Multilingual Toxic Comment Classification

Lynn D. Kong
ldkong@stanford.edu

Abstract

Freedom of speech on the Internet has also led to a pervasive presence of toxic comments in online discussions and a need to detect their presence in any language. While previous work focused on classification models for English input or other pre-defined language input, we investigate whether English-trained toxic comment classification models can be language agnostic. We trained 6 models on English inputs and validated against non-English inputs. We found that models do not perform exceptionally well non-English inputs, but the mislabeling errors was not evenly distributed across languages. This is likely due to degree of similarities between each of the non-English language and English. Based on the model performance, we conclude that existing English-only trained and non-English validated toxic classification models likely will not perform well on non-English inputs.

1 Introduction

Freedom of speech on the Internet has also led to a pervasive presence of toxic comments in online discussions. A toxic comment is defined as a rude, disrespectful, or unreasonable comment that is likely to make other users leave a discussion [8]. Toxic comment detection will facilitate and maintain the openness of the online community. Furthermore, a language-agnostic detection model will be vastly more effective towards moderating all aspects of the Internet.

2 Previous Work

Previous work come in two-fold: toxicity text classification and multilingual classification. The english-only training data set in this problem has been studied by Aken et. al [1], wherein they applied an ensemble of logistic regression, LSTM RNN, GRU RNN, and CNN to classify toxicity. Most approaches to multilingual text classification involves some form of machine translation from source language in which the model is trained on to a target language[10] [7] [5]. Exploration on the combined topic of multilingual toxic text classification has been limited.

3 Dataset and Features

Our training and validation data sets are provided by the Multilingual Toxic Comment Classification Kaggle Challenge, sponsored by Jigsaw. For the baseline model, we used a training set contains 223549 samples, with the input being a English text comment string, and the labels being toxicity (binary). 21384 or 9.4% of the samples are labelled toxic (Fig. 1, left).

validation data set contains 8000 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. 1,

CS230: Deep Learning, Spring 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

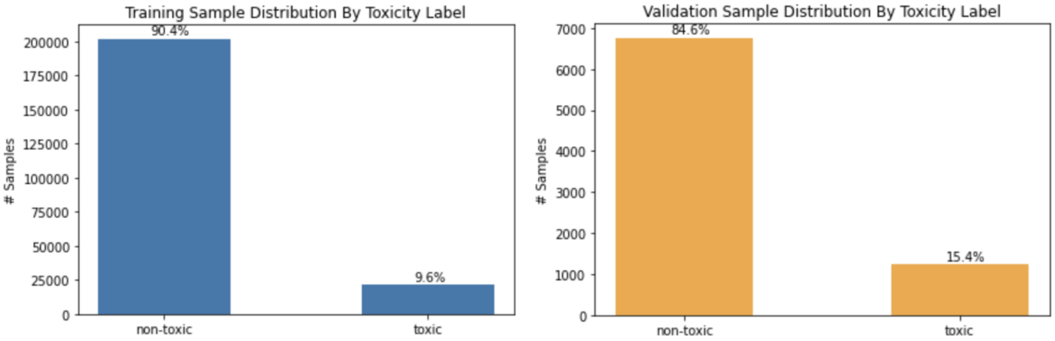

Figure 1: Left: Training sample distribution out of 223549 samples. Right: Validation sample distribution our of 8000 samples.

right). The raw data set also included language of origin for each sample (Appendix Fig. 5) for data analysis. However, as the model is meant to be language-agnostic, we did not include the provided language of origin in our input.

The input text of the training and validation data samples are encoded as feature vectors with max length 512 using the DistilBERT [11] multilingual cased tokenizer. This is a first pass choice for encoding, and we may explore other methods of encoding as next steps. We chose DistilBERT because one of our next steps is to fine-tune a pre-trained multilingual DistilBERT[3] model, and DistilBERT is meant to be "smaller, faster, cheaper and lighter". The training samples are shuffled and grouped into batches of 32.

4 Method and Experiment

We have experimented with six different models, all of which are described in Table 1. Note that model S is the baseline model from milestone 1. The foundation model framework is a bidirectional LSTM structure shown below.We chose to have this bidirectional LSTM structure because it is a well-established text classification model [9] [2]. It is the basic model used in IMDB reviews sentient classification in Tensorflow demos.

1. Embedding layer
   (a) From scratch
   (b) DistilBERT pre-trained weights
2. Bidirectional LSTM layer
   (a) Number of LSTM layers varied between 1, 2, and 4.
3. Self-Attention layer
   (a) Without Self-Attention layer
   (b) With Self-Attention layer
4. ReLU activation layer
5. Sigmoid activation layer (output)

With this foundation model structure, we iterated on multiple models by varying three aspects. The first aspect is whether to train the embedding layer from scratch or with pre-trained weights, so to leverage transfer learning.

