Northeastern University
College of Engineering

Course Information

Natural Language Processing

IE 7374-03-17817

FALL 2020

Project Paper

Mitul Shah

Abstract:

Understanding the emotions portrayed in text has many applications, like, sentiment analysis of user-product reviews, analyzing customer feedback, etc.

GoEmotions is a dataset curated by Google, with 58,000 Reddit comments, manually annotated with one or more of 27 emotions, e.g. anger, confusion, love. The goal is to predict the multi-labelled emotions from the textual comments.

In this work, the data is first preprocessed and used the Glove pretrained word vectors to form the embedding matrix for this corpus. The BiLSTM models achieves an average F1-score of 0.2767 across the multi-labels.

Introduction:

One of the many limitations of a machine that limits its ability to explain human behavior is the lack of emotional context. Different users display sentiment in different ways, and many of these comments are very short and contain multiple meanings (which will be labelled as neutral). But, there are also clearly mentioned phrases that indicate exact emotions. The neural approach will take key considerations of all of above factors.

This study has various applications:

- User Feedback- Studying human emotions from short feedback given by users through forms, surveys & reviews
- Social Listening- Understanding emotions to know what people think about certain products or services
- Customer Support Interaction- Evaluating Customer support data like logs & transcripts to understand Customer Emotions

Major Steps in this work:

- The merged data is sampled 20% for faster computation during training
- Using pretrained word embeddings (GloVe) to form the embedding matrix for this corpus
- LSTM model : LSTM is used for learning long-distance dependency between words. It can be a promising algorithm to capture the sentiments from a sentence. The bidirectionality helps us capture relationships between words in both the directions.

Dataset Link:

https://github.com/google-research/google-research/tree/master/goemotions

Research Paper Link: https://arxiv.org/pdf/2005.00547.pdf

The three links (for 3 datasets) provided in the GitHub repository have been first merged and then split in training and testing.

The dataset overview:

text	id	author	ubredd	link_id	parent_	ated_	ater_i	very_	admiration	amusement	anger	annoyance	approval	caring	confusion	
He isn't as big, but he'…	ecz…	Ral…	cri…	t3_…	t1_…	1.5…	3	Fal…	0	0	0	0	0	0	0	0
That's crazy; I went to a …	ed5…	Bea…	Tee…	t3_…	t1_…	1.5…	23	Fal…	0	1	0	0	0	0	0	0
that's adorable asf	ef9…	Red…	tra…	t3_…	t3_…	1.5…	73	Fal…	0	1	0	0	0	0	0	0
"Sponge Blurb Pubs Quaw Ha…	ed1…	Tia…	you…	t3_…	t1_…	1.5…	54	Fal…	0	1	0	0	0	0	0	0
I have, and now that you…	ed9…	[de…	Ask…	t3_…	t1_…	1.5…	36	Fal…	0	0	0	0	0	0	0	0
I wanted to downvote thi…	ee5…	Son…	tim…	t3_…	t1_…	1.5…	81	Fal…	0	0	0	0	0	0	0	0

Background:

- Research work by Dorottya Demszky and others show that their fine-tuned BERT achieves an average F1-score of .46 (std=.19). The model obtains the best performance on emotions with overt lexical markers, such as gratitude (.86), amusement (.8) and love (.78). The model obtains the lowest F1-score on grief (0), relief (.15) and realization (.21), which are the lowest frequency emotions. The BiLSTM model was also used for the task which performed significantly worse than the BERT model.
- It was also discovered that less frequent emotions tend to be confused by the model with more frequent emotions related in sentiment and intensity (e.g., grief with sadness, pride with admiration, nervousness with fear).
- Transfer learning experiments were also conducted to show that the data generalizes across domains and taxonomies.
- Only batch size and learning rate parameters were used for tuning the model. Tuned params: epochs =4, batch size =16, lr =5e-5 yields best performance

Approach:

- Data Loading:. Fraction (20%) of the entire data is only used, since the size of data is huge and training time would be 2+ hours. Taking fraction decreased the training time to 20mins.
- Data Preprocessing:

1. A small list of stop words is manually written to avoid losing relevant information necessary for sentiment analysis. Some stopwords have been removed by glancing at the dataset. This stopwords removal process will increase model performance.
2. The text is converted to lower case and then tokenized.
3. Choosing num_words: num_words is the amount of words in the dictionary. This was chosen by looking at the tokenizer results. After the first 7000 most occurring words in the dataset, we start to hit typos, strange acronyms and words that are only used once. The result of this is that the amount of words covered by the word embedding accuracy, increases from 62% to 95%.
4. ,maxlen: maxlen is the maximum length of the input text from reddit. maxlen is chosen as 20 to avoid information loss

- Word Embedding:
  1. Pretrained word vectors from GloVe (with 300 Dimensions) are used for creating embedding matrix
  2. The 0^{th} index is reserved for padding and last index for out of the word vocabulary
- Model Building:

  Layer 1: An embedding layer of a vector size of 300 and a max length of each sentence is set to 20.

