

Multilabel Text Classification with Transfer Learning

Prepared by
Tasin Hoque (1605029)

Problem Definition

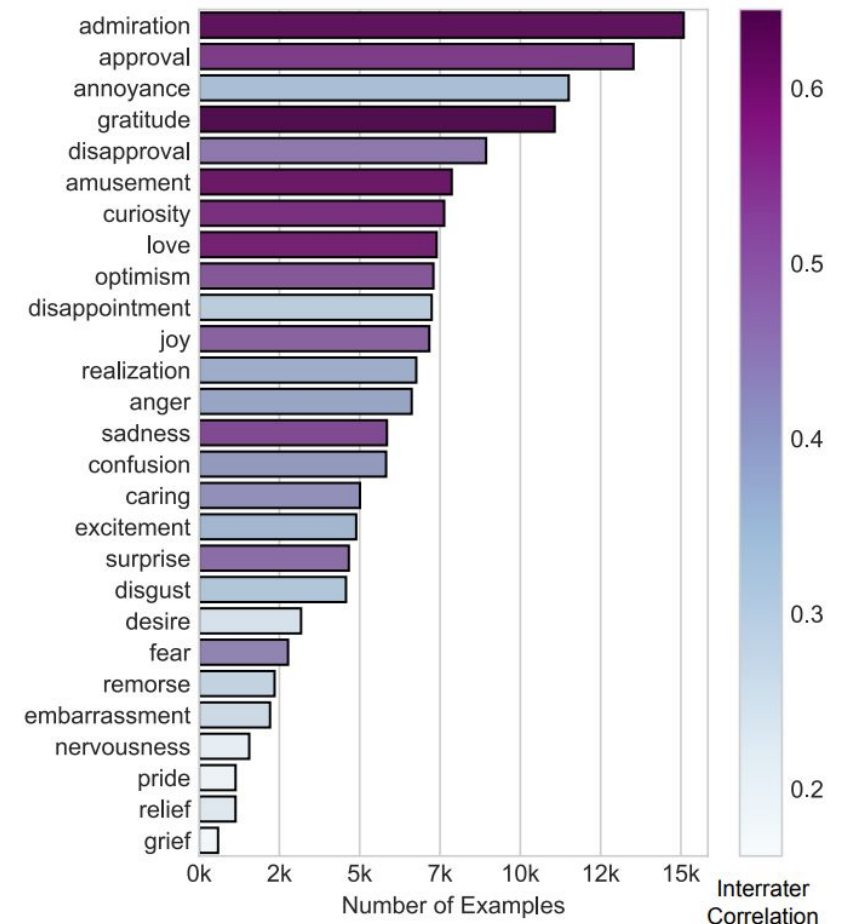
- Given a piece of text, we want to assign emotion labels to it, or mark it as neutral.

Sample Text	Label(s)
OMG, yep!!! That is the final answer. Thank you so much!	gratitude, approval
I'm not even sure what it is, why do people hate it	confusion
Guilty of doing this tbph	remorse
This caught me off guard for real. I'm actually off my bed laughing	surprise, amusement
I tried to send this to a friend but [NAME] knocked it away.	disappointment

Table 1: Example annotations

Dataset

- [GoEmotions](#): A corpus of Reddit comments.
- 58k entries
- 28 categories
- Maximum word sequence length: 30
- 82 unique raters



Approach Overview

- Tokenizing text
- Conversion to word embeddings
- Pretrained transformer as base model
 - DistilBERT
 - RoBERTa Base
 - RoBERTa Large
- Classification layer on top of it
 - Softmax
 - Thresholding (0.15 threshold)
- Fine-tuning after unfreezing the whole model

Loss Function

- Masked Language Modeling (MLM) loss function in original BERT
- Cross-entropy loss function for our case
 - Adapted for multilabel classification by treating each class as a binary classification problem, and computing the loss separately for each class.
- Y be the set of all possible classes
- y_i be the binary label for the i -th class
- z_i be the predicted probability of the input belonging to the i -th class
- L be the loss

$$L = - \sum_{i=1}^{|Y|} [y_i * \log(z_i) + (1 - y_i) * \log(1 - z_i)]$$

Performance on Test Set

- DistilBERT (268 MB)
 - 45% macro-F1
 - 15 epochs, 64 batch size, 5e-5 learning rate
- RoBERTa Base (501 MB)
 - 46.9% macro-F1
 - 4 epochs, 64 batch size, 5e-5 learning rate

Performance on Test Set (Cont.)

