

Natural Language Processing: HW3 report

Taewook Kang

Department of Computer Science, Hanyang university (Student ID: 2021025205).

1. Introduction

As language models advance, optimizing decoding strategies to generate high-quality sentences and designing prompts to elicit accurate responses are becoming increasingly important. In the first part of this report, five different decoding strategies were tested with the LLaMA-3.2-1B model to complete three types of unfinished sentences: factual, inferential, and emotional. The results were analyzed for coherence and repetitiveness – key challenges of decoding tactics – to highlight the characteristics of each strategy. In second part of this report, zero-shot, few-shot, and chain-of-thought (one-shot and zero-shot) prompting methods were compared by evaluating the model's performance on simple arithmetical questions to demonstrate the effectiveness of each approach. Finally, comparisons were made between the base model and the instruction-tuned model of LLaMA 3.2 to observe the impact of instruction fine-tuning.

2. Experiments

2.1. Environments and model information

All experiments are conducted on Google Colab with an NVIDIA T4 GPU (15GB VRAM). I used LLaMA 3.2 (released in September 2024), the latest version of Meta's open-source language model LLaMA [1], known for its powerful capabilities. I specifically used LLaMA-3.2-1B-base and LLaMA-3.2-1B-Instruct models, to load both simultaneously on 15GB VRAM GPU for convenient evaluations. Both models and tokenizers are loaded via Hugging Face interface.

2.2. Decoding strategies

To compare decoding strategies, I selected five unique methods: greedy search, beam-search decoding, top-p sampling [2], contrastive search [3], and DoLa [4] decoding.

Greedy search. Greedy search predicts the next token by selecting the one with the highest probability at each step, without considering alternative paths. This approach is simple and efficient, but it can sometimes lead to less diverse or repetitive outputs.

Beam-search decoding. Beam-search decoding selects k tokens at each step, where k denotes the beam width. For each

```
===== Factual information =====
Prompt      : The Eiffel Tower is located in
Gold sentence : The Eiffel Tower is located in Paris, France.

Greedy search      : The Eiffel Tower is located in the 7th arrondissement of Paris,
Beam-search decoding : The Eiffel Tower is located in the 7th arrondissement of Paris,
Top-p sampling     : The Eiffel Tower is located in Paris, France.
Contrastive search  : The Eiffel Tower is located in the 7th arrondissement of Paris,
DoLa decoding      : The Eiffel Tower is located in the 7th arrondissement of Paris,

===== Inference-based explanation =====
Prompt      : When it rains, people take
Gold sentence : When it rains, people take umbrellas.

Greedy search      : When it rains, people take shelter under their cars, but the same doesn't
Beam-search decoding : When it rains, people take shelter in their homes.
Top-p sampling     : When it rains, people take shelter under a roof.
Contrastive search  : When it rains, people take shelter in the open, and when it rains,
DoLa decoding      : When it rains, people take cover in the basement.

===== Emotional expression =====
Prompt      : When he won the prize, he felt
Gold sentence : When he won the prize, he felt proud.

Greedy search      : When he won the prize, he felt like he was the best person in the world.
Beam-search decoding : When he won the prize, he felt like he had won the lottery.
Top-p sampling     : When he won the prize, he felt he had won the lottery.
Contrastive search  : When he won the prize, he felt like he had been given a gift.
DoLa decoding      : When he won the prize, he felt like a new man.
```

[Figure 1] Outputs of each decoding strategy

candidate path, it iteratively evaluates the next set of probable tokens, expanding each path with the highest-probability options. At the final step, it selects the path with the highest overall probability as the final sentence. This method can create more varied outputs than Greedy search; however, it may still produce repetitive outputs that lack a natural flow.

Top-p sampling. Top-p sampling [2] samples the next token from a subset created by selecting tokens in descending order of probability until their cumulative probability meets or exceeds a threshold p . It can reduce repetition by randomly sampling tokens from a carefully selected pool. Nevertheless, if the hyperparameter p is set incorrectly, the semantic coherence of the generated text is compromised as this method relies on a stochastic process. Also, finding an ideal p value that avoids the drawbacks of greedy search and sampling is difficult.

Contrastive search. Contrastive search [3] selects the next token by considering both the predicted probability like in greedy search to maintain semantic coherence, and its similarity to the preceding context to prevent model degeneration. By balancing sentence coherence and variation, this approach can generate diverse sentences while effectively preserving their overall meaning. However, since it must calculate the cosine similarity between each candidate token and every token in the context, it has high computational requirements.

