Evaluation of Recurrent Neural Network architectures for abusive language detection in cyberbullying contexts

Filip Markoski[1], Eftim Zdravevski[1], Nikola Ljubešić[2], Sonja Gievska[1]
[1]*Faculty of Computer Science and Engineering*
Skopje, Macedonia
filip.markoski45@gmail.com, eftim.zdravevski@finki.ukim.mk, sonja.gievska@finki.ukim.mk
[2]*Department of Knowledge Technologies, Jožef Stefan Institute,*
nikola.ljubesic@ijs.si
Ljubljana, Slovenia

***Abstract*—Cyberbullying is a form of bullying that takes place over digital devices. Social media is one of the most common environments where it occurs. It can lead to serious long-lasting trauma and can lead to problems with fear, anxiety, sadness, mood, energy level, sleep, and appetite. Therefore, detection and tagging of hateful or abusive comments can help in the mitigation or prevention of the negative consequences of cyberbullying. This paper evaluates seven different architectures relying on Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) gating units for classification of comments. The evaluation is conducted on two abusive language detection tasks, on a Wikipedia data set and a Twitter data set, obtaining ROC-AUC scores of up to 0.98. The architectures incorporate various neural network mechanisms such as bi-directionality, regularization, convolutions, attention etc. The paper presents results in multiple evaluation metrics which may serve as baselines in future scientific endeavours. We conclude that the difference is extremely negligible with the GRU models marginally outperforming their LSTM counterparts whilst taking less training time.**

Keywords—Deep Learning, NLP, RNN, LSTM, GRU, Abusive Language Detection, Hate Speech, Cyberbullying

I. INTRODUCTION

Cyberbullying is the use of technology to harass, threaten, embarrass, or target another person. It includes posting online threats and mean, aggressive, or rude texts, personal information, pictures, or videos designed to hurt or embarrass someone else. Online bullying can be particularly damaging and upsetting because it is usually anonymous, and therefore, hard to trace and control. Online bullying and harassment can be easier to commit than other acts of bullying because the bully does not have to confront his or her target in person. Online bullying, as any other kind of bullying, can lead to serious long-lasting problems. The stress of being in a constant state of upset or fear can lead to problems with mood, energy level, sleep, and appetite. It also can make someone feel jumpy, anxious, or sad. If someone is already depressed or anxious, cyberbullying could lead to much more serious consequences.

For these reasons, providing a systematic solution that can recognize and tag textual comments that represent some sort of hateful or abusive content can be valuable in the prevention and mitigation of their consequences. [1]

Previous works have already applied various approaches aiming to tackle this kind of tasks. Most successful approaches for this task usually employ Recurrent Neural Networks (RNNs). RNNs are Deep Neural Networks (DNN) that are adapted to sequence data, i.e. to input and output of variable length. RNNs contain loops in the hidden layer to retain information from a previous time step which will later be used to predict the value of the current time step. This retention of information makes the neural networks extremely deep and thus makes them difficult to train to capture long-term dependencies because the gradients tend to either vanish or explode and thus the RNNs are prone to exploding or vanishing gradients. [1]

The most prominent ways to reduce the negative effects of training RNNs are either to design a better learning algorithm than stochastic gradient descent, such as a powerful second-order optimization algorithm or design an improved activation function such as the LSTM architecture, which was developed and proposed in [2] which proves to be an effective way of dealing with the vanishing gradient problem and thus became a standard, or the GRU architecture which was proposed in [3] and shares many similarities to the LSTM architecture whilst still employing different circuitry. Other ways of dealing with the problems faced by RNNs are to perform regularization of the RNN's weights that ensures that the gradient does not vanish, to entirely stop learning the recurrent weights and finally, to very carefully initialize the RNN's parameters, such as in [4] and [5].

In this paper, we attempt to evaluate the two most prevalent architectures as answers to dealing with the vanishing gradient problem whilst training on sequential data. The approaches we have chosen are LSTM and GRU in the context of other components. Our aim is to see which architecture performs better when its most defining component is an LSTM module, or a GRU module. We perform this evaluation on two abusive language detection tasks whilst situating the most defining component in architectures which incorporate various neural network mechanisms such as bi-directionality, regularization, convolutions, attention etc. We draw our conclusions on the basis of a variety of evaluation metrics, which may subsequently serve as baselines for future research.

