IS-7033-Topics: Natural Language Processing (NLP) – Multi-Modal Multi-Label Propaganda Classification in Online Memes

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

Abstract

This report covers our effort to tackle Task 6 of the 2021 Semantic Evaluation (SemEval) Competition (SemEval et al., 2021) entitled *Detection of Pervasive Techniques in Text and Images*. We further focused on sub-task 3, which utilizes text and image classification. We chose to focus on this task because it addresses the proliferation of fake news and propaganda on the Internet via a modern and convenient distribution channel: the Internet Meme. The goal of the competition is to use state of the art NLP techniques to detect different types of propaganda as defined by the SemEval team. Additionally, this task is further complicated by the fact that the specified classification of the meme is multi-label in nature. The problem is also multi-modal, both textual and visual data types given the nature of Internet Memes. Other challenges present for this task are the small volume of data available for model training and evaluation. Furthermore, the data classes are also imbalanced. We built, trained, and tested deep neural network frameworks to as solutions to this task.

Keywords: multi-modal, multi-label, classification, deep learning, BERT, DistilBert, computer vision, natural language processing.

1 Introduction

With the democratization of the public forum as a result of the proliferation of the internet starting in the 1990s, the control of the traditional media (i.e. television, radio, publishing companies, etc) was slowly loosened up over the next decade. The initial years brought information sharing to the masses and, as a principal, this is a win for free speech and democracies globally. However, with zero controls in place, due to the spirit of openness and good intentions, it was only a matter of time before anti-democratic agents found a way to use this spirit of openness to create chaos and lawlessness. One medium commonly used for "fake news" and misinformation is the Internet Meme, or meme for short. A meme is defined as "an amusing or interesting item (such as a captioned picture or video) or genre of items that is spread widely online especially through social media." This media can be easily transmitted over social media networks and reach vast numbers of users. This type of information dissemination can be classified as propaganda.

Figure 1: The image on the left (taken from krqe.com) shows a playful take on Bernie Sanders during the 2021 presidential inauguration. The image on the right (taken from the Semeval 2021 Task 6 development images) shows a more targeted meme meant to illicit negative responses or behaviors.

The goal of propaganda is to spread misinformation to achieve a certain goal. Propaganda publishers use psychological (emotional appeal) and rhetorical techniques (logical fallacies) to capture their audiences and illicit certain responses and behaviors. By appealing to audience emotions and by misusing logical rules, propagandist can induce audiences to form an emotional bond, thus suspending rational analysis and come to the wrong conclusion. Figure 1 shows a comparison between a relatively innocent meme and a more targeted meme. Deceptive memes efficiently enable propagandists by using an image that is blended with text to "reinforce/complement a technique in the text"

* Equal contribution. Listing order is random.

to enable one or several "persuasive techniques."

The goal of the SemEval 2021 Task 6 and the focus of our project (subtask 3) 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. - **Note: This task will not be covered by this project but it was included for completeness.**

- 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.

1.1 Research Question

Machine learning engineers are tasked with training computers to learn and understand various domains of the human experience. A sub-field of linguistics, computer science and artificial intelligence (AI), natural language processing (NLP) is the computer understanding and manipulation of human based languages. Because human language can be conveyed across multiple human senses (sight, sound, and touch through Braille), there are many avenues to pursue for research. With regards to this project, the research proposed is as follows: Classify the various techniques of propaganda in memes using multi-modal AI methods. These methods will include models and algorithms used in NLP and computer vision.

1.2 Project Motivation

Given the proliferation of propaganda using modern channels such as the internet and various subchannels as social media, 4chan, dark web, etc., a need has arisen to identify memes that contain harmful messages that spread false information, drive violent behavior, and lead to lawlessness. The goal of this project is to utilize NLP and computer vision to efficiently and accurately classify memes in a multi-modal, multi-label setting that are considered to be propaganda. Multi-modal can be defined as communication that uses multiple semiotic systems. Semiotic is the method of how meaning is communicated. In the case of this project, images and text. The task is to model and classify them according to a particular set of domain values (listing of propaganda techniques). The images and text can potentially have multiple domain values assigned (multi-label classification). This project is considered the end of the line as it the last part of a bigger effort for propaganda identification, which (in our view), is as follows:

1. Gather memes from source (Facebook, Twitter, Instagram, 4chan, Dark Web, Other).

2. Determine meme type.
   (a) non-propaganda (i.e. funny, an exaggeration, truthful/factual, etc.)
   (b) ambiguous (i.e. it might might be heading to a bad place)
   (c) propaganda (i.e. false information, elicits violence, leads to lawlessness, etc)

3. Classification of ambiguous and propaganda type memes.

The overall motivation is to assist with identifying and labeling of propaganda to prevent misinformation, violence and lawlessness. More succinctly, the research project involves using natural language processing and computer vision methodologies to create a deep neural network framework that will classify the various techniques of propaganda in memes using multi-modal AI methods.

The rest of the report will be as follows: in Section 2 we will discuss related works in NLP and computer vision, along with open research topic of memes. In Section 3 we will discuss the data and methods used in this project. In Section 4 we discuss our experimental setup and evaluation metric. In Section 5 we discuss our experimental results. In Section 6, we discuss challenges we faced and list items for future work. Finally, in Section 7 we conclude our findings and discuss next steps.

