Image Captioning and Hateful Meme Detection

Heqing Ran[1], **Zhen Tao**[2], **Xiaokun Yu**[3]

[1,2,3] Department of Computer Science, Master's Candidates, Columbia University

Abstract

Image captioning combines the most advanced deep learning models in computer vision and natural language processing. In this project, we apply image captioning to the *Hateful Memes Challenge* dataset to identify hateful content. We first developed a combined model of CNN and LSTM to generate captions. We used pre-trained Inception V3 model for image encoder to extract image features. Then we built a bidirectional LSTM model to generate captions. We trained the model over *Flickr 8k* dataset and saved the best model for transfer learning. For *Hateful Memes* dataset, we use the pretrained model to generate image caption. We built a BERT model as our sentiment classifier. In order to evaluate the performance of the joint model, we also implemented several conventional deep neural network to directly classify the image data as our base models.

Keywords: *Image Captioning; Long-Short Term Memory; Convolutional Neural Network; Transfer Learning*

1. Introduction

Detecting harmful content has always been a challenge in the tech industry as well as our society. *Hateful Memes* dataset is a multimodal image dataset proposed by Meta AI such that a unimodal classifier would struggle to classify them correctly. It contains over 10,000 images both from real examples shared online and artificial ones generated by AI. It is difficult for machine learning models to classify due to different modalities. The dataset contains 8500 training samples, 500 validation samples and 1000 test samples. It also contains the label and text script for each image.

Our aim is to train different classifier models to do classification tasks on predicting the label of the meme images in the hateful memes dataset. (1: hateful, 0: not-hateful). We would try different deep-learning computer vision models to predict the labels based on the images. In addition, we would also go further and attempt to generate captions for these meme images using LSTM model. And we will then use a pre-trained BERT classifier to predict the labels for the memes based on the image caption and text description of the images (without the images themselves).

We want to try different approaches to the classification of the hateful memes task because this task could be very challenging due to the nature of the memes. Sometimes memes could be hateful in a subtle, not obvious or sarcastic way, making it hard to be detected by the models trained with images or captions themselves. Therefore, we want to find out which model we tried would perform better on the prediction.

2. Related work

Recent methods revolves around building a combined model that processes texts and images separately and combines the results for model prediction. One of the participants of *Hateful Memes Challenge* proposed a model that uses bidirectional LSTM based text processing archetecture and a ResNet based image processing architecture.[1]. However, even though the model's performance was 8.24% better than the baseline model, it was still below 50%. Another popular approach is to use multimodal transformer model which achieved a significant improvement in accuracy.

It combines several early fusion multimodal transformers with an ensemble method to achieve a better result. is shown that early fusion transformer model such as UNITER performs very well for this multimodal dataset[2].

3. Methods

As mentioned in the introduction, we built a image caption model and ran it on Hateful Memes dataset. Then we use a BERT classifier to predict whether a meme contains hateful content. We also examined several deep neural network models for classifying images directly without reading the texts and used them as our baseline models.

3.1. Image Caption Model on Flickr-8k

The first part of the proposed model is to encode Flickr-8k images into feature vectors. The proposed method uses the second-to-last layers of InceptionV3 with pretrained weights. When working with text data, the first step to do is always tokenization which consists of splitting an entire text block into small units known as tokens. The captions in the data file are already tokenized, so we can split them by white spaces. We conducted data cleaning and convert each token to lowercase. Next, we created a lookup table from the training data mapping words to integer indices in order to encode input and output sequences using numeric representations. We create the dictionaries int-to-word and word-to-int which can map tokens to numeric ids and vice versa. Then we built a model to predict the next word given a partial sequence and image input.

We used bidirectional LSTM to encode the input sequence from both directions and predict the output. We concatenated text input and image input after adding an embedding layer for input texts and projecting encoded images to a hidden layer. To produce the training sample in a proper format, we implemented a training batch generator that returns an input/output pair ([image inputs, partial sequence inputs], next word outputs). Then we implemented an image decoder that returns a complete sentence given an encoded input image. The model was trained on the Flickr-8k dataset and the best model was saved for future use.

3.2. CNN-LSTM-BERT Sentiment Classifier

We aim to use the BERT classifier to predict the label of the hateful memes dataset. The task for this part consists of two steps: first, generate the caption for the meme images datasets using the LSTM trained on Flickr-8k. Second, use BERT classifier to predict label based on the image captions and the image text description.

For image captioning for hateful memes, we use the LSTM model we trained on Flickr-8k mentioned in section 3.1. The proposed method uses the bidirectional LSTM trained on Flickr 8k to generate captions for images on Hateful Memes dataset. For the data preparation, we implemented a class called HatefulMemes to get images, labels and texts. We used the same trick to save encoded images and loaded them when training. The training set of Hateful Memes is extremely large, to reduce the time for loading data, we saved the dataset as npy file and load the dataset as Pandas Dataframe.

