (SVM) [11], Random Forest [12], Naive Bayes [13], Logistic Regression (LR) [14]) rely on manual feature engineering and are incapable of capturing the contextual features in toxic comments. However, even though deep learning models have become more popular, traditional models have not vanished. Several studies [39,40] show that LR performs better in the low resource setting, while the power of deep learning can only be fully released with enough annotated training data. Additionally, conventional feature-based methods preserve a model's interpretability to some degree, which is not presented by most deep learning models.

2.3.2. Deep Learning Models

Toxic comment detection is commonly treated as a many-to-one sequence modeling problem, which is addressed by a recurrent neural network (RNN) [41]. Two RNN variants, long short-term memory (LSTM) [17], and gated recurrent unit (GRU) [18] are also popular choices due to their ability to overcome the gradient explosion and vanishing problem that exist in the vanilla RNN model. Bi-LSTM and Bi-GRU [42] are known for their potential to capture backward and forward contextual features. BERT [43], built on the Transformer model [37], adopts the multi-headed attention mechanism which allows the model to learn how each word in a sentence is attended by every other word to enrich contextual understanding. BERT has demonstrated SOTA performance in numerous NLP tasks [43], including toxic comment detection [44].

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 [46], word [15], or even sentence-level [47]. 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 [46] 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. [47] propose a CNN-based model that can learn not only the word-level contextual information but also the feature information at the global sentence-level granularity. The model outperforms the word-level encoding model in terms of model perplexity on English and other language datasets.

Recent studies have also explored the effect of utilizing external knowledge that allows a detection model to integrate handcrafted domain keywords into training. Pamungkas et al. [48] proposed a joint model, based on Facebook's Multilingual Unsupervised and Supervised Embeddings (MUSE) [49], which leveraged Hurtlex [50], a multilingual lexicon of toxic words, to help detect hate speech. The results showed that the injection of domain knowledge could boost the performance, especially in detecting positive samples (i.e., toxic texts).

2.3.3. Transfer Learning via Masked Language Models

Using a pre-trained model to encode contextual information embedded in the raw data helps the model better understand the meaning of a character/word/sentence within its context. As a breakthrough language model, BERT employs MLM that allows self-supervised training on large text corpora. MBERT [43] is the pre-trained model of BERT on corpora in multiple languages. Despite the power of BERT and MBERT, Liu et al. [51] show that BERT is undertuned and present RoBERTa, an optimized version of BERT, by carefully setting the training parameters, pushing the SOTA on various tasks. XLM [52] enhances RoBERTa by adding TLM into the pre-training. The same authors of XLM also present 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 [52]. Several of the latest studies have also employed XLM-R for toxic text analysis [22,32] and obtained SOTA performance. In this paper, we adopt the MBERT and XLM-R pre-trained models and conduct fine-tuning with a fusion strategy, which shows performance elevation.

2.3.4. Model Fusion

Model fusion is a commonly used trick to improve prediction performance by aggregating several existing classifiers [53]. Gao and Huang [54] propose fusing LR and Neural Network classifiers. Zimmerman [55] investigated a fusion of models with different hyper-parameters. In this study, we apply fusion on different loss functions as well as different pre-trained models, 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.

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

3. Multilingual Toxic Text Detection Model Based on Multi-Model Fusion

The model, based on a large-scale multilingual pre-training model, uses multiple training methods and multiple pre-training model fusion methods to uncover contextual features that can distinguish toxic texts from ordinary ones. The pre-training model fusion method can characterize different expressions of the same text and improves the system's generalization ability to recognize multilingual text categories. The overall neural architecture of the proposed fusion model is depicted in Figure 1. There are three major phases for the proposed learning pipeline, including text pre-processing, model pre-training, and detection. The pre-processing module transforms a word of an input sentence into a digital form required by the subsequent module. The pre-training model precipitates the common features of the multilingual text detection task. Finally, the detection layer outputs a binary result: non-toxic vs. toxic. The design details of the proposed learning pipeline are covered in the following subsections.

3.1. Text Pre-Processing

In this stage, the original English text corpus is converted to a vector that is easily processed by a learning algorithm. The process preserves as much of the semantic logic, grammatical structure, and intrinsic linguistic information of the input text as possible, while reducing the loss of information. The process mainly consists of the following seven steps.

3.1.1. Translation

We translate the English comments in the training set into the target languages via an off-the-shelf software translator. In other words, each comment in English is translated into six versions of the comment in the six target languages. Modern software translators based on deep learning can preserve most linguistic features for the target languages, although certain translation inaccuracy brings noise to the training and affect the performance of the downstream task. After translation, the augmented training set is seven times the original training set in size. In addition, to facilitate model fusion, we randomly draw 15% of samples from the augmented training set to form a fusion validation set, which is used to determine the fusion weights of multiple models. The rest 75% of samples are used for fine-tuning.