2 Literature Review

2.1 Meme Research

There have been previous studies regarding the multi-modal use of online text and imagery data. (Kruk et al., 2019) demonstrated the use of multi-modal architectures on Instagram images and their associated captions to classify them based on three

taxonomies: intent, contextual, and semiotic. They also showed that using both the image data and the text data together to make predictions leads to better accuracy when compared to using the data separately. Because of this, as we will discuss in Section 3, we also test both image only and text only models. Unlike the previously mentioned study, memes present a challenge for NLP due to their multi-modality nature in conjunction with the usage of humor and sarcasm (Suryawanshi et al., 2020).

Memes specifically were the subject of study in Facebook's *Hateful Memes Challenge* (Kiela et al., 2020). With the vastness of the internet, it is practically impossible for humans to sift through countless of memes to identify hate speech. Thus, there is the need for automation through AI. The goal of this challenge was to construct and implement multi-modal models for classifying hate speech. The authors of the challenge designed their dataset such that uni-modal models would under perform multi-modal models. Even then, as a baseline, they showed that even state-of-the-art (SOTA) models struggled compared to humans. Over 3,000 people and teams participated with (Lippe et al., 2020; Velioglu and Rose, 2020; Muennighoff, 2020; Zhong, 2020) placing 4th, 3rd, 2nd, and 1st in the challenge, respectively.

In addition to the challenge, memes are still a very open research topic. (Afridi et al., 2020) conducted a survey to offer a generalized view of the challenges modeling memes present, what the current advanced techniques for classification are, and discuss ongoing research into memes classification, memes reasoning, memes semantic entailment, and multi-modal fusion and co-learning. Furthermore, other research such as *Detecting Hate Speech in multi-modal Memes* (Das et al., 2020) and *Detecting Hateful Memes Using a Multi-modal Deep Ensemble* (Sandulescu, 2020) continue to prove that memes and multi-modal classification are in further need of innovation and automation due to the complexity of the problem. A major driving force is to realize human-like level accuracy for this problem while also maintaining performance of the implementation system.

2.2 Computer Vision

Ever since their introduction in the late 1990s (Lecun et al., 1998), convolutional neural networks (CNNs) have been at the forefront of SOTA 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 embeddings 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.

2.3 Natural Language Processing

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, significant amounts of data are also required to perform well. SOTA 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,

we decided to use a Bidirectional Encoder Representations from Transformers (BERT) based model, see Figure 2. 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, we would only need to fine tune the model with only one more output layer. One can have a base pre-trained BERT model and then only change the last layer to optimize the require task. If one has multiple tasks, one would have the same pre-trained model and just different output layers (Devlin et al., 2019a).

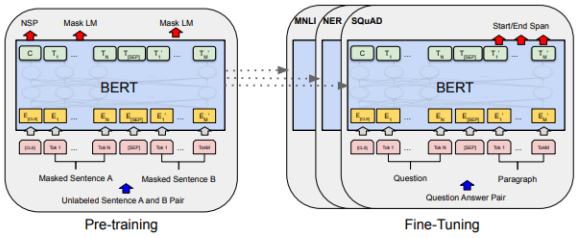


Figure 2: Architecture of BERT (Devlin et al., 2019a).

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 distillation process which involves taking the base model (teacher) to the DistilBERT (student) model.

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 created a model framework that utilizes both the image and text data together. Based on the co-occurrence matrix displayed in Figure 3, we can see that certain labels are highly dependent on another. However, for experimentation, we also treated them as independent. While we expected a multi-modal model to have better results, we also tested image only and text only models as benchmarks for the framework.

3.1 Data

The data is provided by SemEval 2021 for Task 6 github[1] is collected and stored as a json structured object. Each entry consists of the text in a meme, the various propaganda technique labels assigned to the meme, and an associated image file name. 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. In addition to the base data, we augmented the text data with additional derived data. The first metadata attribute was for hate speech classification. To do this, we utilized the python library HateSonar. This library uses input text and outputs a class of neither, hate speech, or offensive language. We encoded these

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

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.

as 0, 1, and 2, respectively. This classification is based off of (Davidson et al., 2017). The second metadata attribute was for sentiment classification. To do this, we used a python library published by HuggingFace, the same publishers of BERT & DistilBERT. This library leverages a pre-trained, fine-tuned model on sst2, which is a GLUE task. The library uses input text and outputs a class of negative or positive. We encoded these as 0 and 1, respectively.

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 3, 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

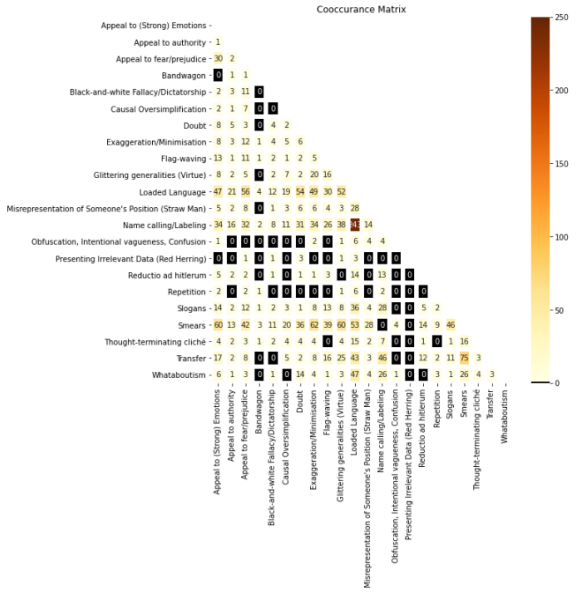


Figure 3: Label Co-Occurrence Matrix of Multi-Label Data.

