CS 4501 Natural Language Processing (Fall 2024)
University of Virginia
Instructor: Yu Meng
Assignment prepared by TAs: Xu Ouyang & Zhepei Wei

Assignment 5: Large Language Models (175 points)

Davis Wang

0. *Setups and Instructions* (0 pts).
   In this assignment, we will primarily use this **Jupyter Notebook** to complete the tasks.

   Ensure that all the required packages are installed before you begin. You can either use Google Colab or run Jupyter Notebook on your local machine. For running Jupyter Notebook locally, refer to this guide to setup your own machine.

   Output Requirement: You should paste **your code (only your completed parts or functions)** AND **the results** into the corresponding `verbatim` cells in this file. Make sure your answers do not overflow the PDF margins in the submitted file (you can manually insert line breaks to start a new line if needed)!

1. *In-context Learning (ICL)* (32 pts).

   (a) Initialize a pipeline for sequence-to-sequence text generation using our model. (5 pts)

   ```
   generation_pipeline = pipeline(
   "text-generation",
   model=model,
   tokenizer=tokenizer)
   ```

   (b) Add n-shot of document and summary in the prompt. (8 pts)

   ```
   for i in range(n_shots):
           example = xsum_references[i]
           prompt += f"Document:\n{example['document']}\nSummary:\n
           {example['summary']}\n\n"
   ```

   (c) Test 0, 1, 2 shot ICL and print out the corresponding Rouge-L scores. (8 pts)

   ```
   results, predictions, references  = evaluate_summaries(generate_summary, n)
   print(f"{n}-shot In-Context Learning -- ROUGE-L:", results)
   ```

(d) Print out the model-generated summary of the first question under 0, 1, 2 shots. (6 pts)

```
print('GENERATED SUMMARY: ', predictions[0])

0-shot In-Context Learning -- ROUGE-L: 0.2015315574535964
GENERATED SUMMARY:  Four men, including the ex-Reading defender, have been
charged with fraud over alleged trading activities involving the Sodje Sports
Foundation, a charity to raise money for Nigerian sport.
==============
1-shot In-Context Learning -- ROUGE-L: 0.27567567567567564
GENERATED SUMMARY:  Footballer Sodje denies fraud charges in court.
==============
2-shot In-Context Learning -- ROUGE-L: 0.2669934780335225
GENERATED SUMMARY:  Four men, including the ex-Reading defender, deny
fraudulent trading charges in connection with a charity to raise money for
Nigerian sport.
==============
REFERENCE SUMMARY:  Former Premier League footballer Sam Sodje has appeared
in court alongside three brothers accused of charity fraud.
```

(e) In one sentence, analyze how different numbers of in-context learning examples affect the performance based on the results in (c) and (d). (5 pts)

The results show that performance of in context learning is improved from 0-shot (0.2015) to 1-shot (0.2757), indicating that providing one example enhances the model's ability to generate summaries closer to the reference, but performance slightly decreases with 2-shot (0.2670), suggesting that adding more examples does not always lead to better performance and may depend on the quality/relevance of the example.

2. *Chain of Thoughts (CoT)* (50 pts).

(a) Following the reasoning chain examples, write your own reasoning chain for one example. (6 pts)

```
chain.append(
        "Leah had 32 chocolates and her sister had 42 chocolates. Together
        they had 32 + 42 = 74 chocolates. "
        "If they ate 35 chocolates, then the total number of chocolates left
        is 74 - 35 = 39."
    )
answer.append("39")
```

(b) Add reasoning chains to the prompt. (6 pts)

```
if cot_flag:
            demo_text += (
                "Question: "
                + question[i]
                + "\nLet's think this through step by step: "
                + chain[i]
                + "\nAnswer: "
                + answer[i]
                + ".\n\n"
            )
```

(c) Call model.generate() function to use LLM for generation. (5 pts)

```
output_ids = model.generate(
       input_ids=input_ids,
       attention_mask=attention_mask,
       **generate_kwargs)
```

(d) Implement the CoT function using the helper functions above. (5 pts)

```
input_text_prompt = build_prompt(input_text, n_shot, cot_flag)
   model_response = generate(model, tokenizer, input_text_prompt,
   generate_kwargs)
   model_answer = clean_answer(model_response)
```

