Hate Speech Detection

Abdul Waheed

November 15, 2020

0.0.1 Results

Model	Accuracy	F1-Score	Notebook Version
MultinomialNB	0.75	0.71	NA
Ensemble (AdaBoost)	0.75	0.70	NA
BiGRU	0.87	0.93	V4
RoBERTa	0.76	0.85	V5
BERTTweet	0.85	0.92	V6
XLNet + GRU	0.76	0.85	V7

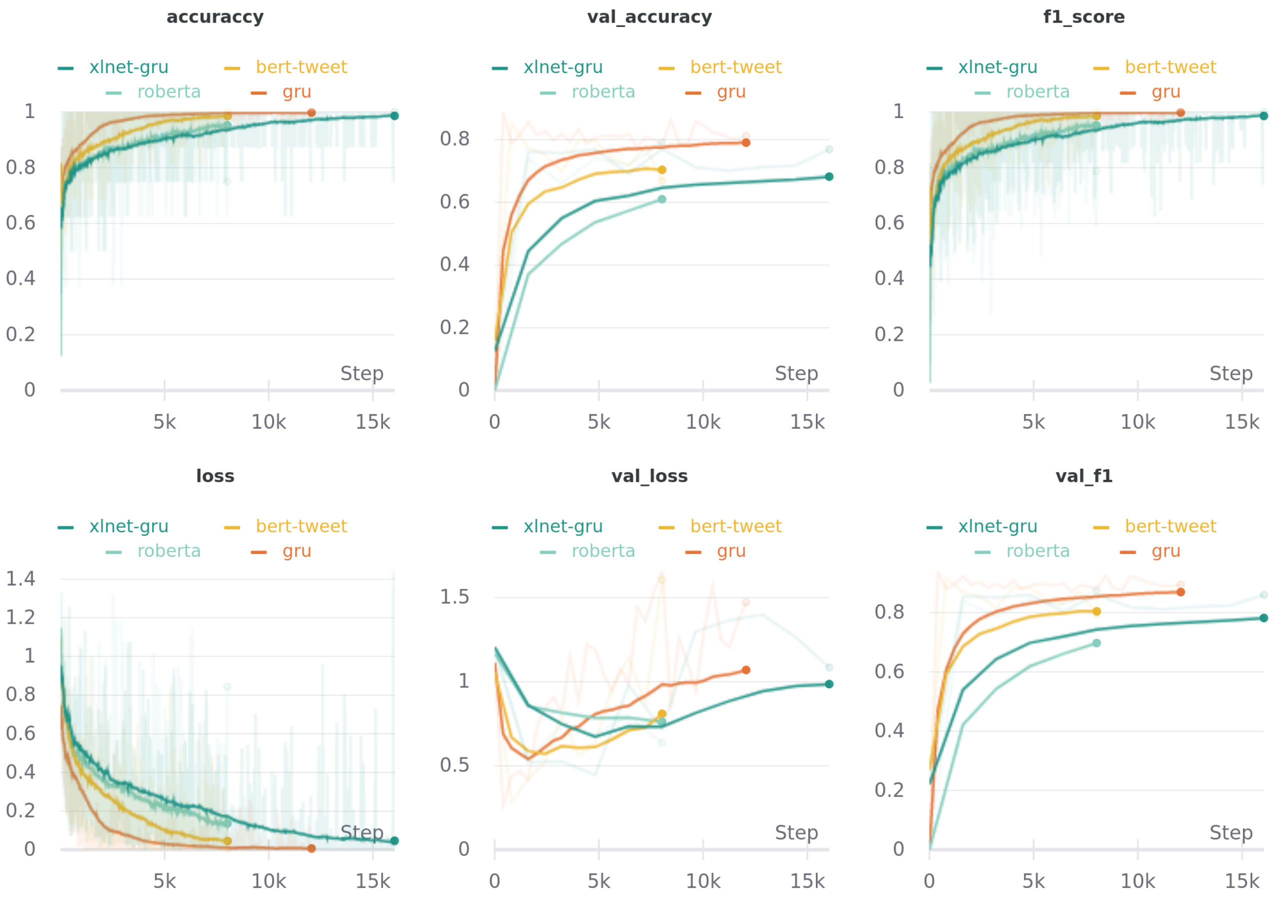
0.0.2 Discussion

The task was text classification problem, which can be solved by any machine learning model, deep learning models, or modern pre-trained nlp models using transfer learning. Out of 6 experimented approaches, first two models in the result table are trained using traditional tf-idf feature vectors and remaining models use high dimensional word embeddings as feature. Surprisingly Bidirectional Gated Recurrent Unit performs best, possibly due to shorter length of sequence which do not causes context loss or vanishing/exploding gradient problem. Another surprise results came out of the experiment is that classical machine learning models with tf-idf vectors are at par with pre-trained large language models (both auto regressive as well as masked). I have used a BERT based model which was pre-trained on ˜1B tweets, as expected it performs quite better and is at par with best performing model. Although these results can be improved by tweaking the training configuration but one can not expect huge performance gain that what makes hate speech detection problem bit complex, hence it becomes very crucial to take care of the nature of the problem during modeling. Large pretrained models seems overfiting on this task due to overparametrization. Here is the possible solutions we can try in order to improve the performance:

- **Data:** If we collect some meta data about user and tweet we can treat this problem as a multimodal which might significantly improve the performance but first we will have to validate the correlation between these meta data and independent variable Label. Also this might induce bias in the system.
- **Representation:** Other than contextualized embedding we can try graph representation both text data as well as meta data and see whether it helps or not.
- **Model**: As we have seen that BERTTweet performs quite well on the task, we can try other similar model and by tweaking the training config we can expect significant improvement in the performance.

Note:These models can be used as baseline.

0.0.3 Training Logs



0.0.4 Tools Used

- **Language:** Python3.6
- **Libraries/Packages:** Transformers, PyTorch, Scikitlearn, PyTorchLightning, WandB
- **Hardware:** 16GB RAM, Core i7 Intel CPU, RTX 2060 (6GB) and V100 GPU (16GB)
- **Others:** JupyterLab, Kaggle, GoogleColab, WandB Logger

0.0.5 Links

- Kaggle Notebook
- WandB Panel
- Github

References

[1] Weights and Biases, https://wandb.com

[2] Transformers by Hugginface https://huggingface.co/transformers/