We use the same image encoder and predict generator as we used for the Flickr-8k dataset to encode the image and use the same image decoder to generate the caption. Then we save the image caption results along with the original hateful memes dataframe as a new column. Since the model is trained entirely on Flickr-8k dataset, the result of image captioning is fair. The image caption generator is able to generate a full sentence with correct logic and a fair guess of the image content. For example, it could recognize "man", "women", "dog" with simple description such as "smile", "in a red shirt", "walking on the street", but it can't recognize things such as "frog", "hat" due to the limitation of the Flickr-8k dataset.

The next step is to use BERT classifier to predict the labels. The features we use are the image captions and texts for the hateful memes. We load the hateful meme dataset as Pandas Dataframe including the captions. The model concatenates image captions generated by the joint Inception-LSTM model and the text descriptions of the dataset together as one learning feature.

The proposed method uses a pretrained BERT model (Bidirectional Encoder Representations from Transformers) which takes the concatenated text information of the image (caption + description) and predicts a probability for the label. We then set a threshold which, for the predicted probability that is greater then the threshold, the predicted result would be classified as 1 (hateful), otherwise 0 (non-hateful). To find the best threshold, we iterate different thresholds between 0 to 1 and compared the predicted results with the actual labels. The accuracy of the prediction is 0.69.

3.3. Image Classification on Hateful Memes

The proposed method uses four deep neural network model as image classifiers to classify hateful memes. The idea is to use a singular classification model as a baseline model. We apply domain transfers for these models. The first model is VGG-16 based. We froze the layers with pre-trained weights and train the network on image data. Then we unfroze those layers and train the whole network again to fine-tune those parameters. We repeated the same procedure for ResNet50, ResNet101, and DenseNet121 and recorded accuracies and losses. We compiled the network using binary cross entropy on our logistic sigmoid and an SGD/momentum optimizer. To overcome overfitting and achieve better performance, we added batch normalization and dropout layers after each dense layer. We also used validation accuracy in the ModelCheckpoint callback as a metric to save the best model observed during training.

4. Result and Discussion (10pt, bold)

4.1. CNN-LSTM-BERT Performance

The multimodal method acchieved 0.6989 test accuracy and 0.752 test AUC which is significantly higher than the baseline model performance given by Meta AI.

Train accuracy	Validation accuracy	Test ACC	Test AUC
0.72	0.6537	0.6989	0.752

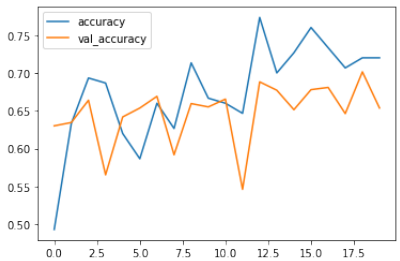

Figure 1. train, validation accuracy

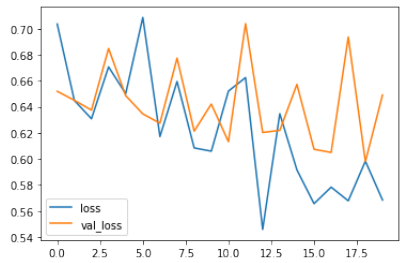

Figure 2. train, validation loss

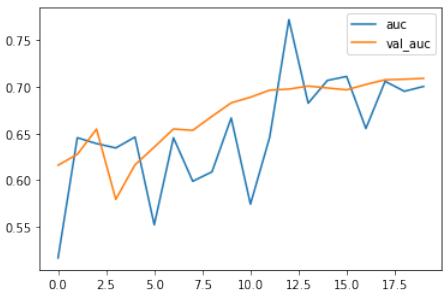

Figure 3. Train, validation AUC

4.2. Image Classification Performance

The image classification models performs surprisingly well compared to the base model given by Meta AI. During the training process we found that the accuracy for validation set was unexpectedly low. Then we found out that the subset of data used for validation set was known to contain mislabeled data. Therefore instead of using the given validation set, we simply divided the train sample into test and accuracy with ratio 8:1:1. After changing the validation set, the result improved significantly

Model	Test loss	Test accuracy
VGG-16	**0.5575**	**0.7188**
ResNet50	0.6073	0.6729
ResNet101	0.5901	0.6753
DenseNet121	0.6250	0.6553

From the table above one can see the best image classification model is VGG-16.

