Multilingual Toxic Comment Detection and Classification ENGR-E 533 Fall 22

Shriya Mandarapu, Akhila Sakiramolla, Karthik Shathiri
shmanda@iu.edu, asakiram@iu.edu, kashat@iu.edu
Link to Github code Here

***Abstract*—In the growing global community that is the internet, conversational toxicity is a major issue. It is important, now more than ever, to develop tools that detect and filter toxic content. Although much work is done in English, there's still much to achieve in other languages. Unavailability of toxic comments data in languages other than English proves to be a challenge.**

This paper is a comprehensive comparative analysis where we investigate how a deep learning model trained on toxic comments in English generalizes to other popular languages. We are using the dataset from a Kaggle competition hosted by Conversation-alAI/Jigsaw "Multilingual Toxic Comment Classification".

We start with LSTM and Bi-LSTM models for baseline and try multiple complex architectures such as BERT and its various versions. We found that XLM-RoBERTa performed best with 0.913 AUC score on validation set which contains 3 languages.

I. Introduction

Social media and online platforms have provided individuals with the means to put forward their thoughts and freely express opinions on various matters. Lately, there have been several instances where this freedom of speech is misused to spread hate and hurtful comments causing more harm than good. Disrespectful comments containing explicit language can be categorized into Toxic, Severely Toxic, Obscene, Threat, Insult, and Identity Hate.

To protect users from being exposed to this offensive language, companies now need to start flagging comments and filtering out content or blocking users found guilty to prevent more cases. Several Machine Learning models have been developed and deployed to filter out the unruly language and protect internet users from becoming victims to online harassment and cyberbullying.

There have been a few kaggle competitions and work done on finding best methods to classify comments into different categories, attempts at removing unintended bias that creeps into models associating names of frequently attacked identities with toxicity. Here we attempt to apply the vast research on multilingual translation to classify comments in different languages.

II. Dataset

We are using data sets from a kaggle competition hosted by Jigsaw : Multilingual Toxic Comment Classification. For the baseline model, we use a training set that contains 220K data samples (transformer based models are trained on 100K samples only), with primary input being English text comment strings collected from either Civil Comments or Wikipedia talk page edits, and binary labels against different categories of toxicity. Test and Validation dataset contains majority of non-english language comments.

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

Fig. 1. Train Data

Our validation dataset requires us to classify into either toxic or not. Hence we are only considering toxic column in our train dataset for training our model.

	id	comment_text	lang	toxic
0	0	Este usuario ni siquiera llega al rango de ...	es	0
1	1	Il testo di questa voce pare esser scopiazzato...	it	0
2	2	Vale. Sólo expongo mi pasado. Todo tiempo pasa...	es	1
3	3	Bu maddenin alt başlığı olarak uluslararası i...	tr	0
4	4	Belçika nın şehirlerinin yanında ilçe ve belde...	tr	0

Fig. 2. Validation Data

21384 or 9.4% of the samples are labelled toxic in training set. validation data set contains 8K samples, with the input being a non-English text comment string, and the label being a binary toxicity label. 1230 or 15.4% of the samples are labelled toxic.

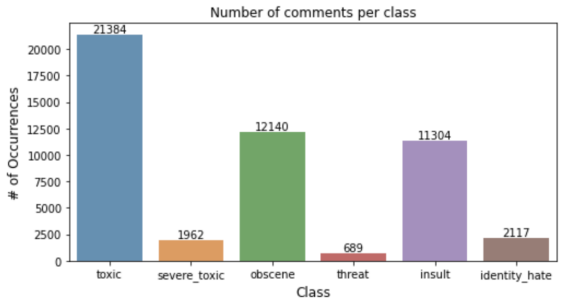

Fig. 3. Train Dataset Class Breakdown

Although train data is primarily English, validation data has equally distributed data points in 3 different languages, majorly non-English.