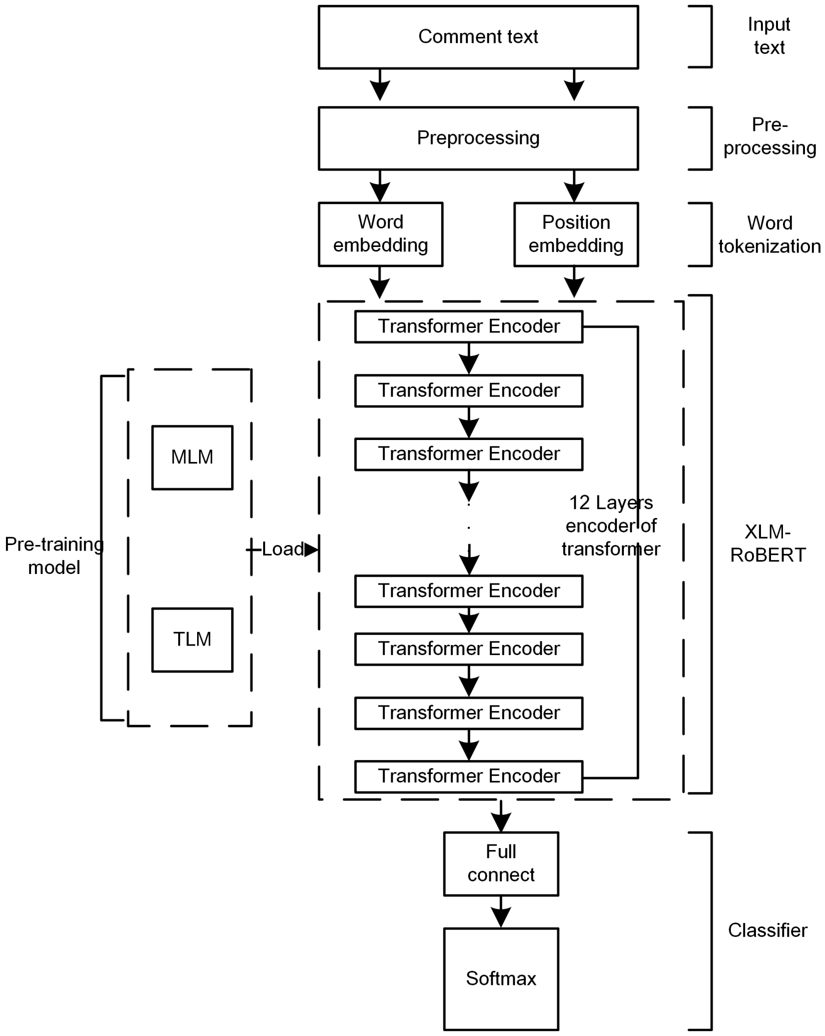


Figure 1. The Proposed Neural Architecture for Multilingual Toxic Text Detection.

3.1.2. Word Segmentation

Sentences are tokenized to separate words by white spaces. All punctuation marks are also kept since they could carry semantic information. This step is common in various NLP models that treat text classification as a sequence modeling task, and word tokens are processed sequentially.

3.1.3. Text Purification

The original text corpus contains many redundant/noisy text information like web-page characters and repetitive tone words, causing problems such as high RAM usage, long training time, and hard convergence of training. To improve the dataset quality and make training more efficient, we purify the text corpus by eliminating the useless tokens. Specifically, we build a seed lexicon and apply a regular expression to perform a precise and fuzzy matching to filter out the noisy tokens.

3.1.4. Sample Equilibrium

There is a big-gap ratio of 1:10 between the positive (toxic) and negative (non-toxic) samples in the dataset, which is not conducive to the learning of positive sample features. To this end, we performed random negative sampling on the negative samples, adjusting the training ratio of positive and negative samples to a relatively appropriate ratio of 1:9, which was an optimal ratio determined by empirical results. Specifically, after testing the ratio options from 1:9 up to 1:1, we found that the ratio of 1:9 yielded the best performance. Since the number of positive samples does not change, a higher ratio means that more negative samples are left out and do not contribute to the training. The ratio of 1:9 was shown to achieve a decent trade-off between rebalancing sample distribution and maintaining sufficient training data. An imbalanced dataset is commonly seen in toxicity analysis, making accuracy unsuitable for performance evaluation. We thus focus on F1, which is more indicative given skewed data distribution.

3.1.5. Lexicon Solidification

This step aims to build a comprehensive lexicon for the task. First, the lexicons of the MBERT and XLM-R pre-trained models are combined to obtain a base lexicon. Second, the word tokens that appear in the Jigsaw dataset but do not exist in the base lexicon are also added to form the final lexicon. This step ensures that each unique token is stored in the final lexicon.

3.1.6. Word Embedding

Word tokens are converted to vectors of dimension d before being fed into the neural network. In particular, an input sentence is scanned, and for each token, the algorithm utilizes its index in the solidified lexicon obtained from the previous step to generate a vector.

3.1.7. Position Embedding

Unlike the recurrent neural network, the multi-headed attention network used in our model of this study is unable to directly encode the positional information of tokens; therefore, the module adds position vectors to preserve the order of words in a sentence. Specifically, the position embedding PE can be calculated as follows

$$PE(pos_{2i}) = sin\left(\frac{pos}{10{,}000^{\frac{2i}{d}}}\right) \tag{1}$$

$$PE(pos_{2i+1}) = cos\left(\frac{pos}{10{,}000^{\frac{2i}{d}}}\right) \tag{2}$$

where pos denotes a token's position, d denotes the embedding dimension, and $2i$ and $2i + 1$ refer to the indices of the embedding vector. The design ensures that the generated embeddings are distinct for all positions with a given dimension d.

3.2. Pre-Training and Fine-Tuning Multilingual Models

In this phase, the MLM [37] and TLM [38] pre-training methods are utilized. In particular, MBERT [37] is pre-trained using MLM, and XLM-R [38] is pre-trained via both MLM and TLM. We then fine-tune the pre-trained two models on the training set for toxicity detection.

3.2.1. The BERT Language Model

Both MBERT and XLM-R adopt the base neural architecture of BERT, which consists of twelve layers of the Transformer encoder, as shown in Figure 1. Equipped with the multi-headed attention layer, BERT generates contextualized embeddings for the downstream tasks.

3.2.2. Pre-Training with Masking-Based Language Modeling

The MLM training stitches together the text in sequential order, then randomly masks part of the vocabulary, replaces the original vocabulary with [Mask] characters, and uses them as the target vocabulary to predict the corresponding loss value. The MLM training is shown in Figure 2 below.