The second aspect is whether to include self-attention layer after LSTM layer. Inclusion of attention layer to build context-aware models have been shown to improve toxic comment classification [4].

Finally, the third aspect is the number of LSTM layers, which varied between 1, 2, and 4. We investigated this aspect because of inspiration from approaches to Neural Machine Translation encoder.Given that we would like to perform classification on multiple languages, our task can be considered a classification after the NMT encoding layer. As mentioned in Google's NMT description [12], deeper LSTM encoding and decoding layers (up to 8 layers) better captures the subtle irregularities from in source and target languages. Given the time and resource constraints, we chose to investigate model P, which was the best performing model of the first 4 models in Table 1, with 2 and 4 LSTM layers.

Notation	Description
S	Trained from scratch, no attention
SA	Trained from scratch, with attention
P	pre-trained, no attention
PA	pre-trained, with attention
P_2	pre-trained, no attention, 2 LSTM layers
P_4	pre-trained, no attention, 4 LSTM layers

Table 1: Model notations and descriptions.

We computed loss with binary cross entropy and optimized with Adam optimizer [6].

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log\left(p\left(y_i\right)\right) + \left(1 - y_i\right) \log\left(1 - p\left(y_i\right)\right)$$

We implemented the model on Google Colab (Github) without using TPUs. We trained on batches of 32 samples, with the initial learning rate of 0.001. We trained with programmatic learning rate adjustment on loss plateaus using `tensorflow.keras.callbacks.ReduceLROnPlateau`. For models S, SA, P, PA, we trained for 10 epochs. For models P_2, P_4, we trained for 20 epochs because they are more complex models.

5 Results

We first evaluated the models S, SA, P, PA before choosing to increase model complexity on the best performing model (based on ROC-AUC score). Over 10 epochs, we observed that training accuracy was gradually increasing for all models. However, validation accuracy was pretty much stagnant or decreasing for all models, with the highest of 0.8499 (Fig. 2, top left, middle left). Similarly, training loss was pretty steadily decreasing while validation loss was stagnant or increasing (Fig. 2, top right, middle right). We chose to calculate the ROC-AUC score for performance comparison, and was able to see a steady increase in all models except PA (Fig. 3, left).

We then evaluated the models P_2 and P_4. These models were trained over 20 epochs instead of 10 epochs to account for the increase in model complexity. We observed a trend of increasing training accuracy and decreasing validation accuracy over time, similar to previous models (Fig. 2, bottom left). However, because of the longer training session, we were able to observe that the validation accuracies was unstable until epoch 14, at which point they began to around 50-60%. We also observe that ROC-AUC score for both models was increasing over training and plateaued around epoch 14 (Fig. 3, right).

6 Analysis

A worse performance with validation was not entirely unexpected because of the linguistic difference in data. We expected models with pre-trained multilingual DistilBERT embedding weights will perform better in terms of accuracy because it will be more language agnostic, but for the most part, models P and PA performed on par with model S and SA. Model P did perform better in terms of ROC-AUC score than all other models.

Through error analysis, we confirmed that the 84% validation accuracy at initial epochs for almost all models are the result of predicting all validation samples as non-toxic. This explains why some models' validation performance stayed stagnant - their learned features did not translate across languages. For the models that do show changes in validation performances over time, it is interesting to observe the difference in how the performance changed. For models P, SA, P_2, and P_4, there was a sudden drop in validation validation to <65%, followed by a sudden increase. It looks as if the