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.

3.2 Image Modeling

One of the hardest challenges with the image data for this problem was the lack of it. Deep learning 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 4. In this model, the image embeddings 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

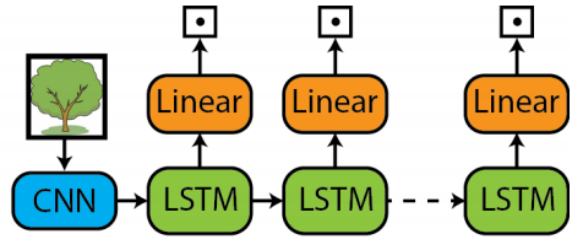


Figure 4: Architecture of (Zanoci and Andress, 2017)'s Binary Decoder RNN Model.

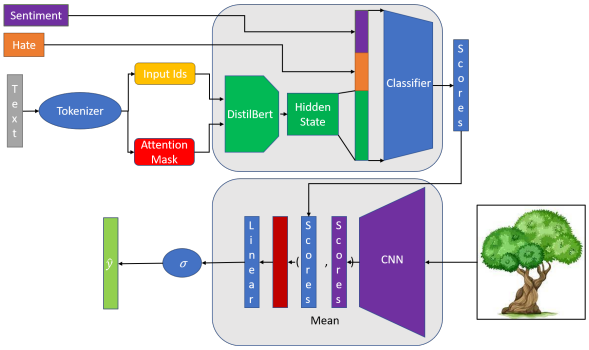
case), where the hidden state at each step is fed into a fully connected layer and outputs a single scalar, with the end result being a vector of length 22. Unlike in the original paper, we used a sigmoid activation at the end of each fully connected layer.

3.3 Text Modeling

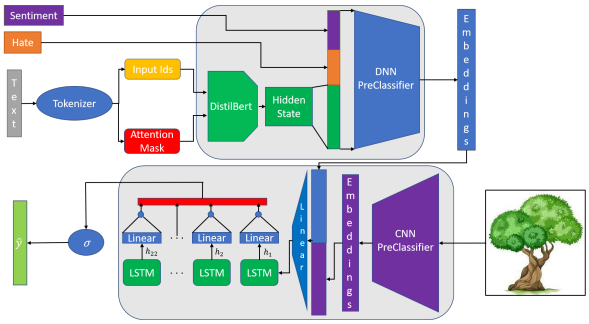
We took an experimental approach with the methodology for the text modeling component. The baseline text model utilized simple linear and tree model approaches from python package sklearn: KNN, Decision Tree, Bagging, Random Forest, Boosting, Multinomial NB, SVC, and Power SVC. No special optimizations were applied to these model algorithms, other than to permit parallelism to improve performance.

We then built a BERT model with minimal fine tuning. 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). The BERT Model architecture consists of 3 components. Layer 1 represents the pre-trained Bert Model. Layer 2 is the drop out set to 0.5. Layer 3 is a fully connected task layer which inputs at 762 and has a final output of 22.

To reduce the high computer hardware requirements for intense multi-modal processing, we 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. The BERT Model Architecture consists The DistilBERT Model architecture consists of 3 components. Layer 1 represents the pre-trained DistilBERT Model. Layer 2 is the drop out set to 0.2. Layer 3 is a fully connected task layer which inputs at 1024 and has a final output of 22. For more details, see Section 4.



(a) Architecture for treating labels as independent.



(b) Architecture for treating labels as dependent.

Figure 5: Multi-modal modeling architectures.

3.4 Multi-Modal Fusion

Figure 5 summarizes the architectures used for multi-modal modeling. When the labels were treated as independent, we took the average between the class score outputs from the DistilBERT and CNN model. This new vector was then processed through one linear layer and the output was activated by a sigmoid to get the final output $\hat{y}_i$. When the labels were treated as dependent, the flattened, preclassifier embeddings from the DistilBERT and CNN were concatenated together. This new vector was then passed through a linear layer to match the hidden layer size of the LSTM component. This "shrunk" vector was then passed to the LSTM component, linear layers, and sigmoid activation to get the final output $\hat{y}_i$.

3.5 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 Experiments

4.1 Experimental Setup

To train the text models, 15-fold cross-validation was applied, with each fold training for 7 epochs. To optimize, Adafactor was used. An initial learning rate was not specified because Adafactor determines the best learning rate using a sub-linear memory cost algorithm that also minimizes memory utilization. Hyperparameters for DistilBERT (distilbert-base-uncased) were a hidden size of 3072, DistilBERT output is 768 plus two additional feature dimensions from sentiment and hate, 6 layers, 6 attention heads, 1024 Max Position Embeddings, and was pre-initialized via the distillation process from BERT. The fully connected (task specific) layer inputs at 768, upscales to 1024 and then back to the number of classes. DistilBertTokenizer(distilbert-base-uncased) was used to parse the words and is based on word piece.