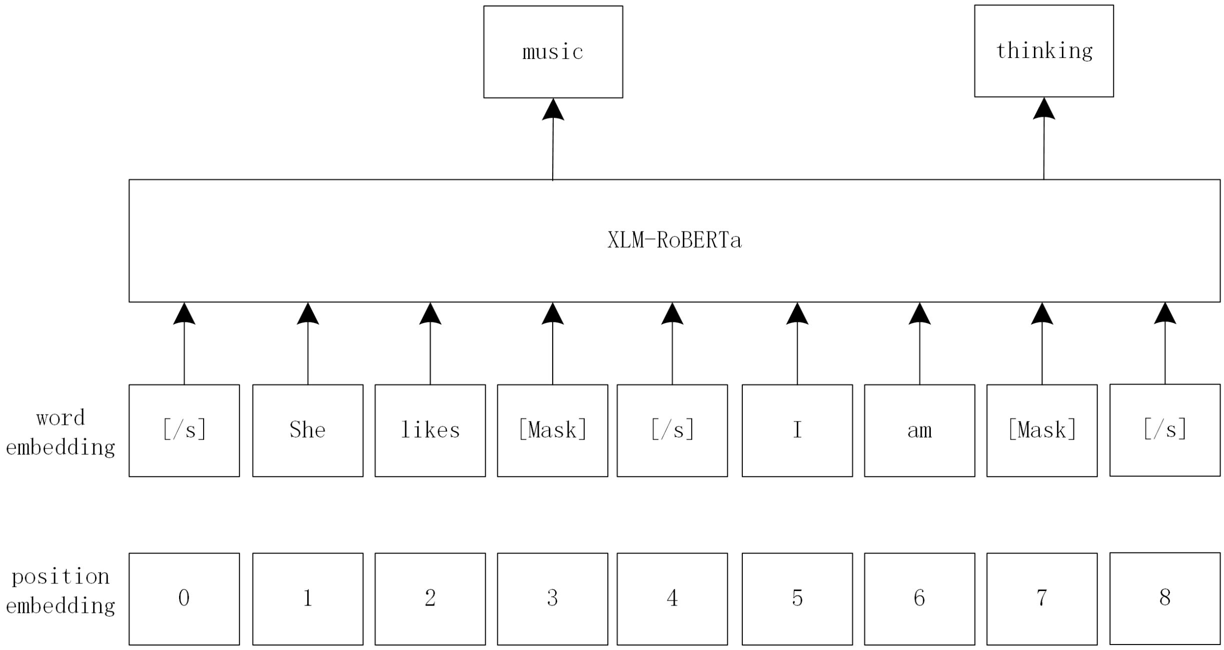


Figure 2. MLM training.

3.2.3. Pre-Training with Translation-Based Language Modeling

As shown in Figure 3, the TLM training inherits from the MLM, which splices the English training corpus with the corresponding translated corpus of other target languages and randomly masks the words and replaces them with [Mask] characters with a certain probability. The location embedding information of the target language sentence is reset to facilitate alignment with English sentences. The TLM method aims to learn the word-level mapping relationship between English and other target languages, which improves prediction performance.

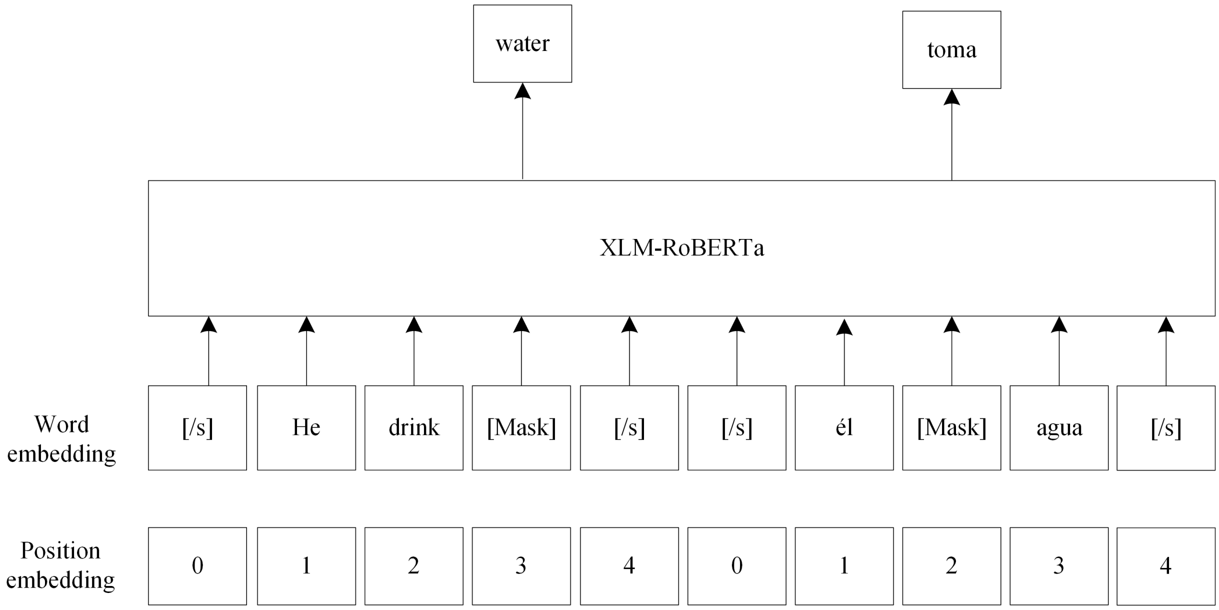


Figure 3. TLM training.

