Hateful Memes Detection: CS 7643

Fangtingyu Hu, Yafen Zhang, Sheng Luo, Zhiyang Xia
Georgia Institute of Technology

Abstract

In this project, we implement different kinds of multimodal models on the hateful memes detection task. We first train and test the performance of models with different fusion position: late fusion, mid fusion (Concat Bert) and early fusion (MMBT and VisualBert). Second, we build three kinds of mid fusion models (Concat Bert, Allign Bert, Cross Bert) and test their performance. For the VisualBert model, we test two types of pretrainings: Multimodal and Unimodal pretrainings. Finally, we expand the task to finer grained hateful memes task and implement the VisualBert technique to both the single and multi Task modeling.

1. Introduction/Background/Motivation

1.1. Motivation

Online hate speech has widely impacted our societies. As such, social media companies have been developing automatic methods for censorship purposes due to its big impact on our daily lives [12]. In addition, the approach to find, rate, and remove hate content must be reliable and accurate so that the balance between ensuring open discussions and protecting people's freedom can be achieved.

Moreover, detecting online hate is difficult and machine learning models have raised concerns about the performance, robustness, generalization and fairness. To advance the capability of deep learning classification and capture the detailed basis for each classification, the fine-grained hatefulness detection model is studied. The model is developed toward modeling the sub-tasks of multiple labels, e.g. protected categories (women, black people, immigrants), and type of attack (inciting violence, dehumanizing, mocking).

Our work can provide an insightful comparison among models and hyper-parameters for both binary and fine-grained hateful memes detection and helps to identify the recommended fusion methods for tasks that require a holistic view between different modalities (e.g., text and images), such as hateful meme classification.

1.2. Introduction and Background

Hateful memes are a growing problem as social media platforms can be used to spread hateful speech. Recent research has explored various approaches to hateful meme classification, some studies developed unimodal models to detect hateful messages, such as Image-Region [7], Text-Bert [1], and Image-Grid [2]. Others have focused on using multimodal information, such as the text and images in the memes, to improve the performance of the classification models. Recent ones include MMBT [3], Concat Bert, and Visual Bert [4]. Many other studies have explored the use of transfer learning, where pre-trained models are fine-tuned on hateful meme data, to improve the efficiency and effectiveness of the classification.

Fusion is important in hateful meme classification. Intuitively, fusion represents the way that image and text interact. There are three major fusion approaches in the multimodal deep learning model: early fusion, mid-fusion, and late-fusion.

As the name suggests, early fusion combines information at input layer of the model, which allows the model to learn interactions between image and text from the beginning. Take the supervised multi- modal bitransformer (MMBT) model for example [3]. The MMBT model (Figure 4) uses the power of the bitransformer's ability to employ self-attention to look at the text and the image at the same time. It concatenates linear projections of Resnet output with BERT token embeddings into a sequence as Transformer input. The inputs containing the three encodings from both the image and text are transferred into the transformer and then using the classifier to get the output labels.

Mid-fusion allows image and text to interact in the middle of the training process before they reach the final prediction stage. The detailed exploration on different mid-fusion models is discussed in the later sections.

Late fusion combines outputs processed separately at the output layer, just like ensembled unimodal models. Late fusion allows flexibility in each model but restricts interactions between image and text information. The Late-fusion model utilizes Resnet-152 as the image encoder and BERT

base uncased transformer as the text encoder and a two-layer MLP for classification. The mean of the image and text classifiers is the final output.

For any task that pertains close relationship (but not necessarily high correlation) between different modalities, a good fusion approach can substantially improve model performance. However, there lacks a comprehensive analysis on how model fusion types influence the hateful meme detection. We aim to fill this gap by reviewing average model performance by early, mid, and late fusion approaches, and experiments different mid-fusion architecture to provide a comparative analysis on fusion influence. By doing so, we can identify the preferable fusion types that contribute to hateful memes classification. Also, potential new fusion ideas are investigated for better outcomes.

Aside from different model architectures and feature extraction methods, hyper-parameter tuning and the use of pre-trained model is also essential in improving model performance. Some top-performing implementations are combinations of all of those components [11, 13]. Recent development is made on fine-grained hateful speech detection on two aspects: one is detecting protected categories (race, disability, religion, nationality, sex), and another is detecting attack types (contempt, mocking, inferiority, slurs, exclusion, dehumanizing, inciting violence). This multi-label and multi-class task has imposed additional complexity on the challenge.

