Hateful Memes Detection : CS 7643

Hassan Naveed
Georgia Institute of Technology
hnaveed3@gatech.edu

Saeb Hashish
Georgia Institute of Technology
shahish3@gatech.edu

Dan Tylukti
Georgia Institute of Technology
dtylutki3@gatech.edu

Jake Banigan
Georgia Institute of Technology
jbanigan3@gatech.edu

Abstract

Users of social media generate massive amounts of content daily. While most of this content is positive or benign, some of this content is created to express hate towards certain groups of people or individuals. Whether by a social media company's own prerogative to create a positive experience for its customers, legal responsibility imposed by governments, or public demands, such companies usually want to identify and remove hateful content from their platforms. Given the sheer amount of content created, it is not feasible to rely entirely on human moderators. Therefore, automated techniques like machine learning can be employed to detect hateful content. The challenge for machine learning solutions is that much content is not simple text, but multimodal combinations of media like memes which combine one or more images with some overlaid caption. Interpreting the meaning of a meme is easy for most humans, but difficult for machines, as understanding may require contextual information that a human person would likely know but a machine might not. In this paper, we describe findings from several experiments in which we employed novel deep learning techniques or existing techniques combined in new ways attempting to improve on state-of-the-art (SOTA) baselines for predicting whether memes are hateful or not.

1. Introduction

In this paper we tackle the problem of classifying memes as hateful or non-hateful, a problem presented as the Hateful Memes challenge by Meta [5]. Since this problem requires using information from both text and images, we explore different techniques for representing text and images. By doing so we test whether more recent (and better performing) representations contribute to improved classification accuracy.

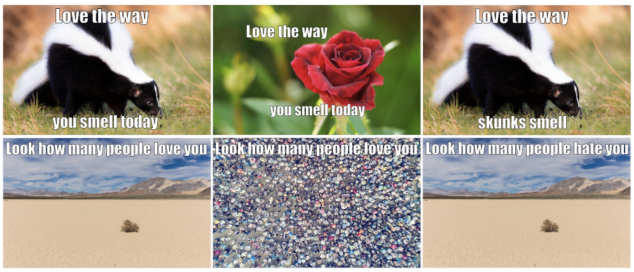

Figure 1. (Left) Mean memes (Middle) Benign Image Confounder (right) Benign Image Confounder. This example shows that using a unimodal approach, that is, the text and image alone, would not fare well in classifying hateful memes.[5]

The Hateful Memes challenge was created by Meta to serve a dual purpose; first, to mitigate the real-world problem of hate speech on its social media platforms by identifying and removing hateful content [5], and second, to facilitate the development of better multimodal analysis techniques. The hateful memes dataset contains memes posted on Facebook and Instagram which are a combination of one or more images with overlaid text. The dataset includes both hateful and non-hateful memes. Meta defines hate speech as: ...anything that directly attacks people based on what are known as their "protected characteristics" — race, ethnicity, national origin, religious affiliation, sexual orientation, sex, gender, gender identity, or serious disability or disease. [5] What makes this problem difficult for a machine to solve is that memes have a combination of text and an image which both contribute to the meaning of the meme. To help the models learn, Meta researchers incorporated benign text and image confounders (depicted in Figure 1), which change the intent of the meme (hateful or non-hateful) by replacing the image or text, to ensure successful models are truly capable of multimodal analysis.

The need for multimodal analysis to detect hateful memes was confirmed by the initial approaches tried by re-

Type	Model	Validation Acc.	Validation AUROC	Test Acc.	Test AUROC
	Human	-	-	84.70	-
Unimodal	Image-Grid	50.67	52.33	52.73±0.72	53.71±2.04
	Image-Region	52.53	57.24	52.36±0.23	57.74±0.73
	Text BERT	58.27	65.05	62.80±1.42	69.00±0.11
Multimodal (Unimodal Pretraining)	Late Fusion	59.39	65.07	63.20±1.09	69.30±0.33
	Concat BERT	59.32	65.88	61.53±0.96	67.77±0.87
	MMBT-Grid	59.59	66.73	62.83±2.04	69.49±0.59
	MMBT-Region	64.75	72.62	67.66±1.39	73.82±0.20
	ViLBERT	63.16	72.17	65.27±2.40	73.32±1.09
	Visual BERT	65.01	74.14	66.67±1.68	74.42±1.34
Multimodal (Multimodal Pretraining)	ViLBERT CC	66.10	73.02	65.90±1.20	74.52±0.06
	Visual BERT COCO	65.93	74.14	69.47±2.06	75.44±1.86