It should be noted that we experimented with adding additional fully connected layers, and varying the shape of the layers to determine if that would enable further performance gains. We 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, we 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).

To train the image and multi-modal models, 10-fold cross-validation was applied, with each fold training for 30 epochs. To optimize the image only model and CNN/CNN-RNN component of the multi-modal models, Adam with $\beta_i = (0.9, 0.999)$ was used. Adafactor was still used for the DistilBERT component of the multi-modal models. For Adam, and initial learning rate of 0.01 was specified, with this being reduced by a factor of 10 if the validation loss did not improve after 5 epochs. For the CNN-RNN models (for both image only and multi-modal), the hyperparameters for the LSTM component were hidden sizes of 500 and 700, one layer, and dropout rates of 0 and 50%. Every combination of hyperparameters was tested. Following (Zanoci and Andress, 2017), the weights of the linear layers directly before and after the LSTM component were initialized using Glorot initialization (Glorot and Bengio, 2010) given by

$$W_{ij} \sim U[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$$

where *W* is the weight and *n* is the size of the previous layer, and the biases were set to 0.

4.2 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. This is the same metric used for evaluation in the SemEval competition. During training, any model with the highest $F1_{micro}$ score across folds & epochs is chosen as the best model for each experiment.

5 Results

5.1 Text Models

Overall model results for each text model built, see Table 3. Utilizing DistilBERT provided an improvement of 13% for the F1 Micros score and 116% improvement for the F1 Macro Score over the non-linear (Random Forest) benchmark model. For more details for non-linear random forest benchmark model, see Figure 6a. For more details for BERT implementation, see Figure 6b. DistilBERT, along with additional enhancements, exceeds the performance of the Random Forest Benchmark and BERT models. For more performance details, see Figure 6c.

Scores	Non-Linear (Random Forest)	BERT	DistilBERT
F1 Score (Micro)	0.444	0.433	0.503
F1 Score (Macro)	0.06	0.117	0.13

Table 3: Text Only Model F1 Scores.

	ResNet18	ResNet50	ResNet101	DenseNet121	DenseNet169	DenseNet201	VGG11BN	VGG16BN	VGG19BN
Base	0.707229	0.812034	***0.883382***	0.733876	0.777941	0.835812	0.644767	0.656075	0.658844
(500, 1, 0)	0.470711	0.127919	0.462373	0.48165	0.457857	0.472843	**0.499164**	0.46946	0.459317
(500, 1, 0.5)	0	0.475725	0.473389	0.416573	0.498636	0.465937	**0.512687**	0.47865	0.482405
(700, 1, 0)	0.41535	0.47439	0.443569	**0.502509**	0.44314	0.511247	0.486578	0.485497	0.489477
(700, 1, 0.5)	0.020202	0.479973	0.48362	**0.508486**	0.062159	0.491883	0.452803	0.44209	0.468468
Base+DB	**0.631852**	0.607781	0.604472	0.59478	0.602711	0.607857	0.610274	0.586806	0.607994
(500, 1, 0)+DB	0.498339	0.474506	0.512115	0.501454	0.475177	**0.527632**	0.467877	0	0.427017
(500, 1, 0.5)+DB	0.450973	0.48704	0.450273	0.494915	0.469402	0.403941	0.471981	**0.516699**	0.469638
(700, 1, 0)+DB	**0.513257**	0.414343	0.472959	0.497942	0.497937	0.511659	0.469206	0.377737	0.468903
(700, 1, 0.5)+DB	0.469136	0.499326	0.471248	0.462522	**0.511045**	0.508731	0.472906	0.431925	0.478511

Table 4: F1 micro scores of the best models during training, per experiment, per model. **Bold** = Best by Experiment. Underline = Best by Model. ***Bold, Underline, Italics*** = Best Overall.

	ResNet18	ResNet50	ResNet101	DenseNet121	DenseNet169	DenseNet201	VGG11BN	VGG16BN	VGG19BN
Base	0.829384	0.913295	0.869565	0.852547	0.908163	***0.939815***	0.877108	0.900222	0.878453
(500, 1, 0)	0.58651	0.555891	0.537468	0.586907	**0.628571**	0.566038	0.569697	0.554572	0.569733
(500, 1, 0.5)	0.552707	0.550296	0.57346	0.568116	0.560694	**0.587896**	0.569697	0.561934	0.571429
(700, 1, 0)	0.48913	0.573964	0.579208	**0.606516**	0.581006	0.550964	0.571429	0.591549	0.561934
(700, 1, 0.5)	0.299625	0.587879	0.570588	0.578378	0.279412	0.564972	0.590799	0.584541	**0.595642**
Base+DB	**0.714667**	0.657143	0.645892	0.676923	0.662757	0.712934	0.704142	0.650847	0.693548
(500, 1, 0)+DB	0.575064	0.559767	**0.619048**	0.564565	0.572193	0.60355	0.559611	0	0.591224
(500, 1, 0.5)+DB	0.564103	0.563953	**0.613333**	0.587537	0.584112	0.5953	0.557864	0.566929	0.551136
(700, 1, 0)+DB	0.549133	0.552941	0.580952	0.581522	0.581006	**0.604396**	0.595122	0.583133	0.587571
(700, 1, 0.5)+DB	0.557185	0.593968	0.567335	**0.63285**	0.625641	0.574924	0.605505	0.590164	0.564841

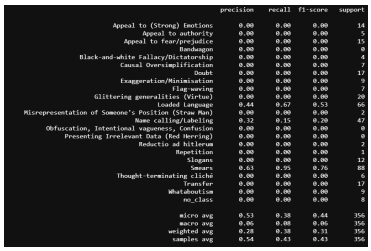
Table 5: F1 micro scores of the best models during validation, per experiment, per model. **Bold** = Best by Experiment. Underline = Best by Model. ***Bold, Underline, Italics*** = Best Overall.