When modeling multiple sub-tasks from the same data set, it is possible to include all tasks into one model, i.e. multi-task model, as shown in Figure 1. The rationale is to share the knowledge between different tasks by using a shared architecture to promote model quality [6]. Another motivation of this approach is to reduce model training resources.

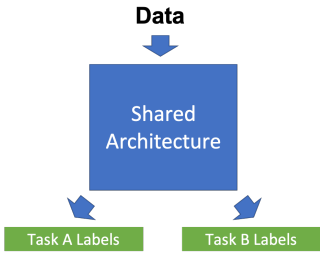

Figure 1. Multi-task model scheme.

1.3. Project Objectives

In this project, we are trying to achieve the following objectives: 1) Learn about the structures and characteristics of models used in hateful meme classification tasks. 2) Explore hyperparameter tuning to improve the performance of existing models. 3) Explore the better way to combine text and image information for better performance. 4) Identify the best model for fine-grained hateful meme classification tasks. 5) Explore multi-task learning with deep neural networks.

1.4. Data Source

We use the facebook hateful memes dataset (here) and fine grained annotation (here) for the shared task modeling. The dataset has three groups of labels:

(1) Hate: not_hateful, hateful. (2) Protected category: religion, race, sex, nationality, disability, pc_empty. (3) Attack type: dehumanizing, inferiority, inciting_violence, mocking, contempt, slurs, exclusion, attack_empty.

It should be noted that (1) labels are balanced, however, (2) and (3) labels are imbalanced. (Table 1 and Table 2)

label	train	validation	test
not_hateful	5493	254	341
hateful	3007	246	199
Total	8500	500	500

Table 1. Hateful Memes Dataset: hatefulness [5].

	fine-grained attributes	train	validation	test
Attack type	dehumanizing	1318	104	121
	inferiority	658	35	49
	inciting_violence	407	23	26
	mocking	378	29	35
	contempt	235	6	10
	slurs	205	4	6
	exclusion	114	8	13
Protected category	religion	1078	77	95
	race	1008	63	78
	sex	746	46	56
	nationality	325	20	26
	disability	255	17	22

Table 2. Hateful Memes Dataset: fine-grained labels [5].

2. Approach

All of our implementations are built upon MMF framework [8], which is a modular framework from Facebook AI, powered by Pytorch, for multimodal vision and language research. It contains built-in state-of-art vision and language models such as ViLBERT, TextCaps, etc. In this work, we have developed multi-modal and multi-task models in MMF using its modular configuration, debugged the data input set-ups, added multi-label vocabulary file and developed scorers for computing the prediction performance.

2.1. Mid-Fusion Experiments

To get a deep understanding about the role of fusion in hateful memes classification, we survey previous models and summarize their performance based on fusion types (Table 12). After reviewing the statistics, mid-fusion appears to be favorable in the hateful-meme classification

task. Therefore, to explore the boundary of mid-fusion and offer some new insights about what kind of mid-fusion is the best choice, we conduct comparative experiments by using different types of mid-fusion in hateful meme tasks, while holding other model components constant. We build Concat-Bert, Align-Bert, and Cross-Bert models from scratch and implement them under MMF framework.

As shown in Figure 2 – model architecture, we use ResNet152 to encode images and BERT to encode texts, after going through a projection training layer, we applied a mid-fusion layer (in three different types, shown in Figure 3) before feeding it to the MLP classifier. Details discussed below.

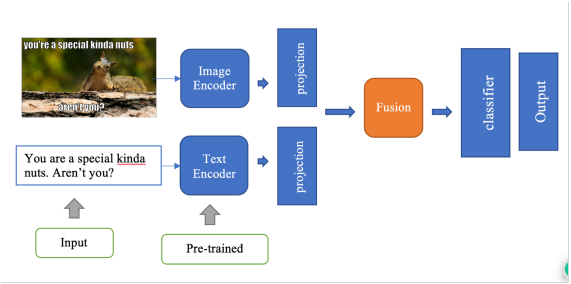

Figure 2. Model architecture for Concat-Bert, Align-Bert, and Cross-Bert.

