III. Model documentation and write-up

Information included in this section may be shared publicly with challenge results. You can respond to these questions in an e-mail or as an attached file. Please number your responses.

1. Who are you (mini-bio) and what do you do professionally?

Ekaterina: I am an Assistant Professor at the Institute for Logic, Language and Computation (ILLC) at the University of Amsterdam, where I lead the Amsterdam Natural Language Understanding Lab. Previously, I was a Leverhulme Early Career Fellow at the University of Cambridge and a Research Scientist at the University of California, Berkeley. I received my PhD in Computer Science from the University of Cambridge. My research is in the area of natural language processing, with a specific focus on machine learning for natural language understanding tasks. My current interests include few-shot learning and meta-learning, joint modelling of language and vision, multilingual NLP and societal applications of NLP, such as hate speech and misinformation detection.

Helen: I am an Assistant Professor at King's College London and a Visiting Researcher at the University of Cambridge. Previously, I was an Affiliated Lecturer and a Senior Research Associate at the Department of Computer Science and Technology of the University of Cambridge, a Fellow and Director of Studies in Computer Science at Murray Edwards College, and a Newton Trust Teaching Fellow at Girton College, Cambridge. I hold a PhD in Natural Language and Information Processing from the University of Cambridge. Current research interests include transfer and multi-task learning, few-shot learning, continual learning, dialogue systems, and explainable machine learning, as well as machine learning for real-world applications and non-canonical forms of language.

George: I am currently working as part of a machine learning research team for a medical diagnosis start-up. Previously, I was a short-term Visiting Researcher at King’s College London working on multimodal approaches for hate speech detection. I hold a PhD in Pure Mathematics from the University of Glasgow and Master’s degree in Applied Mathematics from the University of Cambridge.

Nithin: I am a recent graduate from the MSc AI program at the University of Amsterdam. I worked on meta-learning for few-shot and continual learning in NLP as part of the master thesis.

DRIVENDATA

Phillip: I'm a first-year PhD student at the University of Amsterdam working on temporal causality and deep learning. Before starting the PhD, I have completed a Master degree in Artificial Intelligence at the University of Amsterdam.

Santhosh: I am a Machine Learning Engineer (NLP) at Slimmer AI, and I am part of the team responsible for building and maintaining various AI products used by Springer Nature. I

graduated from University of Amsterdam with a Masters degree in Artificial Intelligence in 2019 and my thesis was on joint modelling of emotion and abusive language detection.

Shantanu: I am currently working as an AI Research Scientist at ZS. Before this role, I did my Masters in Artificial Intelligence from University of Amsterdam. My masters thesis was on graph-based modeling of online communities for misinformation detection in collaboration with Facebook AI and King's College London.

2. **What motivated you to compete in this challenge?**

We had never got a chance to work on a multi-modal problem and we saw this challenge as the perfect opportunity to get acquainted with the field and solve an important social problem in the process. Learning jointly from the two modalities poses its own unique set of challenges and this competition was a great learning experience. Also, the monetary reward for top 5 teams served as an incentive to be competitive and improve our solution to the best of our abilities.

3. High level summary of your approach: what did you do and why?

The late-fusion baselines for the task were shown to underperform whereas the early fusion systems seemed promising. Thus, we experimented with two early-fusion pretrained models, namely UNITER, OSCAR as well as LXMERT. We found that UNITER performed the best, which could perhaps be attributed to its diverse set of pretraining tasks. We observed that simple fine-tuning on the HatefuMemes dataset led to poor performance on text confounders in the training set whereas the model could overfit on the image confounders and non-confounders. Thus, we decided to upsample these text confounders to give additional weight to these examples. Additionally, we added more weight to the hateful examples to combat the class imbalance. Furthermore, we performed cross-validation style training for better generalizability of the model. The final prediction was obtained as a weighted ensemble where the ensemble weights were optimized using an evolutionary algorithm (EA) on the development set predictions. We also observed that the dev_seen split had a higher percentage of truly multimodal examples than the training set. Thus, we included part of the dev_seen set into the cross-validation training. Specifically, for each fold, we split the dev_seen set in two halves, and used one part for training, and the other for testing. Thereby, we included text confounder pairs with different labels together to increase the confounder amount for training. This means that examples that make a confounder pair are not split between the train and test set. Finally, the predictions on the test parts were used for the EA optimization.

4. Copy and paste the 3 most impactful parts of your code and explain what each does and how it helped your model.

1. The use of pretrained UNITER as the base model was one of the most impactful choices that helped us improve over the provided baselines. Here we show the model initialization using the pretrained model:

2. Using cross-validation style training along with a weighted ensemble where the weights are optimized using an evolutionary algorithm also helped increase the performance further. Here we show the call to the ensemble optimization algorithm:

```
if EA_IMPORTED:
    logger.info("Starting EA to find optimal weights...")
    ea_score, ea_config = EA_ensemble_finder(eval_func=eval_func,
                                             num_weights=len(dev_preds),
                                             individual_scores=dev_scores)
```