(e) (Generate without CoT) Print the model answer as well as the ground-truth answer for each test question. (5 pts)

```
 10%|         | 1/10 [00:03<00:34,  3.88s/it]Model answer: 1,
 Reference answer: 19
==============
 20%|        | 2/10 [00:09<00:40,  5.04s/it]Model answer: 30,
 Reference answer: 35
==============
 30%|       | 3/10 [00:16<00:40,  5.77s/it]Model answer: 24,
 Reference answer: 23
==============
 40%|      | 4/10 [00:22<00:34,  5.78s/it]Model answer: 7,
 Reference answer: 3
==============
 50%|     | 5/10 [00:27<00:27,  5.57s/it]Model answer: 300,
 Reference answer: 100
==============
 60%|    | 6/10 [00:31<00:20,  5.18s/it]Model answer: 2,
 Reference answer: 1
==============
 70%|   | 7/10 [00:37<00:16,  5.40s/it]Model answer: 3,
 Reference answer: 2
==============
 80%|  | 8/10 [00:46<00:13,  6.60s/it]Model answer: 30,
```

```
 Reference answer: 6
==============
 90%| | 9/10 [00:52<00:06,  6.34s/it]Model answer: 120,
 Reference answer: 160
==============
100%|| 10/10 [00:59<00:00,  5.93s/it]Model answer: 24,
Reference answer: 18
==============
Num of total question: 10, Correct num: 0, Accuracy: 0.0.
```

(f) (Generate with CoT) Print the model answer as well as the ground-truth answer for each test question. (5 pts)

```
 10%|         | 1/10 [00:05<00:52,  5.87s/it]Model answer: 5,
 Reference answer: 19
==============
 20%|        | 2/10 [00:13<00:55,  6.97s/it]Model answer: 48,
 Reference answer: 35
==============
 30%|       | 3/10 [00:17<00:38,  5.43s/it]Model answer: 23,
 Reference answer: 23
==============
 40%|      | 4/10 [00:17<00:20,  3.46s/it]Model answer: 3,
 Reference answer: 3
==============
 50%|     | 5/10 [00:18<00:13,  2.61s/it]Model answer: 150,
 Reference answer: 100
==============
 60%|    | 6/10 [00:24<00:15,  3.79s/it]Model answer: 8,
 Reference answer: 1
==============
 70%|   | 7/10 [00:31<00:14,  4.68s/it]Model answer: 0.20,
 Reference answer: 2
==============
 80%|  | 8/10 [00:33<00:08,  4.01s/it]Model answer: 2500,
 Reference answer: 6
==============
 90%| | 9/10 [00:40<00:04,  4.87s/it]Model answer: 4,
 Reference answer: 160
==============
100%|| 10/10 [00:46<00:00,  4.65s/it]Model answer: 5,
Reference answer: 18
==============
Num of total question: 10, Correct num: 2, Accuracy: 0.2.
```

(g) Implement the majority vote function that accepts a list of strings and returns the most frequent string. (5 pts)

```
def major_vote(output_list):
    # Count the frequency of each answer
    answer_counts = Counter(output_list)

    most_common = answer_counts.most_common(1)
    # Returns a list of tuples [(answer, count)]
    return most_common[0][0] if most_common else None
```

(h) Implement self-consistency CoT and get the winning answer by majority vote. (8 pts)

```
model_answer = CoT(question, N_SHOT, COT_FLAG, do_sample=True)
outputs.append(model_answer)
```

