Understanding Human Emotions: The DistilBERT-CNN Way

Manas Ranjan Sahoo

Chelsea Salyards

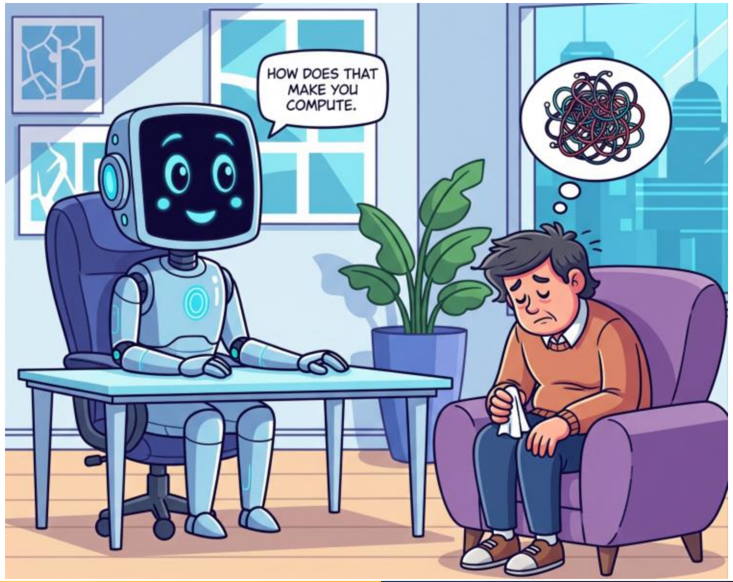
Deep Learning (Summer Semester, 2025)

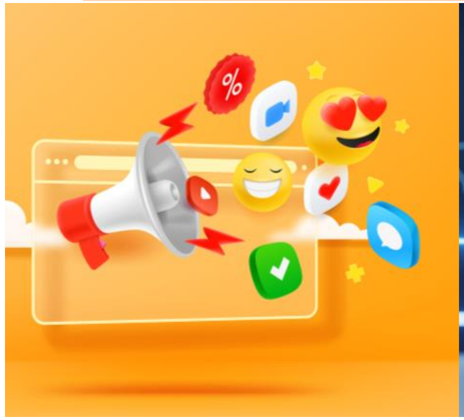
Dr. Youakim Badr

PennState

Why Fine-Grained Sentiment Analysis?

- Sentiment analysis is crucial for modern AI
- Vital for applications like AI therapists, chatbots, social media monitoring for harmful content, recommendation engines
- Elevating AI from transactional to more human-like empathetic mode will increase adoption

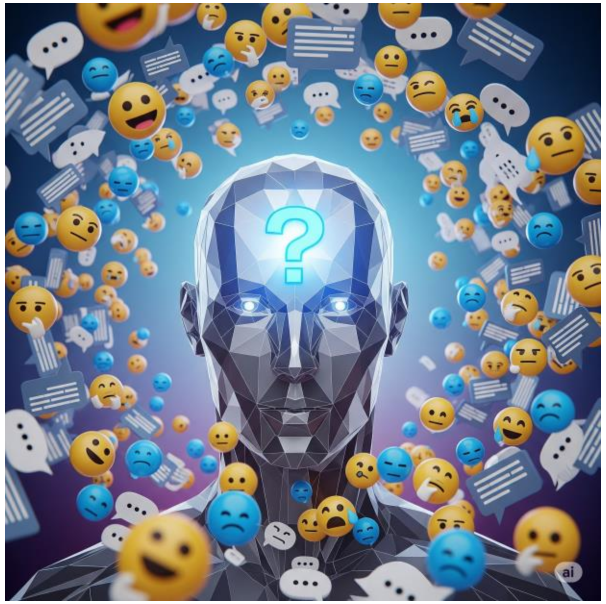




AI Generated Images

Challenges in Emotion Recognition

- Despite decades of research, accurate sentiment analysis is still challenging
  - Nuances like sarcasm and irony
    That's a brilliant idea !
  - Importance of context
    That's just great!
  - Cultural and linguistic differences
    Gift (EN) vs Gift (Nordic)
  - Lack of high-quality, large-scale dataset


AI Generated Image

Opportunity At Hand

- GoEmotions is the largest human-annotated dataset of its kind
  - Made up of 211,225 entries from Reddit
  - Multiple raters rated 58,000 of these entries
  - Rated entries were used to train their original BERT model for sentiment analysis.
- Our objective – Improve the existing sentiment analysis model using a novel deep learning architecture

Positive		Negative		Ambiguous
admiration 👏	joy 😃	anger 😡	grief 😢	confusion 😕
amusement 😂	love ❤️	annoyance 😒	nervousness 😬	curiosity 🤔
approval 👍	optimism 🤞	disappointment	remorse 😔	realization 💡
caring 🤗	pride 😌	disapproval 👎	sadness 😞	surprise 😲
desire 😍	relief 😅	disgust 🤮		
excitement 🤩		embarrassment 😳		
gratitude 🙏		fear 😨		

Credit: [Neurohive.io]

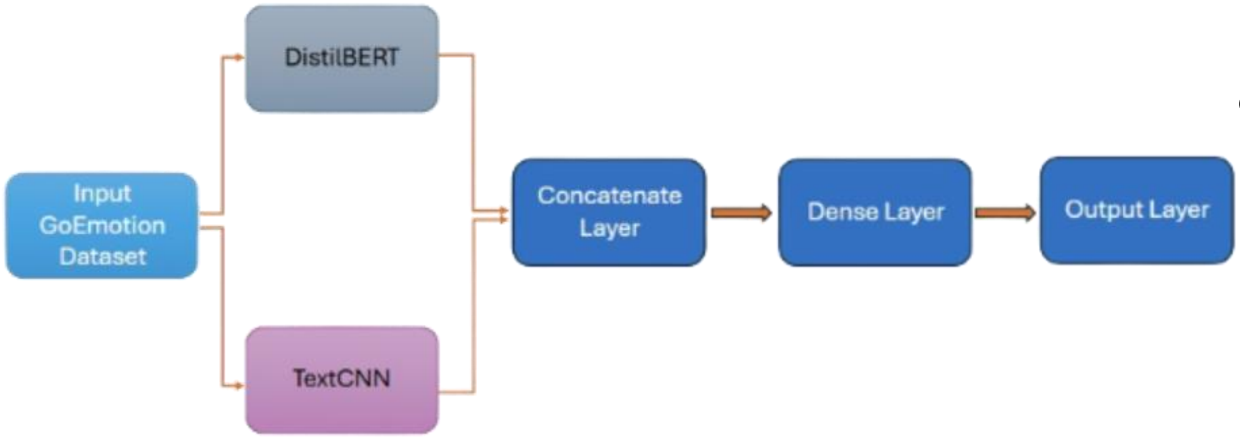
PennState

Overview of Solution / Contributions

Creating a novel deep learning architecture

Prior Work (Google Research)

