IS-7033-Topics: Natural Language Processing (NLP) –Midterm Report

John Rodriguez and **Ian Scarff**
University of Texas at San Antonio
One UTSA Circle, San Antonio, TX 78249

Abstract

The mid-term paper covers the progress status update of the project team as they explore the multi-modal challenge as presented by SemEval 2021 Task 6 challenge. The proposed solution includes training an image and text model separately and then creating a fused multimodal model. The included topics are the current supporting current literature research, methodology, loss function, data, preliminary results and evaluation metrics.

1 Introduction

This mid-term paper serves as our project update. The project involves using natural language processing and computer vision methodologies, we are focused on the SemEval 2021 Task 6 challenge. Our work will demonstrate the use of mutli-modal models for multi-label classification of memes. More specifically, the task is to classify the various techniques of propaganda in memes using multimodal AI methods. These methods will include models and algorithms used in NLP and computer vision.

The goal of the SemEval 2021 Task 6 and the focus of our project is to perform the following empirical research:

- Task 1 - Multi-label Classification Problem - Using only the text component of a meme, identify which of the 20 techniques are utilized.

- Task 2 - Multi-label Sequence Tagging - Using only the text component of the meme, identify which of the 20 techniques are utilized and also the spans of text covered by each technique. - **Note: This task will not be covered by this project but it was included for completeness.**

- Task 3 - Multi-modal Problem - Using both image and text component of a meme, identify which 22 techniques are utilized.

This update will cover our current literature research, methodology, loss function, data, preliminary results and evaluation metrics.

2 Literature Review

Ever since their introduction in the late 1990s (Lecun et al., 1998), convolutional neural networks (CNNs) have been at the forefront of the state-of-the-art in computer vision tasks. In addition to multi-class classification, these architectures have been applied to multi-label settings in recent years. Deep convolutional ranking (Gong et al., 2014a) assigns weights to the loss using a top-*k* ranking objective. Hypotheses-CNN-Pooling (Wei et al., 2016) uses multiple hypothesis region proposals to make predictions, which are then aggregated by max pooling. However, these processes treat labels as independent and do not model label correlations. There have been many attempts to effectively model the label co-occurrence dependencies (Gong et al., 2014b; Li et al., 2014; Li et al., 2015; Guo and Gu, 2011), but these models fall short because they actually model label semantic redundancy and pairwise label correlations.

In 2016 (Wang et al., 2016) introduced the CNN-RNN framework. This framework uses the image embedding produced by a CNN as the inputs to a RNN (Recurrent Neural Network), which model conditional relationships between time-steps. These type of models are able to learn a joint image-label representation to identify the semantic label dependencies. In the paper, Long Short Term Memory (LSTM) cells were used for the RNN implementation. They also employed a beam search algorithm to find the best prediction path. (Zanoci and Andress, 2017) used this methodology in their paper, but took a more greedy approach to decoding rather than using beam search. In their paper they used a label-only RNN model, a captioning RNN model, and a binary decoder RNN model.

In contrast to image modeling, text modeling has it's own unique set of challenges; however, for tasks that most human beings take for granted, these have proven to be quite difficult in artificial neural networks. Additionally, given the nature of this task, it has also proven to be quite challenging for human beings to derive an understanding of what they are being presented. Like in image modeling, a significant amounts of data are also required to perform well. State of the art for text modeling has gone from neural language models(2001), to word embedding (2013), neural network models (2013), to sequence to sequence (2014), to attention (2015), to memory-based networks [LSTM] (2015) and finally, pre-trained language models (2018) (Ruder, 2018).

