Project Report : Hateful-memes

Yan Jiang
Georgia Institution and Technology
North Ave NW, Atlanta, GA 30332
yjiang347@gatech.edu

Ruochen Zhu
Georgia Institution and Technology
North Ave NW, Atlanta, GA 30332
rzhu78@gatech.edu

Abstract

Massive memes spreading in social media can convey different information and can even affect user's behaviors in the real world, therefore it is of great significance to filter out potential hateful memes. However, the classification of hateful memes suffers from enormous data and reversal underlying meaning, making it impossible to be processed by humans and challenging to be detected by machines. Originated from Facebook's hateful memes challenge, which provides 10,000+ new samples of multimodal content, this project aims to explore recent approaches that better capture the underlying meanings of hateful memes. In this report, we used VisualBert which combines vision and language in the "BERT" approach. We seek to advance the line of research by exploiting and tuning multimodal models with different amounts of extracted features utilizing Facebook's detectron and hyperparameter optimization using majority voting for robust final performance. In addition, we have documented our experiments, and analyzed our results with comparison to other baseline models. Our ensemble VisualBert model achieved 0.725 in accuracy and 0.661 AUCROC on the unseen test set. The broader impacts of this project shed light on a myriad of multimodal applications including visual question answering and facilitate the better understanding of multimodal reasoning. The code and notebook links go to see Appendix A.

1. Introduction/Background/Motivation

Memes, combining images with texts, have emerged as a popular form of expression worldwidely on social media platforms [19]. However, hateful memes attacking people directly or indirectly based on characteristics (such as gender, nationality, ece, religion, sexual orientation, disability, and disease or others) can cause malignant social impacts and affect behaviors of users in the real world [7].

Essentially, given the potential detrimental impacts, we need to prevent the spread of hateful memes. Meanwhile, it is impossible to manually tackle hateful memes by humans due to the enormous data for classification. To the end of mitigating the adverse impacts of online hateful speech, developing AI systems to automatically detect hateful memes is therefore of remarkable importance. The main challenge lies in the detection of multimodal hateful interpretations, which is intrinsically difficult due to the joint representation of visual and language understanding/reasoning domains (V+L) [12]. The majority of hate speech research focuses mostly on text. However, Harmless words combined with harmless images sometimes can finally generate harmful interpretations, such as "love the way you smell" combined with a skunk image [7]. With such interactions between the image part and the text part in the meme, conventional learning systems perform below expectations.

According to Facebook Research, existing state-of-the-art models for hateful memes detection are listed in Table 1. Most multimodal systems have adopted either a late-fusion (LF) or an early-fusion (EF) approach to deal with dual modalities. Late-fusion methods would utilize unimodal models to process visual and language signals independently and then combine their features before the final classification [12]. While early-fusion models such as MMBT [6], ViLBERT [13], and VisualBERT [10] , conversely, utilize complex structures to process modalities jointly within the model architecture. Multimodal models can be used in two forms, either pre-trained with unimodal models, or pre-trained with multimodal models.

Advanced performance on the detection of hateful memes have been made recently. Some of the investigated studies are discussed as follows.

In 2020, Rou Zhu from the Alfred La. [21] implemented a versatile ensemble of VL-BERT [16], UNITER [3], VILLIA [4], and ERNIE-Vil [20] models. Moreover, face features were extracted by Mask-RCN to generate race and gender labels, which in turn are introduced towards the inputs. The enhanced method has achieved 0.845 AUROC on the hateful memes detection dataset.

Phillip Lippe et al. [12] reported an AUROC score of 80.53 by using upsampling of contrastive examples and ensemble learning with pretrained Faster R-CNN to improve performance. This research group focused on three early-sufion pretrained models: LXMERT [17], UNITER [3], and OSCAR [11], which are recent popular multimodal learning models pre-trained on various V+L tasks such as visual question answering (VQA) and image captioning.