2.1.1 Image Encoder & Text Encoder

An image input is passed through a pre-trained image encoder based on ResNet 152 to extract image features. ResNet 152 is a convolutional neural network that takes advantage of residual learning and skip connections (He et al., 2015), it is pre-trained on ImageNet-1k at resolution 224x224. The dimension of the embedding finally returned by the image encoder is 2048. Similarly, text input is passed through a pre-train text encoder based on BERT("bert-base-uncased") to extract text features. BERT [1] was a bidirectional transformer pre-trained on unlabeled corpus in a self-supervised way (i.e., masked language modeling (MLM), next sentence prediction (NSP)). The dimension of the embedding finally returned by the text encoder is 768.

The purpose of those encoders is to generate an image/text vector to represent image/text information.

2.1.2 Projection Layer

Text/Image embeddings were fed into their own trainable projection layer before fusion. This simple linear layer is used to further extract features from image and text vectors. This layer can also help to align dimensions between the image and text before fusion is applied. The output dimension is n (we set n=32 in our experiment).

2.1.3 Fusion

Three types of mid-fusion are shown in Figure 3, which represent different ways that text and image information integrates:

a. Concatenation. The baseline concatenation fusion is commonly used in the Concat-Bert model [14], which simply concatenates text and image information (of dimension n) to form a new vector (of dimension 2n). This method does not allow direct cross-interactions between text and image vectors.

b. Align. The align fusion allows element-wise multiplication between image and text projection vectors. If summing those up, intuitively, it represents the alignment angle (i.e., cosine-similarity) between text and image information. The fusion output dimension n is the same as fusion input n.

c. Cross. The Cross fusion is an extension of align fusion, which not only includes the element-wise inner product between image and text vectors but also all the outer products (see examples in Figure 3c). The fusion output dimension is n^2, which is the largest among the three fusion types. Align fusion can be viewed as a special case of Cross fusion with only the diagonal elements included. Intuitively, Cross-Bert is able to allow high flexibility in image and text interactions and carry more extensive information in their correlations.

The new idea in this project is to compare those different types of mid-fusion to evaluate model performance in a comparable way. We hold other model components constant, such as the pre-trained encoders, the in and out dimension of projection layers, hidden-layer dimensions of the MLP classifier, batch size, and learning rate, to make the results more comparable. Due to the high volume of information contained in cross-fusion and high level of interactions allowed in cross and align fusion, we believe they should generate higher performance than the baseline (Concat-bert). Hence, we present the following hypotheses:

Hypothesis 1: Cross-Bert and Align-Bert achieve better model performance the baseline Concat-Bert.

Hypothesis 2: Cross-Bert performs better than Align-Bert.

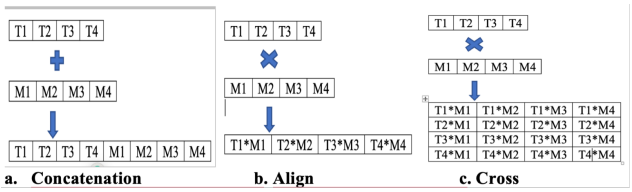

Figure 3. Mid-fusion types – concatenation (a), align (b), cross (c). T1,T2,T3,T4 represents text vector, M1,M2,M3,M4 represents image vector.

2.1.4 Classifier

The fusion output is passed through a MLP classifier (multi-layer perceptron classifier) for final classification labels. We implement a simple two-layer MLP classifier with a hidden dimension of 16, along with the cross-entropy loss function. The final output is a binary label: hateful vs non-hateful.

2.2. Hyperparameter tuning & transfer learning

We performed hyperparameter tuning over the learning rate, batch size, and warm-up. The tuned hyperparameters and the corresponding validation and testing results are shown in Table 3. To some degree, smaller batch size and learning rate would result in better performance.

We explored multi-modal systems (Fusion, Concat Bert, MMBT, VisualBert). These models generally have a vision modal which takes image as input and output image feature embeddings, and a language modal takes words as input and output the words embeddings. The output from two modals are merged and is further used to predict the binary labels

Moreover, we also investigated the impact of pre-trained models on the final performance with the early fusion model Visual BERT. Visual BERT is based on the single stream BERT model with multiple transformer blocks where image and language are combined by a Transformer to allow the self-attention to discover alignment between vision and language [1] [10] [4]. When Visual BERT is pre-trained with a multimodal like COCO, it was introduced with bounding boxes, segmentation, and key points for 80 common categories to advance in object detection [9]. Therefore, it is expected that the multimodal pretraining (Visual BERT COCO) would result in better performance than the unimodal pretraining.

