**Natural Language Processing**

**HW3 - Decoding, Prompting and Instruction Tuning**



**Computer Software Engineering(컴퓨터소프트웨어학부)**

**2020081421**

**Kim Sung Hyuk(김성혁)**

1. **[Model Explanation]**

**Model :** Llama 2 7B-hf(Lagre Language Model Meta AI)

* **Base model :** “meta-llama/Llama-2-7b-hf”
* **Instruction Fine-tuned Model :** “meta-llama/Llama-2-7b-chat-hf”

LLaMA 2 (Large Language Model Meta AI) is a family of open-source large language models developed by Meta. It offers various parameter options, allowing the selection of a suitable model(7B parameters, most smaller one in LLaMA2) based on the given hardware(e.g. GPU, RAM)

As the model is based on a Transformer Decoder, it supports auto-regressive generation.  
Although LLaMA 3 supports multiple languages, only English will be used for this homework3 , that making LLaMA 2 sufficient.

LLaMA 2 stands out for its variety of options, ease of use, and strong performance, which were key factors in selecting this model for the task.

1. **[Decoding algorithms]**

* **3 unfinished test sentences :**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

* 'Hanyang University is~'
* 'Difficulty of learning Korean compare to English is~'
* 'Alpine ski is~'

**2-1. Greedy Search**

* It is simplest decoding method.
* Selects the highest probability as its next word **at each timestep** t.

**[Output Result]**  
**텍스트, 폰트, 영수증, 대수학이(가) 표시된 사진

자동 생성된 설명**

**[Pro]**

* Simple and fast

**[Con]**

* Repetition Happen
* Miss high probability words hidden begind a low probability word -> We can alleviate by using **Beam Search(2-2)**

**2-2. Beam Search**

* On each step of decoder, keep track of the k most probable partial translations. (k : beam size)
* All hypothesis has score(log probability)
* Decode untill **reach timestep T** or **at least n completed hypothesis**

**[Output Result]**텍스트, 스크린샷, 영수증, 폰트이(가) 표시된 사진

자동 생성된 설명

**[Pro]**

* Reduces the risk of missing hidden high probability word sequences
* Work well in task such as Machine Translation or Text Summarization

**[Con]**

* Does not guarantee to find optimal solution(or most likely output)
* Repetition Happen -> we can solve it by **no repeat** **n-gram(2-3)**
* Not good at open-ended generation such as dialog and story generation

**2-3. No Repeat N-Gram**

* Simple option for not to generate(or reduce) repetition on Beam Search Decoding

**[Output Result]**  
**텍스트, 스크린샷, 폰트, 영수증이(가) 표시된 사진

자동 생성된 설명**

**[Pro]**

* Help avoid generating repetitive phrases or words by restricting the model from repeating n-grams (a sequence of n consecutive words)
* By limiting repeated structures, the generated text often sounds more fluent and human-like

**[Con]**

* Preventing repetition too aggressively might result in a loss of context or coherence
* If the restriction is too stringent, it might force the model to generate less optimal or less accurate outputs

**2-4. Vanilla Sampling**

* Randomly picking the next word wt according to its conditional probability distribution

**[Output Result]**  
텍스트, 영수증, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명  
**[Pro]**

* Repetitiion is not occuring as much as in Beam Search or Greedy Decoding Algorithms

**[Con]**

* Model often generate incoherent gibberish beacuse Vanilla sampling makes every token in the vocabulary an option.

**2-5. Top - K Sampling**

* Sample from the top k tokens in the probability distribution.
* Incresing k -> Diverse, risky output
* Decresing k -> Safe, generic output

**[Output Result]**텍스트, 스크린샷, 영수증, 대수학이(가) 표시된 사진

자동 생성된 설명

**[Pro]**

* Much more human like text compare to vanila Sampling

**[Con]**

* Cannot **dynamically adapt** the k value  
  **2-6. Top - p (Nucleus) Sampling**
* Sample from all tokens in the top p cumulative probability mass

**[Output Result]**  
텍스트, 영수증, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명  
**[Pro]**

* Number of words in the set can dynamically increase and decrese
* Diversity increses

**[Con]**

* Sensitive to p value.

**3. [Prompting Base Model vs Instruction fine-tuned Model]**

- Let’s look how our base model and instruction fine tuned model react to the complex question

- **Using Top-k decoding algorithm** to generate answer

- Basic question, Few-Shot prompt, CoT prompt

**Complex step by step question :**

"Kim has 3 boxes of apples, with each box containing 10 apples. Kim gives 1 box of apples to Jung and eats 3 apples from one of the remaining boxes.”