Another remarkable study conducted by Niklas Muennighoff was using a visual token to alleviate the differentiation between text and image contents. Moreover, stochastic weight averaging was added to stabilize training. Finally, Vi-

sualBERT [10], UNITER [3], ERNIE-Vil [20] and OSCAR [11] models were ensembled to improve the performance.

In addition, Riza Velioglu and Jewgeni Rose [18] have provided a simple solution compared to others, with only using VisualBERT [10] backbone. After pre-training VisualBERT on Conceptual Captions, the model was fine-tuned with additional dataset, and was then optimized using hyper-parameter search and majority voting. This method achieves 0.811 AUROC with an accuracy of 0.765.

Among all the research we have investigated, Veliglu and Rose's work has captured our interest since they have achieved a robust performance with a single VisualBERT model. Following the Occam's razor principle, we decide to implement and explore the hateful memes detection using VisualBERT architecture and majority voting technique. In this project, we propose to implement the multimodal hateful memes detection with achieving the following objectives: a detailed comparison with baseline approaches, using feature extraction to facilitate the optimization process, conducting majority voting for obtaining the model with the best performance, and a thorough analysis of experimental results from a structured perspective.

Type	Model
	Human
Unimodal	Image-Grid Image-Region Text BERT
Multimodal (Unimodal Pre-training)	Late Fusion Concat BERT MMBT-Grid MMBT-Region ViLBERT VisualBERT
Multimodal (Multimodal Pre-training)	ViLBERT CC Visual BERT COCO

Table 1: Baseline State-of-the-Art Models[7]

2. Dataset Details

The dataset is provided by Facebook Hateful-Memes Challenge, which pertains to multimodal vision-and-language. it can be downloaded from this site. It is noted that license agreement has to be agreed before downloading it because of ethical concerns. It contains 12140 images, and have already been separated into five datasets: train_default, dev_seen, dev_unseen, test_seen, and test_unseen. The details of datasets are shown in Table 2

Each record in the dataset contains five columns: id, image, text, and corresponding label. Based on the calculation, there are only 12140 images but summation of those five datasets is 12540. After digging the reason, we found there are 400 images overlap between dev_seen and dev_unseen datasets. Therefore, we decided to use train_default to train our model, dev_unseen as validation dataset to tune our model and use test_unseen dataset to test our model. Because those three datasets have similar distribution (percentage of hate rate).

	Total #	Hate #	Non-hate #	Hate%
train_default	8500	3019	5481	35.52%
dev_seen	500	247	253	49.40%
dev_unseen	540	200	340	37.04%
test_seen	1000	490	510	49.00%
test_unseen	2000	750	1250	37.50%

Table 2: Dataset Details

3. Approach

In this study, our general goal is training a model to classify hateful and non-hateful memes, and we would like to learn how the multimodal model works as well. In order to achieve this goal, we utilized VisualBert [10] as the base model and optimized it. There are four main steps in the optimization: image encoding/feature extraction, model training, majority voting ensembling, and evaluation/testing. Meanwhile, we also run baselines including as a benchmark to help us to evaluate our models The following sections will show details for each step.

3.1. Baseline Models

In this research, MMF was used as a starter codebase to explore state-of-the-art vision-and-language models. For baseline evaluation, we explore six baseline models from different categories [7] with training-validation and testing results : (1) Unimodal Image; (2) Late Fusion (The mean of the unimodal ResNet-152 and BERT output); (3) Concat-BERT (concatenate ResNet-152 features with BERT and an MLP); (4) MMBT (Multimodal BiTransformers); (5) ViLBERT CC (pre-trained on Conceptual Captions); (6) Visual-BERT COCO (pre-trained on COCO image). Due to the computation limit, each model was trained with 3000 rather than 22000 updates. The models were evaluated with evaluation metrics described in section 3.5.