2.3. Explore refined-grained modeling

For fine-grained and shared tasks modeling, we explore different models (MMBT vs VisualBert) and compare refresh vs transfer learning (using pretrained) learning approaches. To explore multi-task modeling, we merge the two tasks' modeling (protected groups & attack types) into a single model by including both labels. Each label is set as a single binary classification. The loss function is the sum of binary cross-entropies for all labels. We use Adam's optimizer. The learning rate is applied as 2e-5. A warm-up period of 300 batches is used with the warm-up learning ratio as 0.3. The batch size is 32. The models generally overfit for these tasks. The training is stopped when the AUROC does not improve after 1000 batches.

We evaluate the performance by accuracy, AUROC, and f1 score. Since each task comes with multiple labels, we further examine the confusion matrix for understanding its performance on sub-categories.

3. Experiments and Results

3.1. Hyperparameter Tuning and Model Survey

We conduct a grid search for hyperparameters based on the Visual Bert model (Table 3). We also compare the training approach with parameters pretrained from uni- or multi-modal models. With transfer learning (TL) of pretrained parameters from the multi-modal model with COCO dataset, we found that the model performance is superior (Table 4). This is expected because Visual Bert is essentially multi-modal type, and parameters transferred from multi-modal models should lead to better results.

Learning Rate	Batch Size	Warmup Ratio	Accuracy	AUROC	F1 Score
1e-5	32	0.6	0.6907	0.7278	0.5399
5e-5	32	0.6	0.7000	0.7369	0.5371
1e-4	32	0.6	0.6963	0.7271	0.5060
5e-5	16	0.6	0.7000	0.7290	0.5525
1e-4	16	0.6	0.7074	0.7210	0.5798
5e-5	32	0.3	0.7259	0.7528	0.5488
5e-5	32	1	0.7167	0.7407	0.5666

Table 3. Hyperparameters tuning on Visual BERT COCO.

Model	Validation			Test		
	Accuracy	AUROC	F1 Score	Accuracy	AUROC	F1 Score
TL from multi-modal pretrained	0.7259	0.7528	0.5488	0.7180	0.7584	0.5580
TL from unimodal pretrained	0.6704	0.6884	0.4765	0.7030	0.7516	0.4780

Table 4. Visual BERT: pretrained on unimodal vs multi-modal

We tested a few models: Late Fusion, concat bert, MMBT and Visual Bert (Table 5), and we found that Visual Bert performs the best. This is probably because the Visual Bert model effectively merges information by language and image embeddings, and it also applies the self-attention mechanism and ordering information (position embedding). These techniques help to extract useful information for correctly classifying hateful or not.

Model	Validation		
	Accuracy	AUROC	F1 Score
Late Fusion	0.6407	0.6498	0.3782
Concat Bert	0.6431	0.6503	0.3888
MMBT	0.6574	0.6611	0.3071
Visual BERT	0.6704	0.6884	0.4765

Table 5. Model performance survey

3.2. Mid-Fusion Experiment

Next, we investigate the "fusion" mechanism in multi-modal system.

3.2.1 Mid-fusion Setup

All models are trained based on the train split and are validated on "dev seen" of hateful memes datasets. Training

and evaluation are done via MMF libraries. We use Colab GPU with High-Ram to execute our experiments. MMF library has an in-house Concat-Bert model and we use that as the base but modify it in the following ways: adding a trainable projection layer before concatenation happens and altering the fusion methods to define other models – Align-Bert and Cross-Bert. All three models share similar configurations (see details in Table 6), to make them as comparable as possible. Experiments take about 1-2hrs per model for a combined training and validation session.

Batch size:32	Classifier: MLP
Epochs (max): 12	Loss: cross-entropy
Optimizer: AdamW	Pertained image encoder: ResNet152
Learning rate:5e-5	Pertained text encoder: Bert-base-uncased
Epsilon:1e-8	Projection layer output dimensions:32
	Fusion dimensions: Concat-Bert: 64; Align-Bert: 32; Cross-Bert: 1024

Table 6. Configuration for models used in mid-fusion experiments (Concat-Bert, Align-Bert, and Cross-Bert).

3.2.2 Mid-fusion Results

Model performance is measured in 3 metrics: AUROC, accuracy, and binary-f1. Table 7 shows the results. It is clear that align and cross fusion achieve better modal performance than simple concat fusion for hateful meme classification: Align-Bert gives the highest AUROC value of 0.6017, followed by Cross-Bert with an AUROC of 0.5680. Both out-perform Concat-Bert which has an AUROC of only 0.5232.