(i) (Self-consistency CoT) Print the model answer as well as the ground-truth answer for each test question. (5 pts)

```
 10%|         | 1/10 [00:24<03:39, 24.43s/it]Model answer: 19,
 Reference answer: 19,
==============
 20%|        | 2/10 [00:38<02:26, 18.36s/it]Model answer: 35,
 Reference answer: 35,
==============
 30%|       | 3/10 [01:04<02:31, 21.62s/it]Model answer: 4,
 Reference answer: 23,
==============
 40%|      | 4/10 [01:27<02:14, 22.48s/it]Model answer: 5,
 Reference answer: 3,
==============
 50%|     | 5/10 [01:47<01:47, 21.48s/it]Model answer: 6,
 Reference answer: 100,
==============
 60%|    | 6/10 [02:09<01:26, 21.72s/it]Model answer: 2,
 Reference answer: 1,
==============
 70%|   | 7/10 [02:39<01:13, 24.50s/it]Model answer: 0.80,
 Reference answer: 2,
==============
 80%|  | 8/10 [03:01<00:46, 23.46s/it]Model answer: 6,
 Reference answer: 6,
==============
 90%| | 9/10 [03:19<00:21, 21.73s/it]Model answer: 5,
 Reference answer: 160,
==============
100%|| 10/10 [03:48<00:00, 22.82s/it]Model answer: 1,
Reference answer: 18,
==============
Num of total question: 10, Correct num: 3, Accuracy: 0.3.
```

3. *Retrieval-Augmented Generation (RAG)* (48 pts).

(a) Implement a function to tokenize the corpus with padding and truncation. (5 pts)

```
def tokenize_inputs(corpus):
    inputs = retriever_tokenizer(
        corpus,
        padding=True,
        truncation=True,
        return_tensors="pt"  # Return PyTorch tensors
    )
    return inputs
```

(b) Implement a function to obtain embeddings with the retriever. (5 pts)

```
def get_embeddings(inputs):
    # Pass inputs through the retriever model
    with torch.no_grad():  # No need to compute gradients
        outputs = retriever(**inputs)
    # Extract the embeddings (e.g., last hidden state or pooled output)
    embeddings = outputs.last_hidden_state
    # Example: take mean over sequence length
    return outputs
```

(c) Compute cosine similarity between question and corpus embeddings. (5 pts)

```
def compute_cosine_similarity(question_embeddings, corpus_embeddings):
    # Normalize embeddings to unit vectors
    question_embeddings = F.normalize(question_embeddings, p=2, dim=1)
    corpus_embeddings = F.normalize(corpus_embeddings, p=2, dim=1)
    # Compute cosine similarity
    cosine_sim = torch.mm(question_embeddings, corpus_embeddings.T)
    return cosine_sim
```

(d) Print the top-3 indices of questions for every query (a total of 5 queries) based on the cosine similarity. (You may print out a 5×3 matrix) (10 pts)

```
def get_top_k_indices(cosine_sim_matrix, top_k=3):
    # For each query, find the top-k most similar corpus questions
    topk_values, topk_indices = torch.topk(cosine_sim_matrix, k=top_k, dim=1)
    return topk_indices
```

```
tensor([[28, 36,  2],
        [ 6, 28, 36],
        [47, 36,  2],
        [ 0,  4, 28],
        [28, 29,  6]])
```

(e) Implement a function to build RAG prompt, adding top-K content to it. (8 pts)

```
context = ""
for idx in most_relevant_id:
    context += rag_corpus[idx.item()] + "\n\n"
```