Table 1. Model Performance of models tested by Kiela et al [5] which our models are compared to.

searchers at Meta (see Table 1). Unimodal approaches, i.e., those that relied solely on either the text or the image from the meme, such as features from one of ImageGrid, Image Region or Text BERT did not perform as well as multimodal approaches that utilized both Image and Text. Similarly, models with late fusion (ConcatBERT, or visual BERT with initial layers frozen) were outperformed by models with late fusion. Simple models were shown to perform well without little utilization of the image [11] and hence lacked a truly multimodal analysis. Truly multimodal models that utilized a pretrained language model in combination with a pretrained vision model were shown to perform best on the hateful memes problem. However, there is still more room for improvement as current state of art models fall short of the human baseline.

Since the Hateful Memes challenge paper was published in 2021, new models for both text and image analysis have been created which our team has implemented and tested to improve upon Meta's initial results. Models we create that are successful in improving accuracy could help Meta correctly identify and remove more hateful content from their site.

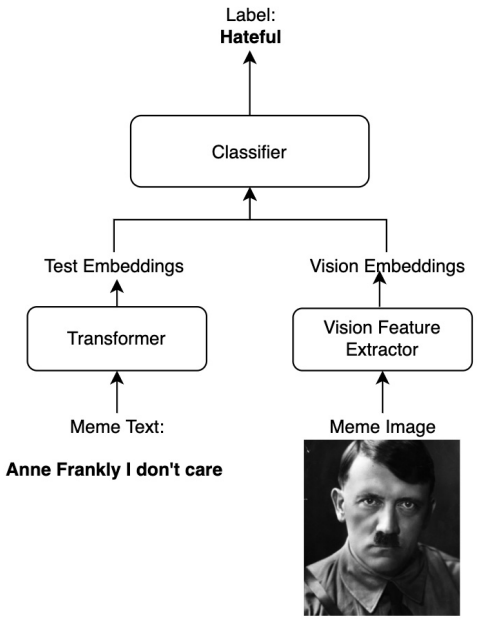

Figure 2. A high level overview of our approach. The modules we experimented with changing include the *Text embedding transformer*, *Vision Feature Extractor* and *Classifier*

2. Approach

The purpose of our team's project was to research new techniques for creating text embeddings, image embeddings, and classification which had not yet been previously applied to the challenge of detecting hateful memes to create combinations yielding novel multimodal models for detecting hateful memes. This approach is depicted in Figure 2.

Our approach was to create a baseline multimodal model which used BERT text embeddings, CLIP image embeddings, and a DistilBERT classification head (i.e., a linear projection), then create several other models that varied the text embedding model, the image embedding model, or the classification head used. Whenever we experimented with one component of the multimodal model, such as the vision model portion, then we only changed this component while keeping other components of the model the same. The hyperparameters that we could change included the learning rate, number of epochs, and the batch size, but we kept these the same when comparing different types of models for the

same component. This approach enabled us to directly compare the contributions to performance of different models for each component in the full multi-modal model. The loss function we employed to train our models was binary cross entropy (BCE) loss, which was calculated after running output logits through a sigmoid function. Implementations of the transformers were used from the "transformers" library provided by huggingface.

The goal of this approach was to build and evaluate novel combinations of multimodal models with the hopes of outperforming the current best models described in the Hateful Memes Challenge [5] as seen in Table 1. As per our knowledge, some of the embeddings were not experimented with. There is also no study comparing the importance of varying how different types of embeddings compare to each other in the Hateful Memes problem.

We did anticipate one-to-one comparisons would be difficult if we used the classification head from a particular model (for example BERTforSequenceClassification), which is why we kept the classification head constant when making comparisons between embeddings. Some of the first things tried were improved text-embeddings, and most of these did not seem to make statistically significant improvements to our predictions.

3. Experiments and Results

As described in the previous section and Figure 2, our experiments varied one component of the overall model at a time and evaluated performance using classification accuracy and AUROC scores on the test data. These experiments and initial results for each component is shown in the subsections below:

3.1. Text Embeddings

The standard natural language processing (NLP) model employed by both the unimodal and multimodal models tested by Meta (previously Facebook) researchers mostly employed Bidirectional Encoder Representations from Transformers (BERT) for creating text features. Since the publication of BERT in October 2018 [3], other high-performing models have been created that are entirely new or built on top of BERT and which may outperform BERT in some use cases. The models that we identified as potential replacements to BERT for creating high quality text embeddings included ERNIE, XLM, and RoBERTa. To measure the relative effectiveness of each NLP model in improving the predictions of memes as hateful or not, our team tested our baseline model which utilized BERT, and then replaced BERT with each of the other models in subsequent tests. All hyperparameters were kept the same across each test.

