Hateful Memes Classification: Deliverable-I Report

Akshay Goindani (20171108) Preet Thakkar (20171068) Vaibhav Garg (20171005)
Sagar Joshi (2020701007)
Team 6

March 15, 2021

1 Problem Recap

Problem Statement: The goal is to predict whether a meme is hateful or non-hateful. This is essentially a binary classification problem with multimodal input data consisting of the the meme image itself (the image mode) and a string representing the text in the meme image (the text mode).

Input and Output: Given a rgb meme image, and a string representing the english text in the meme image, the trained model will output the probability that the meme is hateful.

Motivation:

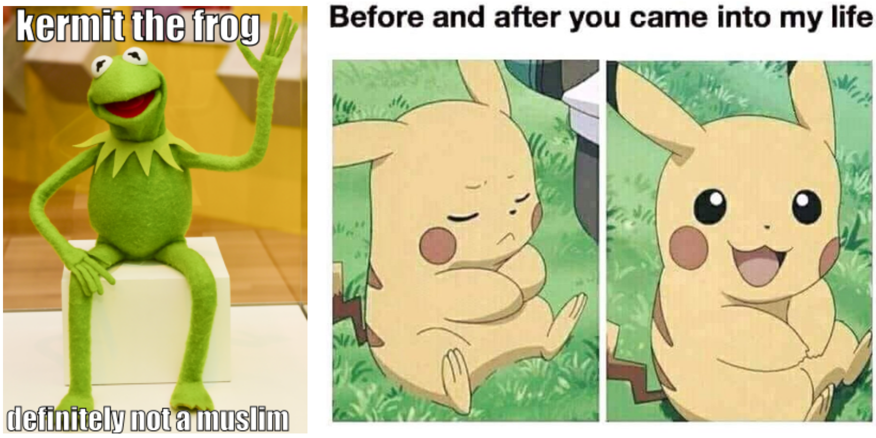


Figure 1: Both the images are fairly popular memes but clearly, one induces hate and negativity whereas the other one promotes love and positivity.

Internet memes have become a major form of communication and expression, they are an essential part of social media's popular culture. Modern-day social media platforms are full of memes as they are very easy to consume and also have the associated humor aspect. Figure 1 contains examples of such commonly shared memes. Moreover, these platforms have made them incredibly easy to share, two clicks are literally all it takes to make sure all your friends/followers have the same meme on their feeds. But like with any other type of content in social media, they also can lead to the spread of hate speech and propaganda. At the end of the day, these memes become an indirect source for our information consumption and play a significant role in shaping our belief systems.

The scale at which such content is produced makes it impossible for humans to go through all the memes and filter out the distasteful ones and hence there is an urgent need for fast and scalable automated solutions. Facebook AI's Hateful Memes Challenge [3] was one of the very first large-scale attempts to explore this problem and the results prove just how challenging this problem is, interpreting a meme requires a lot of cultural, political, and contextual knowledge and multimodal understanding since it is both the text and the image combined that brings out the real meaning of the meme. Another important thing to note here that even the human annotators themselves had a hard time classifying the memes and an average time of 27 minutes was spent per meme in the annotation process. And lastly, even though the results from previous research might look promising, but at the scale of the internet that still equates to a lot of hateful memes still being shared openly.

2 Implementation of different Baselines

We will primarily use the Facebook AI's Hateful Meme Dataset [3] for training and testing our proposed models. The dataset is created in such a way that, uni-modal approaches that use only the image or text features will fail. Therefore the proposed approaches are multi-modal i.e., they use both image and text features. As mentioned in the project outline document, we have first checked the output of the baselines and then followed and implemented the approach from mainly two papers.

2.1 Baselines in the Hateful Memes Challenge Paper

2.1.1 Intuition

In the paper that introduced the Hateful Memes Challenge [3], there were 11 baseline models provided with their AuROC and accuracy scores. As a golden benchmark for hateful memes classification task, the paper provided human accuracy score of 84.70% and AuROC of 0.8265. The annotators had a Cohen's Kappa score of 67.2%, indicating moderate inter-annotator agreement. By providing the 11 baselines using SOTA architectures, the paper provided the intuition behind the difficulty of the task. Firstly, there was a visible gap between the performance of unimodal approaches as compared to multimodal approaches indicating the necessity to make use of both the modalities and secondly, there was a huge gap between the performance of the best performing multimodal model and the human score for the task. This gave the motivation to first try checking the performance on provided baselines and then try approaches to fill in the existing gap with human performance.

