Meme Caption Generator: Project Report

By Nursulu Sagimbayeva

In this project, I am exploring two approaches for meme caption generation. The code for the project and instructions for running are available on GitHub.

First approach: Sentiment-based Caption Generation

One feature that I noticed while brainstorming about the task is that the memes often have **nothing to do with the background.** Therefore, descriptions of the images might be not very helpful, or only very vaguely useful. For example, a picture of a dog smiling might have tons of different captions - which have nothing to do with the dog itself. But what matters is the fact that it is a very **positive** picture, and this "vibe" can be exploited in describing many different situations.

In the “Successful Kid” meme, the captions are not related to the kid itself, but more to his “cool vibes”. The **same caption** could have been used on the “cool cat” meme.

For example, if the caption is: *"POV: your face when you get your dream internship"*, it doesn't really matter if there is a smiling person on the image, or a smiling dog, or a smiling frog. Thus, my first approach was to **detect the sentiment** of the image using a vision-language model. CLIP-base model performed well on this task zero-shot.

- I choose **10 labels** for different emotions (because of limitations of possible label input in this CLIP version). American psychologist Paul Ekman identified six basic emotions: anger, disgust, fear, happiness, sadness and surprise. These emotions can also be **linked well to facial expressions.** I slightly modify the list of emotions.
- After choosing sentiments, I manually provide **10 examples** of meme captions for each sentiment. This is necessary as there is no dataset available with a fine-grained labeled sentiments of memes (or labels are not the same as we want).
- This way, my model is able to do **in-context learning.** I choose open-source **LLaMA-3-8B** (instruction-tuned) due to its relatively small size and good balance between creativity and sticking to the in-context examples. However, any other bigger model can be utilized.
- I choose to give captions in the same format: **"POV: ..."** This is done to achieve a more predictable behavior of the model and make it easier to parse the model's output.

The expected outcome in this approach is that it can **generalize well** to different images, since it is able to capture the sentiment and abstract from more specific context.

Adding a caption to the image

I use “Anton” font, as it is something many people associate with memes.
If the text is too long, it is split into 2 lines - top and bottom, and the size of the font changes depending on the amount of text.

- One presumably small but in fact annoying issue was **parsing the model's output into a proper caption.** Outputs don't always follow a given template; there is noise, different formats, different usage of punctuation, which makes it hard to rely on RE library.
- Thus, providing only examples that are of form **"POV: ..."** turned out helpful during caption extraction. Even though it narrowed the output space of the model, the outputs became more predictable..

I also tried producing images in the form of demotivators with a black background, but it doesn't look quite good (not only because the captions are misplaced, but in general). Meme author: Llama3-8B-it

Some results examples

Second approach: meme style adapters

As you probably noticed, a shortcoming of the sentiment approach is that it **doesn't care about the details** in the context. E.g., if you input an image of a smiling doctor or an image of a smiling dog, it might still produce the *same caption* (since the sentiment is "happy"). But what if we want a doctor-related meme?

To address that, I introduce another approach: **training adapters that learn specific genres of memes.**

- Training adapters is an efficient technique since only a small portion of the model is updated (in comparison to full fine-tuning). Moreover, we can train **several adapters** with different sentiments/meme styles, and easily switch between them at inference.

After looking through many datasets, I stopped at **Memes900k**[1]. It contains 300 meme templates, each with 3000 captions available from different users.

First, I label each of 300 meme templates with a sentiment (using CLIP, notebook provided here). Then I focus only on two sentiments: "**angry**" and "**happy**" (purely because of time constraints, to narrow the task).