- BERT – large transformer/encoder-based model widely used in natural language processing applications



Initial Draft Architecture

Our Modifications

- Upgrading the architecture to a hybrid multi-input single-output model
    - Use DistilBERT as an input replacing BERT for a faster equally-effective version to capture global and local patterns
    - Use TextCNN as another input using CNN with an added focus to capture local context through n-grams

PennState

Architecture In Details

- Replacing BERT
  - BERT is a very large model, and our machines do not have the resources to run it effectively
  - To mitigate this, chose to migrate to smaller BERT variant, DistilBERT
  - 40% smaller than BERT
  - 60% faster in inference timing
  - Still retains 97% of the F1-score of the normal BERT implementation (Sanh et al. 2019)

PennState

Architecture In Details

- Adding a CNN input arm
  - Various studies supported the idea of hybrid models (including BERT-CNN models having superior results to BERT only models
    - Abas et al.
    - Duong, Lebret, and Abererécole.
    - Gou, Lei.
  - Chose TextCNN as basis for implementation
    - CNN originally invented for computer vision
    - TextCNN is a variant that was designed for use with text

PennState

Architecture In Details

- Interpreting the results
  - The output is a multilabel indicator matrix
  - Predicts one or many of the 28 emotions (27 emotions + neutral) as an outcome

Positive		Negative		Ambiguous
admiration 👏	joy 😃	anger 😡	grief 😢	confusion 😕
amusement 😂	love ❤️	annoyance 😒	nervousness 😬	curiosity 🤔
approval 👍	optimism 🤞	disappointment	remorse 😔	realization 💡
caring 🤗	pride 😌	disapproval 👎	sadness 😞	surprise 😲
desire 😍	relief 😅	disgust 🤮		
excitement 🤩		embarrassment 😳		
gratitude 🙏		fear 😨		

Credit: Neurohive.io

Related solutions / State of Art

- Guo, Yuan and Li, Hongmei. "Research on Sentiment Analysis of Online Course Evaluation Based on EN-BERT-CNN"
  - Uses BERT-CNN hybrid model to learn sentiment from a dataset of online course reviews
  - Evaluates BERT-CNN versus other hybrid architectures
  - Recognizes strengths of BERT-CNN, but notes weakness to bad datasets
- Wang, et al. "A BERT-based sentiment analysis model for depressive text"
  - Evaluates using plain BERT for conducting sentiment analysis to identify depression in written text
  - Acknowledges the issue of unbalanced data and utilizes data augmentation as a solution
    - Also acknowledges using weights as another solution, which was the methodology used in GoEmotions

Related solutions / State of Art

- Kian Long Tan, Chin Poo Lee, Kalaiarasi Sonai Muthu Anbananthen, and Kian Ming Lim. "RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis With Transformer and Recurrent Neural Network"
    - Involves using a different type of BERT hybrid architecture – using LSTM
    - Uses RoBERTa BERT variant, an improved BERT model that is more effective than original, but very large
    - Methodology with LSTM did prove effective, which shows potential avenues for improvement on machines with greater resources/more time to train
- Kokab, et al. "Transformer-based deep learning models for the sentiment analysis of social media data"
    - Another alternative hybrid architecture using BERT; uses BERT and CBRNN
    - Describes the advantages of such an architecture and strengths/weaknesses of others

PennState

Problem and Challenges

- Compatibility issues with outdated APIs (Tensorflow 1 and Estimators) in original GoEmotions project
  - Refactored code to Tensorflow 2 and Keras
- Compatibility issues with TextCNN project (written in PyTorch)
  - Refactored code to Tensorflow/Keras
- HuggingFace updates
  - HuggingFace very recently updated their model weight format which broke our code late in our development
    - Unexpected complication after we thought our code was essentially finalized
    - Needed to pull newly released version of safetensors to resolve
    - Needed to alter code to download/store pretrained DistilBERT model instead of redownloading it every time to mitigate this from happening again

PennState

Problem and Challenges

- System limitations
  - Intended to resolve by just migrating to smaller BERT model: DistilBERT, still struggled
  - Ultimately migrated code to Jupyter Notebook and running code in cloud environment

Data Collecting

- Data was sourced by GoEmotions team consisting of researchers from a collaboration of Stanford Linguistics and Google Research
- Final dataset contains 58,000 Reddit comments labeled with 27 emotion categories + neutral
    - Selected subreddits based on popularity, likelihood of demonstrating a variety of emotions versus neutral or overly positive/negative tones, less likelihood of offensive/harmful content
    - Comments categorized by at least three independent raters
        - More added if raters did not agree

