SemEval-2019 Task 6 Sub-Task A: Emotion Classification with GRU and BERT

Ines Pancorbo
Georgetown University
ip221@georgetown.edu

Abstract

This paper presents my findings and results for SemEval-2019's task 6 sub-task A. Task 6 was based on the Offensive Language Identification Dataset, which contains more than 14,000 English tweets and was itself divided into 3 sub-tasks (A, B, C). GRU, LSTM, and BERT based neural networks were considered to classify tweets. However, BERT was considered to be the recommended approach given its higher macro F1 and accuracy scores of 81.85 percent and 85.81 percent, respectively. Given this F1-score, my proposed approach is ranked between the 1st and 2nd place (out of 104) of the scoreboard of the competition.

1.Introduction

Twitter is usually treated as a platform for online debates, where individuals express their opinions and are therefore, often attacked for doing so. Twitter, as well as other platform providers, usually aims to remove or prevent these attacking posts. Doing so manually can be costly and time-consuming, so automatic detection is a nice alternative.

In this paper, I present my results for the *SemEval 2019 Task: Identifying and Categorizing Offensive Language in Social Media* on the Offensive Language Identification Dataset. This task was divided into 3 subtasks (A, B, C) and I focused on sub-task A, in which the goal was to classify tweets as offensive (OFF) or non-offensive (NOT) posts. Offensive posts include insults, threats, and posts containing any form of untargeted profanity. Each instance is assigned one of the following two labels.

- Not Offensive (NOT): Posts that do not contain offense or profanity;
- Offensive (OFF): Posts that contain any form of profanity or a targeted offense (insults, threats, and posts containing profane language or swear words).

The corpus provided by the organizers consists of 14,100 tweets in English. The data collection methods used to compile the dataset used in *OffensEval* is described in Zampieri et al. The 14,100 English tweets were divided into a training set of 13,249 tweets and a testing set of 860 tweets. See the table below for more details.

	Train	Test	Total
OFF	4,400	240	4,640
NOT	8,840	620	9,460
Total	13,240	860	14,100

Table 1: The distribution of the data

The official evaluation measure for task 6 sub-task A was the macro F1 score. Teams in this competition used models that ranged from traditional machine learning, such as SVM or logistic regression, to deep learning, such as CNN, RNN, Bi-LSTM, attention-based models such as ELMo and BERT. Below is a pie chart summarizing models used.

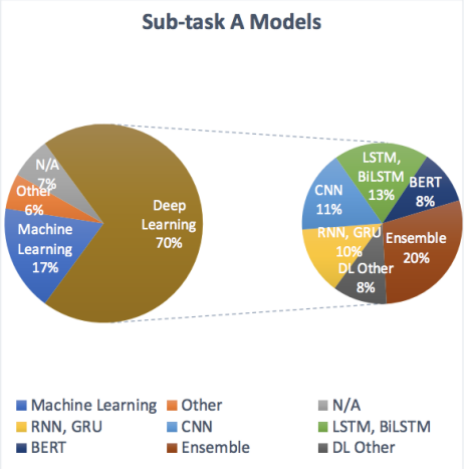


Figure 1: Pie chart of systems used in the competition

Source: Zampieri et al.

104 teams participated in Sub-Task A: Among the top ten teams, seven used BERT but with variations in hyperparameters and approaches to preprocessing. The top team used BERT-base-uncased with default-parameters, trained for 2 epochs, and used a maximum sentence length of 64. The top team achieved an F1 score of 82.9 percent on the test dataset. BERT seemed to perform well on this Sub-Task, as the top nonBERT model was ranked 6th and consisted of an ensemble of CNN, Bi-LSTM and Bi-GRU with Twitter word2wec embeddings. Find below, the scoreboard for Sub-Task A. Further, find below the confusion matrix for the top team (F1 score of 82.9 percent).

Sub-task A	
Team Ranks	**F1 Range**
1	0.829
2	0.815
3	0.814
4	0.808
5	0.807
6	0.806
7	0.804
8	0.803
9	0.802
CNN	**0.800**
10	0.798
11-12	.793-.794
13-23	.782-.789
24-27	.772-.779
28-31	.765-.768
32-40	.750-.759
BiLSTM	**0.750**
41-45	.740-.749
46-57	.730-.739
58-63	.721-.729
64-71	.713-.719
72-74	.704-.709
SVM	**0.690**
75-89	.619-.699
90-96	.500-.590
97-103	.422-.492
All NOT	**0.420**
All OFF	**0.220**
104	0.171