Within the memes with these sentiments, I choose templates that:

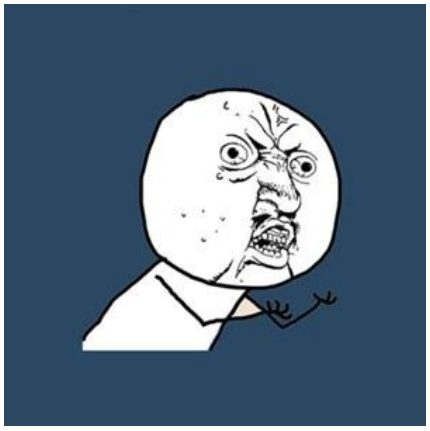
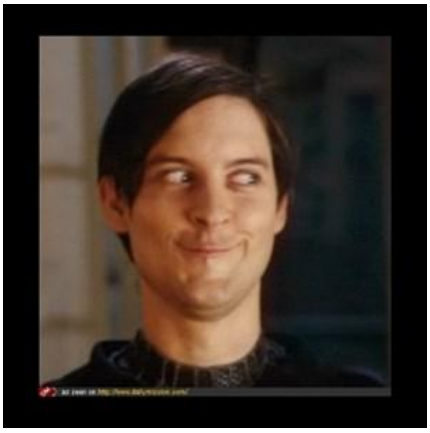
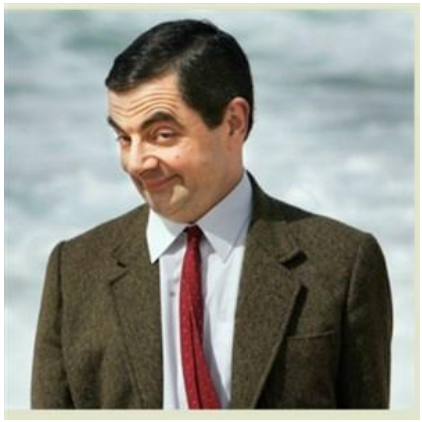
1. **Have the most recognizable form:** This is done so that the model is able to learn the pattern better with fewer examples. Some meme captions differ too much from each other even within one template.

[1] Presented in Borovik, Ilya & Khabibullin, Bulat & Kniazev, Vladislav & Pichugin, Zakhar & Olaleke, Oluwafemi. (2020). DeepHumor: Image-Based Meme Generation Using Deep Learning. 10.13140/RG.2.2.14598.14400.

2. **Are not too image-specific** (e.g. some template that has a very narrow topic of captions)
3. **Don't require complex text overlay** (e.g. the meme below would require more complex object detection)

As a result:

- For "angry" memes, I choose templates "Y U No" and "Y U So". They express rage or irritation at some situation. Examples:
    - bed is comfy in the morning, why u no comfy at night?
    - online game y u no lag for them
    - Internet Explorer, Y U no be good
- For "happy" memes, I choose templates "If you know what I mean" and "Tobey-maguire". They express happiness after doing or saying something sneaky. Examples:
    - GUESS WHAT'S COMING MY BIRTHDAY
    - the face your friend gives you when the teacher says to pair up
    - SEE CUTE GUY SMILES AT YOU SMILE BACK WITH THIS FACE

Meme templates: *Y U NO/ Y U SO*, *Tobey-maguire*, *If you know what I mean*

Note: *in my opinion, "angry" adapters perform better, since they had a more consistent template for every caption, while "happy" meme captions were more diverse and hence their patterns are less consistent.*

After selecting templates and examples for each meme type, I want to teach the adapter to generate **topic-specific captions**. I want to create a dataset with **topics** as inputs and **captions** as outputs. The Meme900k dataset provides captions, but not topics of captions. Hence, I label topics for each caption using **Llama-3-8B-Instruct** Inference API. For each meme template, I manually label 10 random captions and provide them as examples to Llama. The process is captured in this notebook.

```
Hobos <sep> Y u no earn money?
100%|██████████| 5/5 [00:02<00:00,  2.04it/s]
a person struggling financially
```

A funny and politically correct topic that Llama generated for the caption

- One more strength of the adapters is that they are able to help the model learn with a very small amount of examples. In my case, I labeled only around 600 samples for each sentiment. Partly this was dictated by a query limit == 300 queries/hour in HF Inference API.
- Here is the **Colab notebook** where I tuned Gemma's adapters.