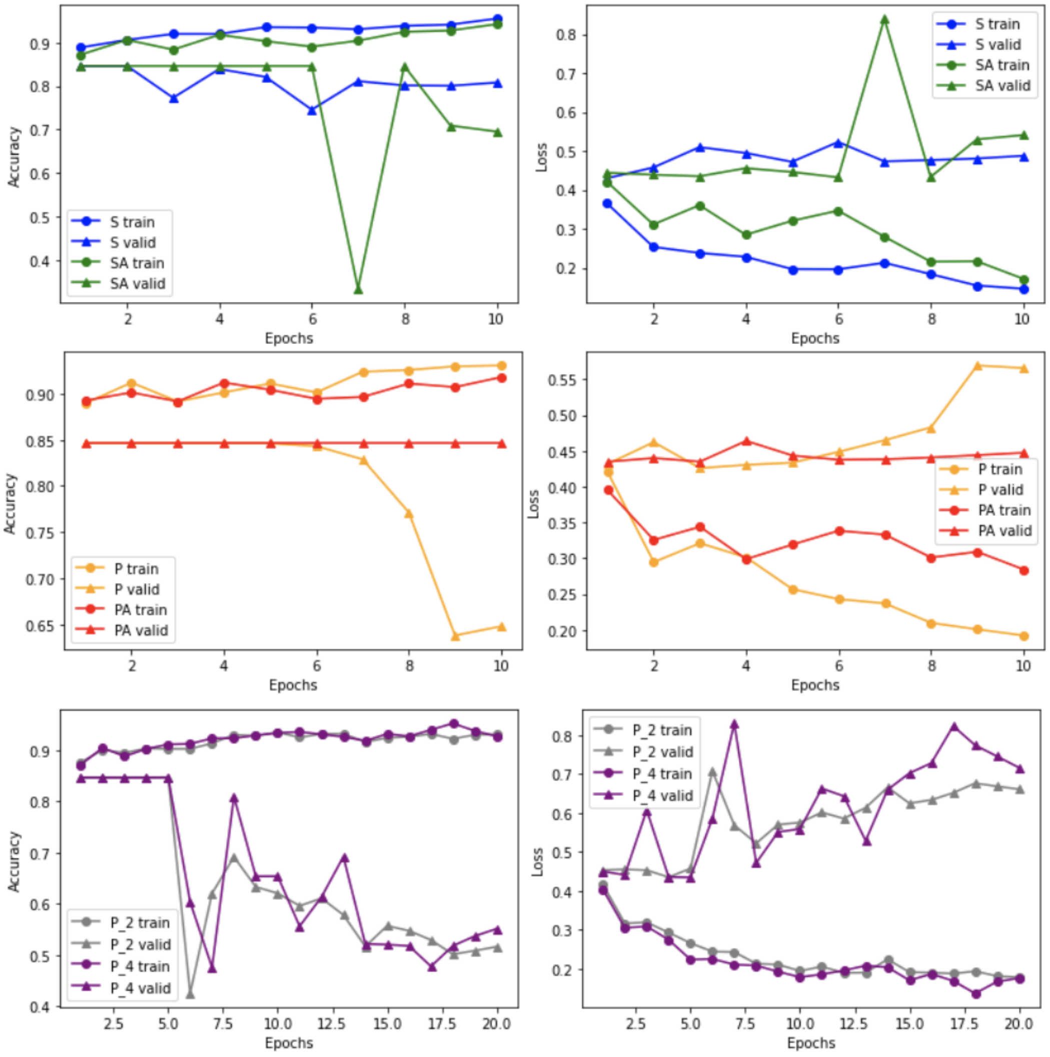

Figure 2: Top and middle left: Training and validation accuracies of models S, SA, P, and PA over 10 epochs. Top and middle right: Training and validation losses of models S, SA, P, and PA over 10 epochs. Bottom left: Training and validation accuracies of models P_2 and P_4 over 20 epochs. Bottom right: Training and validation losses of models P_2 and P_4 over 20 epochs.

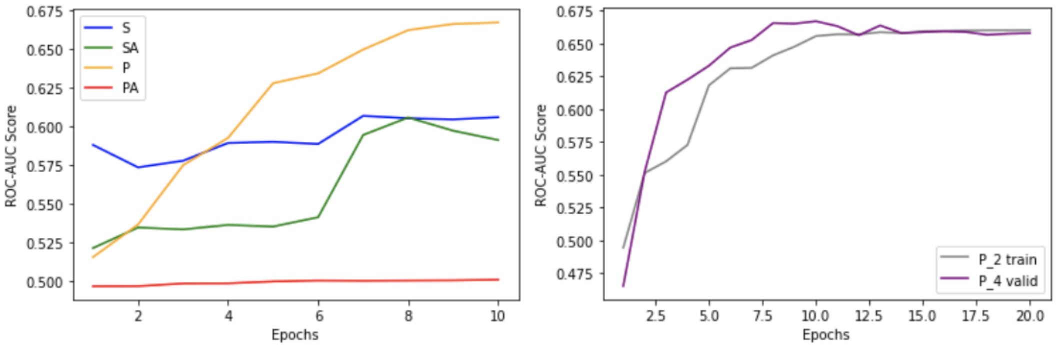

Figure 3: Left: ROC-AUC score for validation of models S, SA, P, and PA over 10 epochs. Right: ROC-AUC score for validation of models P_2 and P_4 over 20 epochs.

models are stabilizing after an initial disturbance. In contrast, the baseline model S was noticeably changing in validation performance but did not have the sudden drop in performance. Pre-trained model P validation performance was stagnant throughout training. It would be interesting to examine in the future why some models experience some sort of jolt-and-stabilization in the training process, while others don't.