In order to maximize the likelihood of success, the project team decided to use a Bidirectional Encoder Representations from Transformers (BERT) based model, see Figure:1. BERT builds upon other NLP techniques and is focused on utilizing a pre-trained model that permits sentence level tasks such as natural language inference and paraphrasing. The goal is to predict the relationships between sentences and analyze them together. BERT can also perform token-level tasks such as name entity recognition and question answering. The model architecture improves upon previous architectures by enabling bidirectional capabilities implemented using a 'masked language model.' The masked language model(MLM) randomly masks some of the tokens taken from the input and attempts to predict the id based only on context. The MLM enables the model to join the left and right context to form a deep bidirectional transformer. Because the model is pre-trained, the project team would only need to fine tune the model with only one more output layer. You can have a base pre-trained BERT model and then only change the last layer to optimize the require task. If you have multiple tasks, you would have the same pre-trained model and just different output layers (Devlin et al., 2019a).

DistilBERT is a BERT style model that has gone through what is termed knowledge distillation process(Sanh et al., 2020) Knowledge distillation is a model compression technique to reduce the size of the model while reproducing the behavior of the original model. Quantization, is another method of model compression which focuses on reducing the resolution of the weights utilized in the model by adjusting their bit size, how they are represented in memory, thus reducing weight bandwidth. In DistilBERT, knowledge is transferred using the distilization process which involves taking the base model (teacher) to the DistilBERT (student) model.

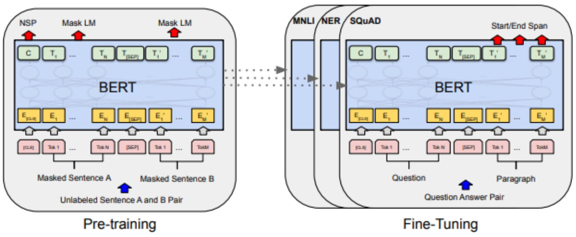


Figure 1: Architecture of BERT (Devlin et al., 2019a).

The next step removes the token-type embeddings and the pooler while also reducing a number of layers by a factor of 2. The reduction in layers reduces the overall number of model parameters. Lastly, the student model is initialized from the teacher as they have a common dimensionality.

With a 40% reduction in size, a 60 % increase in speed, and retains 97% of the learned knowledge, DistilBERT is a great compromise for minimizing computational costs, see Table: 1.

Task	BERT	D-BERT	% of Orig.
GLUE (score)	79.50	77.00	0.97
GLUE (sec)	668.00	410.00	0.61
Tsk - IMDB (score)	93.46	92.82	0.99
Tsk - SQuAD (F1)	88.50	85.80	0.97
Parms (MM)	110.00	66.00	0.60

Table 1: Bert vs. DistilBert Comparison (Sanh et al., 2020)

3 Methodology

For this task, we have to model both the image and text data together. Based on the co-occurrence matrix displayed in Figure: 2, we can see that certain labels are highly dependent on another. However, for experimentation, we also treat them as independent. While we expect a multimodal model to have better results, we also tested image only and text only models.

3.1 Image Modeling

One of the hardest challenges with the image data for this problem is the lack of it. Deep learn-

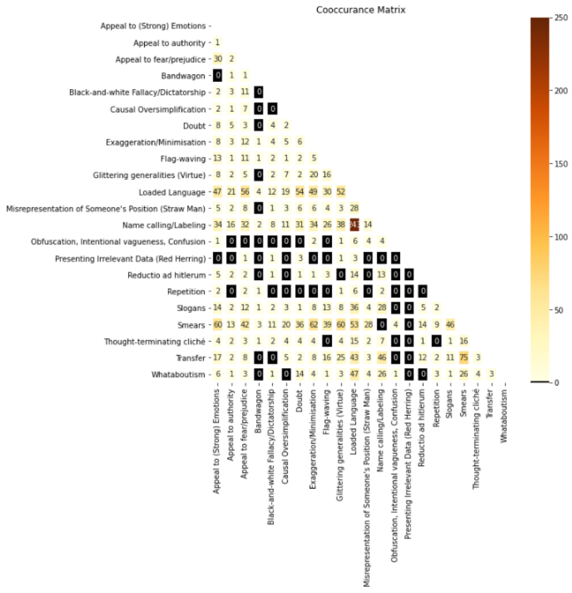


Figure 2: Label Co-Occurrence Matrix of Multi-Label Data.