3.1.1 BERT

BERT is an NLP model designed by Google researchers to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications [3]. It was created in 2018 and has since been the basis for several other NLP models and variations. Surprisingly, in our experiments, the multimodal model that utilized BERT for creating text embeddings was the second highest performer in terms of accuracy, and the highest performer in terms of AUROC when using a learning rate of 1e-5 and 200 epochs.

3.1.2 RoBERTa

Robustly Optimized BERT Pretraining Approach (RoBERTa) is an improved version of BERT developed by Meta researchers which was created after these researchers determined that BERT was significantly undertrained and could perform much better with more tuning [6]. For our models, we tried the roberta-base and roberta-large-mnli checkpoints, the latter of which was specifically pre-trained for text sequence classification.

Our experiments resulted in modestly better performance for roberta-large-mnli which would seem to make sense given its tuning for sequence classification and it being a larger model overall. However, neither of these were the best performing models out of those tested.

3.1.3 XLM-RoBERTa

Cross-Language Model – Robustly Optimized BERT Pretraining Approach (XLM-RoBERTa) is an extension of the the RoBERTa model created by Meta designed to improve cross-lingual language understanding of the RoBERTa model [2]. In our experiments, XLM-RoBERTa performed at about the same level as the other RoBERTa models, with slight accuracy improvement over the roberta-base model.

We hypothesize that there was minor difference compared to the other RoBERTa-like models because the memes in our dataset contained English text, so having a cross-lingual model did not provide much benefit.

3.1.4 ERNIE

ERNIE is an enhanced language representation model which improves upon other NLP models by incorporating knowledge graphs for better contextual understanding [10]. This model was intriguing to our team because the task of predicting if a meme is hateful requires significant cultural

TEXT ENCODER	VISION ENCODER	CLASSIFICATION HEAD	LEARNING RATE	NUM EPOCHS	TEST AUROC	TEST ACC	TEST LOSS
ERNIE	CLIP-VIT-BASE	DistilBERT	1.00E-05	200	0.7367627	0.649	0.01199
BERT	CLIP-VIT-BASE	DistilBERT	1.00E-05	200	0.7433653	0.642	0.01245
ROBERTA-BASE	CLIP-VIT-BASE	DistilBERT	1.00E-05	200	0.7064506	0.608	0.01209
ROBERTA-LARGE-MNLI	CLIP-VIT-BASE	DistilBERT	1.00E-05	200	0.6950260	0.621	0.01237
XLM-ROBERTA	CLIP-VIT-BASE	DistilBERT	1.00E-05	200	0.6981473	0.616	0.01189

Table 2. Performance of our multimodal model with various NLP models for creating text embedding

TEXT ENCODER	VISION ENCODER	CLASSIFICATION HEAD	LEARNING RATE	NUM EPOCHS	TEST AUROC	TEST ACC	TEST LOSS
BERT	CLIP	DistilBERT	1.00E-05	100	0.74336	0.642	0.01245
BERT	Beit	DistilBERT	1.00E-05	100	0.68280	0.598	0.0116
BERT	ViT	DistilBERT	1.00E-05	100	0.69983	0.606	0.01218
BERT	Captioning + BERT	DistilBERT	1.00E-05	100	0.67280	0.58	0.0122

Table 3. Performance of our multimodal model with various Vision models for creating vision embeddings

knowledge since memes often reference real-world and fictional entities, symbols, and ideas. Therefore, a model such as ERNIE should yield better performance over other models without such a benefit. In our experiments, ERNIE did provide the highest accuracy but similar AUROC to BERT after 200 epochs of training each which was surprising. However, the accuracy and AUROC of our model continued to increase to 0.652 and 0.746534614, respectively, after 1000 epochs while using ERNIE as our text embedding method. Therefore, ERNIE was the most effective NLP model for creating text embeddings in our multimodal model.