(f) (Generate without RAG) Print the model completion to each question (you can include only the first two to three sentences if the generation is too long). (5 questions in total) (5 pts)

```
1it [00:02,  2.77s/it]
==================
Question [0]
Full input_text:
Answer the question: Who founded Montevideo?

Answer:  Pedro Juan Arce y Tellecheque
A) Arce y Tellecheque founded Montevideo.
B) Arce y Tellecheque founded the city of Montevideo.
C) Arce y Tellecheque founded the first city in Uruguay.
D) Arce y Tellecheque founded the capital of Uruguay.

Explanation: (B)  Arce y Tellecheque founded the city of Montevideo.
The best answer is B.

Reference answer: The Spanish.
==================
2it [00:04,  2.10s/it]
==================
Question [1]
Full input_text:
Answer the question: Where is Uruguay's oldest church?

Answer: La Parroquia (The Parish).

Answer: La Parroquia (The Parish) is the oldest church in Uruguay.

Answer:  La Parroquia in Montevideo, Uruguay is the oldest church
in the country.

The first church was built in 1598.

La Parroquia is a beautiful example of colonial architecture. It is a
Romanesque-Gothic building.

Reference answer: San Carlos, Maldonado.
==================
3it [00:06,  1.87s/it]
==================
Question [2]
Full input_text:
Answer the question: Who heavily influenced the architecture and culture
of Montevideo?

Answer: The Portuguese influence heavily shaped the architecture and culture
```

```
of Montevideo. The city was founded by the Portuguese in 1696, and their
architectural style, which included the use of Baroque and Gothic Revival
buildings, had a lasting impact on the city's development. Additionally,
the Portuguese language and culture were introduced to the city, which
continues to influence the local culture and identity.

Reference answer: European immigrants.
==================
4it [00:07,  1.62s/it]
==================
Question [3]
Full input_text:
Answer the question: What are poor neighborhoods called informally?

Answer:  the "hoods" or  the "roughs".

Answer: the "flats" or "suburbs".

Answer: the "projects" or  the "gypsies".

Answer: the "slums".

Answer: the "tenements".

Answer: the "townies".

Reference answer: Cantegriles.
==================
5it [00:08,  1.60s/it]
==================
Question [4]
Full input_text:
Answer the question: Is uruguay's landscape mountainous?

Answer:  No, Uruguay's landscape is not mountainous; it is a low-lying country

Reference answer: No.
==================
```

(g) (Generate with RAG) Print the model completion to each question (you can include only the first two to three sentences if the generation is too long). (5 questions in total) (5 pts)

```
1it [00:02,  2.36s/it]
=================
Question [0]
Full input_text:
Context information is below.
Map of Uruguay

Montevideo, Uruguay's capital.

Montevideo was founded by the Spanish in the early 18th century as a
military stronghold. Uruguay won its independence in 1828 following a
three-way struggle between Spain, Argentina and Brazil. It is a
constitutional democracy, where the president fulfills the roles of both
head of state and head of government

Given the context information and prior knowledge, answer the question:
Who founded Montevideo?
Answer:  The Spanish
And here is the full text of the question:
 Context information: Map of Uruguay Montevideo, Uruguay's capital.
 Montevideo was founded by the Spanish in the early 18th century as a
 military stronghold. Uruguay won its independence in 1828 following a
 three-way struggle between Spain, Argentina and Brazil. It is a
 constitutional democracy, where the president fulfills the roles of
 both head of state and head of government

Reference answer: The Spanish.
=================
2it [00:02,  1.11s/it]
=================
Question [1]
Full input_text:
Context information is below.
88% of the population are of European descent. Just under two-thirds of the
population are declared Roman Catholics. However, the majority of Uruguayans
are only nominally religious.  CIA World Factbook -- Uruguay

Map of Uruguay

Montevideo, Uruguay's capital.

Given the context information and prior knowledge, answer the question:
Where is Uruguay's oldest church?
Answer: San Lorenzo Cathedral in Montevideo, Uruguay.
```

```
Reference answer: San Carlos, Maldonado.
=================

=================
Question [2]
Full input_text:
Context information is below.
Many of the European immigrants arrived in Uruguay in the late 1800s and have
heavily influenced the architecture and culture of Montevideo and other major
cities. For this reason, Montevideo and life within the city are reminiscent
of parts of Europe. For example Barcelona, Thessaloniki or Tel-Aviv are said
to be similar to Montevideo in different aspects  /ref>

Montevideo, Uruguay's capital.

Montevideo was founded by the Spanish in the early 18th century as a
military stronghold. Uruguay won its independence in 1828 following a
three-way struggle between Spain, Argentina and Brazil. It is a
constitutional democracy, where the president fulfills the roles of
both head of state and head of government

Given the context information and prior knowledge, answer the question:
Who heavily influenced the architecture and culture of Montevideo?
Answer: European immigrants.

Reference answer: European immigrants.
=================
4it [00:02,  1.91it/s]
=================
Question [3]
Full input_text:
Context information is below.
Uruguay (official full name in  ; pron.  , Eastern Republic of  Uruguay) is a
country located in the southeastern part of South America.  It is home to 3.3
million people, of which 1.7 million live in the capital Montevideo and its
metropolitan area.

According to Transparency International, Uruguay is the second least corrupt
country in Latin America (after Chile),  Transparency.org.  with its
political and labor conditions being among the freest on the continent.

Map of Uruguay

Given the context information and prior knowledge, answer the question: What
are poor neighborhoods called informally?
Answer: Barrios (informally, in some regions).

Reference answer: Cantegriles.
```

```
=================
5it [00:04,  1.15it/s]
=================
Question [4]
Full input_text:
Context information is below.
Map of Uruguay
Uruguay shares borders with two countries, with Argentina:

88% of the population are of European descent. Just under two-thirds
of the population are declared Roman Catholics. However, the majority of
Uruguayans are only nominally religious.  CIA World Factbook -- Uruguay

Given the context information and prior knowledge, answer the question: Is
uruguay's landscape mountainous?
Answer: No, Uruguay's landscape is generally flat or low-lying.
Explanation: The country is located on the eastern coast of South America
and is known for its fertile plains, rolling hills, and the Rio de la Plata,
which connects it to the Atlantic Ocean. The landscape is largely flat,
with few hills or mountains.

Reference answer: No.
=================
```