	ResNet18	ResNet50	ResNet101	DenseNet121	DenseNet169	DenseNet201	VGG11BN	VGG16BN	VGG19BN
Base	0.395062	0.398366	0.383158	0.404235	0.359914	0.390688	0.426694	**0.458788**	0.359223
(500, 1, 0)	0.489749	0.489749	0.489749	**0.524345**	0.478936	0.488688	0.489749	0.489749	0.489749
(500, 1, 0.5)	0.489749	0.489749	**0.529577**	0.489143	0.497238	0.490308	0.489749	0.490352	0.489749
(700, 1, 0)	0.489749	0.489749	**0.529577**	0.509378	0.489192	0.477509	0.488584	**0.529577**	0.489749
(700, 1, 0.5)	0.283647	0.489749	0.489749	0.485393	0.24159	0.482993	**0.529577**	**0.529577**	**0.529577**
Base+DB	0.51512	**0.521028**	0.515294	0.493317	0.515901	0.496368	0.492234	0.492234	0.492234
(500, 1, 0)+DB	**0.529577**	0.489749	**0.529577**	0.488532	0.515658	0.46934	**0.529577**	NaN	**0.529577**
(500, 1, 0.5)+DB	0.489749	0.489749	**0.529577**	0.483616	0.528015	0.524138	0.488636	0.488636	0.488636
(700, 1, 0)+DB	0.489749	0.489749	**0.529577**	0.5	0.497863	0.486425	0.527985	0.527985	0.527985
(700, 1, 0.5)+DB	0.489749	0.529577	0.489749	***0.530233***	0.524715	0.472564	0.529577	0.529577	0.529577

Table 6: F1 micro scores of the best models during testing, per experiment, per model. **Bold** = Best by Experiment. Underline = Best by Model. ***Bold, Underline, Italics*** = Best Overall.

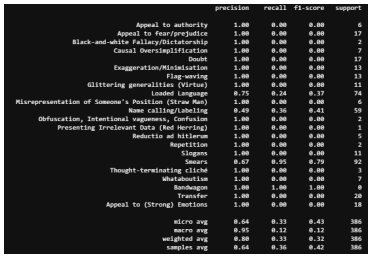
5.2 Image & Multi-Modal Models

F1 micro scores for image only and multi-modal models during training, validation, and testing are displayed in Tables 4, 5, and 6, respectively. In each of these tables, Base represents the base CNN model, treating labels as independent. CNN-RNN



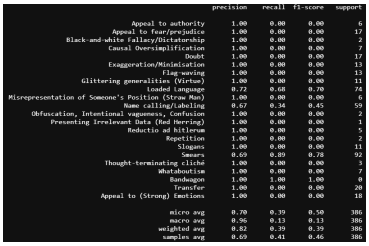
	precision	recall	f1-score	support
Appeal to (Strong) Emotions	0.00	0.00	0.00	14
Appeal to authority	0.00	0.00	0.00	5
Appeal to fear/prejudice	0.00	0.00	0.00	15
Bandwagon	0.00	0.00	0.00	0
Black-and-white Fallacy/Dictatorship	0.00	0.00	0.00	4
Causal Oversimplification	0.00	0.00	0.00	7
Doubt	0.00	0.00	0.00	17
Exaggeration/Minimisation	0.00	0.00	0.00	9
Flag-waving	0.00	0.00	0.00	7
Glittering generalities (Virtue)	0.00	0.00	0.00	20
Loaded Language	0.44	0.67	0.53	66
Misrepresentation of Someone's Position (Straw Man)	0.00	0.00	0.00	2
Name calling/Labeling	0.32	0.15	0.20	47
Obfuscation, Intentional vagueness, Confusion	0.00	0.00	0.00	0
Presenting Irrelevant Data (Red Herring)	0.00	0.00	0.00	0
Reductio ad hitlerum	0.00	0.00	0.00	2
Repetition	0.00	0.00	0.00	1
Slogans	0.00	0.00	0.00	12
Smears	0.63	0.95	0.76	88
Thought-terminating cliché	0.00	0.00	0.00	6
Transfer	0.00	0.00	0.00	17
Whataboutism	0.00	0.00	0.00	9
no_class	0.00	0.00	0.00	8
micro avg	0.53	0.38	0.44	356
macro avg	0.06	0.08	0.06	356
weighted avg	0.28	0.38	0.31	356
samples avg	0.54	0.43	0.43	356

(a) Text Model - Non-Linear Random Forest Benchmark Implementation. Note: The dataset was updated by the SemEval and the benchmarks were not re-run.