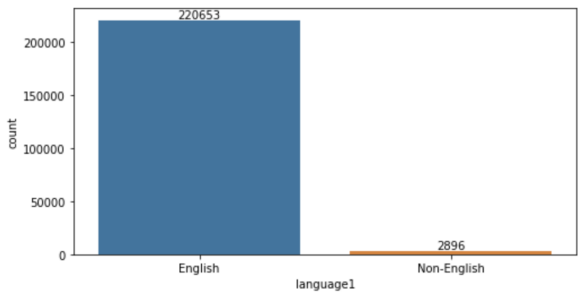

Fig. 4. Train Data Language Breakdown

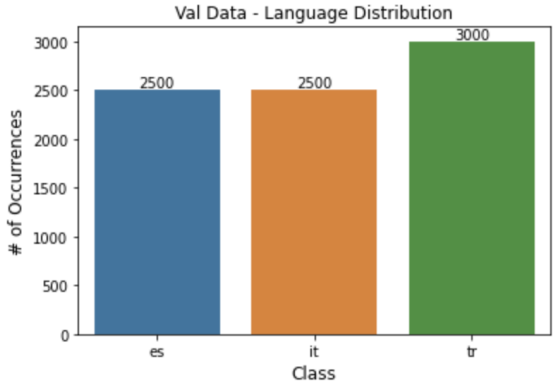

Fig. 5. Validation Data Language Breakdown

III. Approach

A. Pre Processing

In both training and validation datasets, we observed that the comments had unnecessary information like usernames, links, and punctuations and we got rid of them by performing some text preprocessing steps. In order to use the text data for modeling, we then tokenized the comment data. As the comments are of different length, we padded the sequences to have an equal length of 512 for all the comments.

For the model to be able to interpret the text, we encoded the text into features vectors and extracted the word embeddings using the pre-trained DistilBert base multilingual cased tokenizer, which was trained on Wikipedia in 104 different languages. This helps in leveraging transfer learning. We explored other word embeddings like GloVe (Global Vectors for Word Representation) and FastText, but went ahead with DistilBert as it has multilingual embeddings. We then created input vectors of int32 type for the model.

The training data had a large imbalance as non-toxic comments were around 90% of the data and toxic comments were around 10%. To account for this, we used the class_weights method which penalizes the imbalance by giving more emphasis to the minority class.

For the language models, we observed that data cleaning had no difference in the performance hence we used an unedited dataset. We kept the max sequence length to 192 for the comments. Comments are either truncated or padded accordingly and tokenized using tokenizers for respective models from the HuggingFace transformers library.

TABLE I
Baseline Architechtures with Epochs: 10, Optimizer: Adam, Learning Rate: 0.001,Loss - Binary Cross entropy

Architechture	Val Accuracy	ROC-AUC Score	Val Loss
LSTM(2 Layer)	0.84	0.49	0.69
LSTM(4 Layer)	0.72	0.5	0.69
Bi-LSTM(2 Layer)	0.56	0.72	0.82
Bi-LSTM(4 Layer)	0.71	0.6	0.6

B. Baseline Architecture

We implemented LSTM architecture as our baseline model. The Long Short-Term Memory (LSTM) network is a type of recurrent neural network that can learn order dependence in sequence prediction tasks. We tuned the model by changing parameters like the number of LSTM layers, the number of hidden units in dense layers, and padding. We finally implemented 2 models with 2 and 4 LSTM layers and 2 hidden layers.

The input of this model is the 768-dim distilbert pre-trained weights. The 2 hidden layers have a ReLU activation function and the output layer has a sigmoid activation function for binary classification. We used binary cross entropy as the loss function and Adam optimizer with an initial learning rate of 0.001 for optimization. We evaluated the performance of the model over 10 epochs and we observed that the train and validation accuracy increased for both models, but with some irregularities. We observed ROC-AUC of around 0.5 for both models, which is not considered good as the model is unable to classify the minority and majority classes.