When we trained P_2 and P_4 for longer epochs, we observed that the ROC-AUC score plateaued around the same time as the stabilization in validation accuracies and losses. We wonder if this is a sign that these models needs finetuning at a different hyperparameter spectrum than what we have currently examined.

We also looked specifically at the validation predictions for P_2 and P_4 and found the distribution of mislabeled samples across languages is uneven (Fig. 4). Specifically, there is a low number of mislabeling of Spanish samples and a high number of mislabeling of Turkish samples. This is likely due to the degree of similarity between English and Spanish. In contrast, Turkish samples are probably highly mislabeled because of its lack of similarity to English.

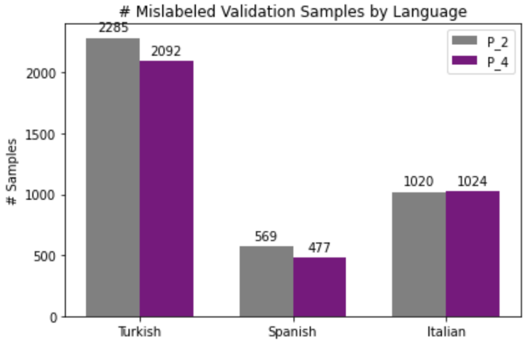

Figure 4: Mislabeled P_2 and P_4 validation output by language.

7 Conclusion and Future Work

Based on the performances of the six models, we conclude that it is unlikely that toxic comment classification models trained on English inputs and validated on non-English inputs will perform well on non-English inputs. However, we would like to investigate whether including English inputs in the validation data set would impact the model learning. We are also curious why models with pre-trained encoding weights have a dramatic dip in validation performance before sta. Lastly, while there does not exist an open-sourced pre-trained multilingual NMT yet, but should it be available in the future, we would like to investigate the performance of multilingual toxic comment classification models that uses pre-trained encoder weights of a multilingual NMT.

8 Contribution

Lynn Kong worked on the entirety of this project.

References

[1] van Aken, Betty, Risch, Julian, Krestel, Ralf, Löser, and Alexander. Challenges for toxic comment classification: An in-depth error analysis, Sep 2018.

[2] 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.

[3] Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, and Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding, May 2019.

[4] 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.

[5] Guo and Xiao. Cross language text classification via subspace co-regularized multi-view learning, Jun 2012.

[6] Kingma, Diederik P., Jimmy, and Ba. Adam: A method for stochastic optimization, Jan 2017.

[7] Peter Prettenhofer and Benno Stein. Cross-language text classification using structural correspondence learning.

[8] Julian Risch and Ralf Krestel. Toxic comment detection in online discussions, 2019.

[9] M. Schuster and K.k. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[10] Lei Shi, Rada Mihalcea, Mingjun Tian, Mingjun Tian, Rada Mihalcea University of North Texas, University of North Texas, Microsoft Research Asia, and Technical University of Catalonia. Cross language text classification by model translation and semi-supervised learning, Oct 2010.

[11] Victor, Lysandre, Julien, Wolf, and Thomas. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, Mar 2020.

[12] Schuster M. Chen Z. Le Q. V. Norouzi M. Macherey W. Krikun M. Cao Y. Gao Q. Macherey K. Klingner J. Shah A. Johnson M. Liu X. Łukasz Kaiser Gouws S. Kato Y. Kudo T. Kazawa H. Stevens K. Kurian G. Patil N. Wang W. Young C. Smith J. Riesa J. Rudnick A. Vinyals O. Corrado G. Hughes M. Wu, Y. and J. G Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

9 Appendix

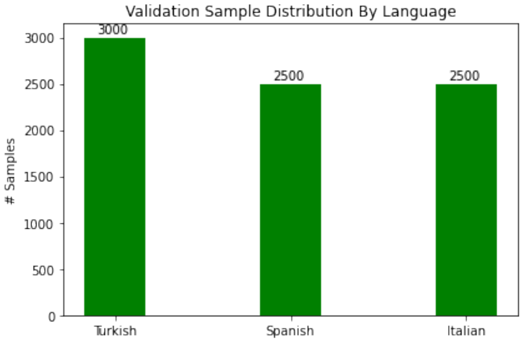

Figure 5: Validation sample distribution by tagged language of origin out of 8000 samples.