NLP2 Project: *Multimodal NLP*

Teaching Assistant: *Shantanu Chandra (shanchandra93@yahoo.in)*

1 Introduction

Motivation Multimodal machine learning is a multi-disciplinary research field which addresses some of the original goals of artificial intelligence by integrating and modeling multiple communicative modalities, including linguistic, acoustic and visual messages. With the initial research on audio-visual speech recognition and more recently with language & vision projects such as image and video captioning and visual question answering, this research field brings some unique challenges for multimodal researchers given the heterogeneity of the data and the contingency often found between modalities.

Assignment In this assignment we will work with a new and challenging problem in the field of multimodal NLP namely, hateful memes detection. Online abuse is an important societal problem of our time and one that is highly correlated with the rise of social media platforms. *Memes* – that have recently emerged as popular engagement tools and which, in their usual form, are image macros shared through social media platforms mainly for amusement – are also being increasingly used to spread hate and/or instigate social unrest, and therefore seem to be a new form of expression of hate speech on online platforms. *Hateful memes* often target certain communities or individuals based on, for example religion, gender, race, or physical attributes; by portraying them in a derogatory manner and/or by reinforcing stereotypes.

However, the detection of multimodal hate speech is an intrinsically difficult and open problem within the joint visual and language (V+L) understanding domain as it requires a holistic understanding of content, where reasoning about image and text is simultaneous. As the example in Figure 1 shows, it is not sufficient to rely on the text or the image modality individually for correct interpretation of content; rather both modalities should be jointly processed to infer the correct meaning of the meme. Not surprisingly, existing and state-of-the-art multimodal systems perform rather poorly on the detection of hateful memes [3].

The task of the assignment is formulated as a binary classification problem, where a meme can belong to one of two classes – *hateful* or *not hateful*.

Figure 1: Example (non-hateful) multimodal meme from the HM dataset [3]. The image is a compilation of assets, including ©Getty Images.

Dataset: For this assignment we will be using the Hateful Memes (HM) dataset [3]. ***The data comes with a licensed agreement and can not be distributed publicly. So please do not host in on any public platforms. Also, please read the user agreement to know what is allowed and what isn't.*** To access the dataset, you will have to register in the Hateful Memes Challenge competition[1] which is a recently concluded NeurIPS-2020 shared task. The competition is now over but the leaderboard is still open until April. In the dataset you will get the train and validation sets, while just the data of the test set (sans the labels). To evaluate on the test set, you have to submit your predictions on the site. Remember you get only one submission per day so use it wisely! You can of course create your own test set from the available splits for internal/unofficial testing. Contact me if you have any doubts regarding this.

Models For the assignment we will be using the transformer architecture based model – UNITER [1]. It is an early-fusion transformer model which utilizes self-attention on the joint image and textual input. It encodes visual features from bounding boxes using an image encoder and encodes word tokens using a text encoder into a common embedding space. For now, if you are not aware of transformer [4] or BERT [2]-like architectures, you can just think of them as black-box multi-layer self-attention layers that you give you a desired output based on the input. You can read about detailed explanations of transformers[2] and BERT[3] in these blog posts.

Deliverables

1. Jupyter Notebook: The notebook should contain the entire pipeline for your experimental setup. Functions or classes are allowed to be defined in Python files externally, as long as the main functionality is listed in the notebook.
2. Short paper: The short paper should contain four pages (references excluded) in the provided format[4]. A suggested page distribution is as follows:
   (a) **Abstract:** summarise the research in a short piece of text that emphasises your contributions and findings (0.1 pages);
   (b) **Introduction:** introduce the reader to your research area, clearly and explicitly mention problem statement and research questions, and summarise your contributions (0.5 pages);
   (c) **Related Work:** summarise research papers relevant for your work. Be brief, since this is a short paper (0.4 pages);
   (d) **Methodology:** this section should detail the tasks or model architecture designed, other design decisions introduced to tackle specific shortcomings, etc (1 page);
   (e) **Experiments and Results:** detail the precise experimental setup used and the results of your evaluation measures so that your work is reproducible (1 page);
   (f) **Discussion:** do some error analysis, design experiments to mitigate or highlight the claims you make in your discussion. Do the findings support your hypothesis? Think of interesting future work directions based on the results and analysis of your work (1 page).
3. Poster presentation, due February 24, 2021, 11:59: Compress the paper's content into a single-page poster that could be presented at a (virtual) conference. Support the textual content through visual aids, such as tables and graphs that facilitate fast understanding of the paper's contributions and main results. You will present this poster via a Zoom session in a presentation of about 5 minutes; details about the poster session will be made available in due time.