Figure 2: F1-scores in the competition
Source: Zampieri et al. 2019

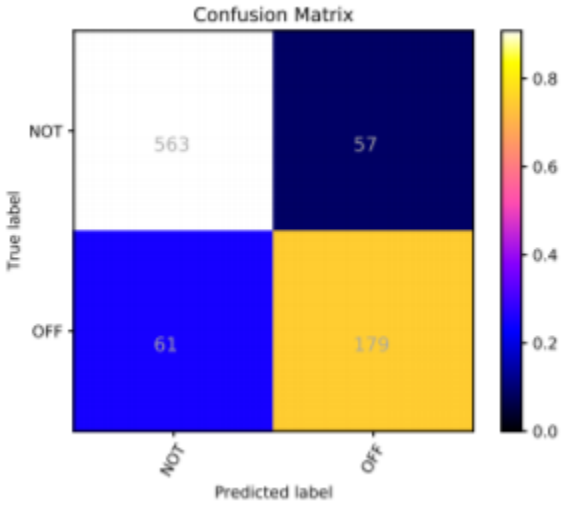


Figure 3: Examples of confusion matrix from the best team
Source: Liu et al. 2019

I mainly experimented with two different types of classifiers. (1) RNN models, making use of bidirectional GRU and LSTM units, due to their capability of sequential processing and ability to retain past information through past hidden states, and (2) a fine-tuned Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2018), therefore avoiding recursion and making use of an encoder/decoder. I ended up relying on BERT given that it reached the best macro F1 score of 81.85 percent when testing the respective models on the test dataset.

2.Approach Description

Sub-task A is a binary classification task. A tweet can be either offensive (OFF) or non-offensive (NOT). The model takes a tweet as input and predicts the corresponding label of that tweet. The given training data was split into 80 percent for training purposes and 20 percent for validation purposes. For this task, the following was performed:

Preprocessing

The labels were transformed from “OFF” / “NOT” to “1” / “0” and emoji unicode was mapped to an English phrase. Lastly, all URL and twitter references, as well as duplicate punctuation and spaces were removed.

Pre-trained word embeddings

Experimentation was done with pre-trained word embeddings, and decisions were made based on the best macro F-1 score on the validation dataset. "Glove.6B.100d" (GloVe as the algorithm, an embedding size of 100, and pre-trained vectors on a 6 billion corpus) was chosen as a result.

2.1 RNN

2.1.1 Model Details

A GRU model was first considered due to its sequential processing ability. The first layer of the GRU model was an embedding layer, initialized with the GloVe 100-dimensional embeddings. “Rnn.pack_padded_sequence” was used on the result of the embedding layer to allow the model to solely process the non-padded elements of the input sequence. The next layer was a bi-directional GRU (hidden size equal to 64, number of layers in RNN unit equal to 2, and dropout in RNN unit equal to 0.2). The output of the GRU layer was a concatenation of the last hidden state from the last word of the post and the hidden state from the first word of the post. A linear layer followed by a sigmoid layer were added subsequently, which produced the final prediction.

The GRU model was trained using 3 epochs, a batch size of 64, Binary Cross Entropy Loss as the loss function, and Adam as the optimizer.

As a note, LSTM units were also considered but they performed slightly worse than GRU units when hyperparameter tuning on the validation set.

2.1.2 Sensitivity Analysis

Sensitivity analyses were conducted on the GRU model’s hyperparameters: hidden size, embedding size, number of layers in Bi-GRU units, batch size, and dropout in Bi-GRU units. The results are depicted below. As can be seen, the model is relatively robust to hyperparameters except, one could argue, to batch size, as we see the greatest fluctuations in F1 score.