We then wanted to increase the model complexity to better classify the labels, so we implemented Bi-LSTM models. Bidirectional LSTM is an extension of the traditional LSTM that improves the performance of sequence classification models. It trains two instead of one LSTM when all time steps of an input sequence are available. In the first case, the input sequence is used as-is, and in the second case, the input sequence has been reversed. As a result, the network will be able to learn more and faster about the problem because it will have additional context. We implemented two models with 2 and 4 Bi-LSTM layers and 2 hidden layers. The input of this model is the 768-dim distilbert pre-trained weights. The 2 hidden layers have ReLU activation function and the output layer has a sigmoid activation function for binary classification. We used binary cross entropy as the loss function and Adam optimizer with an initial learning rate of 0.001 for optimization. We evaluated the performance of the model over 10 epochs and observed comparatively better performance than LSTM models. We observed that the training accuracy increased for both models but validation accuracy decreased, with the highest of 0.8. We observed ROC-AUC of around 0.7 for both models, which is

better than the previous models.

C. Transformer Models

BERT, which stands for Bidirectional Encoder Representations from Transformers is a bidirectional transformer pre-trained using a combination of masked language modeling objective and next sentence prediction on a large corpus comprising the Toronto Book Corpus and Wikipedia.
Bert is an encoder stack of transformer architecture. A transformer architecture is an encoder-decoder network that uses self-attention on the encoder side and attention on the decoder side. Bert base has 12 layers in the Encoder stack. These are more than the Transformer architecture described in the original paper (6 encoder layers). Bert architectures (Base and Large) also have larger feedforward networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively) than the Transformer architecture suggested in the original paper. It contains 512 hidden units and 8 attention heads. Bert_base contains 110M parameters while Bert_large has 340M parameters.Here we have used BERT multilingual base cased.

BERT multilingual base cased is a pre-trained model on the top 104 languages with the largest Wikipedia using a masked language modeling (MLM) objective. This model is case-sensitive: it makes a difference between English and English.

RoBERTa builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates.
RoBERTa is a transformers model pre-trained on a large corpus in a self-supervised fashion. This means it was pre-trained on the raw texts only, with no humans labeling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts. Here we are using XLM-RoBERTa.

XLM-RoBERTa model is pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages.

DistilBERT is a small, fast, cheap, and light Transformer model trained by distilling Bert base. It has 40% less parameters than bert-base-uncased, and runs 60% faster while preserving over 95% of Bert's performances as measured on the GLUE language understanding benchmark. Configuration objects inherit from PretrainedConfig and can be used to control the model outputs.Here we are using DistilBert base multilingual cased.
DistilBert base multilingual cased model is a distilled version of the BERT base multilingual model. The model is trained on the concatenation of Wikipedia in 104 different languages listed here. The model has 6 layers, 768 dimensions, and 12 heads, totalizing 134M parameters (compared to 177M parameters for mBERT-base). On average, this model referred to as DistilmBERT, is twice as fast as mBERT-base.

The last hidden layer output of the language models are mean and max pooled, followed by Dropout, linear layer, and sigmoid to output probability of toxic class.

TABLE II
PRETRAINED MODELS

Pre-trained Model	Hyperparameters	Results
XLM Roberta	Epochs: 2 Optimizer: AdamW With linear LR scheduler Learning Rate: 2e-5 train_batch_size : 64 valid_batch_size : 64 Loss: Binary Cross entropy	val_loss : 0.39 val_auc: 0.91 val_accuracy: 0.87
Bert multilingual cased	Epochs: 2 Optimizer: AdamW With linear LR scheduler Learning Rate: 2e-5 train_batch_size : 64 valid_batch_size : 64 Loss : Binary Cross entropy	val_loss : 0.58 val_auc: 0.86 val_accuracy: 0.85
DistilBert multilingual cased	Epochs: 2 Optimizer: AdamW With linear LR scheduler Learning Rate: 2e-5 train_batch_size : 64 valid_batch_size : 64 Loss : Binary Cross entropy	val_loss : 0.52 val_auc: 0.85 val_accuracy: 0.85

IV. EXPERIMENTS

A. Configurations

We performed several experiments with different models and parameters. We primarily worked on Kaggle notebooks with GPU. We trained our models for 220k comments with binary labels. We made predictions on validation data of 8k comments and collected the results. The proportion of toxic labels in the validation dataset was 1:15.