II. RELATED WORK

There exist many empirical comparisons performed on RNN architectures, such as LSTM or GRU. In [2], the authors evaluated multiple models, namely LSTM, GRU and tanh-RNN, with all approximately the same number of parameters and trained using RMSProp on a suite of sequence modelling tasks, namely, tasks of polyphonic music modelling and speech signal modelling. The authors concluded that although GRU produced superior results to the other models overall, the difference was not too great as to lead to a firm conclusion of which model is best.

Deeming the LSTM's architecture to be 'ad-hoc', in [3], the authors perform an ablation study and an empirical evaluation of LSTM, GRU and LSTM-mutated architectures which they produced using an evolutionary architecture

search, more extensive than the architecture search conducted by [4] in which the authors performed fewer experiments with small models. From the ablation study and model results from [3], it was shown that the forget gate in the LSTM architecture is most important and that its removal results in drastically inferior performance, except in language modelling. Furthermore, the authors noted that initializing the bias of the forget gate to be a number between 1 and 2 leads the LSTM models to have very comparable results to that of the GRU models, thus closing the performance gap between the LSTM and GRU models.

In [5], it is shown that LSTM models with a large number of parameters take up a considerable amount more training time than their GRU counterparts whilst still producing similar results. Their models used ReLu as an activation function and the Adam optimization algorithm.

In comparison to [2], we utilize more appropriate parameter initialization strategies, employ the use of a more robust parameter optimization algorithm and a plethora of architectures leading to perhaps a more reliable empirical comparison between the LSTM and GRU components.

III. METHODOLOGY

The two data sets used within this study, namely the Wikipedia and Twitter data sets. Further, this section describes the data preprocessing, the appropriate evaluation metrics for both data sets and describes the generalized model architectures and the specific LSTM or GRU models which are manifestations of those architectures.

A. Toxic Wikipedia Comment Data Set

The data set, described in Table I, used to train and evaluate the models is the same one used in [6] and offered publicly as part of the Toxic Comment Classification Challenge competition. [7] The multi-labeled data set is comprised of a training set containing 159571 entries and of a testing set comprised of 153164 entries. The six labels, each presented in a separate column that are provided in the training set and need to be predicted in the testing set, are the following: 'toxic', 'severe_toxic', 'obscene', 'threat', 'insult', 'identity_hate'. From the entire training set, only 16225 entries are labeled with any of the aforementioned labels, meaning that the labels often overlap. The training set has a class-imbalance problem, in relation to this, the authors of [8] present a "real-life" distribution of abusive language use via surveying available abusive language data sets. From these data sets, one can see that they are usually comprised of an overwhelming majority of non-abusive entries. Additionally, sentence length does not seem to be a significant indicator of toxicity which is in accordance with the conclusion of [9], which is that word-length distribution features provide little to no improvement in a model's predictive abilities.

TABLE I. LABELS FOR THE TOXIC WIKIPEDIA COMMENTS TRAINING DATA SET (MULTI-LABELED DATA SET WITH OVERLAPPING LABELS)

Label	*toxic*	*severe toxic*	*obscene*	*threat*	*insult*	*identity hate*
Count	15294	1595	8449	478	7877	1405
%	9.6	1.0	5.3	0.3	4.9	0.9

B. Twitter Data Set

We use the same data set as [10], which is comprised of approximately 100000 tweets of which only 61194 were able to be retrieved using the Twitter API. Of the ones retrieved, in a single class column (unlike the Wikipedia data set which separates each label in a separate column), 63% are annotated as 'normal', 19% as 'abusive', 14% as 'spam' and 4% as 'hateful' with exact counts shown in Table II.

Each tweet text was further processed using a tweet normalization tool[1] which directly optimizes the vocabulary and in turn the models' power to generalize by replacing usernames with a single token '<user>' and changing words such as 'goooood' to 'good' for example.

The data set was split into a training, validation and test data set each consisting with 76%, 4% and 20% of the data respectively.

TABLE II. LABELS FOR THE TWITTER DATA SET

Label	*abusive*	*hateful*	*spam*	*normal*
Count	11766	2461	8561	38407
%	19.2	4.0	14.0	62.8

C. Data Preprocessing

Each of the comments was represented with a padded indexed representation of itself. Keras[2] Tokenizer was used to perform the tokenization, indexing and padding of each comment, in which the vocabulary was limited to the most frequent 20000 tokens and each comment was padded to a maximum length of 200 indices.