	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	0.85	0.33	0.32	386
samples avg	0.64	0.36	0.42	386

(b) Text Model - BERT Implementation.



	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

(c) Text Model - DistilBERT Implementation.

Figure 6: Text Model Implementations.

	precision	recall	f1-score	support
Appeal to (Strong) Emotions	0.23	0.33	0.27	18
Appeal to authority	0.00	0.00	0.00	10
Appeal to fear/prejudice	0.40	0.14	0.21	14
Bandwagon	0.00	0.00	0.00	2
Black-and-white Fallacy/Dictatorship	0.00	0.00	0.00	10
Causal Oversimplification	0.07	0.33	0.12	3
Doubt	0.17	0.43	0.25	21
Exaggeration/Minimisation	0.00	0.00	0.00	30
Flag-waving	0.08	0.07	0.08	14
Glittering generalities (Virtue)	0.16	0.10	0.12	31
Loaded Language	0.55	0.52	0.54	98
Misrepresentation of Someone's Position (Straw Man)	0.20	0.17	0.18	6
Name calling/Labeling	0.40	0.43	0.42	67
Obfuscation, Intentional vagueness, Confusion	0.00	0.00	0.00	3
Presenting Irrelevant Data (Red Herring)	0.00	0.00	0.00	3
Reductio ad hitlerum	0.00	0.00	0.00	3
Repetition	0.00	0.00	0.00	3
Slogans	0.57	0.20	0.30	20
Smears	0.66	0.74	0.70	117
Thought-terminating cliché	0.06	0.14	0.08	7
Transfer	0.00	0.00	0.00	14
Whataboutism	0.00	0.00	0.00	10
micro avg	0.41	0.39	0.40	504
macro avg	0.16	0.16	0.15	504
weighted avg	0.38	0.39	0.37	504
samples avg	0.41	0.42	0.37	504

(a) Base ResNet50

	precision	recall	f1-score	support
Appeal to (Strong) Emotions	0.00	0.00	0.00	18
Appeal to authority	0.00	0.00	0.00	10
Appeal to fear/prejudice	0.00	0.00	0.00	14
Bandwagon	0.00	0.00	0.00	2
Black-and-white Fallacy/Dictatorship	0.00	0.00	0.00	10
Causal Oversimplification	0.00	0.00	0.00	3
Doubt	0.00	0.00	0.00	21
Exaggeration/Minimisation	0.00	0.00	0.00	30
Flag-waving	0.00	0.00	0.00	14
Glittering generalities (Virtue)	0.00	0.00	0.00	31
Loaded Language	0.52	1.00	0.69	98
Misrepresentation of Someone's Position (Straw Man)	0.00	0.00	0.00	6
Name calling/Labeling	0.00	0.00	0.00	67
Obfuscation, Intentional vagueness, Confusion	0.00	0.00	0.00	3
Presenting Irrelevant Data (Red Herring)	0.00	0.00	0.00	3
Reductio ad hitlerum	0.00	0.00	0.00	3
Repetition	0.00	0.00	0.00	3
Slogans	0.00	0.00	0.00	20
Smears	0.63	1.00	0.77	117
Thought-terminating cliché	0.00	0.00	0.00	7
Transfer	0.00	0.00	0.00	14
Whataboutism	0.00	0.00	0.00	10
micro avg	0.57	0.43	0.49	504
macro avg	0.05	0.09	0.07	504
weighted avg	0.25	0.43	0.31	504
samples avg	0.57	0.49	0.50	504

(b) ResNet50-LSTM(500, 1, 0)

	precision	recall	f1-score	support
Appeal to (Strong) Emotions	0.00	0.00	0.00	18
Appeal to authority	0.00	0.00	0.00	10
Appeal to fear/prejudice	0.50	0.21	0.30	14
Bandwagon	0.00	0.00	0.00	2
Black-and-white Fallacy/Dictatorship	0.00	0.00	0.00	10
Causal Oversimplification	0.00	0.00	0.00	3
Doubt	0.00	0.00	0.00	21
Exaggeration/Minimisation	0.00	0.00	0.00	30
Flag-waving	0.50	0.14	0.22	14
Glittering generalities (Virtue)	0.00	0.00	0.00	31
Loaded Language	0.71	0.77	0.74	98
Misrepresentation of Someone's Position (Straw Man)	0.00	0.00	0.00	6
Name calling/Labeling	0.51	0.63	0.56	67
Obfuscation, Intentional vagueness, Confusion	0.00	0.00	0.00	3
Presenting Irrelevant Data (Red Herring)	0.00	0.00	0.00	3
Reductio ad hitlerum	0.00	0.00	0.00	3
Repetition	0.00	0.00	0.00	3
Slogans	0.33	0.05	0.09	20
Smears	0.68	0.84	0.75	117
Thought-terminating cliché	0.00	0.00	0.00	7
Transfer	0.25	0.07	0.11	14
Whataboutism	1.00	0.10	0.18	10
micro avg	0.63	0.44	0.52	504
macro avg	0.20	0.13	0.13	504
weighted avg	0.43	0.44	0.42	504
samples avg	0.60	0.46	0.48	504