We find support for Hypothesis 1. This is expected because align fusion and cross fusion allow direct interactions between text and image vector information. Such interaction facilitates the model to process information more efficiently and learn the correlation between image and text. Surprisingly, we do not find support for hypothesis 2. Though cross-fusion carries more information than aligned fusion (because align-fusion is the diagonal case of cross-fusion), it appears that additional information becomes a burden rather than values. It is also possible that extra information requires a more complicated classifier to decipher, while we only implement a simple two-layer MLP classifier in this experiment.

As noted in Table 6, the fusion dimensions are different in 3 models: for a projection dimension of n, concat-bert has 2n features, align-Bert only has n features, while Cross-Bert requires n^2 features. As a result, Cross-Bert requires a more extensive computation resource. For simplicity of our training (and limiting resources), we set n=32, but in real practice, we recommend selecting a larger number to avoid compressing the text/image information too fast too early.

It is also important to note the limitation of our model: this fusion experiment does not try to generate state-of-art performance, but aims to test the performance of a baseline model under three types of fusion methods. We adopt a simple architecture - only a single-layer projection before fusion, and a two-layer classifier - without additional feature extraction or data augmentation. Also, due to limited computation resources, we only perform 12 epochs of training with a small batch size of 32. Thus, the performance of our models cannot (and should not) catch up to those presented earlier.

Model	Fusion type	Evaluation ("Dev seen")			Total parameters
		AUROC	Binary f1	Accuracy	
Concat BERT	Concatenation	0.5232	0.2456	0.6019	167,717,634
Align BERT	Align (dot product)	0.6017	0.3636	0.6630	167,717,122
Cross BERT	Cross (outer product)	0.5680	0.0913	0.6315	167,786,370

Table 7. Model performance using different types of mid-fusion..

3.3. Fine Grained & Single-/Multi-Task Modeling

3.3.1 Single-Task Modeling

We explore models and training approaches for the fine grained classification task (Table 2). We found that VisualBert gives slightly better performance over the MMBT model. As for training with pretrained weights (transfer learning, TL) vs without pretrained model (model refresh), we found that transfer learning gives better performance for both Task A and B, probably because the learning tasks on other dataset help the model to capture features for these new learning tasks.

Performance-wise, we got accuracy and AUROC ~0.9. However, f1 scores are ~0.6, which suggests that precision and recall are not well balanced. We further examine the confusion matrices for each label. For Task A (protected groups), the pc_empty label (no hatefulness against any protected group) is generally modeled well, along with race, sex, religion (Table 9). However, for disability and nationality, the model heavily leans towards predicting negative but missed the positive samples (i.e. low recall). For Task B (attack types), the model successfully predicts for attack_empty, dehumanizing, slurs, but recall is very bad for exclusion, mocking and contempt predictions (Table 10). The reason could be that such categories' positive cases are very sparse in this dataset that the models generalize them well. A possible improvement could be using up-sampling approach to enrich the sparse labels.

	Task A - protected category			Task B - attack type		
Majority Baseline	0.7			0.72		
	accuracy	AUROC	f1	accuracy	AUROC	f1
MMBT (TL)	0.877	0.898	0.643	0.899	0.904	0.564
VisualBert (refresh)	0.877	0.899	0.634	0.894	0.897	0.556
VisualBert (TL, Single Task)	0.882	0.899	0.635	0.913	0.919	0.644
VisualBert (TL, Task A+B)	0.886	0.897	0.651	0.915	0.913	0.637

Table 8. Fined-grained modeling results: single & multi-task modeling.

pc_empty

Gold Values	FALSE	115	84
	TRUE	88	253
		FALSE	TRUE
		Predicted Values	

race

Gold Values	FALSE	441	36
	TRUE	40	23
		FALSE	TRUE
		Predicted Values	

sex

Gold Values	FALSE	486	8
	TRUE	24	22
		FALSE	TRUE
		Predicted Values	

religion

Gold Values	FALSE	440	23
	TRUE	43	34
		FALSE	TRUE
		Predicted Values	

disability

Gold Values	FALSE	521	2
	TRUE	17	0
		FALSE	TRUE
		Predicted Values	

nationality

Gold Values	FALSE	519	1
	TRUE	18	2
		FALSE	TRUE
		Predicted Values	

Table 9. Confusion matrices for Task A – Protected groups (VisualBert, TL, single task).