D. Evaluation Metric

1) Toxic Wikipedia Comments Data Set

The models had to predict a probability for each of the six possible columns, each representing one of the labels. This was evaluated using the area under the receiver operating characteristic curve (ROC-AUC) which was calculated after each epoch for each of the models to get the metrics for the training set predictions. Additionally, the ROC-AUC evaluations are also provided for the private and public Kaggle testing sets evaluated by the Kaggle platform. Although we deem the classification metrics used on the Twitter data set as more appropriate, we could not calculate the same due to not having the corresponding labels for the test set, thus we only resort to the ROC-AUC scores returned for each submission of predictions by the Kaggle platform in the form of private and public scores.

2) Twitter Data Set

The models had to predict a probability for each of the four possible labels with the prediction being a one-hot encoded vector whose only active components correspond to the label assigned with the highest probability. This was evaluated using the following metrics: accuracy, F1-micro, F1-macro, weighted precision, and weighted recall scores.

E. Models

Each of the models was constructed using the python deep learning library Keras. In total, there are seven model architectures containing a recurrent neural network layer which manifests a total of 12 models, that is, all the model architectures once with an LSTM component as the most representative component, and similarly, once with a GRU

[1] https://github.com/cbaziotis/ekphrasis

[2] https://github.com/keras-team/keras

component. Each of the architectures use an Embedding layer which transforms the indexed words with a vector representation of size 50. We chose not to work with pre-trained word embeddings as the primary goal of our experimentation is to isolate the benefits of different architectures. The architectures described through their dual manifestations are:

1. (unidirectional) LSTM / GRU – Unidirectional approach to the language task (Fig. 2).
2. (bidirectional) Bi-LSTM / Bi-GRU – A bidirectional variant of the first model architecture (Fig. 2).
3. (bi-then-conv) Bi-LSTM-CNN / Bi-GRU-CNN – Following the bidirectional recurrent neural network layer, the model extracts one-dimensional convolutions, performs global average pooling and global max pooling, concatenates them and used the resultant vector to infer a prediction (Fig. 3).
4. (bi-conv-uni) Bi-LSTM-CNN-LSTM / Bi-GRU-CNN-GRU – In addition to the Bi-RNN-CNN architecture we add another RNN component that attempts to learn on the convolved information and afterwards infer a prediction (Fig. 4).
5. (convolutional) Multi-CNN-Bi-LSTM / Multi-CNN-Bi-GRU – Contrary to some of the previous architectures, we use one-dimensional convolutional layers of kernel sizes 1, 2, 3 and 5 with the hopes to derive unigram, bigram, trigram and 5-gram features which shall later be used in training the RNN component of the architecture (Fig. 5).
6. (conv-attention) Conv-Att-LSTM / Conv-Att--GRU – Two pairs of a kernel-size-three convolutional layer and max-pooling layer precede an attention mechanism right before the RNN component (Fig. 6).
7. (attention) Attention-LSTM / Attention-GRU – Only an attention mechanism is added before the RNN component (Fig. 7).

In Figures 2 through 7, the shape of the tensor is described below the name of the component in the architecture.

Each of the models is trained for three epochs with a batch size of 256 and optimized with an Adam optimizer with a default learning rate of 0.001 on Google Collaboratory Tensor Processing Unit (TPU) runtime, which most likely offered a TPU v2 device with 8 GiB of high-bandwidth memory, two TPU cores and one matrix unit for each TPU core. For the Toxic Wikipedia Comments data set, as each label is presented in its separate column, the output layer utilizes a sigmoid function, thus it outputs independent probabilities for each label-column, this type of output is in tandem with a binary cross-entropy loss function. For the Twitter data set, the output layer utilizes a Softmax function which is paired with a categorical cross-entropy loss function ensuring the model can assign a probability for each possible label. The rate for any drop-out mechanism, including the recurrent drop-out, ranges from 0.1 to 0.2.

According to the example of [11], each Batch Normalization layer has been positioned before any singular Drop-Out layer. For the recurrent neural networks, the activation function is tanh and the recurrent activation function is Sigmoid. All other hidden layers have ReLu as an activation function.