3.2. Image Embeddings

The Meta researchers extracted image region features using Faster RCNN [8], which was released in 2015. Like our approach with text embeddings, we decided to try some more recent models in the space. Our initial hypothesis was that better image embeddings to contribute to model performance. The vision embeddings tried included:

- **CLIP**: This model was proposed by Radford et al [7]. The approach self-trained on (image, text) pairs from 400 million pictures on the internet. By predicting which caption goes with each image, the model was able to learn effective feature representations.
- **BERT Pretraining of Image Transformers (BEiT)**: Developed by Microsoft Research, Beit follows another self-supervised learning approach [1]. The model performs BERT-like pretraining by "tokenizing" the original image, masking portions of it and then using a transformer to recover the originally masked tokens.
- **Vision Transformer (ViT)**: The vision transformer was the first of its kind, and a step away from CNN's on computer vision tasks [4]. It is trained in a supervised way on ImageNet data. It has since been surpassed by newer models such as BEiT and Masked AutoEncoders by performance on ImageNet data.
- **Captioning**: As a novel approach, we tried converting the image into captions, and encode the text instead of vision models. The captions were created using catr [9] and then turned to text embeddings using BERT.

The pre-training checkpoints are marked in the appendix section A. The results from these experiments (shown in table 3) do show that the version of image embeddings does make a difference. More recent state of the art techniques, for example CLIP which has shown to work well with a range of datasets, generally perform better than previous methods. We expected Beit embeddings to perform better than ViT due its improved result in object detection tasks, but the results are not statistically significant in this case. The captioning method unfortunately did not perform well. Future work could try training the captions in an end to end way, with the meme text as segment 1 and the image caption as segment 2. This approach would be similar to how input is encoded in problems such as VQA. Overall, these experiments do suggest that using CLIP should be the preferred method when creating feature embeddings.

TEXT ENCODER	VISION ENCODER	CLASSIFICATION HEAD	TEST ACC
BERT	CLIP	DistilBERT	0.642
BERT	CLIP	VisualBERT	0.6087
BERT	BEIT	DistilBERT	0.598
BERT	BEIT	VisualBERT	0.593
ERNIE	CLIP	DistilBERT	0.649
ERNIE	CLIP	VisualBERT	0.6087

Table 4. Performance of our multimodal model with various Classification Heads

3.3. Classification Head

The concatenated visual and text embeddings are fed to the classification head to interpret and provide predictions. Two kinds of heads are implemented: DistilBERT and VisualBERT. DistilBERT consists of two linear layers followed by a sigmoid with ReLU activation and dropout in between. VisualBERT consists of 2 self-attention encoder layers loaded with the pretrained weights of the final two layers of huggingface's VisualBERT model, followed by a linear layer and a sigmoid. The VisualBERT head has roughly 20 times more parameters than the DistilBERT head (40M vs 2.3M) and the benefit of pretraining. This large difference in size and pretraining translates to similar accuracies obtained by the VisualBERT head with 10 times less training epochs as the DistilBERT head, as shown in Table 4.

The StepLR scheduler is used to finetune the model with a starting rate of 2e-5, gamma of 0.7 and step size of (epochs/4). This creates a schedule that reduces the learning rate by a factor of gamma every fixed number of steps. The representation capacity of the model is drastically increased, and we obtain some results that are comparable to the multimodal pretrained models in Kiela et. al [5] with the best model making use of visual bert-style pretraining and knowledge graph pretraining. The performance of the VisualBert head models vary more widely with some not improving over the linear DistilBERT head while others improve by a wider margin indicating that the quality of the representations provided by the unimodal embeddings affect the predictions of the multimodal head significantly. Perhaps unsurprisingly, models with visual embeddings from unsupervised pretrained models (CLIP, BeiT, ViT-MAE) outperform models with supervised pretraining (ViT). Similarly, models pretrained with more context like knowledge graphs (ERNIE) or visual features (VisualBERT), instead of just masked language tasks outperform models with only language pretraining (BERT, XLM)

3.4. Final Results

After conducting our experiments for individual components, we were able to take our best performing text model, vision model, and classification head to create our final multi-modal model. This multi-modal model consisted of an ERNIE model for text embedding, BEiT for image embeddings, and a VisualBert classification head. Setting the learning rate to 2e-5 and number of epochs to 24 yielded the highest accuracy of 0.7172. This is the last model shown in Table 4.

4. Conclusion

In our experiments, our best performing final model is not a combination of the best performing sub-parts. For example, VisualBERT did not appear to outperform the DistilBERT head (which was just two linear layers). Similarly, CLIP embeddings seemed superior ViT and BeiT embeddings. However, the best model does not include either of these. We suppose there is a complex interaction of these sub-parts that we could not fully understand.