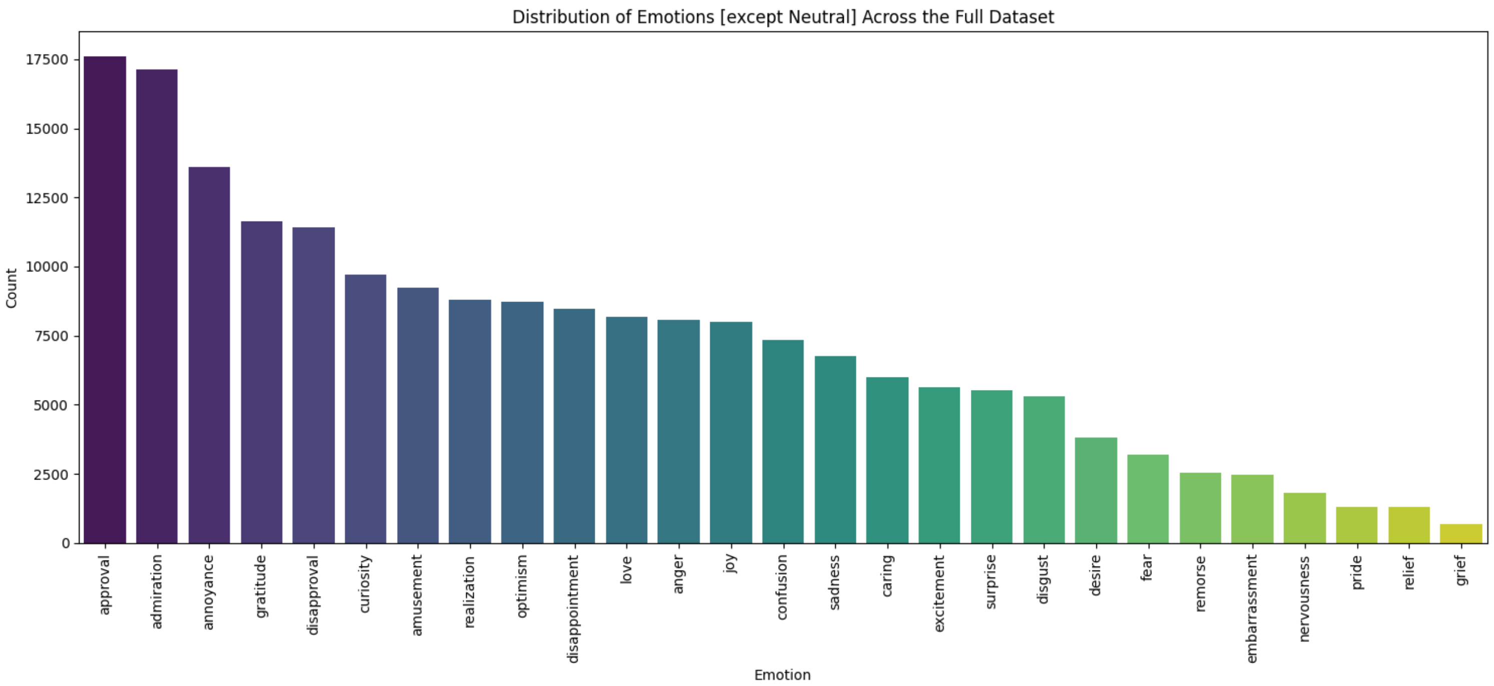
Number of examples	58,009
Number of emotions	27 + neutral
Number of unique raters	82
Number of raters / example	3 or 5
Marked unclear or difficult to label	1.6%
Number of labels per example	1: 83% 2: 15% 3: 2% 4+: .2%
Number of examples w/ 2+ raters agreeing on at least 1 label	54,263 (94%)
Number of examples w/ 3+ raters agreeing on at least 1 label	17,763 (31%)

Credit: Google Research

Data Preprocessing

Data cleanup:

- Original GoEmotions team performed several tasks to preprocess their data
  - Removed offensive/harmful words from collected data, while keeping profanity to help train negative emotions
  - Removed and filter text that was too ambiguous for the raters to detect emotion
  - Data regularization on Reddit comment length to down sample to more consistent number of tokens



Data Preprocessing

- Handling uneven categories through sentiment relationship:
  - Number of examples in emotion categories remain uneven, even with specific data selection efforts
  - Original team used weights and bias algorithms to allow emotion categories with rarer labels to learn from related emotions
    - Sentiment grouping
      - All 27 emotions used as categorization labels were organized into higher level groupings, and model would learn from those groupings
      - Model can extract info and train based on other emotions
  - Additional class imbalance is handled through a weighted loss function in the custom loss function to penalize misclassification in minor classes.

positive	amusement, excitement, joy, love, desire, optimism, caring, pride, admiration, gratitude, relief, approval
negative	fear, nervousness, remorse, embarrassment, disappointment, sadness, grief, disgust, anger, annoyance, disapproval
ambiguous	realization, surprise, curiosity, confusion

PennState

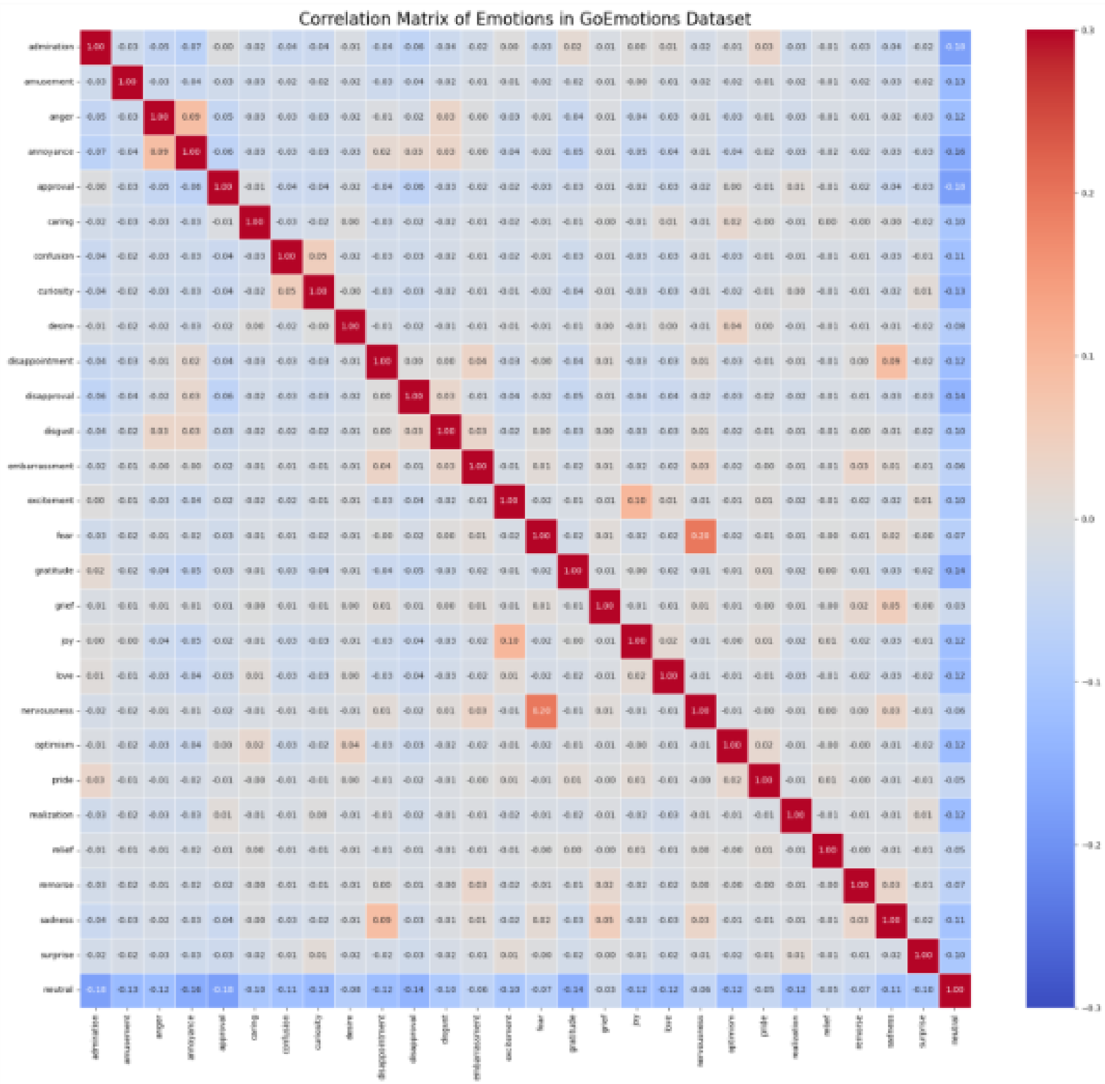
Data Preprocessing

- Handling uneven categories through emotion correlation
  - Correlation grouping
    - While the original work did mention about correlation, the correlation related code wasn't available for use.
    - Recreated the functionality using the details mentioned in the original work
    - Purpose is to show where emotions tend to be expressed together, demonstrating different type of relationship
    - Helps add another valid training point to add learning to rarer emotion categories