We used a linear learning rate scheduler to change the learning rate as the model trained. For LSTM models we used an initial learning rate of 0.001 and for BERT models, 2e-5 which is less than the baseline as they are pre-trained models. Using a large learning rate for BERT gave poor results.

For LSTM models there were around 93M parameters and took about 2 hours for 50 epochs, with a batch size of 512. We observed overfitting after 10 epochs so we limited the final results to 10 epochs. BERT and RoBERTa had around 200M parameters and took about 2 hours for 2 epochs, with a batch size of 64, while DistiBert trained faster. All the models were optimized using Adam optimizer and the loss function was binary cross entropy.

B. Evaluation

For binary classification task, we compared performance across models by calculating metrics like AUC-ROC, precision, recall, F-1 score, and accuracy. We calculated these metrics using tf.keras.metrics for each epoch for all models. For evaluation, we choose AUC (area under curve ROC) as our metric since the data is highly imbalanced.

We assumed that using DistilBERT pre-trained multilingual embeddings will help in better performance of validation data, but this wasn't the case for LSTM models. We observed that high validation accuracy for these models was due to the prediction of all the validation samples as non-toxic. Bi-LSTM

TABLE III
COMPARTIVE RESULTS

Model	Valid ROC-AUC Score
XLM Roberta	0.91
Bert base multilingual cased	0.86
Bert base multilingual uncased	0.85
DistilBert base multilingual cased	0.84
Bi-LSTM(2Layer)	0.75
Bi-LSTM(4Layer)	0.6
LSTM(2Layer)	0.5
LSTM(4Layer)	0.5

models seemed to have performed better than LSTM models as all the predictions were not non-toxic.

As expected, the language models outperform LSTMs. We achieved the best performance from XLM Roberta model with a ROC-AUC of 0.91. This model also had the best validation accuracy and loss of 0.87 and 0.38 respectively. What surprised us is that the performance of DistilBERT and BERT are comparable with DistilBERT being a much smaller and more efficient model. Another important observation is that cased models performed better than uncased validating the fact that toxicity is encapsulated by the case in some form.

V. CONCLUSION AND FUTURE WORK

One of the limitations of our work is not including the embeddings of toxic language observed in the data. We have used only the root word embeddings and may have missed out on using some of the unknown tokens. Our training data primarily consisted of English comments and the validation dataset contained non-English comments. So training the models on English text and validating it on non-English may not give a very good performance. We would like to do a translation of English comments into all target languages and use this dataset as the training dataset and observe the performance.

VI. REFERENCES

1.DistilBert Huggingface
2. Multilingual toxic comment classification Datasets - Kaggle
3. Jigsaw toxic comment classification challenge Kaggle
4. Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, and Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding, May 2019.
5. Victor, Lysandre, Julien, Wolf, and Thomas. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, Mar 2020.
6. M. Schuster and K.k. Paliwal. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681, 1997.
7. Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016.
8. Julian Risch and Ralf Krestel. Toxic comment detection in online discussions, 2019.
9. Lei Gao and Ruihong Huang. Detecting online hate speech using context aware models. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, pages 260–266, Varna, Bulgaria, September 2017. INCOMA Ltd.
10. Guo and Xiao. Cross-language text classification via subspace co-regularized multi-view learning, Jun 2012.
11. Kingma, Diederik P., Jimmy, and Ba. Adam: A method for stochastic optimization, Jan 2017.
12. Peter Prettenhofer and Benno Stein. Cross-language text classification using structural correspondence learning.
13. Bert base multilingual cased Huggingface
14. XLM RoBERTaHuggingface