ing models need extraordinary amounts of data to perform well, in general. (Brigato and Iocchi, 2020) demonstrated that model complexity plays a large role in the ability of CNNs to model small amounts of data. More specifically, less complex models with less parameters tend to work better. To apply this, ResNet18/50/101 (He et al., 2016), DenseNet121/169/201 (Huang et al., 2017), and VGG11/16/19 (Simonyan and Zisserman, 2015) with batch normalization were used as the backbones of our image models. Every model was pretrained on the ImageNet data (Deng et al., 2009). To model the labels independently, the last layer of each model was replaced with a fully connected layer with an output size of 22 and a sigmoid activation.

To model the labels dependently, our work is largely inspired by (Zanoci and Andress, 2017). Specifically their binary decoder RNN model, which is shown in Figure 3. In this model, the features of the image's are fed to an LSTM as the initial hidden state. The model runs for a number of time-steps equal to the number of labels (22 in our case), where the hidden state at each step is fed into a fully connected layer and outputs a single scalar, with the end result being a vector of length 22. Unlike in the original paper, we used a sigmoid activation at the end of each fully connected layer.

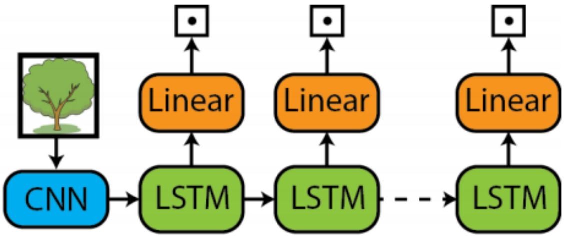


Figure 3: Architecture of (Zanoci and Andress, 2017)'s Binary Decoder RNN Model.

3.2 Text Modeling

The project team took an experimental approach with the methodology for the text modeling component. The baseline text model utilized simple linear and tree model approaches:knn, Decision Tree, Bagging,Random Forest, Boosting, Multinomial NB, SVC,Power SVC.

The team then built a BERT model with minimal fine tuning. To reduce the high computer hardware requirements for intense multi-modal processing, the team decided to use a sub-model called DistilBERT. This would affect the text processing portion of the multi-modal processing. The DistilBERT was also enhanced, and now incorporates additional derived variables and k-fold cross-validation. For more details, please see Section:5 Preliminary Experimentation.

3.3 Multimodal Fusion

Once the model fine tuning of the individual models is completed, the next step will involve merging the output of the DistilBERT model (max probabilities) and using it as an input into the image model. This step has not been attempted yet.

3.4 Loss Function

For our loss function we utilized the Binary Crossentropy Loss given by

$$L(\hat{y}_i, y_i) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)$$

where $\hat{y}_i$ is the i-th value in the model output, y_i is the corresponding target value, and N is the model output size. In this function the target and outputs take on the value 1 for the correct label, 0 otherwise. To do this, we set a prediction threshold of 0.5 for the model output.

4 Data

The data is provided by the SemEval 2020 Task 6 github[1]. The data is spread across three datasets: a training set with 687 observations, a dev set with 63 observations, and a test set with 200 observations (gold labels added later). A total of 36 observations were found to be unlabeled across the three sets; these were removed. The three sets were then combined into one with 914 observations. The frequency of each of the 22 labels is listed in Table 2. Note that since these are multi-labeled data, the sum of the frequencies is greater than 914.

Label	Frequency
Appeal to (Strong) Emotions	90
Appeal to authority	35
Appeal to fear / prejudice	91
Bandwagon	5
Black-and-white Fallacy / Dictatorship	26
Causal Oversimplification	36
Doubt	111
Exaggeration / Minimisation	99
Flag-waving	55
Glittering generalities (Virtue)	112
Loaded Language	492
Misrepresentation of Someone's Position (Straw Man)	40
Name calling / Labeling	347
Obfuscation, Intentional vagueness, Confusion	7
Presenting Irrelevant Data (Red Herring)	7
Reductio ad hitlerum	23
Repetition	14
Slogans	70
Smears	602
Thought-terminating cliché	27
Transfer	95
Whataboutism	67

Table 2: Label Frequencies.