PennState

Data Preprocessing

- Correlation matrix demonstrates calculated relationships
- Highlights of highest values of correlated emotions:
  - Nervousness and fear: .2
  - Joy and excitement: .1
  - Anger and annoyance: .09
  - Disappointment and sadness: .09



PennState

Custom Function Utilization

Custom functions were used to handle data imbalance and fine tune training

- Custom loss function – Incorporates regularization apart from loss calculation
    - Calculates base loss
    - Add weighted-loss function based on class distribution penalizing loss in minority classes
    - Add sentiment regularization if model predicts different sentiment groups together
    - Add correlation regularization if model predicts less correlated emotions together
    - Activation function swaps based on multilabel flag in our config
        - Multilabel: Sigmoid activation function
        - Single label: SoftMax activation function
        - Focused on multilabel classification for our project's evaluation

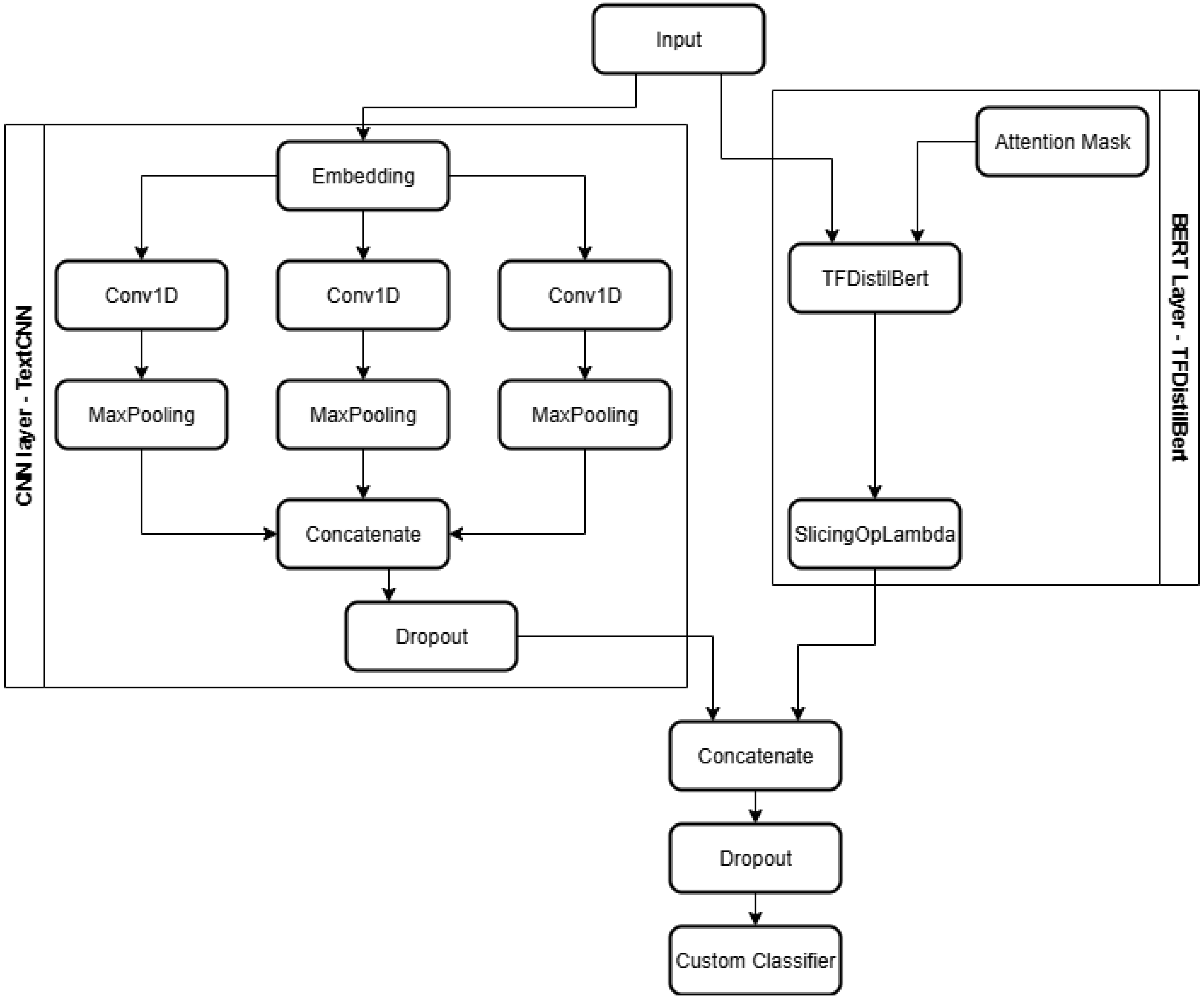
PennState

Custom Function Utilization

Custom functions (cont.)

- Custom learning rate function for AdamW – creates a 2 stage learning rate schedule
    - Starts with warmup for a few steps to stabilize the model (0 to a fixed learning rate)
    - Fine tunes the model for the remaining number of steps with a decay rate (fixed learning rate to 0)
- Custom classifier function – Used in place of Keras Dense layer to predict the emotion using custom weights and biases.
- Custom metrics function - Standard F1 metrics are better for single-label problems and a weighted F1 score metric is better suited for multi-label class imbalanced cases.

Methodology



Final Network Architecture

Multi-Input Single-Output Hybrid Model

- Input 1 – DistilBERT
- Input 2 – TextCNN
- Output – Multi-label emotion prediction matrix

Methodology

TFDistilBert layer

- Input & Attention Mask – Tokenized text and Attention vector for model to focus on real content and not padding
- TFDistilBert – Using self-attention mechanism find the importance of each word with respect to other words to understand global context.
- SlicingOpLambda - To extract the embedding of the first token as a summary vector of text.

