CS 4740 HW 4

Q2.2 We've seen the functionality of <|endofcaption|> before; what do you think is the use of an additional <|endoftext|> special token? What bearing does the <|endofcaption|> token have while pretraining? (Present your views in under three sentences.)

Hint. Recall that due to the use of positional embeddings, we've limited our model to a set maximum sequence length.

The special token <|endoftext|> is for the model to understand when to terminate in the pretraining since there exists no <|endofcaption|> tokens in pretraining. Thus, the <|endofcaption|> token has no bearing in pretraining.

Q4. While masking the attention matrix using the padding mask, we only mask the pad keys and not pad queries; take a moment to convince yourself that this is true. Why is it not needed to mask the pad queries? Would not padding queries have any unintended consequences? Explain in under three sentences; unclear or misleading arguments will receive no credit.

Hint. Think of how the cross-entropy loss is backpropagated from outputs to input tokens, specifically the pad token.

We don't need to mask the pad queries because when we compute the dot product between the attention matrix (key-query matrix) and the values, the pad queries are only contributing to the computation for the pad timesteps, and these padded timesteps aren't contributing to our loss calculation. In contrast, we need to mask the pad keys because these are directly contributing to the token calculations at non padded timesteps, which are then being used to compute loss, and thus these must be masked out.

Q6.2. Explain how the training/validation differs from test inference. At inference time, can we benefit from the parallel processing abilities of our Seagull transformer model, i.e., pass all inputs at once and retrieve the associated outputs in one go? (Answer in no more than three sentences in total.)

Hint. What property of autoregressive language modeling (predicting next token) may/may not affect how the test inference is run.

At inference time, we cannot pass all the inputs in at once because we don't have the caption tokens (we don't have anything after <|caption|>), we only have the inputs for scene and uncanny. Since our model lacks all the inputs, we cannot retrieve the associated outputs in one go, and thus must generate the caption tokens. However, during training/validation we already know the expected caption, so we can pass in all inputs at once and retrieve the associated output.

Using your finetuned model, generate captions for some random test samples from the captions dataset and note down two/three of the most humorous ones below; format: scene, uncanny description, and the model-generated caption. (Maximum score: 2 points.)

This question is intended to have you inspect the generative and humor understanding abilities of the model, and to understand the impact of generation hyperparameters (e.g., temperature, top_k) on the generation diversity/quality.

1. <|scene|> A kid is in a robot suit with multiple hats on. He is making it uncomfortable for the passenger sitting next to him. <|uncanny|> Having a robot suit is uncomfortable to wear and causes space issues in a flight. <|caption|> The purr-fect weather forecast.
2. <|scene|> Two cavemen are sitting in a cave in a mountainous region. They are surrounded by what look like stone wheels. One of the cavemen is taking a bite of the wheel, while the other one holds his stomach as if he's already eaten too much. <|uncanny|> We assume that the wheels are made of stone, but actually, they appear to be doughnuts, which have a similar shape, but are edible and made of dough. <|caption|> Sorry, I was going for a nut-only diet.
3. <|scene|> A cowboy is on a horse. He looks angrily at another cowboy on a saddle. The saddle is in midair and not on a horse. <|uncanny|> There is a cowboy on a saddle in midair with no horse. <|caption|> When I said I wanted to "break out of the blue," I didn't mean it literally.