3. Including part of the dev_seen set in cross-validation was also an important choice that provided a further boost in performance. Here we show the call to the function that creates the cross-validation split. By passing use_dev_set = True, we include the dev set while generating the splits.

```
if not os.path.isdir(crossval_path) or len(glob(os.path.join(crossval_path, "*.jsonl"))) == 0:
    logger.info("Creating cross-validation splits for dev size %i" % dev_size)
    generate_crossval_splits(config['data_path'], dev_size=dev_size, use_dev_set=use_dev_set)
```

5. Please provide the machine specs and time you used to run your model.

- **CPU (model):** Intel Xeon Gold 5118 Processor (16.5M Cache, 2.30 GHz)
- **GPU (model or N/A):** Single Nvidia TitanRTX
- **Memory (GB):** 45 GB
- **OS:** Debian GNU/Linux 10 (buster)
- **Train duration:** 6 hours
- **Inference duration:** 2 mins

DRIVENDATA

6. **Please list any external data or pre-trained models you used to develop your winning submission.**

External Data: None

Pretrained model: UNITER(base)

- Paper: UNITER

- **Official GitHub:** UNITER code

7. What are some other things you tried that didn't necessarily make it into the final workflow (quick overview)?

- We tried UNITER-large as a next step to using UNITER-base in our initial experiments. Overfitting was more severe and led to lower performance on the test set.
- We tried the OSCAR-base and -large as well as LXMERT as our base model. Performance was lower than UNITER.
- We tried a warm-up phase for the pretrained UNITER model as a domain-adaptation step to the hateful memes dataset. We used the UNITER pretraining tasks of MLM, ITM and MRFR on this dataset. We followed it by supervised learning for hateful meme detection task using this new "warmed-up" pretrained model. This gave no improvements.
- We extracted fine-grained labels (eg, recognizing people as Omani, Haiti, etc) using YOLO9000 for the images in the hateful meme dataset. We then concatenated these labels to the meme text during training to give the model precise information of the subjects in the image. Lead to no improvements.
- Looking at the offensive memes, we realized that most of them targeted social groups such as women, disabled, Muslims, Jews, etc. In order to leverage that information, we tried to use the Social Bias Frames dataset to detect the social minority being referred to in the meme through the meme text. To that effect, we first tried using it in an MTL fashion (primary task - hateful meme detection, auxiliary task - social minority detection) which gave no improvements. Next we trained a purely text model (RoBERTa) for the social minority detection using the Social Bias Frames dataset and then appended these labels to the meme text during training UNITER. Again, no improvements in test scores.
- We tried a similar MTL approach with the GoEmotions dataset in an attempt to supplement the model with the emotions associated with each meme (primary task - hateful meme detection, auxiliary task - emotion detection). Lead to no improvement.
- We trained the model using margin ranking loss instead of binary cross-entropy loss. Training was performed on pairs of data points - pairs of text confounders and random pairs of non-confounders. Lead to no improvements.
- Experiments showed that the model performed really well on image confounders while poorly on text confounders indicating that it was utilizing mostly the textual information of the input and ignoring the image features. Thus, we tried to split the internal self-attention of the UNITER transformer into 4 independent chunks - text->text, text->img, img->text, img->img -- to enable different attention dropout probabilities on different subsets of the input. We hoped that a higher text2text dropout with lower img2text and text2img dropout will help, as we overfit on the text input. No improvements were noted under different values of dropouts.

8. Did you use any tools for data preparation or exploratory data analysis that aren't listed in your code submission?

We extracted our own Faster R-CNN features, as opposed to using the ones provided in the MMF library. We have included the code for the same in the submission.

9. **How did you evaluate performance of the model other than the provided metric, if at all?**
   We monitored precision, recall, F1 on the hateful class, AUCROC and accuracy during training. However, the best model was chosen based on just auroc and accuracy.

10. **Anything we should watch out for or be aware of in using your model (e.g. code quirks, memory requirements, numerical stability issues, etc.)?**
    Choose batch size that fits into GPU memory. For larger batch size use the *--gradient_accumulation* parameter (see code). Apart from that no quirks or instability issues. Given the hardware specifications and the configuration provided in the readme file, our approach is perfectly reproducible.

11. **Do you have any useful charts, graphs, or visualizations from the process?** We visualized the evaluation metrics and the loss during training using Tensorboard. Nothing else apart from that.

12. **If you were to continue working on this problem for the next year, what methods or techniques might you try in order to build on your work so far? Are there other fields or features you felt would have been very helpful to have?**
    - Looking at the errors our model makes on the dev_seen set, we find that it lacks in its image understanding capability. Some of the false negatives are memes that are hateful because of the following in the images - black people, differently-abled people, Hitler, Nazi symbolism, Anne Frank, domestic abuse indicators etc. We believe that the Faster R-CNN features do not encode finer details such as these. Thus, training an image feature extractor on a different dataset that has information on racial profiles, disabilities, abuse, well-known persons (e.g., Hitler) etc could help in increasing the performance.
    - Yet another possibility is to fine-tune (parts of) the Faster R-CNN backbone while training UNITER, in an end-to-end fashion. This could perhaps combat the domain shift between ImageNet and the current dataset.

13. **Provide a link to the GitHub repository for your model and source code.**
    https://github.com/Nithin-Holla/meme_challenge

14. **Provide a link to your published academic explanation of your solution.**
    https://arxiv.org/abs/2012.12871