(c) ResNet50+DB

	precision	recall	f1-score	support
Appeal to (Strong) Emotions	0.00	0.00	0.00	18
Appeal to authority	0.00	0.00	0.00	10
Appeal to fear/prejudice	0.00	0.00	0.00	14
Bandwagon	0.00	0.00	0.00	2
Black-and-white Fallacy/Dictatorship	0.00	0.00	0.00	10
Causal Oversimplification	0.00	0.00	0.00	3
Doubt	0.00	0.00	0.00	21
Exaggeration/Minimisation	0.00	0.00	0.00	30
Flag-waving	0.00	0.00	0.00	14
Glittering generalities (Virtue)	0.00	0.00	0.00	31
Loaded Language	0.52	1.00	0.69	98
Misrepresentation of Someone's Position (Straw Man)	0.00	0.00	0.00	6
Name calling/Labeling	0.00	0.00	0.00	67
Obfuscation, Intentional vagueness, Confusion	0.00	0.00	0.00	3
Presenting Irrelevant Data (Red Herring)	0.00	0.00	0.00	3
Reductio ad hitlerum	0.00	0.00	0.00	3
Repetition	0.00	0.00	0.00	3
Slogans	0.00	0.00	0.00	20
Smears	0.63	1.00	0.77	117
Thought-terminating cliché	0.00	0.00	0.00	7
Transfer	0.00	0.00	0.00	14
Whataboutism	0.00	0.00	0.00	10
micro avg	0.57	0.43	0.49	504
macro avg	0.05	0.09	0.07	504
weighted avg	0.25	0.43	0.31	504
samples avg	0.57	0.49	0.50	504

(d) ResNet50-LSTM(500, 1, 0)+DB

Figure 7: Classification reports during testing of ResNet50 for different experiments.

models are represented by (H, L, D), where H is the hidden layer size of the LSTM component, L is the number of LSTM layers, and D is the dropout rate of the LSTM component. Multi-modal models are represented by a CNN/CNN-RNN+DB, where DB denotes the addition of the DistilBERT model. From the tables we can see that the base CNN models performed the best during training and validation, achieving high F1 micro scores, while the CNN-RNN and multi-modal models performed poorer. However, during testing, the F1 micro scores for the base CNN models fell, while most others remained with a 0.1 range. Most notably, the multi-modal models had more high scores across various CNN backbones than the image only models. In fact, DenseNet121-LSTM(700, 1, 0.5)+DB performed the best overall during testing. One perplexing result of the test F1 micro scores is that a large majority of them are exactly the same. To explore this, we examined classification reports for each experiment during training. An example is shown in Figures 7a-d using ResNet50 as the CNN backbone. From this reports, we can see that the base CNN did a decent job classifying various labels, only missing some with very low frequencies. However, for the other experiments, the models missed nearly all labels, at most classifying only 4, with ”Smears” and ”Loaded Language” usually

being classified. These behaviors are practically the same for all models and experiments performed during testing.

6 Discussion

6.1 Challenges

We encountered several challenges that have been discussed throughout this paper: small dataset, highly unbalanced classes (propaganda types), hardware limitations, limited model explainability (apart from published details), and model architectures that can handle multimodal type data. Another challenge of modeling memes is that they are unnatural images.

The BERT class models used in this project were pretrained on BookCorpus and English Wikipedia. However, though this amounts to large amounts of trainable data, the fine-tuning task specifically contains text that is deep in subtleties, humor, internet colloquialisms, and innuendos. This is a very different language structure from what the original model pretraining. Additional pre-training may be required to hone in on specific propaganda techniques to better permit the text architecture to understand the 22 classes of propaganda. Additionally, the nuances between the classes can also pose a problem toe the BERT class models given the nature of the dataset, size and depth.

The CNN models used in this project were pretrained on ImageNet, which contains clear, natural photos. Memes can be very complex with text overlaying parts the image, multiple images placed next to each other that are different, some can be drawn, etc. This "unnatural" structure makes the use of pretrained models difficult. One final challenge in this project was the abstractness of the labels. For example, one of the classes in ImageNet is "peacock." Images of a peacock are easy to learn because the peacocks have common structures and features that can be generalized to other observations of the same class. It is very difficult to tell if the various propaganda classes have common structures and features that can be learned and generalized. For example, what does "Whataboutism" look like? It can look very different from meme to meme.

6.2 Future Work

Given the time constraints of this project, there was much we were not able to experiment with. There were also many things that could have been done differently. The following can be considered as future work:

- **Use enhanced feature engineering for text**. In addition to the augmented hate and sentiment metadata, there are additional opportunities to create additional metadata based on NLP techniques such as Bag of Words, N-gram combinations, word count, prune features that are not necessary, etc.

- **Use text architectural changes**. The text component can also be further optimized by exploring other more recent text modeling techniques such as GPT-3. The text component can also be enhanced using a weighting algorithm to control for the in-balance classes.

- **Use performance enhancements for text**. Additional opportunities exist to further improve performance for the text component by applying weight quantitization and pruning. This would allow for more experimentation in shorter time frames.

- **Use a F_β score as the evaluation metric**. Given by:

$$F_\beta = \frac{(1+\beta^2) \times (P \times R)}{(\beta^2 \times P + R)}$$

  where *P* is precision and *R* is recall, the F_β score is better for unbalanced data. When we care more about minimizing false positives than false negatives (precision given more weight than recall), we set $\beta < 1$. When we care more about minimizing false negative (recall given more wieght), we set $\beta > 1$. In (Zanoci and Andress, 2017), they used a F_2 score.