3.2. Image Encoding/Feature Extraction

Instead of using default features in MMF, we did feature extraction or say image encoding by utilizing the fc6 layer of a ResNeXT-152 based on Mask-RCNN model [5] to extract specific number of boxes of 2048D region-based image features, which is trained on Genome [9] with the attribute prediction loss following [15]. In this research, we extracted 50 boxes and 100 boxes to see how the number of feature extractions affect our model performance. The workflow for image encoding is displayed in Figure 1. When the meme is fed, the image would be patched and text would be tokenized with positional indexes. In this step, visual embeddings are combined with the textual embeddings. Combined embeddings would be fed into transformer layers, and then go through the classifier. The weights $W_n \in \mathbb{R}^{P_x}$ is used to project each image embeddings to D-dimensional token input embedding space [18]:

$$I_n = W_n f(img, n)$$

Where P =2048, D = 768, and f(img, n) is output the nth fully-connected layer in the image encoder

3.3. Model Training

3.3.1 Pre-training

The visualBert is pre-trained on COCO image caption dataset [20], which is provided by MMF GitHub. The pre-trained model based on large-scale dataset – COCO which has larger image size (330K)[20] than images' in hateful-memes(10k), therefore, the pre-trained already has a good shape which saves much effort in following fine-tune steps. This also gives us more possibilities to achieve a high-performance model.

3.3.2 Fine-tune on Hateful-memes datasets

As Bugliarello et al. recently reported intriguing findings suggesting that differences between various vision and language BERT models are mostly due to training data and hyper-parameters [2], pre-training might help optimization.We utilized pre-trained visualBert model on our hateful-memes train_default dataset to fine-tune it, and evaluated trained models on dev_unseen dataset in accuracy, AUROC, and F1 score.

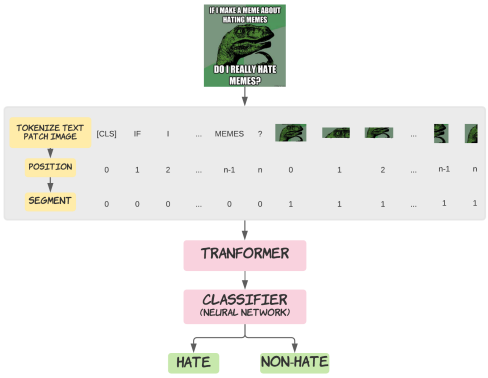

Figure 1: Image Encoding Workflow

3.4. Ensembling Learning

Ensembling learning is a very popular machine learning technique that combines several single trained models in order to obtain one optimal model. Because ensembling combines several models, its robustness and generalizability will be improved. Here, one of ensemble methods named majority voting (also known as hard voting) was employed to help models achieve better performance than a single model. In the majority voting, every single model votes for a class and majority vins. Specifical steps: (1) hyper-pramater search was used to generate several models; (2) sorted them by AUROC in ascending order; (3) selected top 5 models and collect their predictions (votes); (4) applied majority voting; (5) chose max predicted probability among those 5 models when prediction is 1 or say hate level or chose min predicted probability among those 5 models when prediction is 0 or say non-hate level as final probability for the sample.

3.5. Evaluation Metrics

After fine-tune the model, the model was tested on 2000 unseen samples. Since our task can be defined as a binary classification problem, three performance metrics are calculated including accuracy, The area under the receiver operating characteristics curve (AUROC) [1] , and F1 score. Three formula were used as below:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of predictions}$$

$$AUROC = \int_{x=0}^{1} TPR(FPR^{-1}(x)))dx$$

$$F1 = 2*\frac{precision * recall}{precision + recall} = \frac{TPR}{TPR + \frac{1}{2}(FPR + FNR))}$$

Where TPR is true positive rate, FPR is false positive rate, and FN is false negative rate.