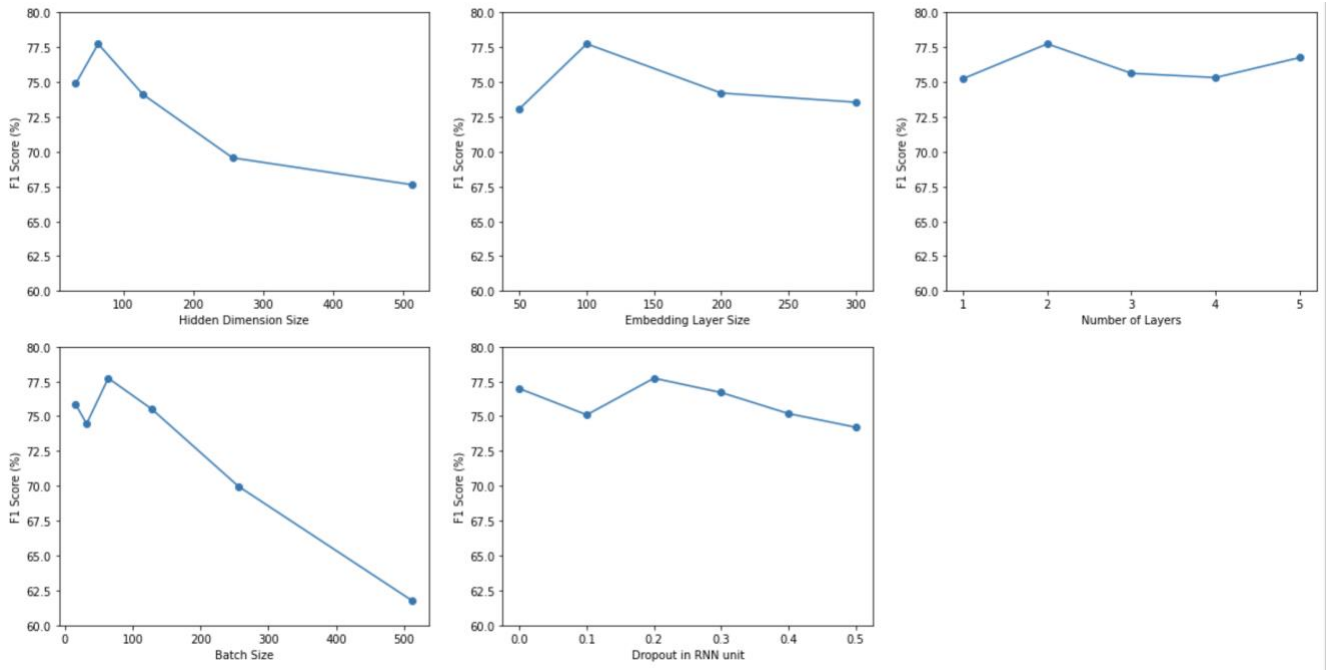


Figure 4: Sensitivity analysis for the Bi-GRU model

2.2 BERT

2.2.1 Model Details

A classifier was also trained by fine-tuning a pre-trained BERT model (huggingface:https://huggingface.co/transformers/index.html) with a linear layer for text sequence classification on top.

Every tweet was used as a text "sentence" of arbitrary length. A maximum tweet length was decided and therefore, shorter tweets were padded.

The BertForSequence Classification model was chosen for training. The documentation for this particular model can be found in the following link: https://huggingface.co/transformers/v2.2.0/model_doc/bert.html.

Consideration was given to "bert-base-uncased" and "bert-large-uncased." When hyperparameter tuning, "bert-large-uncased" did not perform better than "bert-base-uncased" and consequently, chose the smaller model version. The model contained 12 transformer blocks, 12 self-attention heads, and a hidden dimension of 768, which totaled 110 million parameters. The model used BookCorpus (800M words) and the English Wikipedia (2,500M words) as the corpus. Lastly, AdamW was used as the optimizer and Cross-Entropy Loss as the loss function.

2.2.2 Implementation Details

The model was built using PyTorch and a GPU was used for training purposes reducing training time to 7 minutes per epoch. Different batch sizes were considered including 8, 16, 32, 64 and 128 as well as different epochs including 0.9, 1, 2, 3 and 4 and different learning rates including 3e-5, 2e-5, 1e-5, 1e-4, 1e-3, …, etc. The best F1 score on the validation dataset was achieved with 2 epochs, a batch-size of 64, and a learning rate of 2e-5.

In addition, during hyperparameter tuning, the scheduler was also tested. Consideration was given to *get_constant_schedule_with_warmup*, *get_linear_schedule_with_warmup*, *get_cosine_schedule_with_warmup* and *get_cosine_with_hard_restarts_schedule_with_warmup.* The best F1 score on the validation dataset resulted from using the *get_cosine_schedule_with_warmup* scheduler.

Adding weights to the loss function was also considered given the imbalance of the dataset. However, weights did not significantly improve the F1 score on the validation dataset and thus, were not included when testing.

2.2.3 Sensitivity Analysis

Sensitivity analyses were also performed on BERT's hyperparameters, in particular batch size, learning rate, and epochs, to understand the robustness of the model. The results are shown in the graphs below.