attack_empty

Gold Values	FALSE	45	151
	TRUE	30	314
		FALSE	TRUE
		Predicted Values	

dehumanizing

Gold Values	FALSE	434	2
	TRUE	90	14
		FALSE	TRUE
		Predicted Values	

inferiority

Gold Values	FALSE	505	0
	TRUE	34	1
		FALSE	TRUE
		Predicted Values	

inciting_violence

Gold Values	FALSE	517	0
	TRUE	23	0
		FALSE	TRUE
		Predicted Values	

slurs

Gold Values	FALSE	536	0
	TRUE	2	2
		FALSE	TRUE
		Predicted Values	

mocking

Gold Values	FALSE	505	6
	TRUE	22	7
		FALSE	TRUE
		Predicted Values	

exclusion

Gold Values	FALSE	532	0
	TRUE	8	0
		FALSE	TRUE
		Predicted Values	

contempt

Gold Values	FALSE	534	0
	TRUE	6	0
		FALSE	TRUE
		Predicted Values	

Table 10. Confusion matrices for Task B – Attack types (VisualBert, TL, single task).

3.3.2 Multi-Task Model

Comparing single- vs multi-task results, we observe that the metrics are relatively close (Table 8). This is interesting because the multi-task model (Figure 1) that has the same model capacity as the single task one can actually model for both tasks without performance regression. However, we should also point out that the model performance doesn't improve either. That suggests in this model architecture the shared knowledge between modeling protected groups and attack types does not benefit the other.

4. Experience

4.1. Challenges

We anticipated challenges in obtaining data sources when it was initially unavailable from the official website. We contacted the facebook personnel by email to acquire the dataset. Besides, we might encounter problems with the MMF framework as none of us ever worked on it before. We went through a learning process to understand MMF configs and data handling. For the binary hateful meme task, we need to build new models from scratch using PyTorch and adapt it to MMF framework. For the fine-grained modeling, the vocabulary files are missing. We debugged the framework to enable the training. When training and tuning hyper-parameters for our models, we are restricted by the computation power available on Google Colab. For instance, when the batch size exceeds 64, kernel shut down automatically. As a result, we only chose batch sizes of 16 and 32 for tuning.

4.2. Future work

Future research may: use data augmentation to generate more memes automatically. Apply upsampling or introduce additional data to improve performance for sparse labels. Explore impact from the order of text and image vectors. Explore a better classifier for cross-bert model due to high volumes of information generated by cross-fusion technique. Apply other text and image encoders on multi-modal systems.

5. Conclusion

In this work, we explore modeling in the hateful meme challenge on both hateful or not and fine grained classification. We explore various models and investigate the fusion mechanism in multi-modal architecture. Here are conclusions from this study:

- Multi-modal model VisualBert effectively models the hatefulness and fine-grained labels.
- Transfer learning from the models trained on other dataset (e.g. COCO) helps to improve model performance, especially by transferring parameters from similar multi-model architecture.
- "Fusion" is a critical step in merging output from different modals. We found "align" and "cross" are superior to "concat", which is probably because they allow direct interaction between image and language embeddings. "align" performs better than "cross" in our observation, which could be due to that the extra outer-product information is not very useful, or that the model is too simple to make use of the information.
- VisualBert gives reasonable modeling performance on the fine grained tasks (protected groups and attack types). However, there is the caveat that the sparse labels are not generalized well. Improvement could be to upsample the sparse labels or add additional data.
- The multi-task architecture models both protected group and attack types without performance regression. However, in our observation there is no performance lift, which suggests that sharing the knowledge between two tasks does not seem useful in this design.

Student Name	Contributed Aspects	Details
Fangtingyu Hu	Mid-fusion experiment design and implementation and analysis	1.Design models and experiments on the impact of different mid-fusion types on task performance (See section 2.1 of Approach). 2.Conduct mid-fusion experiments: construct Concat-Bert, Align-Bert, and Cross-Bert models from scratch in MMF, run training and validation, and analyze results(See section 3.2 of Experiments and Results). 3.Write and run relevant code on the mid-fusion experiments (See code file "mid-fusion experiments.ipynb"). 4. Literature review.
Sheng Luo	Fine grained modeling, multi-task modeling	Debug MMF package and develop model configs for fine-grained tasks. Develop models for the multi-task model. Write scorer code for calculating metrics for fine grained modeling.
Yafen Zhang	Hyperparameter tuning, model comparison	Hyperparameters tuning over learning rate, batch size, and warmup. Compare different models in the early fusion type, including MMBT vs. Visual BERT and unimodal pretraining vs. multimodal pretraining.
Zhiyang Xia	Environment setup, Fusion type comparison, document finalization	Setup up the MMF environment and preprocess the hateful memes dataset till successfully running of binary hateful memes training and validation. Training models with different fusion position and compare the performance. Finalize the project report into latex template.