3.2.4. Fine-Tuning

The pre-trained MBERT and XLM-R models are fine-tuned on the augmented training set. Specifically, a detection layer consisting of two fully connected layers and a softmax layer is added on top of the BERT neural network. During fine-tuning, the parameters of the pre-trained models are frozen, while the parameters of the detection layer are updated by an optimization algorithm to minimize a specified loss function. The outcome of this phase is a predictive model that can perform multilingual toxicity detection. It is noted that the models obtained in this phase are fused in the next phase to achieve a better performance.

3.3. Model Fusion

We adopt two fusion strategies, including a fusion of different loss functions and a fusion of different pre-training models.

3.3.1. A Fusion of Loss Functions

Since the toxic comment detection task considered in this study is a typical binary classification problem, it is intuitive to utilize the binary cross-entropy (BCE) loss, as shown in Equation (3).

$$L_{\text{BCE}} = -\frac{1}{m}\sum_{i=1}^{m}\Big(y_{(i)}\log(\hat{y}_{(i)}) + (1-y_{(i)})\log(1-\hat{y}_{(i)})\Big) \tag{3}$$

in which m is the training set size, and $y_{(i)}$ and $\hat{y}_{(i)}$ denote the ground truth and the predicted class for the ith sample in the dataset, respectively.

Meanwhile, considering the imbalanced sample distribution, this study also employs the Focal Loss (FL), defined in Equation (4)

$$L_{\text{FL}} = -\frac{1}{m}\sum_{i=1}^{m}\Big(y_{(i)}\alpha(1-\hat{y}_{(i)})^{\gamma}\log(\hat{y}_{(i)}) + (1-y_{(i)})(1-\alpha)\hat{y}_{(i)}^{\gamma}\log(1-\hat{y}_{(i)})\Big) \tag{4}$$

where γ is a coefficient that controls the curve shape of the focal loss function. Using Focal Loss with $\gamma > 1$ reduces the loss for well-classified examples (i.e., with a prediction probability larger than 0.5), and increases loss for hard-to-classify examples (i.e., with a prediction probability less than 0.5). Therefore, it turns the model's attention towards the rare class in case of class imbalance. On the other hand, a lower α means that we tend to give a small weight to the dominating or common class and a high weight to the rare class.

By fusing the focal loss and the BCE loss in a certain ratio, we obtain Equation (5), in which β_1 and β_2 specify the fusion weights.

$$L = \beta_1 L_{\text{BCE}} + \beta_2 L_{\text{FL}} \tag{5}$$

3.3.2. A Fusion of Multilingual Models

Model fusion is a common strategy in machine learning to improve model performance and robustness. In this study, we associate the MBERT and XLM-R pre-trained models with BCE, FL, and a fusion loss (given in Equation (5)) to obtain six model combinations. After fine-tuning, six multilingual models based on different pre-training models and loss functions are obtained. The F1 scores of the six models on the fusion validation set are utilized as fusion weights, and the intuition is that the model with higher F1 presents better performance and should have more votes in the fusion model. A detailed description of the evaluated models is provided in Section 4.3.

4. Experimental Results and Analysis

This section describes the components of our experiment with an overall goal of evaluating the performance of the proposed model by comparing it to a series of other model options.

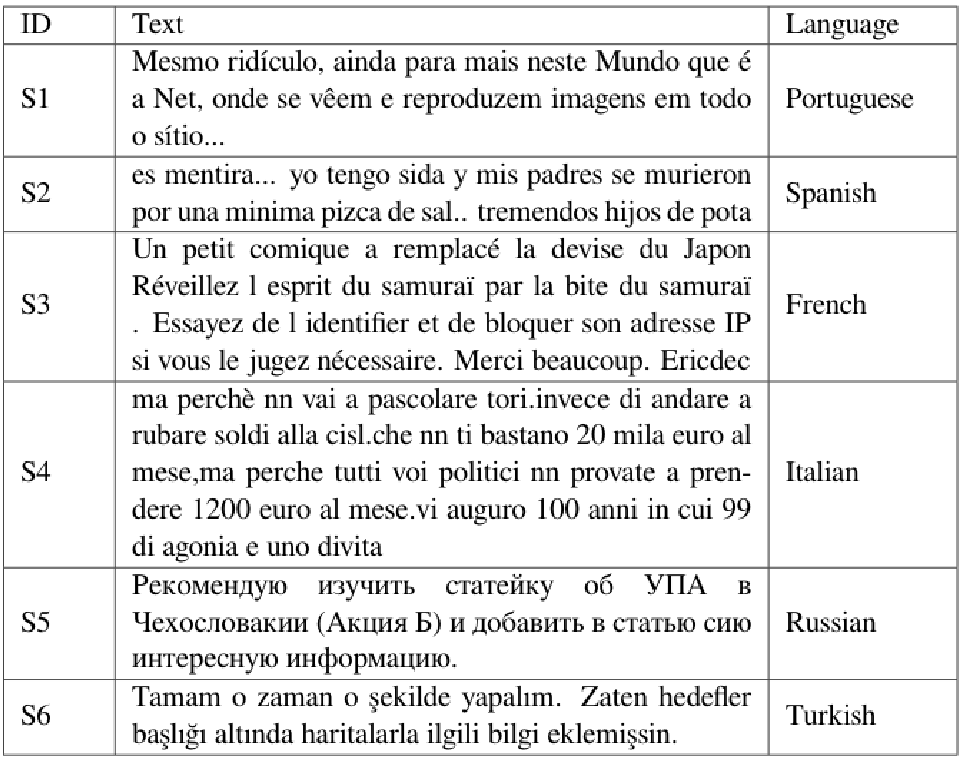
4.1. Dataset

We utilize the Jigsaw Multilingual Toxic Comment dataset, which was created by the Conversation AI team. The dataset was used for a Kaggle competition in July 2020.