2.1.2 Architecture

Among the 11 models used as baselines, three are unimodal approaches (Image-Grid, Image-Region, Text BERT) making use of either image or text modality to make the classification. There are five multimodal architectures (Late Fusion, Concat BERT, MMBT-Grid, MMBT-Region, ViLBERT, Visual BERT) but pretrained on unimodal objective and two multimodal architectures (ViLBERT CC, Visual BERT COCO) pretrained on multimodal objective. For image features in input, ResNET-152 and Faster-RCNN with ResNeXt-152 based features are used and for textual modality, BERT embeddings are used. A quick idea of each of the baseline architectures is given below.

- Image-Grid
  Unimodal image-based classifier which uses convolutional features with average pooling from ResNet-152 architecture.

- Image-Region
  Unimodal image-based classifier which uses features from Faster-RCNN with ResNeXt-152 as the backbone network, and is pretrained on the Visual Genome dataset.

- Text BERT
  Unimodal text-based approach which uses BERT embeddings on the text given as a part of the hateful memes dataset.

- Late Fusion
  A simple multimodal approach where output of ResNet-152 as in Image-Grid and BERT based models is taken unimodally, and their mean is taken as model output.

- Concat BERT
  In this multimodal approach, an earlier fusion of the output of the unimodal ResNet-152 and BERT embeddings is performed by concatenation and an MLP is trained for classification.

- MMBT-Grid
  MMBT is a multimodal supervised bitransformer architecture that consists of individual unimodally pretrained components that is trained to map multimodal image embeddings to text token space. MMBT-Grid uses features from ResNet-152 for image embeddings.

- MMBT-Region
  In this approach, the MMBT transformer uses features from Faster-RCNN as in Image-Region for image embeddings.

- ViLBERT
  ViLBERT is a dual stream multimodal transformer architecture. Here, the VilBERT model without any multimodal pretraining is used. It has BERT initializations for the text stream and uses Faster-RCNN pretrained on Visual Genome dataset to extract image region features.

- Visual BERT
  Visual BERT is a multimodal single stream transformer architecture in which the text and image inputs are jointly processed by a stack of BERT-based transformer layers. It uses Faster RCNN for extracting image features. For this baseline, Visual BERT without any pretraining on multimodal tasks is used.

- ViLBERT CC
  ViLBERT architecture used here is pretrained multimodally on the Conceptual Captions (CC) dataset using two pretraining tasks - masked multi-modal modelling (masking 15% of text and image region inputs and reconstructing them with unmasked inputs) and multi-modal alignment prediction (given a pair of image and text, determine if the text describes the image).

- Visual BERT COCO
  For this baseline, Visual BERT architecture is pretrained multimodally on the Common Objects in Context (COCO) dataset. The two tasks the model is pretrained on are masked language modelling with image (some part of text is masked, and is to be predicted using image regions and unmasked text) and sentence-image prediction (given two captions for an image, while one of them is the right caption for the image, determine if the same holds for the remaining caption as well).

2.1.3 Training

Pretrained models and the codebase and environment needed to replicate the results were provided in the mmf framework by Facebook AI Research. We first tested the performance of the models directly on the pretrained weights, after which we trained all the 11 models to see the improvement in performance. The hyperparameters used were the same as those provided in the challenge paper.

Model	**Batch size**	**No. of epochs**	**Best update**
Image-Grid	32	83	1000
Image-Region	32	83	1000
Text BERT	128	329	1000
Late Fusion	64	166	1000
Concat BERT	64	166	22000
MMBT-Grid	32	83	11000
MMBT-Region	32	83	7000
ViLBERT	32	83	14000
Visual BERT	128	329	1000
ViLBERT CC	32	83	15000
Visual BERT COCO	64	166	2000

Table 1: Training statistics for the 11 baseline models

The models were trained or finetuned for a maximum of 22000 updates, independent of the batch size. The architectural capacity on Ada cluster allowed us to train the models with the batch sizes used in the paper. On an average, each model took approximately 3.5 hours to train for the fixed no of 22000 updates. For every 1000th update, the accuracy, F1-score and AuROC was noted on the validation data. This facilitated in determining the best update after the training process and guided to make observations regarding overfitting. Table 1 shows training statistics for all the 11 models.