- **Use a different CNN-RNN architecture**. In addition to the Binary Decoder RNN model from the previously mentioned paper, the authors also tested a Label-Only RNN model and a Captioning RNN model. These would be interesting to apply to see if they yield better results.

- **Don't use K-Fold cross validation**. The reason for this is due to certain labels having very small frequencies. As explained in Section 3.1, we held out 3 labels during data merging, and then selectively inserted them into

the training and test set. The use of K-Fold cross-validation on the training set meant that for a majority of the folds, none, one, two, or all of those three labels were not predicted during validation. This means that the best performing model was potentially not trained well enough on those labels. Not using cross validation would allow us to be sure that those labels were in both the training and validation set. This would require another layer of dataset manipulation. In order to avoid this, more labeled data would be needed.

- **Remove text from the images before modeling**. Doing this could potentially aid the model in choosing the right features to make a prediction. In a modeling pipeline, this would be done after the text is extracted as its own data.

- **Use a different optimizer for image modeling**. While training of our models, there were many times where the loss and/or evaluation metric did not improve for several epochs. In personal prior experience with image modeling, the Adam optimizer had been ineffective at helping models learn. In these cases, Stochastic Gradient Descent has often performed better than Adam.

- **Don't used models pretrained on ImageNet data**. As discussed in the challenges section, ImageNet contains natural photos. Therefore the pretrained weights are designed for natural photos. Again, memes are highly unnatural. For this modeling task, we could do either of the following: 1) Collect more labeled data similar to the data used in this project. However, labeled meme data of descent size is hard to come by. 2) Use a unsupervised learning task on a large number of memes to learn their representations, then apply those to the classification task. 3) Train directly on the data used in this project.

- **Use a transformer instead of LSTM**. Introduced in 2017, transformers (Vaswani et al., 2017) have been shown to be much better than RNNs. Using these instead of LSTMs may offer better predictions.

- **Implement a beam search**. In the original paper introducing the CNN-RNN architecture, (Wang et al., 2016) used a beam search for predicting each label. In our project and in the paper that motivated our image modeling, a more greedy approach was used. A beam search may have yielded better results.

7 Conclusion & Next Steps

We has made significant progress in tackling SemEval 2021 Task 6 subtask 3, but more work remains. The Binary Decoder RNN is the champion model for the image component. DistilBERT is the champion model for the text component. These two components were fused in the multi-modal model. We were able to show that multi-modal models performed marginally better than their single modal counterparts. While our results may have been less than optimal for us, comparing to the task leaderboard[2], we had similar results. However, we were not under the same data constraints as the competition.

Memes are an active and challenging area of reseach. Their multi-modal and unnatrual qualities make them hard to model, which our methods and project demonstrate. Next steps include reviewing other model methodologies for both text and images, using weights to address the unbalanced data, explore additional NLP metadata augmentation, and provide additional visualizations on the data and to assist with model explainability and insights. Code for everything done so far can be found on the project github[3].

Acknowledgments

John Rodriguez: Many thanks to friends, coworkers, professors and acquaintances who took the time and provided support during this incredible project. Additionally, many thanks to Dr. Rios for his support and mentoring during our project.

Ian Scarff: I would like to thank Dr. Rios for his assistance, time and guidance that he provided during our project.

References

Tariq Habib Afridi, Aftab Alam, Muhammad Numan Khan, Jawad Khan, and Young-Koo Lee. 2020. A multimodal memes classification: A survey and open research issues.

[2]https://propaganda.math.unipd.it/semeval2021task6/leaderboard.php

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

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

Abhishek Das, Japsimar Singh Wahi, and Siyao Li. 2020. Detecting hate speech in multi-modal memes.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *CoRR*, abs/1703.04009.

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.

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.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

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.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes.

Julia Kruk, Jonah Lubin, Karan Sikka, Xiao Lin, Dan Jurafsky, and Ajay Divakaran. 2019. Integrating text and image: Determining multimodal document intent in instagram posts. *CoRR*, abs/1904.09073.

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.

Phillip Lippe, Nithin Holla, Shantanu Chandra, Santhosh Rajamanickam, Georgios Antoniou, Ekaterina Shutova, and Helen Yannakoudakis. 2020. A multimodal framework for the detection of hateful memes.

Niklas Muennighoff. 2020. Vilio: State-of-the-art visio-linguistic models applied to hateful memes.

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/`.

Vlad Sandulescu. 2020. Detecting hateful memes using a multimodal deep ensemble.

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

SemEval, Giovanni Da San Martino, Hamed Firooz, Preslav Nakov, and Fabrizio Silvestri. 2021. Semeval 2021 task 6 on "detection of persuasion techniques in texts and images".

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

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal meme dataset (multioff) for identifying offensive content in image and text. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Riza Velioglu and Jewgeni Rose. 2020. Detecting hate speech in memes using multimodal deep learning approaches: Prize-winning solution to hateful memes challenge.

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.

Xiayu Zhong. 2020. Classification of multimodal hate speech – the winning solution of hateful memes challenge.