I train two Gemma2B adapters - happy and angry - with a train-test split 80:20. **Gemma2B** turned out to result in a good compromise between size and performance.

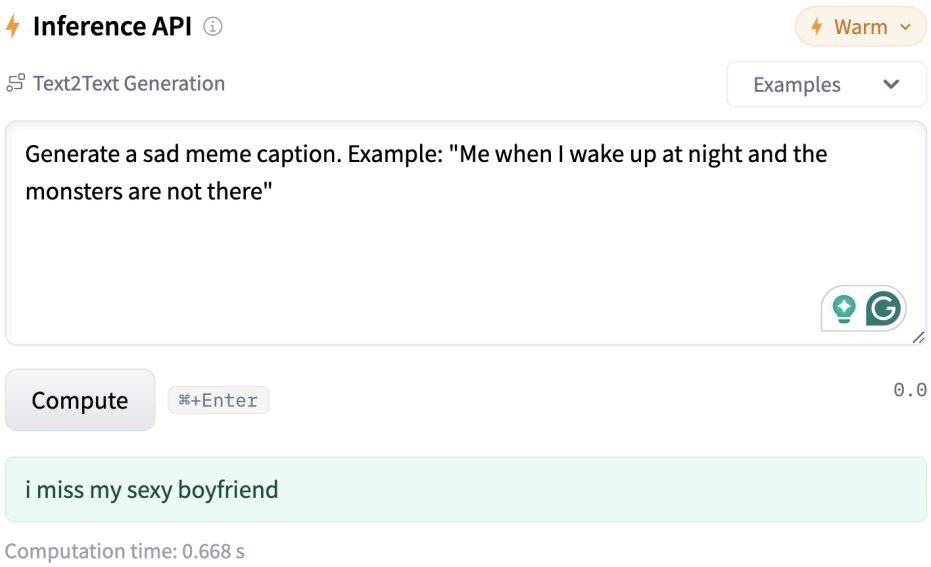
- Before switching to Gemma, I tried to teach other models to do caption generation (e.g., **FLAN-T5 family** - large, XL, and XXL; ...), but they didn't achieve reliable performance neither before training nor after. Notebooks with prefix-tuning of FLAN-T5 are available here (requires scrolling down a bit, in the future it will be a separate notebook).

The adapters are available on my HuggingFace profile:

- Happy
- Angry

```
angry
Actual caption: Why u no get 110% on math test <sep> Now I spank U
prediction:  do no like    s sep>   havek you
angry
Actual caption: Sign pen pad after a charge purchase <sep> do i hit accept or clear?
prediction: eguin  a long of sep>  notn have you? not?
angry
Actual caption: wow capcom Dlc sale <sep> so good, so nice much sale
prediction: w t ID   sep>  much  good better
angry
Actual caption: Sounds Like <sep> A Spiritual Malady
prediction:  like asep>   adys
angry
Actual caption: SHUT UP <sep> PHOTOGRAPHY, TRAVEL, HIP
prediction: IT UP sep> YONESAPHER IS PHVEL, AIR H
```

Some garbage captions from prompt-tuned FLAN-T5 large



FLAN-T5 XXL is also hopeless

- An alternative approach to labeling topics with LLMs could be simply taking hundreds/thousands of different templates, labeling their sentiments, then getting their image descriptions and passing to the model for training. However, this wouldn't fit in the **time/resource constraints** well, and most datasets I found didn't contain the needed amount and quality of images. Thus, I do the labeling myself (well, with open-source LLMs).

Getting image context

For fetching the image context, I use **BLIP-larg**e for image captioning. I use Inference API again to avoid loading heavy models locally.