4. Experiments and Results

4.1. Experiments and Results

We used a modified MMF setting to train baselines: (1) Unimodal Image; (2) Late Fusion ; (3) Concat BERT; (4) MMBT; (5) ViLBERT CC; (6) Visual BERT COCO. The detailed experimental settings and hyper-parameters are provided in Table 7 of Appendix B. of Appendix B. Although it was suggested in previous research [7] that 22000 updates training would provide the best performant models, we have decreased the number of training iterations down to 3000 due to the trade-off between performance and limited computation resources.

As we mentioned in section 3.2, we utilized the fc6 layer of a ResNeXT-152 based on Mask-RCNN model[21]to extract 50 and 100 of boxes of 2048D region-based image features. Pre-trained model used MMF visualBert default setting(see appendix B) on COCO dataset. During train visualBERT, we set different parameters.

- VisualBERT with 50 feature extraction(visualBERT-100) fixed 1000 training iteration, 80 batch size, 0.00005 learn ration, cosine scheduler type, 1000 warmup iterations. While we hyper-parameter searched different lr_ratios (0.3, 0.6), scheduler number of warmup steps (250, 500), and warmup factors (0.2, 0.6).

- VisualBERT with 100 feature extraction(visualBERT-100) fixed 32 batch size, 0.3 lr_ratio, 1000 training iteration, 250 warmup steps, 0.00005 learning ratio. While we hyper-parameter searched warmup iteration (500, 1000), warmup factor(0.1, 0.3), and scheduler type (cosine and linear).

The single model was evaluated by AUROC and accuracy while the final ensemble models were evaluated by accuracy, AUROC, and F1 scores, which formula are shown in section 3.4.

	visualBERT-50			visualBERT-100		
	lr_ratio	warmup_factor	#warmup_step	warmup_iter	warmup_factor	Scheduler_type
Model 1	0.6	0.2	500	1000	0.3	cosine
Model 2	0.6	0.6	500	500	0.3	linear
Model 3	0.3	0.2	250	1000	0.3	cosine
Model 4	0.6	0.6	250	500	0.1	linear
Model 5	0.3	0.2	500	1000	0.1	cosine

Table 3: VisualBERT Models Hyper-parameters

4.2. Baselines

The MMF framework [7]paves a foundation codebase to our baseline implementation. We use a batch size of 32 to train the model due to Colab's memory limitation. During the training, we target the binary cross entropy as the loss function and optimize the cross entropy loss by using Adam optimizer [8] with a weigh update epsilon of $1 * 10^{-8}$. In addition, among three evaluation metrics, we aim at achieving the highest AUROC as the criterion for best update result and early stop during the whole training and validation process. From Table 4, we can observe that Unimodal Image, as the only unimodal model presented in the baseline training, has the worst performance with a test AUROC of 0.5303, since it does not have a text encoder to gain the multimodal representations from the text input. Remarkably, a traditional multimodal model, Late Fusion suffers from the challenge with a low AUROC of 0.5601. Even the Concat BERT model is still performing the classification with struggles. On the other hand, MMBT, ViL BERT CC, and Visual BERT COCO have achieved AUROC over 0.6, which might be attributed to the Faster RCNN image encoder and the intrinsic early fusion pattern during the learning process. Among all six baseline models, ViLBERT CC and VisualBERT COCO have captured our interest as the two best performing baselines with test AUROC of 0.6339 and 0.6476 respectively. Provided with the best performance on the unseen test dataset, VisualBERT COCO is chosen as the backbone model for the further optimization.

Moreover, during the experiments (Appendix C), we have observed some overfitting and unstable performance. One reason behind the unstable performance could be the relatively small epsilon used in the Adam optimizer. Overfitting in multimodal learning could be a challenge to solve. Due to the limited exploration, we currently have no evidence suggesting the source of that overfitting. Research [12], showed that the baseline models might rely more on texture features and could have overfitting on text confounders however, approach using dropout across multiple-attention heads fails to mitigate the overfitting.[12]