2.1.4 Results

Table 2 shows the accuracy and AuROC scores on the validation data, while table 3 shows the scores after finetuning. Some of our observations from the process which are consistent with those in the challenge paper are stated below.

- The increase in accuracy and AuROC scores was too little after finetuning, and for worse was negative for some of the baseline models, implying there has to be a better way for training with multimodal objectives.

- Among multimodal architectures, the difference is not much between the performance of unimodally pretrained and multimodally pretrained models, implying the pretraining of multimodal models can be improved.

- The performance improves with more advanced fusion of the two modalities. For instance, after finetuning, the performance of ConcatBERT beats that of Late Fusion. It is in turn superseded by multimodal architectures like MMBT, ViLBERT, VisualBERT.

Type	Model	Accuracy	AuROC
Unimodal	Image-Grid	61.8%	0.57
	Image-Region	57.6%	0.47
	Text BERT	61.7%	0.61
Multimodal (Unimodal Pretraining)	Late Fusion	67.4%	0.66
	Concat BERT	63.0%	0.61
	MMBT-Grid	66.5%	0.68
	MMBT-Region	66.3%	0.63
	ViLBERT	67.8%	0.65
	Visual BERT	66.8%	0.69
Multimodal (Multimodal Pretraining)	ViLBERT CC	68.0%	0.66
	Visual BERT COCO	67.0%	0.69

Table 2: Results on pretrained weights of baseline models on *val* dataset

Type	Model	Accuracy	AuROC
Unimodal	Image-Grid	59.8%	0.54
	Image-Region	58.0%	0.55
	Text BERT	60.0%	0.58
Multimodal (Unimodal Pretraining)	Late Fusion	63.3%	0.62
	Concat BERT	63.9%	0.63
	MMBT-Grid	64.3%	0.64
	MMBT-Region	66.5%	0.70
	ViLBERT	69.6%	0.69
	Visual BERT	69.2%	0.71
Multimodal (Multimodal Pretraining)	ViLBERT CC	68.9%	0.70
	Visual BERT COCO	69.3%	0.72

Table 3: Results on finetuned weights of baseline models on *val* dataset

- Human AuROC of 0.8265 is still very far from the best baseline AuROC of 0.72. Hence, filling in this gap becomes a guiding force in trying out with further techniques in the project.
- Early occurrences of best update in many models indicates overfitting problem. Given the relatively small size of data as compared to the model architectures used, lesser no of epochs should be sufficient to produce good results.

2.2 Detecting hate speech in memes using multimodal deep learning approaches:Prize-winning solution to hateful memes challenge [12]

2.2.1 Intuition

In a multi-modal approach, both, image and text features are used to make the final prediction. A straightforward approach can be to extract image and text features independently and then combining these features using some function which can be addition, concatenation, a Multi Layer Perceptron, etc. But understanding image and text together is much more complex than this, the image features must be aware of the contextual information from the text and similarly text features must be aware of the contextual information from the image. Therefore, the intuition is to combine the image and text features in such a way that both the modalities are aware of the context from the other one. The self-attention mechanism from VisualBert [5], allows the image features to attend to the text features and vice-versa, hence, we follow the approach mentioned in the paper [12].

2.2.2 Architecture

The VisualBERT[5] architecture is very similar to that of BERT[11], that is, it essentially comprises of a large stack of transformer based encoder layers focusing on self-attention. The only difference is the addition of image-region embeddings along with the word embeddings, which are similarly calculated using addition of a segment embedding, a position embedding and a token embedding as shown in Figure 2. 100 boxes of 2048D region-based image features are extracted from the fc6 layer of a ResNeXT-152 based Mask-RCNN[2], trained on the Visual Genome dataset[4].