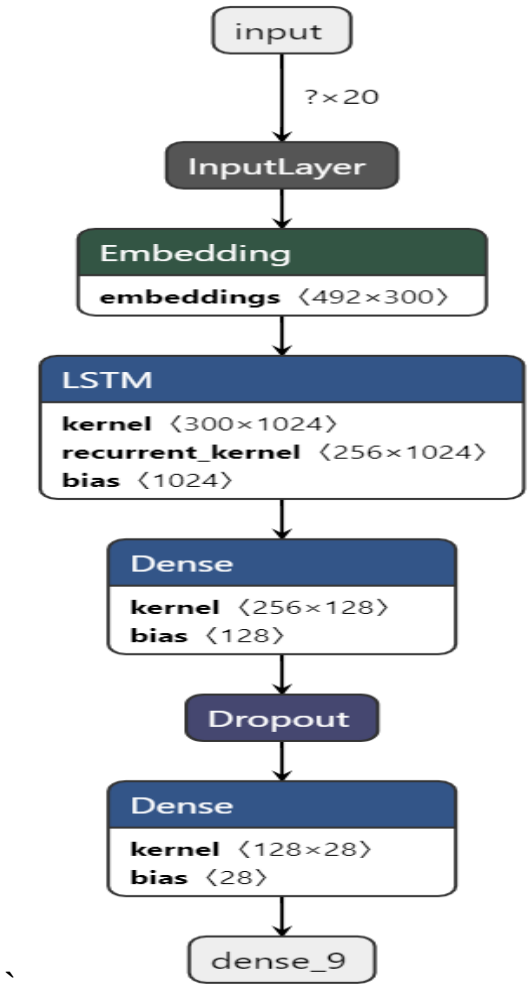
  Layer 2: 256 cell BiLSTM layers, where the embedding data is fed to the network. We add a dropout of 0.2 this is used to prevent overfitting.

  Layer 3: A 128 layer dense network which takes in the input from the BiLSTM layer. A Dropout of 0.5 is added here.

  Layer 4: A 28 layer dense network with softmax activation, each class is used to represent a sentiment category

  Finally, the model is combined with an adam optimizer and binary_crossentropy loss function, for multi-label & multi-class classification

  Once the model is trained and the number of epochs are optimized where difference between training and validation error is minimum. After that we



Structure of the Model

The model is initially fit epochs =15 and batch size =100. The batch size has not been optimized due to computational constraints.

```
Epoch 4/15
296/296 [==============================] - 229s 773ms/step - loss: 0.1182 - accuracy: 0.4162 - val_loss: 0.1259 - val_acc
uracy: 0.3886
Epoch 5/15
296/296 [==============================] - 189s 637ms/step - loss: 0.1131 - accuracy: 0.4379 - val_loss: 0.1267 - val_acc
uracy: 0.3856
Epoch 6/15
296/296 [==============================] - 187s 633ms/step - loss: 0.1079 - accuracy: 0.4559 - val_loss: 0.1307 - val_acc
uracy: 0.3842
Epoch 7/15
296/296 [==============================] - 187s 632ms/step - loss: 0.1028 - accuracy: 0.4812 - val_loss: 0.1345 - val_acc
uracy: 0.3726
Epoch 8/15
296/296 [==============================] - 187s 631ms/step - loss: 0.0979 - accuracy: 0.5016 - val_loss: 0.1366 - val_acc
uracy: 0.3645
Epoch 9/15
296/296 [==============================] - 188s 635ms/step - loss: 0.0932 - accuracy: 0.5227 - val_loss: 0.1391 - val_acc
uracy: 0.3566
Epoch 10/15
296/296 [==============================] - 187s 632ms/step - loss: 0.0891 - accuracy: 0.5431 - val_loss: 0.1508 - val_acc
```