4.3. VisualBERT Optimization

In order to stabilize the predictions and tackle down the overfitting problem, we utilized a R-152-FPN model (Faster RCNN with ResNet architecture) to extract features and to conduct ensemble learning. This model has fine-tuned weights form ImageNet and an inner optimization solver with a weight decay of 0.0001 and a base learning rate of 0.01. The default max iteration for this model is set as 180000. In addition, the model is pre-trained on Visual Genome with attribute loss. To the end of improving generalizability, after feature extraction, the ensemble learning is utilized to combine the top 5 models with majority voting. The resulting classifier is expected to have better performance than any single model we have trained. The AUROC and accuracy on dev unseen set for best 5 models with the two different feature extraction are shown in Table 5, which have confirmed the enhanced performance over the single baseline model after the feature extraction. In Figure 7 of appendix D, we have presented the cross-entropy loss for the top 5 models with two different feature extractions. After refining the hyper-parameters and ensemble learning, the performance has been improved in both feature extractions, achieving AUROC of 0.6489 and 0.6611 for VisualBERT-50 and VisualBERT-100 respectively in Table 6.

4.4. Optimized Visual BERT vs. Baselines

Recall the performance of baselines presented in Table 4, we compare the VisualBERT ensemble models with baseline models. Remarkable results have been revealed that ensemble learning with feature extraction have improved AUROC by 2.08 on unseen test set and have increased Accuracy by 2.04%.

The combining results suggest that the majority voting technique used in ensemble learning and feature extraction overall generate a robust model for the hateful memes detection. More comparison details and results can be found in the Appendix C and D.

	visualBERT-50		visualBERT-100	
	AUROC	Accuracy	AUROC	Accuracy
Model 1	0.7546	0.7278	0.7474	0.6889
Model 2	0.7388	0.6981	0.7289	0.6870
Model 3	0.7328	0.6907	0.7274	0.6889
Model 4	0.7280	0.6889	0.7215	0.6833
Model 5	0.7280	0.6852	0.7207	0.6685

Table 5: VisualBERT Model Performance on Validation Dataset

	Validation		Test		Best Iterations
	Accuracy	AUROC	Accuracy	AUROC	
Unimodal Image	0.6241	0.5808	0.6155	0.5303	1100/3000
Late Fusion	0.6352	0.6395	0.6390	0.5601	800/3000
Concat BERT	0.6296	0.6266	0.6525	0.5828	1600/3000
MMBT	0.6574	0.6726	0.6801	0.6243	1900/3000
ViLBERT CC	0.7056	0.6893	0.6980	0.6339	2600/3000
VisualBERT COCO	0.7019	0.7265	0.7105	0.6476	1600/3000

Table 4: Baseline Metrics

	AUROC	Accuracy	F1 Score
VisualBERT-50	0.6489	0.7125	0.5073
VisualBERT-100	0.6611	0.7250	0.5250

Table 6: VisualBERT Ensemble Model Metrics

5. Discussion and Challenges

5.1. Discussion

Considering the intrinsic multimodal structure of our classification problem with a large dataset. The anticipated challenges and problems are (1) which platform is most cost-effective for processing large datasets in this research; (2) which baseline and benchmark we should use; (3) how to do feature extraction to optimize the model. we proposed to solve expected problems mentioned above by (a) comparing AWS and Google Colab (b) researching the state-of-the-art baseline models and choosing representatives from different types (multimodal vs unimodal, multimodal pre-training vs. unimodal pre-training) (c) investigating the feature extraction mechanisms and searching for applicable image encoding models.