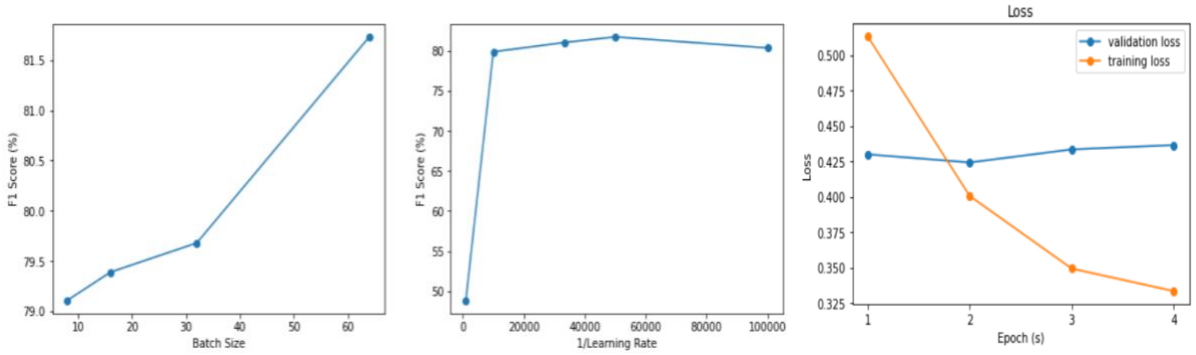


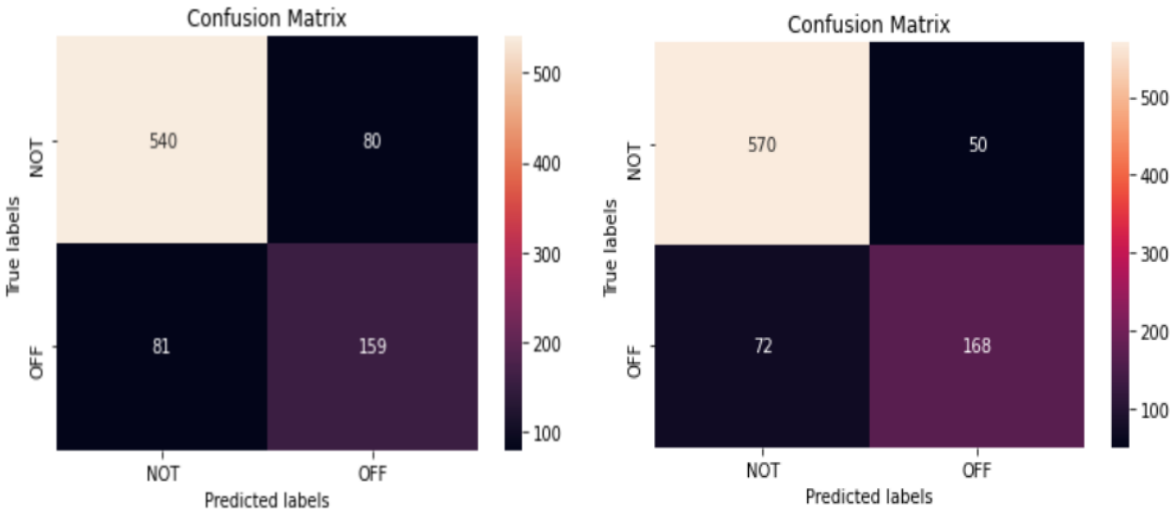
Figure 5: Sensitivity analysis for the BERT model

3. Result Analysis

The results of the respective models on the test set are shown in the table below. As can be seen, BERT outperformed the Bidirectional GRU achieving an F1 score of 81.85 percent and an accuracy of 85.81 percent. The confusion matrices are also shown.

System	F1-score (macro)	Accuracy
Bi-GRU	76.71%	81.28%
BERT	**81.85%**	**85.81%**

Table 2: The F1-scores and accuracies of Bi-GRU and BERT models



Confusion Matrix for Bi-GRU Confusion Matrix for BERT

Figure 6: Confusion matrix for the Bi-GRU model and the BERT model

The table below depicts the performance of the models on each class label (OFF and NOT).

	OFF			NOT		
	Precision	Recall	F1(macro)	Precision	Recall	F1(macro)
Bi-GRU	66.53%	66.25%	66.39%	86.96%	87.10%	87.03%
BERT	**77.06%**	**70.00%**	**73.36%**	**88.79%**	**91.94%**	**90.33%**

Table 3: The F1-scores and accuracies of Bi-GRU and BERT models on “OFF” and “NOT” tweets

From the above table, it can be seen that BERT performed better on both classes with an F1 score of 73.36 percent (OFF label) and 90.33 percent (NOT label). From the table, one can also observe that offensive tweets are relatively harder to classify.

4. Conclusion

This paper addressed the challenges of identifying and classifying offensive tweets. As explained, while both an RNN and a BERT model were considered, the BERT model was better at the binary classification task than the RNN model.

References

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

Liu, Ping, Wen Li, and Liang Zou. "NULI at SemEval-2019 Task 6: transfer learning for offensive language detection using bidirectional transformers." *Proceedings of the 13th International Workshop on Semantic Evaluation*. 2019.

Zampieri, Marcos, et al. “Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval).” arXiv preprint arXiv:1903.08983 (2019).