[1] https://www.drivendata.org/competitions/64/hateful-memes/
[2] http://jalammar.github.io/illustrated-transformer/
[3] http://jalammar.github.io/illustrated-bert/
[4] https://acl2020.org/calls/papers/#paper-submission-and-templates

Suggested Research Directions To get you started with some interesting research directions to tackle the task, you can refer to the list below. Please feel free to contact me to discuss these ideas in more detail if needed. Of course, as always, your own research ideas are also welcomed but make sure to discuss it with me first and get it approved to avoid any obstacles in completing the assignment in time. Feel free to implement more than one of these ideas to push the model performance in the shared task.

1. Try upsampling the text confounders since they are the truly multimodal and typically harder instances to classify. Think about the exact setup of identifying and sampling them in different data splits. You can also think about how else can these typically "hard" examples be leveraged during training.
2. Think of tweaking the loss function - (a) explore loss functions in literature and use it here with the right motivation provided; or (b) introduce your own with the right motivation provided.
3. Look at the vanilla model's misclassifications and try to look for a pattern where the model usually fails. Try to think of ways to mitigate this, for instance by supplementing additional knowledge to the model via other data sources (MTL with other similar tasks, pre-training on a different task or dataset, etc).
4. Implement existing multimodal pre-training tasks and "warmup" the pretrained model with these tasks on the HM dataset before fine-tuning on the classification task. Think of any other effective pre-training that the model can benefit from.

2 Suggested Schedule

To stay on track, we recommend adhering to the following schedule. To get you started with the assignment, we provide you with a code base[5] as well. We recommend to start off with spending more time on reading all the relevant literature before moving to coding the assignment and then finally spend more time writing. The detailed suggested schedule can be found below.

2.1 Week 1

Reading:

- Start off by reading the blog posts on transformers and BERT if you want to familiarize yourself with the architecture and inner workings.
- Thoroughly go through Kiela et al. [3] to get yourself familiarized with the dataset, its properties and the baselines. This will help you understand the problem and its finer details better.
- You can also briefly read through the UNITER [1] architecture that can help you design your research goals better.

Coding:

- Familiarize yourself with the provided code base. Go through the data and training pipelines.
- Run a vanilla UNITER model on the dataset and make sure you can reproduce the results to test the pipeline.

Writing: It is always good to write down the research paper alongside the research process. This week, since most of the time will be spent in reading the relevant literature and familiarizing yourself with the field, it will be a good idea to already draft the "Introduction" and "Related Work" sections. You can also start thinking of your research direction simultaneously as you read about the problem more.

[5] `https://github.com/shaanchandra/NLP-2-Assignment-Multimodal-NLP`

2.2 Week 2

Reading: This week you should finalize your research hypothesis and discuss it with me. You can also read a bit more about your approach in the literature to align your findings and expectations.

Coding: Start coding your research hypothesis. Most of the boiler-plate code is designed such that it can be tweaked seamlessly to accommodate for any kind of additions or changes you might want to do.

Writing: Since you have finalized your research question and started coding it, this would be a good time to prepare a draft of the "Approach" section. This can change over the course of the next 2 weeks as you get your initial results and plan to change things if need be.

2.3 Week 3

Reading: By now you should already have your initial results. You can now start doing some error analysis and start looking into literature for the missing pieces (e.g., new datasets to supplement it with, successful MTL setups, etc).

Coding: This week you should spend time on improving upon your initial results by trying out other stuff. You should also focus on the error analysis of your proposed model and design experiments to support your claims if need be.

Writing: Since you already have the results of your baseline model and your proposed framework, you should start preparing the "Experiments and Results" and "Discussion" sections.

2.4 Week 4: Wrapping up and final touches

Wrap up your project by cleaning the code, adding necessary comments so that it is easy to follow for the reviewer and making sure it runs end-to-end without errors. Finish writing up the paper and discuss any final things before you submit your final copy for assessment.

Good luck and happy learning!

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. In *European Conference on Computer Vision*, pages 104–120. Springer, 2020.

[2] 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.

[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] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.