The data were collected from Civil Comments and Wikipedia, which contain page comment messages from 63 M users and articles, which were manually annotated via crowdsourcing between 2004 and 2015. Specifically, the dataset is divided into three sections:

- The original training set contains 435,775 labeled samples, all in English. After translation, we obtain a total of 3,050,425 labeled samples in the seven languages considered in this task, and fine-tuning is conducted on the augmented training set;
- The validation set contains 8000 labeled samples in three languages, including 3000 Turkish samples, 2000 Italian samples, and 2000 Spanish samples;
- The test set consists of 63,812 unlabeled samples in six languages, including 8438 Spanish samples, 10,920 French samples, 8494 Italian samples, 11,012 Portuguese samples, 10,948 Russian samples, and 14,000 Turkish samples.

Figure 4 shows six comment samples in different languages in the test set.



ID	Text	Language
S1	Mesmo ridículo, ainda para mais neste Mundo que é a Net, onde se vêem e reproduzem imagens em todo o sítio...	Portuguese
S2	es mentira... yo tengo sida y mis padres se murieron por una minima pizca de sal.. tremendos hijos de pota	Spanish
S3	Un petit comique a remplacé la devise du Japon Réveillez l esprit du samuraï par la bite du samuraï . Essayez de l identifier et de bloquer son adresse IP si vous le jugez nécessaire. Merci beaucoup. Ericdec	French
S4	ma perchè nn vai a pascolare tori.invece di andare a rubare soldi alla cisl.che nn ti bastano 20 mila euro al mese,ma perche tutti voi politici nn provate a prendere 1200 euro al mese.vi auguro 100 anni in cui 99 di agonia e uno divita	Italian
S5	Рекомендую изучить статейку об УПА в Чехословакии (Акция Б) и добавить в статью сию интересную информацию.	Russian
S6	Tamam o zaman o şekilde yapalım. Zaten hedefler başlığı altında haritalarla ilgili bilgi eklemişsin.	Turkish

Figure 4. Samples in the test set.

4.2. Evaluation Metrics

In this paper, we use the same evaluation criteria as other multilingual classification tasks, including accuracy, the macro-average precision, the macro-average recall, and the macro-average F1. The latter three are given as follows:

$$P_{ma} = \frac{\sum_{i=1}^{N} P_i}{N} \tag{6}$$

$$R_{ma} = \frac{\sum_{i=1}^{N} R_i}{N} \tag{7}$$

$$F1_{ma} = \frac{\sum_{i=1}^{N} \frac{2\times P_i \times R_i}{P_i + R_i}}{N} \tag{8}$$

where N is the number of categories (two in our case), P_i and R_i are the precision and recall of category i, respectively. P_{ma}, R_{ma}, and $F1_{ma}$ are the macro averages of precision, recall, and F1 score, respectively. Due to the imbalanced sample distribution, the F1 score is a more indicative metric and can reflect the true performance of a model, compared with accuracy.

4.3. Models

In order to verify the validity of the pre-training-based multilingual fusion model proposed in this paper, we designed the following nine sets of comparison experiments:

1. MBERT_BCE: Using MBERT as a pre-training model and BCE loss as the loss function;
2. MBERT_FOCAL: Using MBERT as a pre-training model and focal loss as the loss function;
3. MBERT_MIX: Using MBERT as a pre-training model and the mixed BCE and focal loss at a ratio of 1:1;
4. XLM-R_BCE: Using XLM-R as a pre-training model and BCE loss as the loss function;
5. XLM-R_FOCAL: Using XLM-R as a pre-training model and focal loss as the loss function;
6. XLM-R_MIX: Using XLM-R as a pre-training model and BCE loss and focal loss at a ratio of 1:1 as the loss function;
7. Model-Fusion-1: The two models 3 and 6 are fused with the validation $F1_{ma}$ values used as the fusion weights;
8. Model-Fusion-2: The four models of 2, 3, 5, and 6 are fused with the validation $F1_{ma}$ values used as the fusion weights;
9. Model-Fusion-3: The six models 1-6 are fused with the validation $F1_{ma}$ values used as the fusion weights.

4.4. Benchmarks

We choose Logistic Regression [14], CNN+fastText [45], Bi-LSTM [42], Bi-GRU [42] as the four benchmark models. Logistic Regression represents the conventional learning methods that need manual feature engineering. CNN+fastText represents the CNN-based deep learning models for this task, while Bi-LSTM and Bi-GRU represent RNN-based deep learning models. Due to their superior performance in toxic text detection tasks [22,32], MBERT_FOCAL and XLM-R_FOCAL are used as the SOTA models. By comparing these with the chosen benchmark and SOTA models, we demonstrate the effectiveness of the proposed model fusion strategy.

4.5. Experimental Environment and Parameter Settings

The studied models are implemented using Python 3.6 with Tensorflow v2.2.0 as the deep learning framework and transformers v2.11.0 as the core module to build the neural architecture for MBERT and XLM-R. The models are trained on the Tensor Processing Unit (TPU) servers of the Google Cloud platform. Table 3 shows the training parameters.

Table 3. Model parameter setting.

Parameter Name	Parameter Value
Number of fully Connected layers	2
Number of hidden cells of fully connected layer	768 × 2
Learning rate	1×10^{-5}
Word vector dimension	768
Training batch size	16
XLM-R input sentence length	224
Input sentence length	512

As for focal loss, the two parameters γ and α are set to 2 and 0.2, respectively. The choices of values for γ and α are based on empirical results. In addition, when fusing the loss functions, BCE loss and focal loss are fused at a ratio of 1:1.

4.6. Experimental Results and Analysis

We report the performance results in Table 4, where a total of thirteen models are trained and evaluated.