How many apples does Kim have now?"

**3-1. [Base Model]**

**-model :** “meta-llama/Llama-2-7b-hf”

**3-1-1. Base Model**

**[Input : Complex question]**

**텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**[Output Result]텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명**

* **Resulted in the model generating repetitive sequences**
* **Do not give a proper answer to the given complex question**

**3-1-2. Few-Shot Prompting**

* Using Few-Shot Prompting to our Base Model
* Few-shot prompting involves providing the model with a small number of examples (in this case, 3) within the prompt to help it understand the task. By showing the model how a problem is solved with a few specific examples, the model can generalize from these examples and apply the learned patterns to new inputs

**[Input : Few-Shot Prompt]**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**[Output]**

**텍스트, 스크린샷, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명**

* **Same as the above case, repetition occurred.**
* **Do not give a proper answer to the given complex question**

**3-1-3. Chain-of-Thought(COT) Prompt**

* Using  **Chain-of-Thought(COT)** Prompting to our Base Model
* Chain-of-Thought (CoT) prompting encourages the model to explicitly break down the reasoning process step-by-step, rather than jumping straight to an answer. This is particularly useful for complex tasks that involve multi-step reasoning. By **prompting the model to reason** through the problem, CoT helps the model provide more accurate and logically coherent answers, especially for tasks that require intermediate steps, like math problems(like this case) or logical deductions.

**[Input : Chain-of-Thought(COT) Prompt]**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**[Output]**

**텍스트, 폰트, 스크린샷, 문서이(가) 표시된 사진

자동 생성된 설명**

* **Model attempts to solve the problem step by step in response to the question**
* **Do not give a proper answer to the given complex question.**
* **Closer to the correct answer compared to the few-shot prompt approach**

**3-1. [Instruction fine-tuned model]**

**- “meta-llama/Llama-2-7b-chat-hf”**

**3-2-1. Instruction fine-tuned model**

**[Input : Complex Quetion]**

**텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**[Output]**

**텍스트, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명**

* **Model not only attempts to generate an answer but also tries to provide an explanation**
* **However, it failed to deliver the desired exact answer**
* **Repetition does not occur**

**3-2-2. Few-Shot Prompting**

**[Input : Chain-of-Thought(COT) Prompt]**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**[Output] 텍스트, 폰트, 대수학, 영수증이(가) 표시된 사진

자동 생성된 설명**

* **Model finally provided the correct answer!!!**
* **Additionally, in the Few Shot 3 example provided as input to the model, a detailed explanation was omitted in the answer**
* **Similarly, the model was able to generate the answer while providing only the necessary explanation**

**3-2-3. Chain-of-Thought(COT) Prompt**

**[Input : Chain-of-Thought(COT) Prompt]**

**텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명**

**[Output]**

**텍스트, 폰트, 문서, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

* **Also, Model finally provided the correct answer!!!**
* **Compared to other prompts, it shows significantly better results**
* **Model accurately understood the intent of the question and generated the most appropriate answer, step by step, along with explanations as intended.**

**4. [Compare result with Base model and Instruction fine-tuned model]**

**Base Model**

* Base models are pre-trained on large datasets that cover a broad range of topics and tasks but are not specifically fine-tuned for particular instructions or objectives
* Output might not always align with the user’s specific needs or instructions
* Since they are not instruction-tuned, their responses may lack clarity, precision, or relevance to a particular task

**Instruction Fine-Tuned Model**

* Models are better at interpreting complex instructions and providing more relevant and focused outputs
* With instruction tuning, the model can provide more accurate, contextually appropriate, and consistent responses compared to a base model

**For commercial applications, instruction fine-tuned models are generally more suitable**

* Instruction-tuned models offer more user-friendly and context-aware outputs, enhancing the overall user experience
* In tasks where efficiency is essential, instruction fine-tuning enables the model to follow directions more effectively, reducing the need for extensive post-processing or human intervention.

**5. [Truble Shooting]**

* **Import Llama2 Model :**We cannot use Llama2 the same way as GPT-2 (an open model). To use Llama2, we need to log in to Hugging Face, obtain a token, and use that token to import the Llama2 model.
* **Lack of Memory :**When running the Llama2 7B model, our GPU and RAM may not have enough memory to handle it. By setting torch\_dtype=torch.float16, we can utilize the model efficiently on the given hardware, as it reduces the memory requirements.