- RoBERTa Large (1.43 GB)
 - Run 1
 - 52.9738% ($\approx 53\%$) macro-F1 (4 epochs, 128 batch size, $5e-5$ learning rate), [Checkpoint](#)
 - Run 2
 - 51% macro-F1 (3 epochs, 128 batch size, $5e-5$ learning rate), [Checkpoint](#)
 - 52% macro-F1 (6 epochs, last 3 epochs in a separate run), [Checkpoint](#)
 - 51.8% macro-F1 (9 epochs, in 3 separate runs), [Checkpoint](#)

Curves

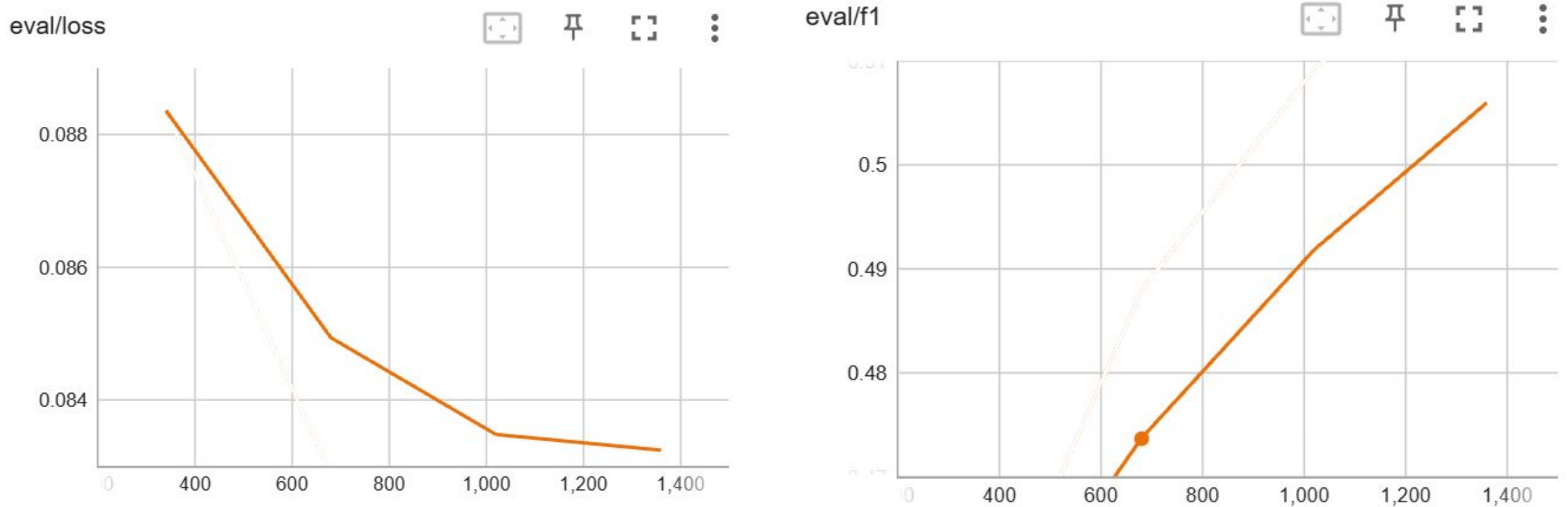


Fig: Loss and F1 curve (validation) for RoBERTa Large (Run 1). The X-axis contains steps, 1400 steps being equivalent to 4 epochs.

Validation Metrics

Epoch	Validation Loss	Accuracy	Precision	Recall	F1
1.0	0.0889	0.4043	0.4807	0.4568	0.4446
2.0	0.0828	0.4113	0.4608	0.5363	0.4868
3.0	0.0813	0.4201	0.5198	0.5612	0.5227
4.0	0.0862	0.4292	0.5012	0.5558	0.5208
5.0	0.0924	0.4329	0.5164	0.5362	0.5151
6.0	0.0956	0.4445	0.5241	0.5328	0.5161
7.0	0.0962	0.4648	0.5138	0.5277	0.5151
8.0	0.0962	0.4462	0.5257	0.5270	0.5203
9.0	0.1029	0.4432	0.5076	0.5249	0.5111

Fig: Validation metrics for RoBERTa Large (Run 2)

Inference

⚡ Hosted inference API ⓘ

🧠 Text Classification

Examples ▾

I like you. I love you

Compute

Computation time on Intel Xeon 3rd Gen Scalable cpu: cached

love	0.981
admiration	0.027
joy	0.013
approval	0.007
gratitude	0.006
disapproval	0.005
realization	0.005
neutral	0.004
excitement	0.004
optimism	0.004

⚡ Hosted inference API ⓘ

🧠 Text Classification

Examples ▾

What a big bird it is

Compute

Computation time on Intel Xeon 3rd Gen Scalable cpu: 0.060 s

admiration	0.331
excitement	0.260
surprise	0.244
neutral	0.176
approval	0.016
curiosity	0.008
joy	0.008
realization	0.005
pride	0.005
annoyance	0.004
optimism	0.004

Comparison

- Baseline model performance (current SOTA): 46% macro-F1
- [HuggingFace best model](#) performance: 53.3% macro-F1
- Our model's performance: 53% macro-F1

Task	Dataset	Model	Metric Name	Metric Value	Global Rank	Result	Benchmark
Emotion Classification	GoEmotions	BERT	Average F1	46	# 1		Compare

Fig: Global ranking of models trained on GoEmotions on “Papers With Code”

Challenges

- During fine-tuning the RoBERTa Large model, a memory error would crash the program at 4th epoch everytime (on Kaggle).
 - Loaded the model from saved checkpoint and resumed training in such a case.
 - Used a fixed seed while shuffling the dataset to avoid any change in order.