Dank Learning: Generating Memes using Deep Neural Networks

Abel L Peirson V, E Meltem Tolunay
CS230 Project — Stanford University
alpv95@stanford.edu, meltem.tolunay@stanford.edu

Motivation

- Memes are ubiquitous in today's day and age; their language and ideologies are in constant flux. Primarily they function as a medium for humor to be shared, utilising cultural (especially subcultural) themes. However, they can also be manipulated to further political ideals, magnify echo chambers and antagonise minorities. It can take a remarkably high level of understanding to produce a good meme (image, culture, language etc.). The contemporary relevance of memes and the high level of understanding required to generate them motivate this project.
- We consider only the image with caption class of meme, an example of which is shown above – leading to a pseudo - image captioning task

Data Preparation

- 400k image-caption-name triples scraped using python from memegenerator.net user created memes.
- 2600 unique images and names in the dataset.
- ~160 unique captions corresponding to each image-name pair.
- Small image distortions applied during training to each example to augment image data.

Approach

1. Form an evaluation set of 105 image - caption (- name) examples which have repeated formats (e.g Boromir 'one does not simply' above)
2. Test both pretrained encoder types: alexnet / inception v3 using perplexity score on evaluation set as metric. Results shown to the right.
3. Tune hyperparameters (e.g number of LSTM layers, optimizer, batch size, LSTM units, training time) using perplexity as the metric then finetune hyperparameters using qualitative evaluation.
4. Finetune pretrained CNN encoders to reach for final performance boost.
5. Use altered beam search with temperature controlled sampling to encourage diversity in generated memes.

Acknowledgments & References

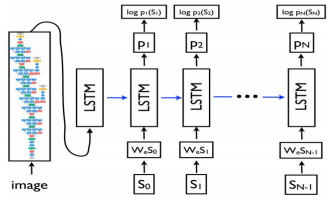
TA/Mentor: Surag Nair (Top right, he needs money apparently)

[1] Vinyals, Toshev, Bengio et al.
Show and Tell: A Neural Image Caption Generator 2017

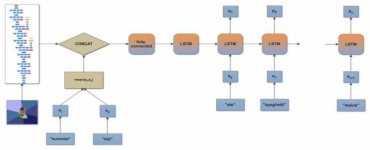
Models

Two model encoder-decoder model variations were explored. Both use pretrained GloVe word embeddings and an LSTM decoder:

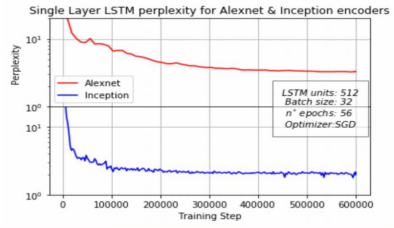
1. A baseline 'show and tell' [1] image captioning architecture which was adjusted to test both an Inception v3 and an alexnet encoder. This conditions a text generating LSTM on an image embedding



2. An 'image+name' model which conditions on both an image representation provided by a CNN (Inception) and a few words describing the meme (the name) which are mapped to pretrained GloVe vectorsand averaged.



Quantitative Results: Perplexity



Qualitative Results: Memes

Memes generated by both models on seen and unseen images. The sentence splitting is done manually. 'AI is the new electricity' are the name arguments for the unseen images.

Analysis

- Conclusions
  - There was little difference between the two models, both performing very well. As can be seen from the table below, generated memes are almost indistinguishable from real ones and similarly funny.
  - Inception outperformed the alexnet in both qualitative and quantitative metrics as an encoder
  - Varying the 'name' input in the second model during generation did not function well as a content control but did increase the meme variety.
- Future work
  - A small but significant fraction of the generated memes were copies or sub-copies of captions in the dataset. This could be ameliorated by improving the beam search mechanism further.
  - Sentence splitting of the generated caption is done manually, this is important for the humorous impact of the meme. Train using a dataset that contains sentence split tokens.

Model	% in data	Perplexity	Hilarity	Differentiability
Seen Images				
Image	17	2.01	7.5	57%
Image+Name	18	2.28	6.8	63%
Unseen Images				
Image	29	-	6.1	-
Image+Name	26		6.9	-

Real Memes scored on average 7.3 in the hilarity test, between 1-10. Differentiability shows how often our human testers were able to distinguish real memes from generated ones, if indistinguishable this score would be near 50%