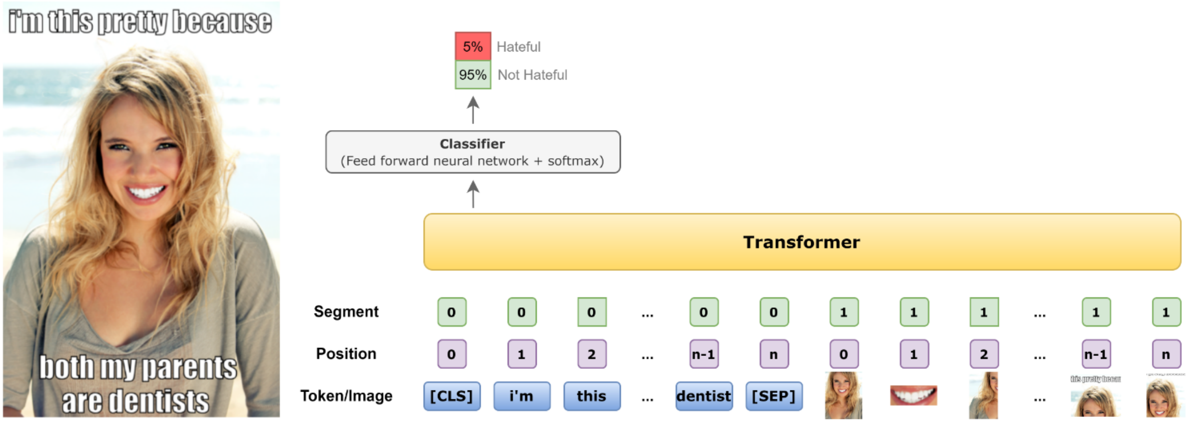


Figure 2: Multimodal transformer architecture, relationships between image regions and captions is learnt implicitly through self-attention.

2.2.3 Training

For stable learning and better scores, we increased the training data by 328 additional memes. We used the memotion dataset [9] for increasing our training data, we picked those memes that were most similar to the memes already present in the Facebook AI's hateful memes training data.

The extracted image regions and text (Sec 2.2.2) are the input to the pretrained visualBert model. Along with the image regions and text tokens, a special token - ["CLS"] is prepended to the input. For the image regions we used a ResNeXT-152 based Mask-RCNN [1] for extracting features. Using the self attention mechanism, the features of image regions, text and the special token are enhanced. From the final layer of the architecture, features of the ["CLS"] token (X) are passed through a classifier which outputs the probability ($\hat{Y}$) of the input meme being hateful and non-hateful. In our case, the classifier is just a fully connected Layer followed by softmax activation:

$$\hat{Y} = Softmax(WX + b) \tag{1}$$

Here, $X \in R^d$, $W \in R^{c \times d}$, $b \in R^c$ and c is the number of classes (2 in our case). W and b are learnable parameters.

Using the ground truth labels (Y) and the predicted probabilities ($\hat{Y}$), we calculate the cross entropy loss. The objective is to minimize the mean cross entropy loss computed over all the training examples in order to find the optimal parameters for the classification layer and fine tune the parameters of the pre-trained VisualBert model. We perform the training for 5 epochs.

The VisualBert model has a lot of hyperparameters, in order to enhance the performance of our model, we tuned few of the hyperaprameters such as batch size, learning rate, warmup steps, warmup type and warmup iterations. The total number of unique combinations for different values of these hyperparameters is 96. Since, we train a single model for only 5 epochs which takes only 40 minutes, therefore we performed a grid search over these 96 combinations of hyperparameters.

Out of these 96 models, we pick top 27 models based on the validation loss. Using these 27 models, we create an ensemble, where the final class label is computed using the outputs of individual models. Majority voting strategy is used to predict the final label.

2.2.4 Results

Dataset	Accuracy	AuROC
Dev Seen	71.2%	0.81
Dev Unseen	71.3%	0.75
Test Seen	66.2%	0.74

Table 4: Accuracy and AuROC score computed on dev seen, dev unseen and test seen using the ensemble of top 27 models using majority voting strategy.

The results in Table 4 are numerically similar to what the paper [12] claims to achieve.