TextCNN layer

- Designed to capture local patterns & features through n-grams
- Embedding layer – tokenizes words into dense vector
- Conv1D – use filters of different sizes to capture local patterns
- MaxPooling – Reduces dimensionality keeping most significant feature
- Concatenate – Create a summary of the most important local features or patterns across all filters.
- Dropout – Regularization technique to prevent overfitting.

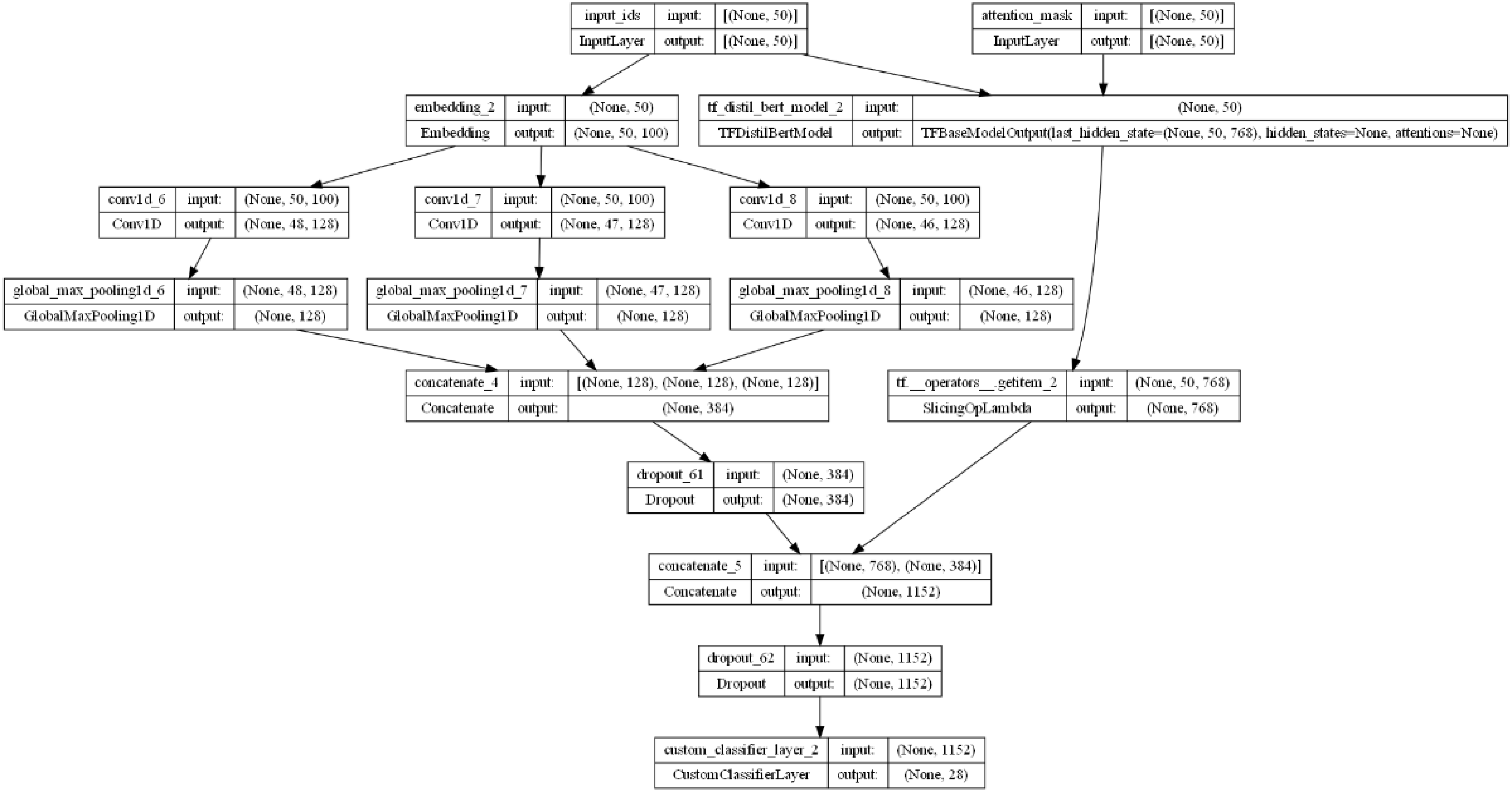
Methodology

Combined Layer

- Concatenate – Output vector of TextCNN (local context) and output vector of DistilBert (global context) are combined into single vector.
- Dropout – Regularization layer to prevent overfitting
- Custom Classifier - The final vector passes through the custom classifier to provide emotion prediction