Inspired from [12] and [13], the layers using the ReLu activation function are initialized with a He uniform distribution, whilst the others are initialized with a Glorot uniform distribution.

It is important to note that Keras follows the advice from [3] and initializes the bias of the LSTM forget gate to 1.

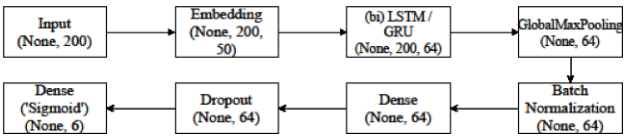

Fig. 2. Architecture for the LSTM and GRU models, as well as their Bidirectional alternatives

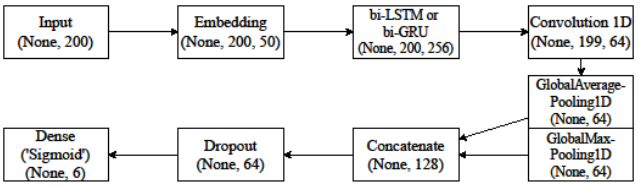

Fig. 3. Architecture for the bi-LSTM-CNN and bi-GRU-CNN models

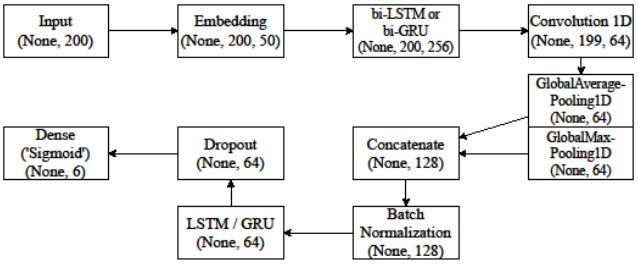

Fig. 4. Architecture for the bi-LSTM-CNN-LSTM and bi-GRU-CNN-GRU models

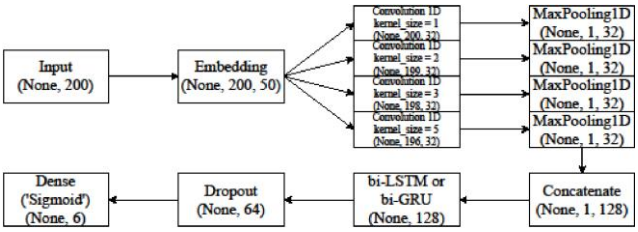

Fig. 5. Architecture for the multi-CNN-bi-LSTM and multi-CNN-bi-GRU models

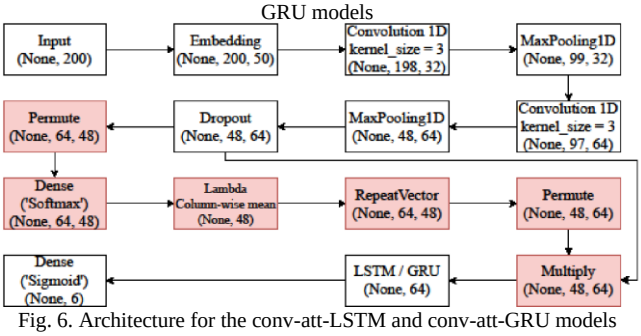

Fig. 6. Architecture for the conv-att-LSTM and conv-att-GRU models (the red sections highlight the attention mechanism)

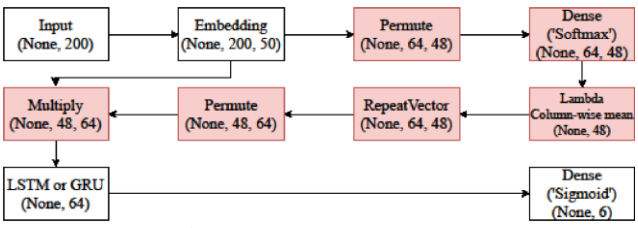

Fig. 7. Architecture for the attention-LSTM and attention-GRU models (the red sections highlight the attention mechanism)

IV. Results

Concerning the Toxic Wikipedia Comments data set, in Fig. 8 we present the ROC-AUC scores for each model for each of the three epochs on a validation data set and the Kaggle private and public testing data sets. We deem the private testing data set ROC-AUC score as most representative of the model's power to generalize as performs its evaluation on a testing set not publicly provided.