5.2. Challenge

In reality, we experienced several more challenges: (1) the biggest one is resource limitation, we chose google Colab as our tech stack and we even paid pro and pro+ features, but it has a daily quota limit, disconnection issue, and memory limit. Based on original plan, we would compare ViLBERT and VisualBERT on different feature extraction models (R-101-FPN and R-152-FPN), and do more hyper-parameter searching such as picking top 8 models among 64 candidates (while the process was killed several times due to limitations). Each feature extraction process would process 12140 images and usually takes 8-10 hours to complete. Although we had successfully extracted features using both R-101-FPN and R-152-FPN models several times, the following training session was terminated with CUDA error. Therefore, such limitations and issues make us give up multiple training with various feature extraction and narrow down search ranges. We even used four google accounts to train models simultaneously for the last few days; (2) the next unanticipated problem we have encountered is dataset upload issue to google drive, since the original competition data download API was deprecated, we had to fill out an agreement and download more than 8G datasets to local and then upload to google drive and we encountered a bunch of failures due to environmental settings; (3) Understanding the format of the dataset and representations with dual visual and language modularity domain is more challenging than we had originally expected

6. Conclusions

6.1. Conclusion

In this project, we aim to facilitate the understanding of multimodal reasoning for solving real-world applications like hateful memes detection. We justify our choice of Visual BERT model as backbone architecture by evaluation the baseline models. Further, we propose to improve the baseline performance through two approaches, feature extraction and ensemble learning. As expected, we have improved the performance of multimodal approaches beyond state-of-the-art baselines by 2.08% of AUROC on test set and have shown the effectiveness of ensemble learning based to improve generalizability and robustness. Although debatable, we think that pretraining offers a way to improve the performance for the multimodal learning. Moreover, ensemble learning improves performance, therefore combining more advanced models could better the performance, however, we must consider the trade-off between performance enhancement and computational limitation with deploying constraints.

6.2. Outlooks

Current state-of-the art multimodal models still have a lot of room for the improvement. A promising approach can utilize entities like faces in memes to generate labels like gender, sex, or even race, and then feed these generated labels to the input to improve performance. Another further research direction for improving model performance could be confounder upsampling and re-weighting loss since the inclusion of benign confounders is a key feature in hateful memes detection dataset.

Moreover, according to Facebook Research [7], it is revealed that 47.1% of protected category is from race or ethnicity and only 0.4% form socioeconomic class can be protected. Therefore, real-world knowledge might help improve the detection of hateful memes, since reasoning from memes usually needs subtle world knowledge.

6.3. Broader Impacts

Developing hateful memes detection ability could help to create a beneficial atmosphere among the network community. As to the influence, this project endows us with a deeper understanding of how multimodal learning can be applied to improve integral understanding of images and languages. Furthermore, this project also has enlightened us with potential solutions for the detection of hateful content and multimodal reasoning. In the wake of the better understanding of multimodal reasoning, we can transfer the multimodal learning on natural language processing, computer vision evaluation and embodied AI development [14]. Meanwhile, the potential misuse of multimodal system needs to be considered, and this risk could be mitigated by establishing precautionary regulations and developing advanced AI systems.

7. Team Contribution

Student	Contributed Aspects	Details
Yan Jiang (yjiang347)	Analyzing Dataset, Implementation, Live Coding, Code Reviews, Results Analysis, Paper Reviews	• Analyzing Dataset • Baseline Training & Test • Image encoding • Models Training & Test • Discussion & Meetings • Documenting Report
Ruochen Zhu (rzhu78)	Obtaining Dataset, Implementation, Live Coding, Code Reviews, Results Analysis, Paper Reviews	• Acquiring Dataset • Baseline Training & Test • Feature Extraction • Models Evaluation • Discussion & Meetings • Documenting Report

Figure 2: Team Contribution

References

[1] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997. 3

[2] Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. Multimodal pretraining unmasked: Unifying the vision and language berts. *arXiv preprint arXiv:2011.15124*, 2020. 3

[3] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 1, 2

[4] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *arXiv preprint arXiv:2006.06195*, 2020. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[6] Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*, 2019. 1

[7] 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. *arXiv preprint arXiv:2005.04790*, 2020. 1, 2, 3, 4, 5

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[9] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 2