The 3 labels with the lowest frequencies are ”Bandwagon,” ”Presenting Irrelevant Data (Red Herring),” and ”Obfuscation, Intentional vagueness, Confusion.” Based on the co-occurrence matrix displayed in Figure 1, if the data was randomly split to create a training and test set, these labels would probably be included in only one of the sets. To compensate for this, these observations were held out at first. Then the rest of the data was randomly split with 80% going to training, the rest to test. The holdout observations were then inserted at random locations into the two datasets, with 3, 4, and 4 observations of ”Bandwagon,” ”Presenting Irrelevant Data (Red Herring),” and ”Obfuscation, Intentional vagueness, Confusion,” respectively, going to the training set and the other 2, 3, and 3 observations, respectively, going to the test set. The images in the data were all of varying sizes. To standardized, images were resized to be 224x224 pixels and normalized to have a value range of [-1, 1]. Finally, the labels in each set were converted to a one-hot encoded vector.

5 Preliminary Experiments

The first set of experiments that have been performed were on the image only and text only models, treating the class labels as independent. For image modeling, models were trained for 30 epochs with batch sizes of 25 and a starting learning rate of 0.01, using 10-fold corss-validation. If the loss had not improved for at least 5 epochs, the learning rate was reduced by a factor of 10. The Adam optimizer with $\beta_i = (0.9, 0.999)$ was used. The model with the best validation score across all epochs and folds was deemed the best.

For the text modeling, baseline models (i.e non-neural network linear models) were built as a comparison that included knn, Decision Tree, Bagging, Random Forest, Boosting, Multinomial NB, SVC, Power SVC. The project team termed these the simple models. The complex models built are DNN in nature. A BERT Model(see Figure: 1 for more details) was the first complex model created as as first step in using state of the art techniques. Unlike the image model, the text model uses Adafactor for the optimizer component due to it having a smaller memory foot print and being tested on transformer style models (Shazeer and Stern, 2018). The Adafactor optimizer can dynamically calculate it's own learning rate and uses a step approach as the model is training. Adafactor also has the ability to prevent weights from going to zero and enable gradient clipping. This is a common problem that occurs when using non-linear activation. The last modification made to this model was implementing quanitizations of the weights. The reason for this was to enable training on the limited hardware available to the project team and to allow for quicker turn around times during experimentation. The weights are currently 64-bit but will likely get pushed down to 16-bit. Results will determine what the final weight bandwidth.

BERT Model Architecture:

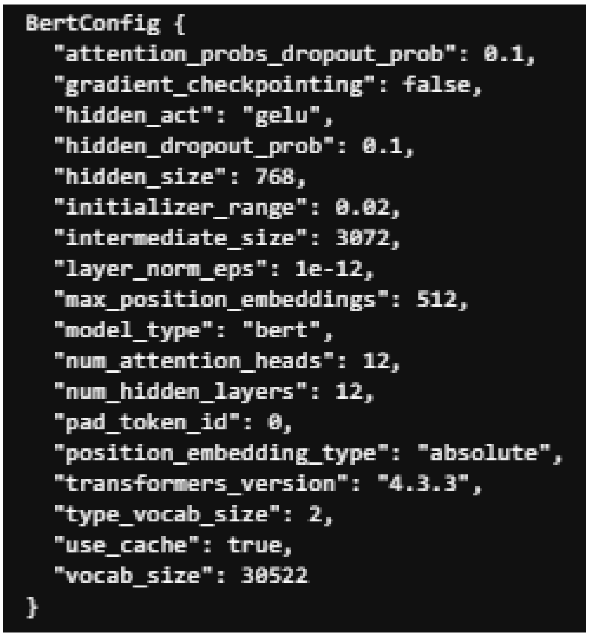
1. Layer_-transformers.BertModel.from_pretrained