Concerning the Twitter data set, in Fig. 9 we present the aforementioned metrics for each of the models. The authors of [14] and [15] note that macro metrics provide a better sense of effectiveness on the minority classes in a class-imbalanced problem, thus we deem the F1-macro score as the most relevant indicator.

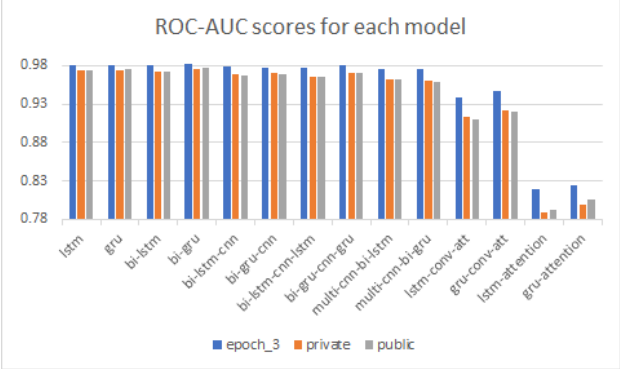

Fig. 8. ROC-AUC scores for each model on the Toxic Wikipedia Comment data set

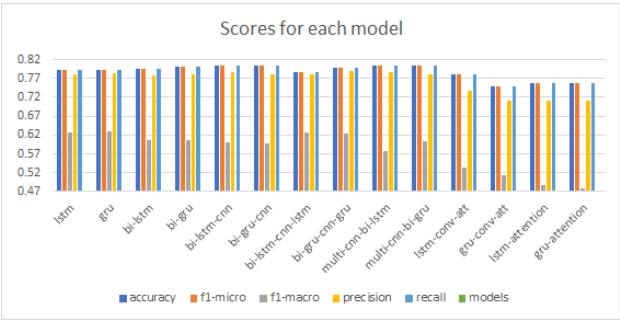

Fig. 9. Accuracy, F1-micro, F1-macro, precision and recall scores for each model on the Twitter data set

Our findings are summarized below:

- Each of the models, for both data sets, began to overfit after the second epoch of training, except for the only-attention models. Possible causes for this are presented in Section V.
- Judging the models on both data sets, looking at the private score for the Toxic Wikipedia Comments data set, and the F1-Macro score for the Twitter data set and awarding points to each recurrent neural component whenever its model scores better than its counterpart, we can say, although the differences are negligible, the GRU models outperformed the LSTM models. The scores are presented in Table III and Table IV, for the Wikipedia and Twitter data sets, respectively.
- In terms of the number of parameters, the LSTM models have significantly more trainable parameters than their GRU counterparts. Details can be found in Table VI.
- In terms of training time, which is closely related to the number of trainable parameters, the GRU models always finished their training faster than their LSTM counterparts. Details can be found in Table V.
- The simpler architectures got the best scores, with the best model for the Wikipedia data set being the Bi-GRU model achieving a private score of 0.975, closely followed by the unidirectional GRU model. For the Twitter data set the best model being GRU, with 0.631 F1-Macro score is followed by Bi-LSTM-CNN-LSTM with 0.626 F1-Macro score.
- The conv-attention and attention architectures trained for significantly less time while achieving decent results, yet still inferior to the rest of the evaluated models. Section V gives possibilities to why the attention-including architectures noticeably underperform in comparison to the other alternatives.

V. Discussion and Future Work

The authors of [6], which contributed the Wikipedia data set, managed to obtain a ROC-AUC score of 0.971 using a DNN architecture with character n-grams. To contrast this, a more traditional machine learning approach can be found in [16], which uses feature construction analogous to [17] which achieved a ROC-AUC score of 0.89 using a logistic regression classifier. Evidently, the Bi-GRU model, with a score of 0.975 here performs sufficiently well with both of these attempts.

Regarding the Twitter data set, which is differently annotated as in [14] or [15], our best model, being the unidirectional GRU, achieved 0.63 F1-Macro score. Even though it does not outperform [18], it does compare well with the reported [19] F1-Micro score of 0.827, whilst our Bi-GRU has 0.802.