Table 4. Performance evaluation on the test set.

Model	Accuracy	Recall	Precision	F1
Logistic Regression [14]	0.8584	0.7874	0.7443	0.7594
CNN+fastText [45]	0.8787	0.8097	0.7587	0.7822
Bi-LSTM [42]	0.8656	0.7939	0.7736	0.7828
Bi-GRU [42]	0.8912	0.8586	0.8015	0.8249
XLM-R_BCE	0.9376	**0.8698** *	0.8129	0.8381
XLM-R_FOCAL (SOTA)	0.9450	0.8232	0.8529	0.8372
XLM-R_MIX	0.9411	0.8514	0.8276	0.8389
MBERT_BCE	0.9094	0.8605	0.7505	0.7907
MBERT_FOCAL (SOTA)	0.9420	0.7381	**0.9035**	0.7943
MBERT_MIX	0.9408	0.8479	0.8274	0.8373
Model-Fusion-1	0.9437	0.8539	0.8360	0.8446
Model-Fusion-2	**0.9469**	0.8344	0.8560	0.8448
Model-Fusion-3	0.9437	0.8548	0.8357	**0.8449**

* The highest score of each metric is marked in bold-face.

The first section of Table 4 displays the performance of the four benchmark models. It is observed that the Logistic Regression model performs the worst, with an accuracy of 0.8584 and an F1 of 0.7594, which does not show any strength compared with its competitors. The rationale behind its weak performance is that traditional learning models like Logistic Regression are unable to understand the semantic and contextual meaning of words, which is crucial in most NLP tasks. On the other hand, CNN+fastText posts a better accuracy (0.8787 vs. 0.8656) and a similar F1 score (0.7822 vs. 0.7828), compared with Bi-LSTM. Bi-GRU performs the best among the four, with an accuracy of 0.8912 and an F1 of 0.8249. According to [57], GRUs train faster and perform better than LSTMs on fewer training data with shorter sequences. Since social texts are generally short and concise, this result can be justified.

The second section of Table 4 shows the results of models 1–6 in Section 4.3, in which XLM-R_FOCAL and MBERT_FOCAL represent the SOTA performance. We find that compared with BCE, Focal loss does help improve the accuracy for both XLM-R and MBERT. However, Focal loss does not benefit the F1 by a significant margin. It is observed that when replacing BCE with the Focal loss, recall drops while precision rises, meaning that using Focal loss helps reduce the false alarms but misses more toxic samples that should have been detected. It is also noted that a mixing of BCE and Focal loss can further elevate the F1 score, especially for MBERT. The result demonstrates that the loss function fusion strategy can effectively combine the strength of both loss functions and lead to a better F1.

The third section of Table 4 presents the results of the fused models. It can be seen that the fusion of different pre-training models does lead to an improvement in both accuracy and F1. After fusing MBERT_MIX and XLM-R_MIX to obtain Model-Fusion-1, the F1 value showed an increase of 0.5% compared with the single XLM-R_MIX model. In terms of accuracy, the Model-Fusion-2 with four fused models reached 94.69%. As the number of fusion models increased, the average macro F1 value also improved steadily by a small margin, with Model-Fusion-2 improving by 0.02% over Model-Fusion-1, and Model-Fusion-3 improving by 0.01% over Model-Fusion-2, reaching 84.49%. Compared with the SOTA models XLM-R_FOCAL and MBERT_FOCAL, which did not use loss function fusion and multi-model fusion, the accuracy improved by 0.19% and 0.49%, respectively, and the average macro F1 value improved by 0.76% and 5.05% respectively. Meanwhile, compared with other models, Model-Fusion-3 could find a balance between precision and recall, and had stable prediction performance on text classification task.

It is also observed that, although the fused model presented the best F1 scores, neither recall nor precision achieved a new high. This result, from another angle, demonstrates the superiority and practicability of the fused models, which showed the smallest recall-precision gaps (less than 2.5%), indicating a decent balance of the two metrics. On the other hand, XLM-R_BCE, the one with the highest recall, posted a gap of 5.69%; MBERT_FOCAL, the one with the highest precision, reported a gap of 16.54%. The larger the gap, the less practical for a model, since it is more focused on optimizing either recall or precision, but not both. In a realistic setting, a model should strive to optimize F1 and narrow the recall-precision gap so that the detector does not throw many false alarms nor miss many toxic samples.

The above experiments show that the introduction of the fusion method based on different loss functions and pre-training models achieved better experimental results and verified the effectiveness of the method on the multilingual toxic text detection task with imbalanced sample distribution.

5. Summary and Prospect

In this paper, we propose a multilingual toxic text detection method based on pre-training model fusion under imbalanced sample distribution. Through text pre-processing, part of the English training corpus is translated into multiple languages, and the proportion of multiple languages in the training set is appropriately expanded, and the input data that are acceptable to the BERT model are then obtained through sample balancing, lexicon solidification, text vectorization, and position vectorization. With the fusion methods based on different loss functions and different pre-training models, the above-mentioned vectorized data are fed into two pre-training models of MBERT and XLM-R, and the final prediction model is obtained. The experimental results show that the introduction of the fusion method based on different loss functions effectively solved the problem of imbalance in precision and recall due to imbalanced samples; while the introduction of the fusion method based on different pre-trained models drew in the respective advantages of the different models, explored the effective features of the multilingual corpus, and achieved a significant improvement in F1 compared with the models without fusion.

The proposed toxic language detection approach can be used to build an automated content moderation system, which can be adopted by any globalized social media platform such as Twitter, Facebook, Instagram, Snapchat, and Discord. These platforms generate a huge amount of social content in a multilingual environment and are in urgent need of a robust moderation system to prevent abusive online language from spreading.