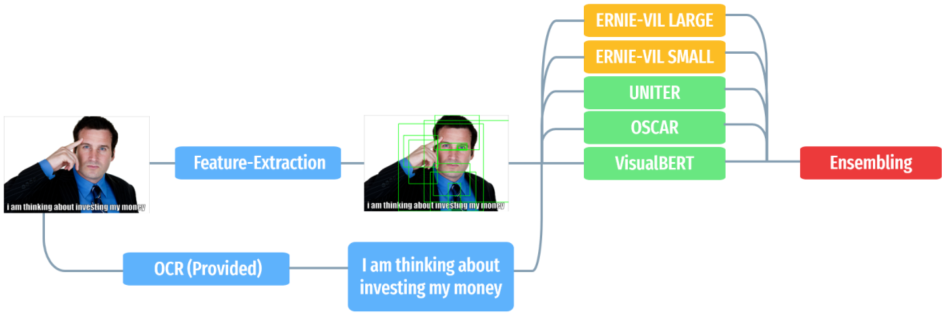
[1] https://dl.fbaipublicfiles.com/pythia/detectron_model/FAST_RCNN_MLP_DIM2048_FPN_DIM512.pkl

2.3 Vilio: State-of-the-art visio-linguistic models applied to hateful memes [8]

2.3.1 Intuition

Intuition for the approach used in Vilio paper is more or less similar to that of the first paper. With the presence of benign confounders, it is evident that any approach will be using Vision+Language machine learning models (which are based on transformer architecture). Vilio proposes an ensemble containing the following models: VisualBERT, UNITER[1], OSCAR[6], ERNIE-ViL[13].

2.3.2 Architecture



State-of-the-art Vision+Language models are of two types: Single-stream and Dual-stream. The former uses a single transformer to process the image and language input at the same time; some examples are VisualBERT, UNITER, OCSAR. The latter relies on separate transformers for vision and language, which are then combined towards the end of the model. LXMERT[10], ERNIE-ViL, DeVLBERT[14], VilBERT[7] are examples of Dual-stream models. Vilio uses a combination of Single-stream and Dual-stream architecture.

2.3.3 Training

The models used features extracted from images using the detectron2 framework by Facebook. For extracting features, the no. of minboxes and max boxes was kept to the same number, either 36, 50, 72 or 100. Text extracted by OCR techniques from the original meme was used as is from the hateful memes dataset. The models were trained on the pretrained weights provided by the authors of the multimodal architectures. For each model, different strategies were used for pre-training phase, the details of which are provided in the paper. The models were trained on the pretrained weights provided by the authors of the multimodal architectures. For every model, finetuning was done using binary cross-entropy loss. For OSCAR, UNITER and VisualBERT, 3 different seeds with different extracted features and for ERNIE-ViL, 5 different seeds with different extracted features were used. The Adam optimizer is used with a learning rate of 1e-5 and 10% linear warm-up steps. Gradients are clipped at 5 for VisualBERT, OSCAR & UNITER and at 1 for ERNIE-ViL. VisualBERT, OSCAR & UNITER are trained for 5 epochs and Stochastic Weight Averaging is used during the last 25% of training. ERNIE-ViL models are trained for 5000 steps. The weights from the last step are taken for all models and used for inference on the test set. The results were processed in an ensembling loop applying simple averaging, rank averaging, power averaging and simplex optimization to produce the final predictions.

2.3.4 Results

Model	AUROC
VisualBERT	0.75
OSCAR	0.76
UNITER	0.77
ERNIE-ViL Base	0.79
ERNIE-ViL Large	0.81

Table 5: Individual Model Performance on Dev Seen

The results in Tables 5, 6 are numerically similar to what the paper claims to achieve.

Dataset	AUROC	Accuracy
Dev Seen	0.81	73.2%
Dev Unseen	0.80	72.0%
Test Seen	0.83	73.6%

Table 6: Ensemble Performance

3 Comparison

We reproduced the results for both the models (visualBert [12], vilio [8]) that we mentioned in our project scope document. Table 6 shows the performance of the vilio model and Table 4 shows results for the visualBert model. It is clear from the Test seen scores that the vilio model outperforms the visualBert model by a large margin. Also, we submitted the predictions of both the models on the datadriven platform [2], and we found that the visualBert model is ranked 81 and vilio model is ranked 9 on the leaderboard for the phase 1 of the competition.
Clearly, Vilio model performs better than the visualBert model, therefore we have decided to move forward with the vilio model and perform all the next set of experiments using it.