Regarding the underperformance of the attention-based models, one possibility might be due to the use of a single attention vector which is shared across the input dimensions. In the computation of the attention vector is a mean operation which possibly cumulatively worsens the feature vectors.

Regarding future work, further empirical evaluations may be performed on the current model architectures using various pre-trained embeddings. Additionally, these evaluations may incorporate different types of data sets and different model architectures. Theoretical analysis also may be conducted as well as an ablation study with the goal of forming a more concrete and reliable comparison. This could be done in the form of a survey in which the various findings are collected, and the conclusion is more significant. The algorithms may also be adapted for real-time classification or abusive language detection, as well as employed by in-production platforms.

VI. Conclusion

In the context of the two abusive language detection tasks, the difference between the LSTM models and their GRU counterparts is extremely negligible. Still, the GRU models train faster due to the smaller number of trainable parameters and thereby, overall, outperform the LSTM models. In agreement with the results obtained by [2], we cannot make a firm conclusion on which of the gating units is better.

TABLE III. SCORES ON THE TOXIC WIKIPEDIA COMMENTS DATA SET (LSTM IS SUPERIOR IN CELLS HIGHLIGHTED WITH BOLD, GRU IS SUPERIOR IN CELLS HIGHLIGHTED WITH ITALIC)

Models	epoch 1	epoch 2	epoch 3	private	public
lstm	0.9732	**0.9798**	**0.9815**	0.9737	0.9734
gru	*0.9738*	0.9787	0.9812	*0.9738*	*0.9748*
bi-lstm	0.9737	0.9777	0.9810	0.9727	0.9717
bi-gru	*0.9772*	*0.9809*	*0.9818*	*0.9751*	*0.9775*
bi-lstm-cnn	**0.9749**	**0.9773**	**0.9787**	0.9694	0.9679
bi-gru-cnn	0.97478	0.9753	0.9766	*0.9706*	*0.9695*
bi-lstm-cnn-lstm	0.9713	0.9750	0.9772	0.9661	0.9648
bi-gru-cnn-gru	*0.9764*	*0.9784*	*0.9815*	*0.9710*	*0.9700*
multi-cnn-bi-lstm	0.9704	**0.9764**	**0.9759**	**0.9624**	**0.9615**
multi-cnn-bi-gru	*0.9728*	0.9750	0.9757	0.9609	0.9595
lstm-conv-att	0.9168	0.9359	0.9386	0.9139	0.9105
gru-conv-att	*0.9307*	*0.9456*	*0.9466*	*0.9220*	*0.9196*
lstm-attention	**0.7872**	0.8126	0.8188	0.7884	0.7924
gru-attention	0.7832	*0.8168*	*0.8245*	*0.7988*	*0.8065*

TABLE IV. SCORES ON THE TWITTER DATA SET (LSTM IS SUPERIOR IN CELLS HIGHLIGHTED WITH BOLD, GRU IS SUPERIOR IN CELLS HIGHLIGHTED WITH ITALIC)

Models	Accura-cy	F1-micro	F1-macro	Preci-sion	Recall
lstm	**0.7939**	**0.7939**	0.6259	0.7810	**0.7939**
gru	0.7928	0.7928	*0.6309*	*0.7838*	0.7928
bi-lstm	0.7948	0.7948	0.6062	0.7773	0.7948
bi-gru	*0.8022*	*0.8022*	*0.6062*	*0.7798*	*0.8022*
bi-lstm-cnn	0.8042	0.8042	**0.6016**	**0.7863**	0.8042
bi-gru-cnn	*0.8046*	*0.8046*	0.5981	0.7814	*0.8046*
bi-lstm-cnn-lstm	0.7880	0.7880	**0.6260**	0.7826	0.7880
bi-gru-cnn-gru	*0.7978*	*0.7978*	0.6231	*0.7890*	*0.7978*
multi-cnn-bi-lstm	**0.8040**	**0.8040**	0.5763	**0.7879**	**0.8040**
multi-cnn-bi-gru	0.8034	0.8034	*0.6044*	0.7798	0.8034
lstm-conv-att	**0.7803**	**0.7803**	**0.5343**	**0.7388**	**0.7803**
gru-conv-att	0.7494	0.7494	0.5118	0.7119	0.7494
lstm-attention	0.7580	0.7580	**0.4878**	0.7110	0.7580
gru-attention	*0.7591*	*0.7591*	0.4793	*0.7124*	*0.7591*