[1] https://github.com/di-dimitrov/SEMEVAL-2021-task6-corpus

(’bert-base-uncased’) [*Note: this contains the architecture for BERT described above]

2. Layer_2-torch.nn.Dropout(0.5)
3. Layer_3-torch.nn.Linear(768, 22)

Further research indicated that other than augmenting the dataset, the hyper-parameter adjustments as part of the fine tuning would have negligible results (Quijano et al., 2021), see Figure: 4.



```
BertConfig {
  "attention_probs_dropout_prob": 0.1,
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "position_embedding_type": "absolute",
  "transformers_version": "4.3.3",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 30522
}
```

Figure 4: Text Model - BERT Hyper Parameters.

The project team then built a DistilBERT model. This model is very similar to the BERT model. However, the team also further refined the modeling process by adding additional derived variables to the forward pass of the model (calculating hate (using SonAR Hate Speech detection) and sentiment (using SentimentAnalyzer)) as augmented numerical representations in the feature space of the DistilBERT model. Additionally, kfold cross validation was also added to the modeling algorithm, along with L2 regularization to minimize over fitting.

DistilBERT Model Architecture:

1. Layer_1-transformers.DistilBertModel. from_pretrained(”distilbert-base-uncased”) [*Note: this contains the architecture for distilBERT described above, which is also a compressed form of BERT]
2. Layer_2-torch.nn.Dropout(0.2)
3. Layer_3-torch.nn.Linear(1024, 22)

It should be noted that the project team experimented with adding additional fully connected layers, and varying the shape of the layers to determine if that would enable further performance gains. The team determined that utilizing one layer (as originally recommended) and with a shape of 1024 was the ideal solution. Experimentation with the activation function was also performed. The following activation functions were tried: tanh, relu, and gelu. Ultimately, the project team found that though using any of theses functions seemed to produce similar outcomes, the gelu activation did resolve the vanishing gradients issue and has built-in dropout regularization(Hendrycks and Gimpel, 2020). It also was the preferred activation in BERT type models (Devlin et al., 2019b).

5.1 Evaluation Metric

To evaluate our results and determine the best model, we use the F1 Micro score given by

$$F1_{micro} = 2 \cdot \frac{P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}}$$

where P_{micro} and R_{micro} are the micro measures of precision and recall, respectively. These are given by

$$P_{micro} = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} TP_i + FP_i}$$

$$R_{micro} = \frac{\sum_{i=1}^{K} TP_i}{\sum_{i=1}^{K} TP_i + FN_i}$$

where TP_i is the true positive rate, FP_i is the false positive rate, TN_i is the false negative rate, and K is the number of classes.

5.2 Results

The results are given for each of the attempted models for both the image and text models.

5.3 Image Model

For the first round of experiments, the class labels were treated as independent. The results of image modeling is summarized in Table: 3. While the models preformed well on the training and validation data, with DenseNet169 being the best in both, they performed poorly on the test data, with ResNet101 being the best and DenseNet169 being the worst. On the test data, all models were not able to predict all of the class

Data/Model	ResNet18	ResNet50	ResNet101	DenseNet121	DenseNet169	DenseNet201	VGG11	VGG16	VGG16
Training	0.7289	0.8116	0.8063	0.7413	**0.8456**	0.7818	0.6529	0.6476	0.6558
Validation	0.8593	0.9019	0.8859	0.8790	**0.9337**	0.9289	0.9050	0.8856	0.8943
Testing	0.3831	0.4023	**0.4306**	0.4227	0.3746	0.3791	0.3920	0.3900	0.4152

Table 3: Image Only Model F1 Micro Scores.

labels. For example, in the best performing model (ResNet101) the labels ”Bandwagon,” ”Black-and-white Fallacy/Dictatorship,” ”Causal Oversimplification,” ”Misrepresentation of Someone’s Position (Straw Man),” ”Obfuscation, Intentional vagueness, Confusion,” ”Presenting Irrelevant Data (Red Herring),” ”Reductio ad hitlerum,” ”Repetition,” ”Slogans,” and ”Whataboutism” where not predicted by the model. There may be a few reasons for this: 1) These labels had some of the lowest occurrences in the data, 2) Because the model treated the labels as independent, a label for one didn’t increase the probability of another. We expect treating the labels as dependent will improve the performance of the models.