DoLa decoding. DoLa decoding [4] is a contrastive decoding strategy to enhance factual accuracy and reduce hallucinations in LLMs. DoLa works by emphasizing factual knowledge through contrasting the differences in logits obtained from

```
----- Zero-shot prompting -----
Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?
A. 1
B. 2
C. 3
D. 4
Answer: B

----- Few-shot prompting -----
Question: Alice bought 4 boxes of oranges, each containing 6 oranges. She ate 5 of them. How many oranges does she have now?
Answer: 19
Question: Tom bought 2 packs of candies, each with 8 candies. He gave 3 to his sister. How many candies does he have now?
Answer: 13
Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?
Answer: 17
Question: John bought 3 packs of oranges, each containing 8 oranges. He gave 5 to his friend. How many oranges does he have now?
Answer: 13
```

```
----- Chain-of-thought -----
Question: Alice bought 4 boxes of oranges, each containing 6 oranges. She ate 5 of them. How many oranges does she have now?
Solution:
1. Calculate the total number of oranges Alice bought: 4 * 6 = 24
2. Subtract the number of oranges Alice ate: 24 - 5 = 19
3. The answer is the remaining number of oranges: 19
Answer: 19

Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?
Solution:
1. Calculate the total number of apples Bob bought: 3 * 5 = 15
2. Subtract the number of apples Bob ate: 15 - 4 = 11
3. The answer is the remaining number of apples: 11
Answer: 11

----- Chain-of-thought (zero-shot) -----
Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?

Let's think step by step.
1. Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them.
2. Bob ate 4 apples, so he has 3 apples left.
3. Bob has 3 apples left, so he has 3 apples.
4. Bob has 3 apples, so he has 3 apples.
5. Bob has 3 apples, so he has 3 apples.
6. Bob has 3 apples, so he has 3 apples.
```

[Figure 2] Zero-shot and few-shot prompting with a base model

specific intervals between layers in the model, helping to reduce inaccuracies and enhance the reliability of generate responses. However, its performance may be limited in creative generation tasks as it primarily focuses on conveying accurate information. Additionally, it may be less effective when applied to smaller models, such as GPT-2 [4, 5].

For analyzing each decoding methods across diverse tasks, I created unfinished sentences in three categories: factual information, inference-based explanation, emotional expression. These sentences are presented in table 1.

Factual information	The Eiffel Tower is located in
Inference-based explanation	When it rains, people take
Emotional expression	When he won the prize, he felt

[Table 1] Three unfinished sentences for different objectives

The number of newly generated tokens was limited to 10, focusing only on completing the given unfinished sentences (i.e. ignored any text generated after the first punctuation mark).

The sentences generated through each decoding strategy are shown in figure 1. All strategies generated accurate sentences when dealing with factual information, as the topic involved well-known landmarks and straightforward details. When continuing inference-based explanations, beam-search decoding and top-p sampling produced the most coherent and least repetitive sentences compared to other methods. Greedy search, in detail, generated a sentence about taking shelter under cars, which is an unnatural choice for sheltering. Also, contrastive search showed the repetitive phrases in the sentence, and DoLa decoding produced a coherent but somewhat unnatural sentence, as people typically do not go to the basement during regular rain. Most strategies generated coherent sentences expressing joy, surprise, and pride in the given unfinished sentence, while DoLa did not. This may be due to its tendency to prioritize precise information over directly expressing emotions, and small models may not be well-suited for the decoding approach.

In summary, beam-search decoding and top-p sampling produced the best decoding results. Greedy search produces incoherent sentences, highlighting its weakness. Advanced methods like contrastive search and DoLa generated sentences

[Figure 3] Chain-of-thoughts (one & zero-shot) with a base model

that were less effective than traditional methods for simpler sentences requiring straightforward inference.

2.3. Prompting

In this section, I compared zero-shot prompting, few-shot prompting, and chain-of-thought prompting by asking the model to solve a simple math question. Specifically, I provided two examples for a few-shot prompt and evaluated cases where the chain-of-thought approach was used with one-shot and zero-shot prompting. The prompt shown in prompt 1 was used as a zero-shot question, serving as the basis for all other methods.

```
Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?
```

[Prompt 1] Question (zero-shot) prompting

Specifically, the following prompt was used as a few-shot prompt.

```
Question: Alice bought 4 boxes of oranges, each containing 6 oranges. She ate 5 of them. How many oranges does she have now?
Answer: 19

Question: Tom bought 2 packs of candies, each with 8 candies. He gave 3 to his sister. How many candies does he have now?
Answer: 13

Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?
```

[Prompt 2] Few-shot prompting

For chain-of-thought, the following prompts was used.

```
Question: Alice bought 4 boxes of oranges, each containing 6 oranges. She ate 5 of them. How many oranges does she have now?
Solution:
```

----- Zero-shot prompting -----

Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?
Answer: 25

Answer: 25

Explanation: Bob bought 3 packs of apples, each containing 5 apples, so he bought a total of $3 \times 5 = 15$ apples. He ate 4 of them, so he has $15 - 4 = 11$ apples left. The answer is 11.

Note: This question is a classic example of a "laboratory experiment" where the variables are Bob's eating and the initial quantity of apples. The question assumes that Bob's eating

----- Few-shot prompting -----

Question: Alice bought 4 boxes of oranges, each containing 6 oranges. She ate 5 of them. How many oranges does she have now?
Answer: 19

Question: Tom bought 2 packs of candies, each with 8 candies. He gave 3 to his sister. How many candies does he have now?
Answer: 13

Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?
Answer: 11

Step 1: Calculate the total number of oranges Alice bought
Alice bought 4 boxes of oranges, each containing 6 oranges. So, the total number of oranges she bought is $4 \times 6 = 24$.