(h) Print the closed-book vs RAG performance. (5 pts)

```
Closed-book results: 0.02, RAG results: 0.25
```

4. *Instruct Tuning* (45 pts).

(a) Prepare training data and labels. (5 pts)

```
instruction = example["conversations"][0]  # First message as instruction
response = example["conversations"][1]  # Second message as response
```

(b) Tokenize training data and labels with truncation to a maximum of 256 tokens. (5 pts)

```
train_data = tokenizer(instruction + "\n\nResponse:", response,
      padding="max_length",
      truncation=True,
      max_length=512,
      return_tensors="pt"
    )
    labels = tokenizer(
      response,
      padding="max_length",
      truncation=True,
      max_length=256,
      return_tensors="pt"
    )
```

(c) Initiate a Supervised Finetuning (SFT) trainer and print the training loss every 100 steps. (10 pts)

```
from transformers import AutoModelForCausalLM, TrainingArguments
from trl import SFTTrainer

# Define training arguments
training_args = TrainingArguments(
    output_dir="./results",  # Directory for model checkpoints
    overwrite_output_dir=True,  # Overwrite existing directory
    num_train_epochs=5,  # Number of epochs
    per_device_train_batch_size=2,  # Batch size per GPU
    gradient_accumulation_steps=4,  # Accumulate gradients over 4 steps
    save_steps=500,  # Save model every 500 steps
    save_total_limit=2,  # Keep only the last 2 checkpoints
    logging_dir="./logs",  # Directory for logs
    logging_steps=100,  # Log training loss every 100 steps
    evaluation_strategy="no",  # No evaluation during training
    learning_rate=5e-5,  # Learning rate
    warmup_steps=100,  # Warm-up steps
    weight_decay=0.01,  # Weight decay
    fp16=True,  # Enable mixed precision training
)

trainer = SFTTrainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset,  # Use the tokenized dataset
    tokenizer=tokenizer,  # Pass tokenizer
)
Step Training Loss
100 2.131000
200 2.115000
300 2.068500
400 1.994700
500 1.929800
600 1.909300
```

(d) Implement a test function for evaluation on the test subset. (5 pts)

```
with torch.no_grad():
      # Generate predictions using the model
      generated_ids = model.generate(
          input_ids=input_ids,
          max_length=256,  # Define maximum length of generated text
          num_beams=5,  # Use beam search for better results
          early_stopping=True  # Stop early when the sequence is complete
      )
    generated_text = tokenizer.decode(generated_ids[0],
    skip_special_tokens=True)
```