4.1. Challenges

- **Running out of GPU memory for certain model variations**: Some of our first attempts at creating a multimodal model failed due to memory overloads on the GPU any of us used individually. Our team was able to fix the issue by moving variables to CPU to free up space on the GPU.
- **Subjectivity of what counts as hateful**: This problem is interesting because even humans have a low accuracy score (0.847) for correctly classifying hateful and non-hateful memes, relative to objective classifications like identifying an animal in a picture. Since the definition of what is hateful is determined by Meta and annotators must be trained on what fits this definition, there is more subjectivity to the labeling which means there may be much more misclassification in the training set and therefore more examples that contradict each other. This makes it more difficult for the model to learn. However, this is a challenge faced by all who are building a model for this problem, not just our team.
- **Difficulty beating SOTA accuracy scores**: Our best performing model achieved accuracy comparable to the SOTA in [5], but was not able to outperform it.

4.2. Successes

Our team collectively learnt a great deal about:

- **Multi-modal models**: We realized features from text and features from images can be combined in several

Student Name	Contributed Aspects	Details
Hassan Naveed	Coding Baseline, Model training, report writing	Implemented the Baseline Linear Layer and the workflow that allowed dumping hidden states from the text and vision embeddings for reuse. Created vision embeddings and reported score (same results as Jake). Conducted training of various model combinations. Wrote Introduction, and Image Embedding section for the final report. Shifted document to Latex and included Figures.
Saeb Hashish	Coding models, model training, proposal writing, report writing	Implemented the base models for DistilBERT and VisualBERT (including code/idea for using hidden state dumping to speed up training) Implemented and trained 7 variations of the models (mostly the VisualBERT variants) Contributed most of the proposal Contributed to the Classification Head and Experience sections of the report.
Dan Tylutki	Team organization, coding, model training, report writing	Created git project and Teams channel, and organized team meetings. Developed code for reading in CT scan images and UNet model (prior to project topic change) Developed code for image captioning and added image captions to training data set. Created two notebooks that fed meme text and captions through text classification models. Conducted training of 18 variations of our team's model to experiment and observe the performance effect of text embeddings created by various NLP models. Wrote outline for the project report and the following sections: Abstract, overview and Text Embeddings in Approach. Contributed to the Introduction section and Our Experience section.
Jake Banigan	Model Training, Image Embeddings	Created ViT and BeiT vision embeddings and trained model used these. Experimented with Fairface classifier. Results did not help so were excluded from the final report.

Table 5. Contributions of team members.

ways. The popular option was to concatenate features from a text model and a vision model then passing through a linear layer. Our novel approach used an image captioning model to convert images to textual descriptions, append the captions with the existing meme text, then pass the full text through an NLP model for text classification.

- **Text Embeddings**: For this problem, creating a way to get contextual information about entities mentioned in the meme text is helpful for improving scores. This can be done using a model like ERNIE which utilizes knowledge graphs.

- **Image Embeddings**: Several image embeddings were utilized in this research such ViT and BEiT, this allowed classifier to be able to look at the image and provide a prediction if there could be association with hate speech.

Although we were not able to defeat SOTA results, we were able to make a multimodal model that was different from Meta's but comparable in results.

5. Work Division

Table 5 shows the contributions provided by each team member.

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. 4

[2] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzman, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale, 2020. 3

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Pre-training of Deep Bidirectional Transformers for Language Understanding, BERT, 2018. 3

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 4

[5] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. *The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes.*, 2021, 2005. 1, 2, 3, 5

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. A Robustly Optimized BERT Pretraining Approach, RoBERTa, 2019. 3

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision ". 4

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv.org, January*, 6, 2016. 4

[9] Saahil Uppal. Catr: Image captioning with transformers. 4

[10] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Enhanced Language Representation with Informative Entities, ERNIE, 2019. 3

[11] Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Simple baseline for visual question answering. arxiv. preprint, 2015. 2

Appendix

A. Pretrained Checkpoints

The pre-trained checkpoints used are shown in Table 1

Encoder Type	Encoder Name	Developer	Checkpoint
Text	BERT	Google	bert-base-uncased
Text	ERNIE	Huawei and Tsinghua University	ernie-2.0-en
Text	RoBERTa	Meta	roberta-base
Text	RoBERTa	Meta	roberta-large-mnli
Text	XLM	Meta	xlm-roberta-base
Vision	CLIP	openAI	clip-vit-base-patch32
Vision	BEiT	Microsoft Research	beit-base-patch16-224
Vision	ViT	Google	vit-base-patch16-224-in21k
Vision	ViTMAE	Meta	Vit-mae-base

Table 6. Pretraining checkpoints for Text and vision models