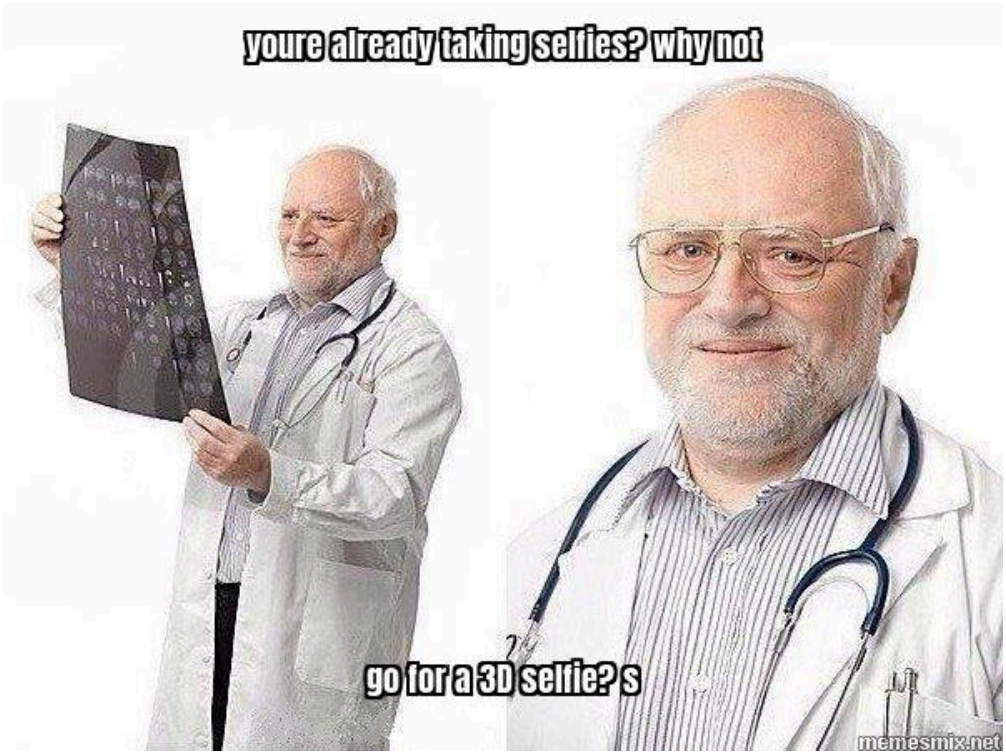
A description by BLIP-large: "arafed man in a suit and tie with his arms open"

In the end, I achieve the following pipeline:

1. A user inputs an image
2. CLIP model detects a sentiment of the image - happy or angry - and Gemma-2B LLM chooses a corresponding adapter
3. BLIP large model fetches a description of the user's image
4. Gemma with a particular sentiment adapter takes the description as an input and generates a caption
5. The caption is laid over the image using the Pillow library (just like in the first approach).

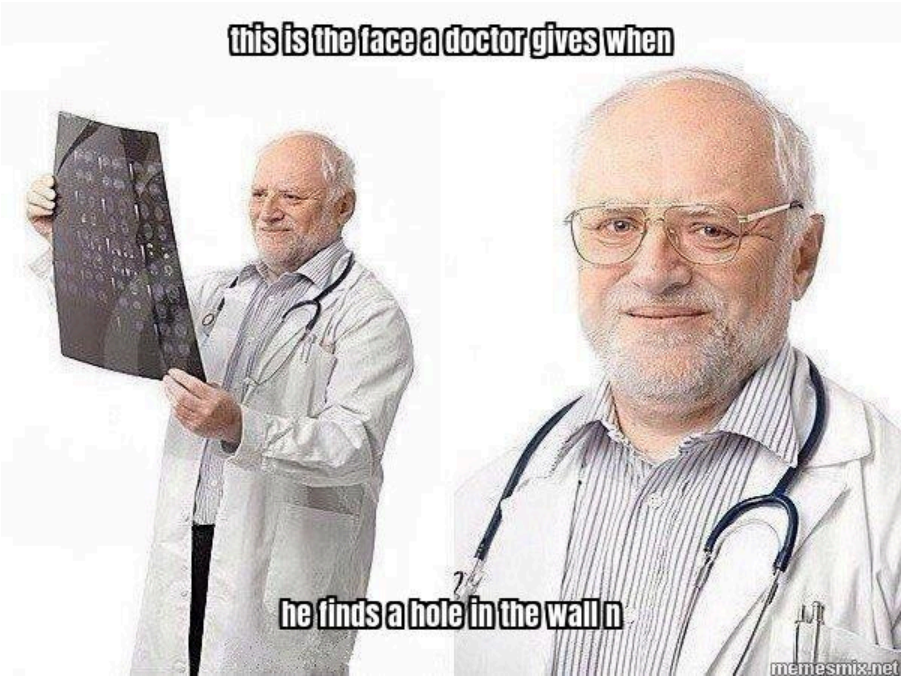
It is also possible to use the second approach via HuggingFace spaces, however, you would have to download the files. The instructions are provided on GitHub repo. I will consider how to improve that aspect to avoid inconveniences.