The fusion strategy can be further improved in two aspects. First, the 1:1 fusion ratio of the BCE and focal loss can be adjusted through tuning or auto-learning to determine an optimal fusion ratio. Second, in the given training corpus, there are small number of multilingual corpus other than English, and the machine translation errors in the supplemental multilingual corpus are transmitted downwards, which degrades the performance to some extent. In the following research on this task, we plan to apply the sample language reconstruction method to make the text language richer and more diversified to reduce the impact of data imbalance.

Author Contributions: Conceptualization and methodology, G.S. and D.H.; software, validation, and original draft preparation, G.S.; review and editing, supervision, funding acquisition, D.H. and Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Program of China (2020AAA0108004) and the National Natural Science Foundation of China (61672127, U1936109). The funding agency has no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data Availability Statement: the Jigsaw Multilingual Toxic Comment Classification dataset supporting the conclusions of this article are available at https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification (accessed on 10 March 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. van Aken, B.; Risch, J.; Krestel, R.; Löser, A. Challenges for toxic comment classification: An in-depth error analysis. *arXiv* **2018**, arXiv:1809.07572.
2. Bashar, M.A.; Nayak, R. QutNocturnal@ HASOC'19: CNN for hate speech and offensive content identification in Hindi language. *arXiv* **2020**, arXiv:2008.12448.
3. Moon, J.; Cho, W.I.; Lee, J. BEEP! Korean Corpus of Online News Comments for Toxic Speech Detection. *arXiv* **2020**, arXiv:2005.12503.
4. Zueva, N.; Kabirova, M.; Kalaidin, P. Reducing Unintended Identity Bias in Russian Hate Speech Detection. *arXiv* **2020**, arXiv:2010.11666.
5. Plaza-del Arco, F.M.; Molina-González, M.D.; Ureña-López, L.A.; Martín-Valdivia, M.T. Comparing pre-trained language models for Spanish hate speech detection. *Expert Syst. Appl.* **2021**, *166*, 114120. [CrossRef]
6. Waseem, Z.; Hovy, D. Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In Proceedings of the NAACL Student Research Workshop, Berlin, Germany, 7–12 August 2016; pp. 88–93.
7. Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. Automated hate speech detection and the problem of offensive language. In Proceedings of the International AAAI Conference on Web and Social Media, Montréal, QC, Canada, 15–18 May 2017; Volume 11.
8. Sharma, S.; Agrawal, S.; Shrivastava, M. Degree based classification of harmful speech using twitter data. *arXiv* **2018**, arXiv:1806.04197.
9. Salminen, J.; Almerekhi, H.; Kamel, A.M.; Jung, S.G.; Jansen, B.J. Online hate ratings vary by extremes: A statistical analysis. In Proceedings of the 2019 Conference on Human Information Interaction and Retrieval, Glasgow, UK, 10–14 March 2019; pp. 213–217.
10. Kajla, H.; Hooda, J.; Saini, G. Classification of Online Toxic Comments Using Machine Learning Algorithms. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; pp. 1119–1123.
11. Greevy, E.; Smeaton, A.F. Classifying racist texts using a support vector machine. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 25–29 July 2004; pp. 468–469.
12. Alfina, I.; Mulia, R.; Fanany, M.I.; Ekanata, Y. Hate speech detection in the Indonesian language: A dataset and preliminary study. In Proceedings of the 2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Jakarta, Indonesia, 28–29 October 2017; pp. 233–238.
13. Kwok, I.; Wang, Y. Locate the hate: Detecting tweets against blacks. In Proceedings of the AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; Volume 27.
14. Saif, M.A.; Medvedev, A.N.; Medvedev, M.A.; Atanasova, T. Classification of online toxic comments using the logistic regression and neural networks models. In *AIP Conference Proceedings*; AIP Publishing LLC.: New York, NY, USA, 2018; Volume 2048, p. 060011.
15. Georgakopoulos, S.V.; Tasoulis, S.K.; Vrahatis, A.G.; Plagianakos, V.P. Convolutional neural networks for toxic comment classification. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence, Patras, Greece, 9–12 July 2018; pp. 1–6.
16. Jubaer, A.; Sayem, A.; Rahman, M.A. Bangla toxic comment classification (machine learning and deep learning approach). In Proceedings of the 2019 8th International Conference System Modeling and Advancement in Research Trends (SMART), Moradabad, India, 22–23 November 2019; pp. 62–66.
17. Dubey, K.; Nair, R.; Khan, M.U.; Shaikh, S. Toxic Comment Detection using LSTM. In Proceedings of the 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC), Bengaluru, India, 11–12 December 2020; pp. 1–8.
18. Mahajan, A.; Shah, D.; Jafar, G. Explainable AI Approach towards Toxic Comment Classification. *EasyChair Preprint*, 26 February 2020.
19. Halim, Z.; Waqar, M.; Tahir, M. A machine learning-based investigation utilizing the in-text features for the identification of dominant emotion in an email. *Knowl. Based Syst.* **2020**, *208*, 106443. [CrossRef]
20. Jia, X.; Deng, Z.; Min, F.; Liu, D. Three-way decisions based feature fusion for Chinese irony detection. *Int. J. Approx. Reason.* **2019**, *113*, 324–335. [CrossRef]
21. Tzogka, C.; Passalis, N.; Iosifidis, A.; Gabbouj, M.; Tefas, A. Less Is More: Deep Learning Using Subjective Annotations For Sentiment Analysis From Social Media. In Proceedings of the 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), Pittsburgh, PA, USA, 13–16 October 2019; pp. 1–6.
22. Ranasinghe, T.; Zampieri, M. MUDES: Multilingual Detection of Offensive Spans. *arXiv* **2021**, arXiv:2102.09665.
23. Ranasinghe, T.; Hettiarachchi, H. BRUMS at SemEval-2020 task 12: Transformer based multilingual offensive language identification in social media. *arXiv* **2020**, arXiv:2010.06278.