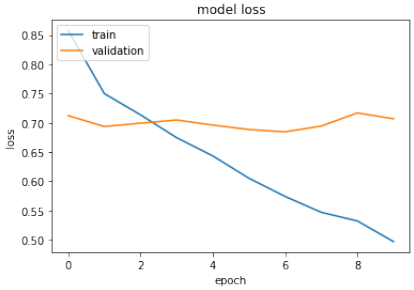

Figure 4. VGG16 loss

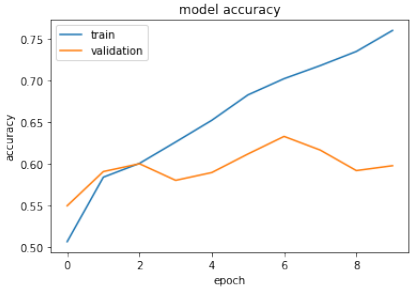

Figure 5. VGG16 accuracy

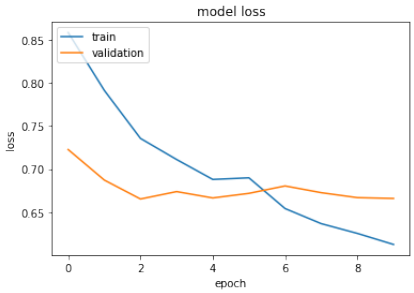

Figure 6. ResNet50 loss

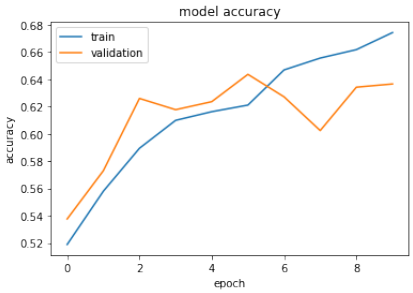

Figure 7. ResNet50 accuracy

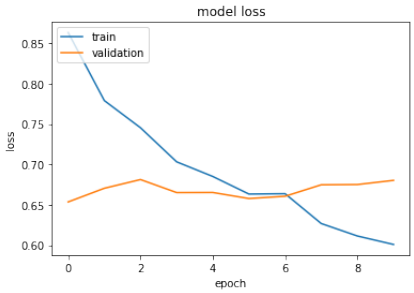

Figure 8. ResNet101 loss

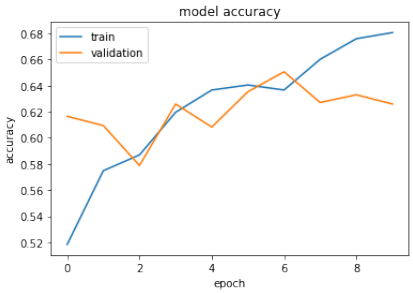

Figure 9. ResNet101 acc

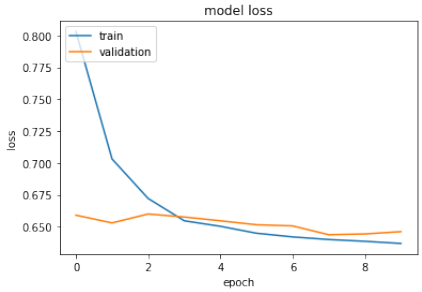

Figure 10. DenseNet121 loss

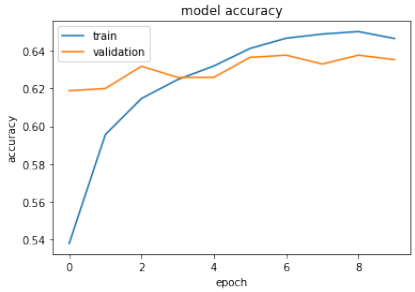

Figure 11. ResNet121 acc

5. Conclusion

The proposed multimodal method achieved a test accuracy of 0.6989 and a test AUC of 0.752, which means it performs well on identfying hateful memes. The result is consistent with other related works. However, there is still space for improvement. In the future, we could experiment on different encoders and decoders, and combine multiple methods by taking the ensemble as suggested in related works. It is also worth mentioning that since the Hateful Memes challenge provided text information for each memes in the dataset, we don't have to implement an optical character recognition (OCR) module. It would be interesting if we could implement one for our model and experiment with the dataset.

For this project, Zhen and Heqing worked on the preprocessing of Flickr dataset and Xiaokun worked on the pre-processing of Hateful Memes dataset. Zhen and Heqing worked on building the model and training on Flickr, Zhen worked on training the model on Hateful Memes dataset and Xiaokun helped with writing part of the model. Heqing and Xiaokun worked on writing image classification models. Each of us spent even effort coming up with this report. Our code can be found on GitHub `https://github.com/taoz0721/dl4cvproject`. Our presentation can be found at `https://drive.google.com/drive/folders/1x1zN0sDcO3zqJgxYCU6K775xJW7fAK2I?usp=sharing`.

References

[1] Constantin, M. G., Pavu, D. S., Stanciu, C., Ionascu, D., Ionescu, B., "Hateful meme detection with multimodal deep neural networks" *2021 International Symposium on Signals, Circuits and Systems*, pp. 1-4, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9497374.

[2] Lippe, P., Holla, N., Chandra, S., Rajamanickam, S., Antoniou, G., Shutova, E., and Yannakoudakis, H. "A multimodal framework for the detection of hateful memes," *arXiv preprint*, arXiv:2012.12871, 2020.