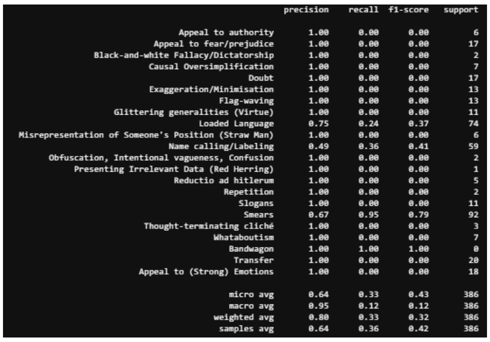
5.4 Text Model

Overall model results for each text model built, see Table: 4.

Scores	Linear	BERT	DistilBERT
F1 Score (Micro)	0.444	0.433	0.503
F1 Score (Macro)		0.117	0.13

Table 4: Text Only Model F1 Micro Scores.

More details for BERT implementation, see Figure: 5.



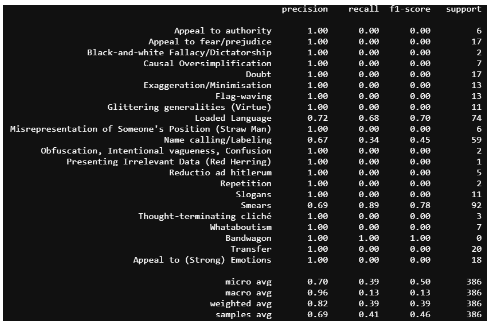
```
                                                    precision    recall  f1-score   support

                              Appeal to authority       1.00      0.00      0.00         6
                         Appeal to fear/prejudice       1.00      0.00      0.00        17
          Black-and-white Fallacy/Dictatorship       1.00      0.00      0.00         2
                     Causal Oversimplification       1.00      0.00      0.00         7
                                             Doubt       1.00      0.00      0.00        17
                     Exaggeration/Minimisation       1.00      0.00      0.00        13
                                       Flag-waving       1.00      0.00      0.00        13
              Glittering generalities (Virtue)       1.00      0.00      0.00        11
                                  Loaded Language       0.75      0.24      0.37        74
Misrepresentation of Someone's Position (Straw Man)       1.00      0.00      0.00         6
                           Name calling/Labeling       0.49      0.36      0.41        59
     Obfuscation, Intentional vagueness, Confusion       1.00      0.00      0.00         2
          Presenting Irrelevant Data (Red Herring)       1.00      0.00      0.00         1
                             Reductio ad hitlerum       1.00      0.00      0.00         5
                                        Repetition       1.00      0.00      0.00         2
                                           Slogans       1.00      0.00      0.00        11
                                            Smears       0.67      0.95      0.79        92
                       Thought-terminating cliché       1.00      0.00      0.00         3
                                      Whataboutism       1.00      0.00      0.00         7
                                         Bandwagon       1.00      1.00      1.00         0
                                          Transfer       1.00      0.00      0.00        20
                      Appeal to (Strong) Emotions       1.00      0.00      0.00        18

                                         micro avg       0.64      0.33      0.43       386
                                         macro avg       0.95      0.12      0.12       386
                                      weighted avg       0.80      0.33      0.32       386
                                       samples avg       0.64      0.36      0.42       386
```

Figure 5: Text Model - BERT Implementation.