24. Becker, K.; Moreira, V.P.; dos Santos, A.G. Multilingual emotion classification using supervised learning: Comparative experiments. *Inf. Process. Manag.* **2017**, *53*, 684–704. [CrossRef]
25. Ousidhoum, N.; Lin, Z.; Zhang, H.; Song, Y.; Yeung, D.Y. Multilingual and multi-aspect hate speech analysis. *arXiv* **2019**, arXiv:1908.11049.
26. Corazza, M.; Menini, S.; Cabrio, E.; Tonelli, S.; Villata, S. A multilingual evaluation for online hate speech detection. *ACM Trans. Internet Technol.* **2020**, *20*, 1–22. [CrossRef]
27. Pamungkas, E.W.; Basile, V.; Patti, V. Misogyny detection in twitter: A multilingual and cross-domain study. *Inf. Process. Manag.* **2020**, *57*, 102360. [CrossRef]
28. Rasooli, M.S.; Farra, N.; Radeva, A.; Yu, T.; McKeown, K. Cross-lingual sentiment transfer with limited resources. *Mach. Transl.* **2018**, *32*, 143–165. [CrossRef]
29. Dong, X.; De Melo, G. Cross-lingual propagation for deep sentiment analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
30. Can, E.F.; Ezen-Can, A.; Can, F. Multilingual sentiment analysis: An RNN-based framework for limited data. *arXiv* **2018**, arXiv:1806.04511.
31. Li, X.; Li, Z.; Sheng, J.; Slamu, W. Low-Resource Text Classification via Cross-Lingual Language Model Fine-Tuning. In *China National Conference on Chinese Computational Linguistics*; Springer: Cham, Switzerland, 2020; pp. 231–246.
32. Roy, S.G.; Narayan, U.; Raha, T.; Abid, Z.; Varma, V. Leveraging Multilingual Transformers for Hate Speech Detection. *arXiv* **2021**, arXiv:2101.03207.
33. Mohammad, F. Is preprocessing of text really worth your time for online comment classification? *arXiv* **2018**, arXiv:1806.02908.
34. Kalouli, A.L.; Kaiser, K.; Hautli, A.; Kaiser, G.A.; Butt, M. A multilingual approach to question classification. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
35. Wang, Z.; Lee, S.; Li, S.; Zhou, G. Emotion detection in code-switching texts via bilingual and sentimental information. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Beijing, China, 26–31 July 2015; pp. 763–768.
36. Ibrahim, M.; Torki, M.; El-Makky, N. Imbalanced toxic comments classification using data augmentation and deep learning. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 875–878.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
38. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised cross-lingual representation learning at scale. *arXiv* **2019**, arXiv:1911.02116.
39. Huang, X.; Xing, L.; Dernoncourt, F.; Paul, M.J. Multilingual Twitter corpus and baselines for evaluating demographic bias in hate speech recognition. *arXiv* **2020**, arXiv:2002.10361.
40. Aluru, S.S.; Mathew, B.; Saha, P.; Mukherjee, A. Deep learning models for multilingual hate speech detection. *arXiv* **2020**, arXiv:2004.06465.
41. Mikolov, T.; Karafiát, M.; Burget, L.; Černockỳ, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Makuhari, Japan, 26–30 September 2010.
42. Ghosh, S.; Kumar, S.; Lepcha, S.; Jain, S.S. Toxic Text Classification. In *Data Science and Security*; Springer: Singapore, 2021; pp. 251–260.
43. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
44. Mozafari, M.; Farahbakhsh, R.; Crespi, N. A BERT-based transfer learning approach for hate speech detection in online social media. In *International Conference on Complex Networks and Their Applications*; Springer: Cham, Switzerland, 2019; pp. 928–940.
45. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:1607.01759.
46. Kim, Y.; Jernite, Y.; Sontag, D.; Rush, A. Character-aware neural language models. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
47. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.
48. Pamungkas, E.W.; Basile, V.; Patti, V. A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. *Inf. Process. Manag.* **2021**, *58*, 102544. [CrossRef]
49. Conneau, A.; Lample, G.; Ranzato, M.; Denoyer, L.; Jégou, H. Word translation without parallel data. *arXiv* **2017**, arXiv:1710.04087.
50. Bassignana, E.; Basile, V.; Patti, V. Hurtlex: A multilingual lexicon of words to hurt. In Proceedings of the 5th Italian Conference on Computational Linguistics, CLiC-it 2018. CEUR-WS, Torino, Italy, 10–12 December 2018; Volume 2253, pp. 1–6.
51. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
52. Lample, G.; Conneau, A. Cross-lingual language model pretraining. *arXiv* **2019**, arXiv:1901.07291.
53. Burnap, P.; Williams, M.L. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy Internet* **2015**, *7*, 223–242. [CrossRef]

54. Gao, L.; Huang, R. Detecting online hate speech using context aware models. *arXiv* **2017**, arXiv:1710.07395.
55. Zimmerman, S.; Kruschwitz, U.; Fox, C. Improving hate speech detection with deep learning ensembles. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018.
56. Zhang, L.; Wu, L.; Li, S.; Wang, Z.; Zhou, G. Cross-lingual emotion classification with auxiliary and attention neural networks. In *CCF International Conference on Natural Language Processing and Chinese Computing;* Springer: Cham, Switzerland, 2018; pp. 429–441.
57. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative study of CNN and RNN for natural language processing. *arXiv* **2017**, arXiv:1702.01923.