Some results examples



X-rays are 3D selfies technically?

Y-U-No template

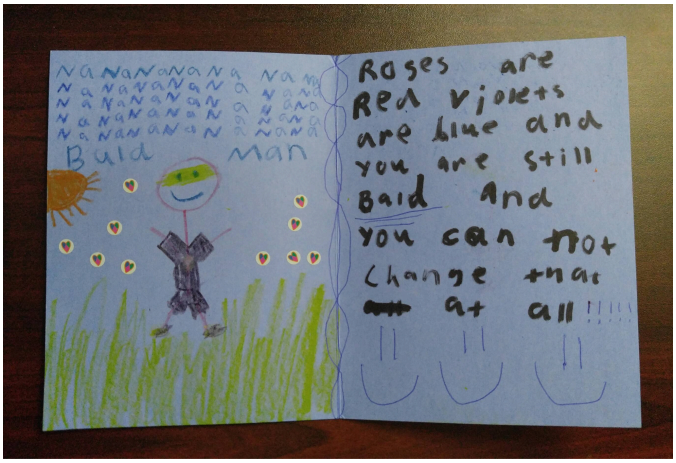

this is the face a doctor gives when
he finds a hole in the wall n
memesmix.net


that little face when he sees his owner arrive
n

Other notes

Datasets

While several meme datasets are available online, not all of them are suitable for our task. Some memes are self-content, and they don't need a text. In our task, however, we are talking specifically about the category of memes that are **IWT[2] (image-with-text)** so that without text the image doesn't make sense.



Examples of images that don't fit the task (Source, Kaggle meme dataset)

Handling noise

Memes900k dataset contained many captions in various languages, so I filtered only **English** captions using langdetect library. Unfortunately, there were captions that were **offensive and inappropriate**. Some captions were filtered by Gemma, as it refused to process them, however, there is no guarantee it detected every possible offensive statement.

[2] https://ojs.aaai.org/index.php/ICWSM/article/view/7287/7141

In production, I would do a **safety-check** for each caption training a classifier with high recall.

Limitations and future work

- Add more adapters for handling different sentiments
- Adapters for more templates - choose randomly different templates within some sentiment during caption generation
- The free option has a limited rate of queries per hour, which is 300. In the more advanced setting (e.g. in production with more GPUs), a model can be stored locally to avoid this limitation.
- Implement a safety check, so that the model doesn't generate offensive or discriminatory memes

Resources used

1. https://huggingface.co/blog/gemma-peft (Gemma tuning)
2. https://medium.com/@samvardhan777/fine-tune-gemma-using-qlora-%EF%B8%8F-6b2f2e76dc55 (Gemma tuning)
3. https://colab.research.google.com/drive/1Ys44kVvmeZtnICzWz0xgpRnrIOjZAuxp?usp=sharing (running Llama efficiently with Unsloth library)
4. Memes900k dataset: Borovik, Ilya & Khabibullin, Bulat & Kniazev, Vladislav & Pichugin, Zakhar & Olaleke, Oluwafemi. (2020). DeepHumor: Image-Based Meme Generation Using Deep Learning. 10.13140/RG.2.2.14598.14400.