4 Challenges Faced

- **Computing resources:** Training the above mentioned models required working with really large Pre-Trained models and large input features making it impossible to train on our local machines. Moreover, due to the resource allocation and data sharing constraints on the institute's GPU cluster, we had to spend a decent amount on copying the data from one node to another. For instance, while getting the results for the Vilio paper, the directory size had grown to more than 100 GB, that too when we smartly shared the feature files between the ensemble models.

- **Problems with available implementation:** For Vilio paper, we faced a few problems in dealing with some of the ensemble models. For VisualBERT, the code was crashing. An issue for the same was created on Vilio's github repo (link). The problem was that feature files provided were not updated according to the Phase 2 data of challenge. This issue was resolved later on.

- **Problems faced in training OSCAR:** There was an error in one of the feature files generated, which had one of the data points not in the right shape. Due to this issue, the training routine kept crashing before we discovered the issue by debugging. This was then resolved by using the corresponding feature file directly provided[3] by the authors of the paper.

- **CUDA Runtime Memory Errors:** In training OSCAR, the training routine was getting crashed after successfully running for 4 epochs because of CUDA Runtime Error for insufficient memory. On reducin g the batch size from 8 to 4, the issue was resolved, however, the training had to be continued from scratch. We suspect this could be because of Stochastic Weighted Averaging being used in the last 25% of training. ERNIE-ViL models were available in PaddlePaddle framework, which kept throwing the memory error for initial batch size of 8 as well as when experimented with batch sizes of 4, 2, 1. Since we didn't have sufficient exposure to the PaddlePaddle framework and it lacks good community support in English, we were not able to train the ERNIE-ViL models and used the trained weights directly provided by the authors to go in the final ensemble.

5 Progress So Far and Future Work

As we had mentioned in the scope document of the Project Idea phase, we have successfully managed to reproduce the SOTA results for our first deliverable and are well on track to complete the project. The code for our experiments is available at github [4].
For the next and the final deliverable, our focus will be on enhancing the above results by exploring more on the ensembling side. After that we have two directions to explore -

[2] https://www.drivendata.org/competitions/64/hateful-memes/page/205/
[3] Pre-extracted features provided on Kaggle
[4] https://github.com/VAIBHAV-2303/Hateful-Memes-Classification

- Improve the AUROC score on the Facebook hateful meme dataset [3] by enriching both the image and text features. We will also try to propose a novel approach for combining the image and text features in order to enhance the performance of the multi-modal architecture.
- Another direction is to create a hateful meme classification model for different languages for e.g., Hindi. Since, enough training data for Hindi is not available to train a model from scratch therefore, our idea is to use a translation module, which will first translate the text in the meme to English and then we will use our implementation of the hateful meme classification model for english, to make the final classification. In this approach we will first extract memes which have text in the foreign language (not English) and then we will do inference on those memes. We will not use the extracted memes for training.

References

[1] 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, 2020.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[3] 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.

[4] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[5] 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.

[6] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks, 2020.

[7] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.

[8] Niklas Muennighoff. Vilio: State-of-the-art visio-linguistic models applied to hateful memes. *arXiv preprint arXiv:2012.07788*, 2020.

[9] Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Bjorn Gamback. Semeval-2020 task 8: Memotion analysis–the visuo-lingual metaphor! *arXiv preprint arXiv:2008.03781*, 2020.

[10] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers, 2019.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, page arXiv:1706.03762, June 2017.

[12] 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.

[13] 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, 2020.

[14] Shengyu Zhang, Tan Jiang, Tan Wang, Kun Kuang, Zhou Zhao, Jianke Zhu, Jin Yu, Hongxia Yang, and Fei Wu. Devlbert. *Proceedings of the 28th ACM International Conference on Multimedia*, Oct 2020.