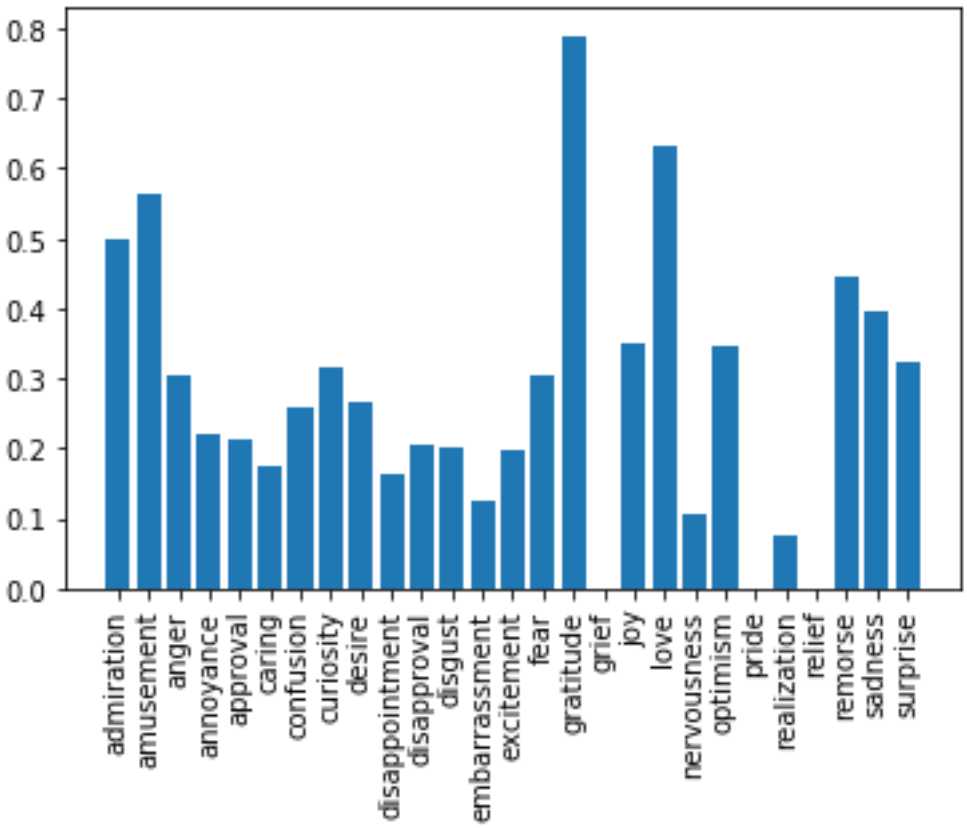
Number of Epochs = 7

The threshold for classifying the emotions can be determined comparing the F1-score for each threshold. The below results show the same:

```
Threshold: 0.1000, Precision: 0.2816, Recall: 0.5573, F1-measure: 0.3742
Threshold: 0.2000, Precision: 0.3794, Recall: 0.4034, F1-measure: 0.3910
Threshold: 0.2500, Precision: 0.4196, Recall: 0.3533, F1-measure: 0.3836
Threshold: 0.3000, Precision: 0.4521, Recall: 0.3123, F1-measure: 0.3694
Threshold: 0.4000, Precision: 0.5081, Recall: 0.2476, F1-measure: 0.3329
Threshold: 0.5000, Precision: 0.5469, Recall: 0.1907, F1-measure: 0.2828
Threshold: 0.6000, Precision: 0.5794, Recall: 0.1458, F1-measure: 0.2329
Threshold: 0.7000, Precision: 0.6127, Recall: 0.1077, F1-measure: 0.1831
Threshold: 0.8000, Precision: 0.6413, Recall: 0.0703, F1-measure: 0.1268
Threshold: 0.9000, Precision: 0.6518, Recall: 0.0345, F1-measure: 0.0655
```

Choosing suitable threshold = 0.2 (This where the F1-score is max)

In this project, F1 score has been used extensively for model evaluation since the data is very imbalanced and F1-score also gives importance to False Negatives and False positives that are very crucial for sentiment analysis.



- Thus we attain an mirco-averaged F1-score across multilabels = 0.3910

- As anticipated, some emotions (amusement, gratitude, love) are way easier to predict than subtle emotions like disappointment, grief, relief
- Threshold = 0.2 gets us the optimal F1-score

Conclusion:

- The preprocessing steps is very crucial because over pre-processing the text will lead to loss of context words. The vocabulary size can be increased to capture more words until the frequency of word does not become singular.
- Model building techniques are studied from different works and general practices. Most common general practices and techniques for building LSTM model are applied in this work.
- The state of the art fined-tuned BERT model achieves F1-score = 0.45 while the BiLSTM in this project achieves F1-score = 0.39 with partial tuning. Further tuning of BiLSTM will result improve the performance.
- Further, state of the art BERT model can be used for making predictions when larger computational power is available
- Further, transfer learning experiments can be conducted, where annotated data from different Emotional Benchmark Datasets can be used to evaluate or further train the model.

References:

@inproceedings{demszky2020goemotions,
 author = {Demszky, Dorottya and Movshovitz-Attias, Dana and Ko, Jeongwoo and Cowen, Alan and Nemade, Gaurav and Ravi, Sujith},
 booktitle = {58th Annual Meeting of the Association for Computational Linguistics (ACL)},
 title = {{GoEmotions: A Dataset of Fine-Grained Emotions}},
 year = {2020}
}