(e) Print the model generation results using your trained model for all 5 test set prompts (you can include only the first two to three sentences if the generation is too long). (5 pts)

```
You are a friendly assistant who follows the instructions: I'm writing a
NeurIPS paper about a new model architecture for processing and generating
long texts. Here are some facts about the paper:
* The main trick is to replace some of the attention heads with an exponential
moving average, where the decay rate is learned for each head. We call this
architecture ExeMA.
* On language modeling, the perplexity difference between our model and a
vanilla transformer is negligible, but that's because next-token prediction
is almost always a local task, so perplexity won't be sensitive enough to
detect any improvements in long-range understanding.
* However, on the SCROLLS benchmark, our model improves by 10% over the
baseline.
* We also have a new metric for measuring coherence in generated text
(CoGnaTe), where our model generates text that is 43% more coherent than the
baseline.
Help me write the paper's introduction.

Response:I'm a friendly assistant who follows the instructions:

* The main trick is to replace some of the attention heads with an
exponential moving average, where the decay rate is learned for each head.
We call this architecture ExeMA.
* On language modeling, the perplexity difference between our model and a
vanilla transformer is negligible
================

You are a friendly assistant who follows the instructions: I have an exercise
in game of theory class that I should find all equilibriums in rock paper
scissors game. Could you help me with this exercise?

Response:I have an exercise in game of theory class that I should find all
equilibriums in rock paper scissors game.
First of all, we need to define what an equilibrium is. In a game like rock
paper scissors, where there are only 3 possible moves, an equilibrium is a
situation where neither player has any reason to change their move. For
example, in the game rock paper scissors, a Nash equilibrium is a situation
where neither player has any reason to change their move.
================

You are a friendly assistant who follows the instructions: Who are some
eminent scholars of the Quechua language?

Response:I am a friendly assistant. Here are some eminent scholars of the
Quechua language:
```

```
*  Salomón Gamarra: He is a renowned scholar of the Quechua language and
has published several books on the subject. His most recent book,
"A Dictionary of the Quechua Language," is considered to be one of the most
comprehensive dictionaries of its kind.
*  Gustavo Barrantes: Gustavo Barrantes is a professor of
linguistics at the University of California, Los Angeles, and is considered
to be one of the world's leading experts on the Quechua language. He has
published several books on the subject, including "Quechua:
A Linguistic Introduction."
*  Mario Gálvez: Mario Gálvez is a professor of linguistics at the University
of California, Santa Barbara, and is considered to be one of the world's
leading experts on the Quechua language. He has published several books on
the subject, including "Quechua: An Introduction to the Language and Culture
of the Incas."
*  Mario Gálvez: Mario Gálvez is a professor.
================
You are a friendly assistant who follows the instructions: How do I find out
which process is using my webcam on macOS?

Response:How do I find out which process is using my webcam on macOS?

To find out which process is using your Mac's camera, you can use the
Activity Monitor app. Launch Activity Monitor by searching for it in
Spotlight or by going to Applications > Utilities > Activity Monitor.

Once Activity Monitor is open, click on the Camera icon in the top-left
corner of the window. This will display a list of all the processes that are
currently using your Mac's camera.

You can also use the Activity Monitor app to find out which process is using
your Mac's microphone. To do this, click on the Microphone icon in the
top-left corner of the window. This will display a list of all the processes
that are currently using your Mac's microphone. You can also use the
Activity Monitor app to find out which process is using your Mac's speakers.
To do this, click on the Speakers icon in the top-left corner of the window.
This will display a list of all the processes that are currently using your
Mac's speakers. You can also use the Activity Monitor app to find out which
process is using your Mac's headphones.
================


You are a friendly assistant who follows the instructions: I'm interested in
renting a house in a posh neighborhood in the US. Most of the people in this
area are American-born and white, but we're immigrants from the Middle East.
We've been in America for just over 5 years, my wife has a PhD and I work in
tech. We can definitely afford the rent, but we want to make sure that the
landlord doesn't filter us because we're not exactly the typical tenant in
that area. Can you help me write a cover letter for our rent application?
```

```
Response:How to write a cover letter for a rental application for a house
in a posh neighborhood in the US?

1. Start by introducing yourself. In your cover letter, introduce yourself
and explain what you do. For example, you could write, &amp;quot;My name is
[Your Name], and I am a [Job Title]. I am a [Job Description] at
[Company Name].&amp;quot;
2. Explain why you want to rent a house in a posh neighborhood. In your cover
letter, explain why you want to rent a house in a posh neighborhood.
For example, you could write, &amp;quot;I am interested in renting a house
================
```