Table 11. Contributions of team members.

6. Work Division

Summary of contributions are provided in Table 11.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 3, 4

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[3] Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, and Davide Testuggine. Supervised multimodal bitransformers for classifying images and text. *CoRR*, abs/1909.02950, 2019. 1, 8

[4] LH Li, M Yatskar, D Yin, CJ Hsieh, and KW Chang. A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 1, 4

[5] Lambert Mathias, Shaoliang Nie, Aida Mostafazadeh Davani, Douwe Kiela, Vinodkumar Prabhakaran, Bertie Vidgen, and Zeerak Waseem. Findings of the WOAH 5 shared task on fine grained hateful memes detection. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH*

2021), pages 201–206, Online, Aug. 2021. Association for Computational Linguistics. 2

[6] Gerard Pons and David Masip. Multi-task, multi-label and multi-domain learning with residual convolutional networks for emotion recognition. *CoRR*, abs/1802.06664, 2018. 2

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1

[8] 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. https://github.com/facebookresearch/mmf, 2020. 2

[9] Amanpreet Singh, Vedanuj Goswami, and Devi Parikh. Are we pretraining it right? digging deeper into visio-linguistic pretraining. *arXiv preprint arXiv:2004.08744*, 2020. 4

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[11] 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

[12] Bertie Vidgen, Alex Harris, Josh Cowls, Ella Guest, and Helen Margetts. An agenda for research into online hate, 2020. 1

[13] Ron Zhu. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. *arXiv preprint arXiv:2012.08290*, 2020. 2

[14] Haris Bin Zia, Ignacio Castro, and Gareth Tyson. Racist or sexist meme? classifying memes beyond hateful. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 215–219, 2021. 3

7. Code Repository

We used MMF framework as the initial start for model building, training, and evaluation. We made the following changes to the code base:

- Built and implemented three mid-fusion multimodal models from scrach, including (our own version of) Concat-Bert, Align-Bert, Cross-Bert.
- Built multi-modal and multi-task models for MMBT, Concat Bert, VisualBert architectures, debugged the data input, added multi-label vocabulary file and developed scorers for computing the prediction performance.

  The link to Facebook MMF repository is at: https://github.com/facebookresearch/mmf.

  The Github Repository to our project is at: [1]https://github.com/EllaHxyz/hm_model

A. Supporting Figures and Tables

Additional Tables and Figures can be found in Appendix.

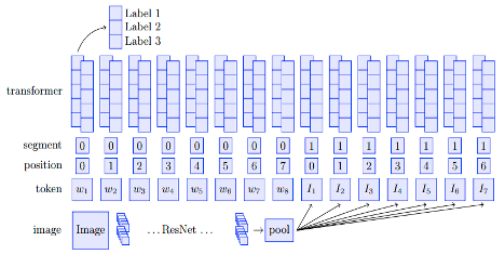

Figure 4. The structure of MMBT model [3].

Early Fusion		Mid Fusion		Late Fusion	
Model	AUROC	Model	AUROC	Model	AUROC
MMBT-GRID	66.73	Concat BERT	65.88	Late Fusion	65.05
MMBT-Region	72.62	ViT-Large-Patch14	79.02	Image-Grid	58.79
ViLBERT	73.02	CLIP	82.62	Image-Region	57.98
VisualBERT	74.14	MOMENTA	79.51	Text-BERT	64.65
UNITER	78.04	CLIP-ViT-L/14	77.3		
LXMERT	72.33	SEER-RG-10B	73.4		
Oscar	72.00	FLAVA	77.45		

Table 12. Summary of AUROC performance of different models, by fusion type. The table organizes various models used in hateful meme classification tasks with their AUROC performance on "dev seen". We observe a comparable performance from early and mid-fusion models, but an inferior performance from late fusion or unimodal models. This is relevant to the characteristics of hateful meme tasks – that emphasize a comprehensive view of text and image information, rather than treating them separately.