[10] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 1, 2

[11] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 1, 2

[12] Phillip Lippe, Nithin Holla, Shantanu Chandra, Santhosh Rajamanickam, Georgios Antoniou, Ekaterina Shutova, and Helen Yannakoudakis. A multimodal framework for the detection of hateful memes. *arXiv preprint arXiv:2012.12871*, 2020. 1, 4

[13] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019. 1

[14] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 6

[15] Amanpreet Singh, Vedanuj Goswami, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Mmf: A multimodal framework for vision and language research, 2020. 2

[16] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 1

[17] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 1

[18] Riza Velioglu and Jewgeni Rose. Detecting hate speech in memes using multimodal deep learning approaches: Prize-winning solution to hateful memes challenge. *arXiv preprint arXiv:2012.12975*, 2020. 2

[19] Amanda Williams, Clio Oliver, Katherine Aumer, and Chanel Meyers. Racial microaggressions and perceptions of internet memes. *Computers in Human Behavior*, 63:424–432, 2016. 1

[20] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graph. *arXiv preprint arXiv:2006.16934*, 1:12, 2020. 1, 2, 3

[21] Ron Zhu. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. *arXiv preprint arXiv:2012.08290*, 2020. 1, 3

A. Appendix: Accounts and Code links

Since the computing resource and Google Colab's limitations, we employed four gmail accounts to work on Colab with this project. The code is available at (links):

- shuweizhao1991@gmail.com
- yanjiang.joy@gmail.com
- yr97@cornell.edu
- zhuruochen0@gmail.com

We used four notebooks to train baseline and VisualBERT models, the links as below:

- Jupyter Notebook Folder for VisualBERT Model Training
  - you can find visualBERT Model Training code and outputs in this Jupyter Notebook
  - You can find final visualBERT model prediction csv in this results
  - Training log information can be found VisualBERT-100 and VisualBERT-50
- Jupyter Notebook Folder for Baseline Evaluation
  - This link goes to Unimodel & Late Fusion baseline training jupyter notebook
  - This link goes to VilBERT & Concat & BERT, MMBT baseline training jupyter notebook
  - This link goes to VisualBERT baseline training jupyter notebook
- Data & Implementation Folder for Baseline Evaluation
- Baseline records

.

We have used the Facebook Research MMF modular framework as an initial start for the project and explored and implemented the ensemble learning from Riza Velioglu and Jewgeni Rosethe of the Team Hate Detection

- Implement the metrics evaluation of Accuracy, F1 Score, and AUROC from scratch.
- Heavily modified code to implement baseline benchmarks from different types of models.
- Modified image encoding to extract different numbers of features and using different feature extraction alternatives.
- Modified ensemble learning and majority voting to obtain the best 5 models.
- Modified MMF format conversion.
- Modified hyper-parameter search for model optimization

B. Appendix:Baseline Implementation Details

- Unimodal Image yaml file
- Late Fusion yaml file
- Concat BERT yaml file
- MMBT yaml file
- ViLBERT CC yaml file
- VisualBERT COCO yaml file

The Table 7 below shows more details for baseline models.