TABLE V. DIFFERENCES IN TOTAL SECONDS TAKEN TO TRAIN EACH MODEL (LSTM IS SUPERIOR IN CELLS HIGHLIGHTED WITH BOLD, WHERE THE DIFFERENCE IS NEGATIVE, GRU IS WITH ITALIC, WHERE THE DIFFERENCE IS POSITIVE)

	Wikipedia Data Set			Twitter Data Set		
Architecture	LSTM	GRU	Differ-ence	LSTM	GRU	Differ-ence
unidirectional	890	812	*78*	277	242	*35*
bidirectional	1703	1402	*301*	546	451	*95*
bi-then-conv	4681	3879	*802*	1519	1235	*284*
bi-conv-uni	5445	4474	*971*	1753	1444	*309*
convolutional	500	507	**-7**	160	164	**-4**
conv-attention	568	495	*73*	178	163	*15*
attention	1154	964	*190*	381	320	*61*

TABLE VI. DIFFERENCES IN THE TOTAL NUMBER OF TRAINABLE PARAMETERS FOR EACH MODEL (GRU HAS FEWER TRAINABLE PARAMETERS IN ALL CASES)

Architecture	LSTM	GRU	Difference
unidirectional	29440	22080	*7360*
bidirectional	58880	44160	*14720*
bi-then-conv	183296	137472	*45824*
bi-conv-uni	232704	174528	*58176*
convolutional	98816	74112	*24704*
conv-attention	33024	24768	*8256*
attention	70030	62670	*7360*

VII. REFERENCES

[1] S. Hinduja and J. W. Patchin, "Bullying, Cyberbullying, and Suicide," *Archives of Suicide Research*, vol. 14, no. 3, pp. 206-221, 2010.

[2] J. Chung, C. Gulcehre and K. Cho, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," 2014.

[3] R. Jozefowicz, W. Zaremba and I. Sutskever, "An empirical exploration of recurrent network architectures," *International Conference on Machine Learning*, vol. 2350, p. 2342, 2015.

[4] J. Bayer, D. Wierstra, J. Togelius and J. Schmidhuber, "Evolving Memory Cell Structures for Sequence Learning," pp. 755-764, 2009.

[5] A. Shewalkar, D. Nyavanandi and S. Ludwig, "Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, pp. 235-245, 2019.

[6] E. Wulczyn, N. Thain and L. Dixon, "Ex Machina: Personal Attacks Seen at Scale," 2016.

[7] "kaggle," [Online]. Available: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data.

[8] P. Mishra, H. Yannakoudakis and E. Shutova, "Tackling Online Abuse: A Survey of Automated Abuse Detection Methods," 2019.

[9] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," in *Proceedings of the NAACL Student Research Workshop*, San Diego, California, 2016.

[10] A.-M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos and N. Kourtellis, "Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior," in *11th International Conference on Web and Social Media, ICWSM 2018*, 2018.

[11] X. Li, S. Chen, X. Hu and J. Yang, "Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift," 2018.

[12] G. Xavier and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research - Proceedings Track*, vol. 9, pp. 249-256, 2010.

[13] K. He, X. Zhang, S. Ren and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *IEEE International Conference on Computer Vision (ICCV 2015)*, vol. 1502, 2015.

[14] P. Mishra, H. Yannakoudakis and E. Shutova, "Neural Character-based Composition Models for Abuse Detection".

[15] Z. Zhang and L. Luo, "Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter," 2018.

[16] M. Todosovska and S. Gievska, "Detection of Abusive Language in OnlineComments," 2018.

[17] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad and Y. Chang, "Abusive Language Detection in Online User Content," *Proceedings of the 25th international conference on world wide web*, pp. 145-153, 2016.

[18] P. Mishra, M. D. Tredici, H. Yannakoudakis and E. Shutova, "Abusive Language Detection with Graph Convolutional Networks," 2019.

[19] J. H. Park and P. Fung, "One-step and Two-step Classification for Abusive Language Detection on Twitter," 2017.

[20] Y. Bengio, P. Simard and P. Frasconi, "Learning long-term dependencies with gradient descent is," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, p. 157–166, 1994.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[22] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," 2014.