PennState

Methodology



PennState

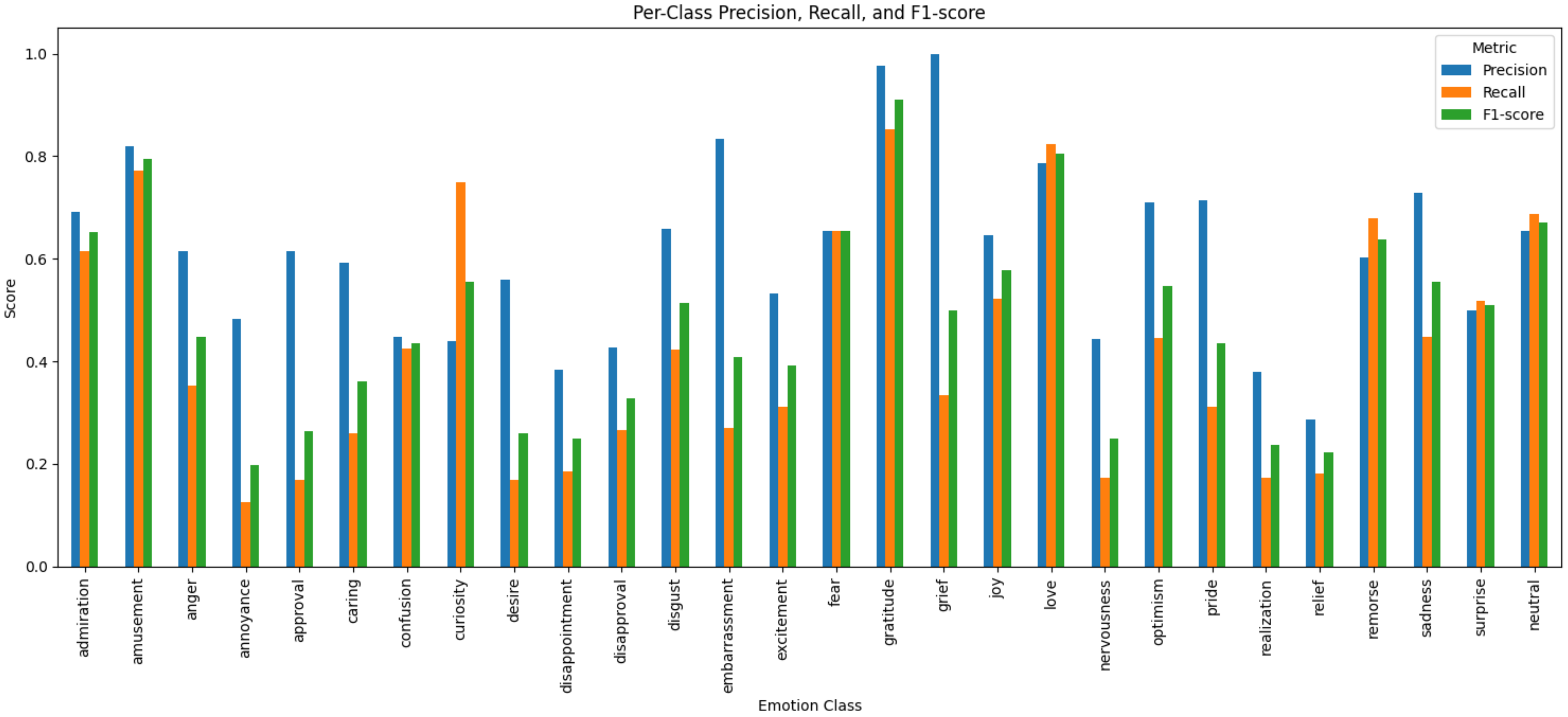
Network Training – validation - testing

- The complete labeled dataset of 58000 records were split as:
- Train – 80%
- Validation – 10%
- Test – 10%
- Additional configurations:
  - Batch size – 16
  - Initial Learning Rate – 5e^-5
  - Learning Rate Warmup – 10% of steps
  - Epochs - 20
  - Early Stopping Patience (to prevent overfitting) – 4 epochs

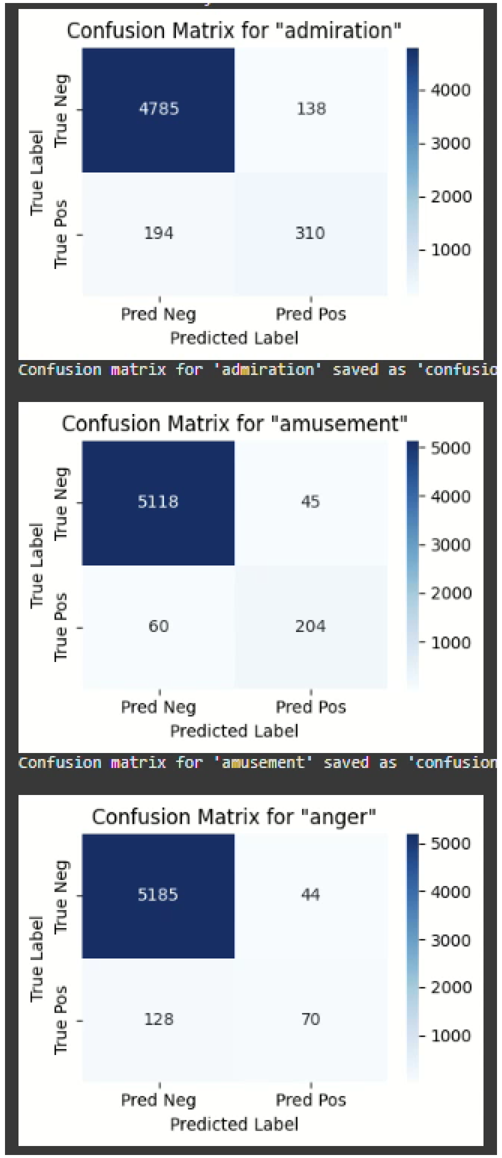
PennState

Outcomes/Findings

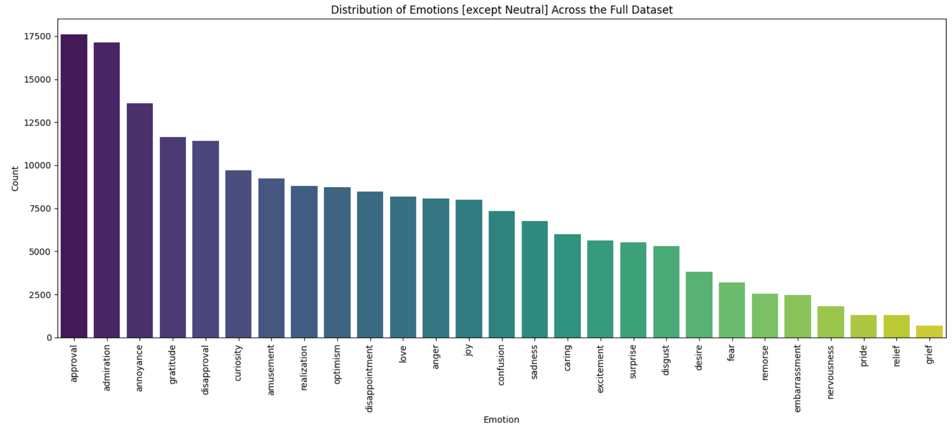
	Precision	Recall	F1-score
admiration	0.691964	0.615079	0.651261
amusement	0.819277	0.772727	0.795322
anger	0.614035	0.353535	0.448718
annoyance	0.481928	0.125	0.198511
approval	0.614583	0.168091	0.263982
caring	0.59322	0.259259	0.360825
confusion	0.448276	0.424837	0.436242
curiosity	0.440083	0.75	0.554688
desire	0.56	0.168675	0.259259
disappointment	0.383562	0.18543	0.25
disapproval	0.427711	0.265918	0.327945
disgust	0.658228	0.422764	0.514851
embarrassment	0.833333	0.27027	0.408163
excitement	0.533333	0.31068	0.392638
fear	0.653846	0.653846	0.653846
gratitude	0.977199	0.852273	0.91047
grief	1.0	0.333333	0.5
joy	0.646154	0.521739	0.57732
love	0.787149	0.823529	0.804928
nervousness	0.444444	0.173913	0.25
optimism	0.709402	0.446237	0.547855
pride	0.714286	0.3125	0.434783
realization	0.378788	0.172414	0.236967
relief	0.285714	0.181818	0.222222
remorse	0.603175	0.678571	0.638655
sadness	0.729167	0.448718	0.555556
surprise	0.5	0.51773	0.508711
neutral	0.653539	0.687185	0.66994