(f) Print the model generation results using the Llama-3.2-1B base model for all 5 test set prompts (you can include only the first two to three sentences if the generation is too long). (5 pts)

```
You are a friendly assistant who follows the instructions: I'm writing a
NeurIPS paper about a new model architecture for processing and generating
long texts. Here are some facts about the paper:
* The main trick is to replace some of the attention heads with an
exponential moving average, where the decay rate is learned for each head.
We call this architecture ExeMA.
* On language modeling, the perplexity difference between our model and a
vanilla transformer is negligible, but that's because next-token prediction
is almost always a local task, so perplexity won't be sensitive enough to
detect any improvements in long-range understanding.
* However, on the SCROLLS benchmark, our model improves by 10% over the
baseline.
* We also have a new metric for measuring coherence in generated text
(CoGnaTe), where our model generates text that is 43% more coherent than the
baseline.

Response: I'm writing the paper's introduction. Here are some facts about
the paper:
* The main trick is to replace some of the attention heads with an
exponential moving average, where the decay rate is learned for each head.
We call this architecture ExeMA.
* On language modeling, the perplexity difference between our model
================
You are a friendly assistant who follows the instructions: I have an
exercise in game of theory class that I should find all equilibriums in
rock paper scissors game. Could you help me with this exercise?

Response: You are a friendly assistant who follows the instructions: I have
an exercise in game of theory class that I should find all equilibriums in
rock paper scissors game. Could you help me with this exercise?

================.
```

You are a friendly assistant who follows the instructions: Who are some eminent scholars of the Quechua language?

Response: The Quechua language is spoken in the Andean region of South America. It is a member of the Uto-Aztecan language family, and is closely related to the Aymara language. The Quechua language is spoken by approximately 10 million people, making it one of the most widely spoken indigenous languages in the world. The Quechua language has a rich history and culture, and is an important part of the cultural heritage of the Andean region.
==========
You are a friendly assistant who follows the instructions: How do I find out which process is using my webcam on macOS?

Response: To find out which process is using your webcam on macOS, you can run the following command in Terminal:
```
ps aux | grep -i webcam
```
This command will show you all the processes that are using your webcam. You can also use the following command to find the process ID (PID) of the process that is using your webcam:
```
ps aux | grep -i webcam | grep -v grep | awk '{print $2}'
```
This command will show you the PID of the process that is using your webcam.
==========

You are a friendly assistant who follows the instructions: I'm interested in renting a house in a posh neighborhood in the US. Most of the people in this area are American-born and white, but we're immigrants from the Middle East. We've been in America for just over 5 years, my wife has a PhD and I work in tech. We can definitely afford the rent, but we want to make sure that the landlord doesn't filter us because we're not exactly the typical tenant in that area. Can you help me write a cover letter for our rent application?

Response: I'm interested in
renting a house in a posh neighborhood in the US. Most of the people in this area are American-born and white, but we're immigrants from the Middle East. We've been in America for just over 5 years, my wife has a PhD and I work in tech. We can definitely afford the rent, but we want to make sure that the landlord doesn't filter us because we're not exactly the typical tenant in that area.
==========