The smaller DistilBERT, along with additional enhancements exceed the performance of the simple and BERT models. More performance details, see Figure: 6.



```
                                                    precision    recall  f1-score   support

                              Appeal to authority       1.00      0.00      0.00         6
                         Appeal to fear/prejudice       1.00      0.00      0.00        17
          Black-and-white Fallacy/Dictatorship       1.00      0.00      0.00         2
                     Causal Oversimplification       1.00      0.00      0.00         7
                                             Doubt       1.00      0.00      0.00        17
                     Exaggeration/Minimisation       1.00      0.00      0.00        13
                                       Flag-waving       1.00      0.00      0.00        13
              Glittering generalities (Virtue)       1.00      0.00      0.00        11
                                  Loaded Language       0.72      0.68      0.70        74
Misrepresentation of Someone's Position (Straw Man)       1.00      0.00      0.00         6
                           Name calling/Labeling       0.67      0.34      0.45        59
     Obfuscation, Intentional vagueness, Confusion       1.00      0.00      0.00         2
          Presenting Irrelevant Data (Red Herring)       1.00      0.00      0.00         1
                             Reductio ad hitlerum       1.00      0.00      0.00         5
                                        Repetition       1.00      0.00      0.00         2
                                           Slogans       1.00      0.00      0.00        11
                                            Smears       0.69      0.89      0.78        92
                       Thought-terminating cliché       1.00      0.00      0.00         3
                                      Whataboutism       1.00      0.00      0.00         7
                                         Bandwagon       1.00      1.00      1.00         0
                                          Transfer       1.00      0.00      0.00        20
                      Appeal to (Strong) Emotions       1.00      0.00      0.00        18

                                         micro avg       0.70      0.39      0.50       386
                                         macro avg       0.96      0.13      0.13       386
                                      weighted avg       0.82      0.39      0.39       386
                                       samples avg       0.69      0.41      0.46       386
```

Figure 6: Text Model - DistilBERT Implementation (preliminary).

5.5 Multimodal Fusion

This step has not been attempted yet due to ongoing refinement of the image and text models.

6 Conclusion & Next Steps

The project team has made significant progress in tackling SemEval 2020 Task 6 github, but more work remains. The Binary Decoder RNN is the champion model for the image model. DistilBERT is the champion model for the text model. The project team is currently refining the fine-tuning process. The next steps for the project team include refining the respective image and text models and then creating the multimodal fusion model and optimizing it. The final step will be to evaluate and document results. Code for everything done so far can be found on the project github[2].

References

L. Brigato and L. Iocchi. 2020. A close look at deep learning with small data.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding.

[2] https://github.com/iscarff123/SemEval2021-Task6.3-ClassProject

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. 2014a. Deep convolutional ranking for multilabel image annotation.

Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. 2014b. A multi-view embedding space for modeling internet images, tags, and their semantics. *Int. J. Comput. Vision*, 106(2):210–233.

Yuhong Guo and Suicheng Gu. 2011. Multi-label classification using conditional dependency networks. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, page 1300–1305. AAAI Press.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Dan Hendrycks and Kevin Gimpel. 2020. Gaussian error linear units (gelus).

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. 2017. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

J. Li, X. Lin, X. Rui, Y. Rui, and D. Tao. 2015. A distributed approach toward discriminative distance metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):2111–2122.

Xin Li, Feipeng Zhao, and Yuhong Guo. 2014. Multi-label image classification with a probabilistic label enhancement model. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, page 430–439, Arlington, Virginia, USA. AUAI Press.

Alex John Quijano, Sam Nguyen, and Juanita Ordonez. 2021. Grid search hyperparameter benchmarking of bert, albert, and longformer on duorc.

Sebastian Ruder. 2018. A Review of the Neural History of Natural Language Processing. http://ruder.io/a-review-of-the-recent-history-of-nlp/.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition.

Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. Cnn-rnn: A unified framework for multi-label image classification. pages 2285–2294.

Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. 2016. Hcp: A flexible cnn framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1901–1907.

Cristian Zanoci and Jim Andress. 2017. Exploring cnn-rnn architectures for multilabel classification of the amazon.