Outcomes/Findings



- Confusion Matrices were created for each emotion to measure precision, recall and F1 score for each emotion
- Metrics weren't entirely related to the frequency of available data as some classes with low data count also has high F1 score
- Metrics weren't entirely related to the sentiment groups of ambiguous as well
- We believe metrics may be related to a combination of number of samples and interrater correlation.
- We also believe F1-score is not always a sufficient criteria for datasets with low data counts for certain classes (e.g., grief)



Positive		Negative		Ambiguous
admiration 👏	joy 😃	anger 😡	grief 😢	confusion 😕
amusement 😂	love ❤️	annoyance 😒	nervousness 😬	curiosity 🤔
approval 👍	optimism 🤞	disappointment	remorse 😔	realization 💡
caring 🤗	pride 😌	disapproval 👎	sadness 😞	surprise 😲
desire 😍	relief 😅	disgust 🤮		
excitement 🤩		embarrassment 😳		
gratitude 🙏		fear 😨		

PennState

Evaluations – Performance Comparison

```
--- Model Performance Summary ---

Emotions the model understands better (higher F1-score):
           Precision    Recall  F1-score
gratitude   0.977199  0.852273   0.91047
love        0.787149  0.823529  0.804928
amusement   0.819277  0.772727  0.795322
neutral     0.653539  0.687185   0.66994
fear        0.653846  0.653846  0.653846
admiration  0.691964  0.615079  0.651261
remorse     0.603175  0.678571  0.638655
joy         0.646154  0.521739   0.57732
sadness     0.729167  0.448718  0.555556
curiosity   0.440083      0.75  0.554688

Emotions the model struggles with (lower F1-score):
               Precision    Recall  F1-score
excitement      0.533333   0.31068  0.392638
caring           0.59322  0.259259  0.360825
disapproval     0.427711  0.265918  0.327945
approval        0.614583  0.168091  0.263982
desire              0.56  0.168675  0.259259
nervousness     0.444444  0.173913      0.25
disappointment  0.383562   0.18543      0.25
realization     0.378788  0.172414  0.236967
relief          0.285714  0.181818  0.222222
annoyance       0.481928     0.125  0.198511
```

Key Findings

- Best performance - gratitude, love, and amusement (higher F1-scores).
- Poor performance – realization, relief, and annoyance (lower F1-scores).
- Low data support possible reason behind lower F1-scores for emotions like nervousness, relief, embarrassment.
- Trade-offs observed - "curiosity" has high recall but lower precision, while "approval" has high precision but low recall.
- The "neutral" emotion has a moderate F1-score, with decent number of False Negatives suggesting failure to identify neutral instances by model but at the same time we see higher values of False Positives as well showing a certain 'bias' for "neutral" emotion.

PennState

Evaluations – Performance Comparison

Emotion	Precision	Recall	F1
admiration	0.53	0.83	0.65
amusement	0.70	0.94	0.80
anger	0.36	0.66	0.47
annoyance	0.24	0.63	0.34
approval	0.26	0.57	0.36
caring	0.30	0.56	0.39
confusion	0.24	0.76	0.37
curiosity	0.40	0.84	0.54
desire	0.43	0.59	0.49
disappointment	0.19	0.52	0.28
disapproval	0.29	0.61	0.39
disgust	0.34	0.66	0.45
embarrassment	0.39	0.49	0.43
excitement	0.26	0.52	0.34
fear	0.46	0.85	0.60
gratitude	0.79	0.95	0.86
grief	0.00	0.00	0.00
joy	0.39	0.73	0.51
love	0.68	0.92	0.78
nervousness	0.28	0.48	0.35
neutral	0.56	0.84	0.68
optimism	0.41	0.69	0.51
pride	0.67	0.25	0.36
realization	0.16	0.29	0.21
relief	0.50	0.09	0.15
remorse	0.53	0.88	0.66
sadness	0.38	0.71	0.49
surprise	0.40	0.66	0.50
macro-average	**0.40**	**0.63**	**0.46**
std	**0.18**	**0.24**	**0.19**

Table 4: Results based on GoEmotions taxonomy.

Google Research

```
Classification Report:
                precision    recall  f1-score   support

    admiration       0.69      0.62      0.65       504
     amusement       0.82      0.77      0.80       264
         anger       0.61      0.35      0.45       198
     annoyance       0.48      0.12      0.20       320
      approval       0.61      0.17      0.26       351
        caring       0.59      0.26      0.36       135
     confusion       0.45      0.42      0.44       153
     curiosity       0.44      0.75      0.55       284
        desire       0.56      0.17      0.26        83
disappointment       0.38      0.19      0.25       151
   disapproval       0.43      0.27      0.33       267
       disgust       0.66      0.42      0.51       123
 embarrassment       0.83      0.27      0.41        37
    excitement       0.53      0.31      0.39       103
          fear       0.65      0.65      0.65        78
     gratitude       0.98      0.85      0.91       352
         grief       1.00      0.33      0.50         6
           joy       0.65      0.52      0.58       161
          love       0.79      0.82      0.80       238
   nervousness       0.44      0.17      0.25        23
      optimism       0.71      0.45      0.55       186
         pride       0.71      0.31      0.43        16
   realization       0.38      0.17      0.24       145
        relief       0.29      0.18      0.22        11
       remorse       0.60      0.68      0.64        56
       sadness       0.73      0.45      0.56       156
      surprise       0.50      0.52      0.51       141
       neutral       0.65      0.69      0.67      1787

     micro avg       0.64      0.53      0.58      6329
     macro avg       0.61      0.42      0.48      6329
  weighted avg       0.63      0.53      0.56      6329
   samples avg       0.56      0.56      0.55      6329
```

Our Performance

Performance observations:

- **F1 score** is the most important metric for imbalanced datasets
- **Small improvement** in F1 score over the original research through the hybrid model
- Hybrid model yielded a **better precision** and **poorer recall** compared to original model
- Sentiment analysis for AI therapy, customer service chatbots **prefer precision scores** over recall scores
- Considering a **higher precision and F1 score**, we consider our architecture an improvement

PennState

Demo !