C. Appendix: Baseline Charts

Figure 3: Baseline Metrics Curves

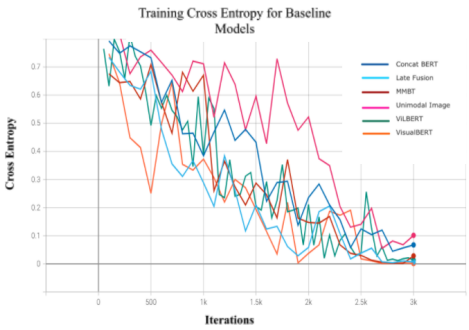

Figure 4: Baseline Cross Entropy Curves

D. Appendix: VisualBERT Metrics Charts

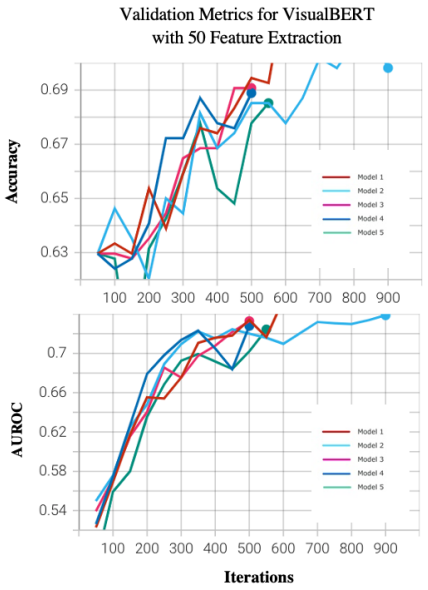

Figure 5: VisualBERT with 50 Feature Extraction Metrics Curves

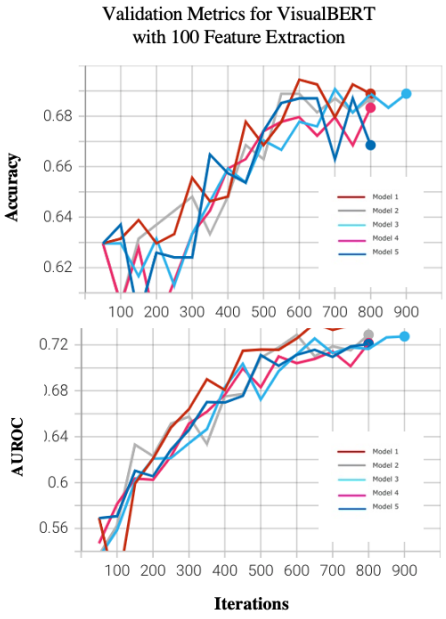

Figure 6: VisualBERT with 100 Feature Extraction Metrics Curve

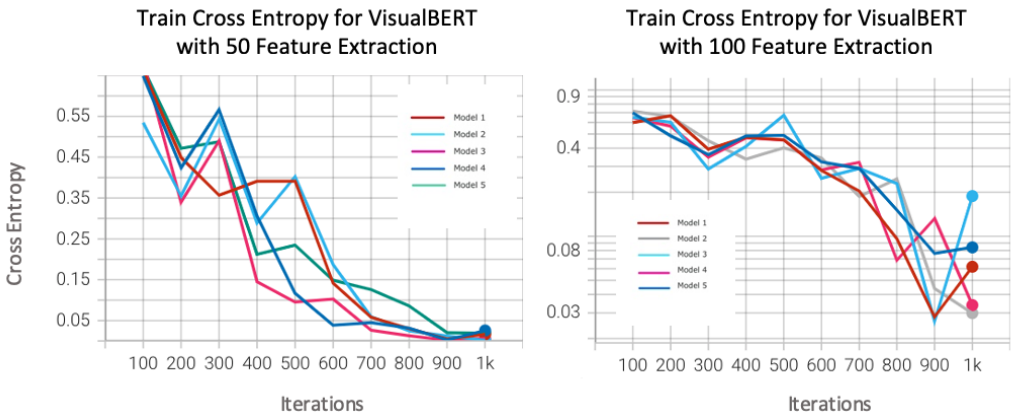

Figure 7: VisualBERT Cross Entropy Curves

	Batch size	Total parameters	Max Updates	Peak LR	Classifier Layer	Text Encoder	Image Encoder
Unimodal Image	32	60312642	3000	1e-5	2	N/A	ResNet-152
Late Fusion	32	170980676	3000	1e-5	2	BERT	ResNet-152
Concat BERT	32	170384706	3000	1e-5	2	BERT	ResNet-152
MMBT	32	115845890	3000	1e-5	1	BERT	FasterRCNN
ViLBERT CC	32	247780354	3000	1e-5	2	BERT	FasterRCNN
VisualBERT COCO	32	112044290	3000	1e-5	2	BERT	FasterRCNN

Table 7: Baseline Model Details