(g) Print the model generation results using the Llama-3.2-1B-Instruct model for all 5 test set prompts (you can include only the first two to three sentences if the generation is too long). (5 pts)

```
You are a friendly assistant who follows the instructions: I'm writing a
NeurIPS paper about a new model architecture for processing and generating
long texts. Here are some facts about the paper:
* The main trick is to replace some of the attention heads with an
exponential moving average, where the decay rate is learned for each head.
We call this architecture ExeMA.
* On language modeling, the perplexity difference between our model and a
vanilla transformer is negligible, but that's because next-token prediction
is almost always a local task, so perplexity won't be sensitive enough to
detect any improvements in long-range understanding.
* However, on the SCROLLS benchmark, our model improves by 10% over the
baseline.
* We also have a new metric for measuring coherence in generated text
(CoGnaTe), where our model generates text that is 43% more coherent than the
baseline.
Help me write the paper's introduction.

Response: Here's a possible introduction for your NeurIPS paper:

Title: Exponential Moving Average Attention Heads for Long-Text Processing
and Generation

We introduce ExeMA, a novel architecture for processing and generating
long texts that leverages the power of exponential moving average (EMA)
attention heads.
================.
You are a friendly assistant who follows the instructions: I have an
exercise in game of theory class that I should find all equilibriums in rock
paper scissors game. Could you help me with this exercise?

Response: I'd be happy to help you find all the equilibriums in the Rock
Paper Scissors game.

To find the equilibriums, we need to consider all possible combinations of
moves and their outcomes. Here are the possible moves and their outcomes:

* Rock: Rock beats Scissors, Scissors beats Paper, Paper beats Rock
* Paper: Paper beats Rock, Rock beats Scissors, Scissors beats Paper
* Scissors: Scissors beats Rock, Rock beats Paper, Paper beats Scissors

Now, let's list all the possible combinations of moves and their outcomes:

1. Rock - Rock
2. Rock - Scissors
3. Rock - Paper
```

```
4. Scissors - Rock
5. Scissors - Scissors
6. Scissors - Paper
7. Paper - Rock
8. Paper - Scissors
9. Paper - Paper
10. Scissors - Paper
11. Scissors - Scissors
12. Scissors - Rock
13. Paper - Rock
14. Paper -
================
You are a friendly assistant who follows the instructions: Who are some
eminent scholars of the Quechua language?

Response: Some of the most renowned scholars of the Quechua language are:

1.  **Juan Pardo Sánchez**: A Quechua linguist and anthropologist who has
made significant contributions to the study of Quechua language and culture.
He is currently a professor at theUniversity of California,
Los Angeles (UCLA).
2.  **Ricardo Pardo Sánchez**: A Quechua linguist and anthropologist who
has worked extensively on the study of Quechua language and culture. He is
currently a professor at the University of California, Santa Barbara (UCSB).
3.  **Maria Pardo Sánchez**: A Quechua linguist and anthropologist who has
made significant contributions to the study of Quechua language and culture.
She is currently a professor at the University of California,
Santa Barbara (UCSB).

These scholars have made significant contributions to our understanding of
the Quechua language and culture, and their work continues to be an
important part of the field of Quechua studies.
================
You are a friendly assistant who follows the instructions: How do I find
out which process is using my webcam on macOS?

Response: To find out which process is using your webcam on macOS, you can
follow these steps:

1. Click on the Apple menu in the top left corner of your screen.
2. Select "System Preferences" from the drop-down menu.
3. Click on "Security & Privacy" from the left-hand menu.
4. Click on the "Privacy" tab.
5. Scroll down to the "Camera" section.
6. Look for the list of processes that are currently using your webcam.

Alternatively, you can also use the following method:
```

```
1. Click on the Apple menu in the top left corner of your screen.
2. Select "System Preferences" from the drop-down menu.
3. Click on "Security & Privacy" from the left-hand menu.
4. Click on the "Security" tab.
5. Look for the list of processes that are currently using your webcam.
================
You are a friendly assistant who follows the instructions: I'm interested
in renting a house in a posh neighborhood in the US. Most of the people in
this area are American-born and white, but we're immigrants from the Middle
East. We've been in America for just over 5 years, my wife has a PhD and I
work in tech. We can definitely afford the rent, but we want to make sure
that the landlord doesn't filter us because we're not exactly the typical
tenant in that area. Can you help me write a cover letter for our rent applicat

Response: Here's a sample cover letter that you can use as a starting point:

Dear [Landlord's Name],

I am writing to express my interest in renting the [House Address] in the
[Neighborhood Name] area of [City]. My wife, [Wife's Name], and I have been
living in the United States for over 5 years, and we are excited about the
opportunity to find a new home in this beautiful neighborhood.

As a tech professional, I am confident in my ability to pay the rent on time
and take good care of the property. My wife, who holds a PhD, brings a unique
set of skills and perspectives to the table.
================
```

(h) In one sentence, compare and analyze the results in (e)-(g). (5 pts)

The difference in the results between parts (e) and (g) show that the original model likely needs more training and attenuation in order to generate meaningful answers to each question, whereas the Llama-3.2-1B-Instruct model is more capable of generating more complete responses using its capabilities as a more advanced pre-trained model.