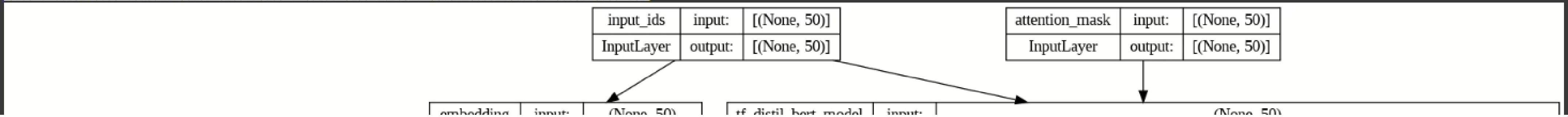
```
# Instantiate the combined model
vocab_size = len(tokenizer.vocab)
embedding_dim = FLAGS.embedding_dim # Embedding dimension for TextCNN
max_seq_length = FLAGS.max_seq_length
num_labels = num_labels

combined_model = build_combined_model(
    bert_config=None, # Not directly used
    seq_length=max_seq_length,
    is_training=True, # Set to True for training
    num_labels=num_labels,
    init_checkpoint=None,
    multilabel=FLAGS.multilabel,
    sent_rels=sent_rels, # Pass the actual sent_rels matrix
    sentiment=FLAGS.sentiment,
    corr_rels=corr_rels, # Pass the actual corr_rels matrix
    correlation=FLAGS.correlation,
    idx2emotion=idx2emotion,
    sentiment_groups=None, # Currently not used in model build
    intensity_groups=None, # Currently not used in model build
    num_train_steps=num_train_steps,
    num_warmup_steps=num_warmup_steps,
    vocab_size=vocab_size,
    embedding_dim=embedding_dim
)

# Display the model summary
combined_model.summary()
tf.keras.utils.plot_model(combined_model, to_file='model_architecture.png', show_shapes=True, show_layer_names=True)
```

```
 custom_classifier_layer (C  (None, 28)                32284     ['dropout_20[0][0]']
 ustomClassifierLayer)

==================================================================================================
Total params: 69601348 (265.51 MB)
Trainable params: 69601348 (265.51 MB)
Non-trainable params: 0 (0.00 Byte)
__________________________________________________________________________________________________
```



Lessons learned and Perspectives

- Reusing pre-existing project code is a lot of work, perhaps more than developing new model on existing dataset from scratch
  - GoEmotions repo has not had meaningful updates for about 4 years
    - Depended on extremely outdated modules, deprecated and even completely defunct APIs
    - Had to choose between pinning our code to extremely outdated libraries, losing access to new developments in AI space, or upgrading and refactoring code to supported APIs
  - Existing code can have incompatibilities in expected ways to run
    - GoEmotions expected to run from Bash command line, we have Windows
      - Possible to get working through Windows Subsystem for Linux or Bash emulator
      - Ultimately migrated to Jupyter Notebook for ease of running and development
- Cache all local dependencies, including remote models
  - HuggingFace updated remote model, broke compatibility with our modules and code and needed upgrades

PennState

Lessons learned and Perspectives

- Running BERT model is very process-intensive and time-consuming, even more than we expected
  - Running non-HuggingFace pretrained model on computer without CPU often took 4 hours per run
- Multilabel classification on uneven datasets is extremely complicated to solve
  - Through own trial-and-error and research papers, learned a lot about potential solutions
- Downloading model file saves GPU compute costs
  - For extensive analysis on predicted data, it is better to download the model and run it locally rather than use paid GPU compute
- Sentiment analysis needs humongous amounts of labeled data.
  - For emotions that fared poorly, data augmentation, additional data labeling, oversampling of minority class or undersampling of majority class was necessary.
  - Investing more time in data preparation and data labelling could possibly yield better results than algorithmic customizations

PennState

Bibliographical References

- Demszky, Dorottya, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 'GoEmotions: A Dataset of Fine-Grained Emotions', 5 2020. http://arxiv.org/abs/2005.00547.
- Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 'DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter', 2019. https://github.com/huggingface/transformers.
- Abas, Ahmed R., Ibrahim Elhenawy, Mahinda Zidan, and Mahmoud Othman. 'Bert-Cnn: A Deep Learning Model for Detecting Emotions from Text'. *Computers, Materials and Continua* 71 (2022): 2943–61. https://doi.org/10.32604/cmc.2022.021671.
- Guo, Yuan, and Hongmei Li. 'Research on Sentiment Analysis of Online Course Evaluation Based on EN- BERT-CNN'. In *Proceedings of the 2024 8th International Conference on Electronic Information Technology and Computer Engineering*, 65–68. ACM, 10 2024. https://doi.org/10.1145/3711129.3711142.
- Duong, Chi Thang, Remi Lebret, and Karl Abererécole. 'Multimodal Classification for Analysing Social Media', 2017.
- Tan, Kian Long, Chin Poo Lee, Kalaiarasi Sonai Muthu Anbananthen, and Kian Ming Lim. 'RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis With Transformer and Recurrent Neural Network'. *IEEE Access* 10 (2022): 21517–25. https://doi.org/10.1109/ACCESS.2022.3152828.
- Kokab, Sayyida Tabinda, Sohail Asghar, and Shehneela Naz. 'Transformer-Based Deep Learning Models for the Sentiment Analysis of Social Media Data'. *Array* 14 (7 2022). https://doi.org/10.1016/j.array.2022.100157.
- Wang, Shuo, Yuqi Lu, Ruijue Luo, Boxuan Li, Yingjie Zhang, and Liang Xiao. 'A BERT-Based Sentiment Analysis Model for Depressive Text'. In *Proceedings of the 2025 International Conference on Generative Artificial Intelligence and Digital Media*, 76–83. ACM, 3 2025. https://doi.org/10.1145/3734921.3734934.
- Kim, Yoon. 'Convolutional Neural Networks for Sentence Classification', 8 2014. http://arxiv.org/abs/1408.5882.

PennState

Bibliographical References

- GoEmotions data set: https://www.kaggle.com/datasets/debarshichanda/goemotions Downloaded June 10, 2025
- GoEmotions source code: https://github.com/google-research/google-research/tree/master/goemotions Downloaded June 10, 2025
- TextCNN: https://github.com/delldu/TextCNN Downloaded July 11, 2025
- HuggingFace documentation: https://huggingface.co/docs/transformers/en/model_doc/distilbert
- Keras documentation: https://www.tensorflow.org/tutorials/estimator/keras_model_to_estimator https://www.tensorflow.org/guide/migrate/migrating_estimator
- F1 Score in Machine Learning. 2025: https://www.futurense.com/Uni-Blog/F1-Score-Machine-Learning.
- Classification Metrics Guide: https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall
- GoEmotions emotions via emojis: https://neurohive.io/en/datasets/go-emotions-google-ai-dataset-for-sentiment-analysis/#:~:text=31%20October%202021,as%20improve%20customer%20support%20services
- Metrics for Sentiment Analysis: https://www.getfocal.co/post/top-7-metrics-to-evaluate-sentiment-analysis-models

PennState

Thank You !

AI–570 – Group 12

Manas Ranjan Sahoo

Chelsea Salyards

PennState