Step 2: Calculate the number of oranges Alice has left after eating 5
Alice ate 5 oranges out of the 24 she bought. So, the number of oranges she has left is $24 - 5 = 19$.

Step 3: Calculate the total number of candies Tom bought
Tom bought 2 packs of candies, each with 8 candies. So, the total number of candies he bought is $2 \times 8 = 16$.

Step 4: Calculate the number of candies Tom has left after giving 3 to his sister
Tom gave 3 candies to his sister. So, the number of candies he has left is $16 - 3 = 13$.

Step 5: Calculate the total number of apples Tom bought
Tom bought 3 packs of apples, each with 5 apples. So, the total number of apples he bought is $3 \times 5 = 15$.

Step 6: Calculate the number of apples Tom has left after eating 4
Tom ate 4 apples out of the 15 he bought. So, the number of apples he has left is $15 - 4 = 11$.

The final answer is: $\boxed{11}$

[Figure 4] Zero-shot and few-shot prompting with an instruct model

1. Calculate the total number of oranges Alice bought: $4 \times 6 = 24$
2. Subtract the number of oranges Alice ate: $24 - 5 = 19$
3. The answer is the remaining number of oranges: 19

Answer: 19

Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?

[Prompt 3] One-shot chain-of-thought prompting

Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?

Let's think step-by-step.

[Prompt 4] Zero-shot chain-of-thought prompting

All outputs were decoded using the base model of LLaMA 3.2 with top-p sampling [2], as this combination demonstrated the most robust performance among the decoding strategies discussed in section 2.2. The maximum number of new tokens was limited to 100 for both zero-shot and few-shot prompts, and 400 for chain-of-thought prompts.

As shown in figure 2, neither zero-shot nor few-shot prompts failed to answer the question correctly. In the zero-shot prompt, the model generated its own answer choices and selected one, but none of the options were correct. The few-shot prompt succeeded in adhering to the format but failed to produce the correct answer and generated unnecessary additional text. In

----- Chain-of-thought -----

Question: Alice bought 4 boxes of oranges, each containing 6 oranges. She ate 5 of them. How many oranges does she have now?

Solution:

1. Calculate the total number of oranges Alice bought: $4 \times 6 = 24$

2. Subtract the number of oranges Alice ate: $24 - 5 = 19$

3. The answer is the remaining number of oranges: 19

Answer: 19

Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?

Solution:

1. Calculate the total number of apples Bob bought: $3 \times 5 = 15$

2. Subtract the number of apples Bob ate: $15 - 4 = 11$

3. The answer is the remaining number of apples: 11

Answer: 11

Question: Emma bought 2 boxes of grapes, each containing 8 grapes. She ate 3 of them. How many grapes does she have now?

Solution:

1. Calculate the total number of

----- Chain-of-thought (zero-shot) -----

Question: Bob bought 3 packs of apples, each containing 5 apples. He ate 4 of them. How many apples does he have now?

Let's think step by step.

Step 1: Bob bought 3 packs of apples, and each pack contains 5 apples. So, he bought a total of $3 \times 5 = 15$ apples.

Step 2: After eating 4 apples, Bob has $15 - 4 = 11$ apples left.

Step 3: Since each apple is 1 unit, Bob now has 11 apples.

Answer: Bob has 11 apples.

[Figure 5] Chain-of-thoughts (one & zero shot) w. instruct model

figure 3, the answers generated by the chain-of-thought prompts are presented. The zero-shot chain-of-thought prompt successfully induced a step-by-step reasoning process, but its content and logic were incorrect, ultimately failing to achieve the goal. The one-shot chain-of-thought prompt arrived at the correct answer by following the reasoning process shown in its example. However, it also produced extraneous text.

As a result, only the one-shot chain-of-thought prompt succeeded producing the correct answer. Zero-shot and one-shot prompts failed to demonstrate the process needed to reach the correct answer and ended up generating content unrelated to the correct answer. The zero-shot chain-of-thought attempted to guide the reasoning process, but the process itself was flawed, resulting in an incorrect answer.

2.4. Instruction tuning

In this section, math question answering was performed under the same conditions as section 2.3., but using the instruct model instead of the base model. Figure 4 and 5 show the generation results of each prompt. Unlike the base model, the instruct

model produced the correct answer for all prompts. Specifically, the instruct model analyzed the logical flow of each example in the few-shot prompt and applied it to the question. In the one-shot chain-of-thought prompt, the model effectively utilized the logical flow of the example, and in the zero-shot prompt, it not only followed a step-by-step approach but also applied appropriate logic at each step, successfully arriving at the correct answer. Consequently, the instruct model demonstrated an outperforming ability to perform logical thinking, in contrast to the base model.

3. Conclusion

For the task of completing simple unfinished sentences with a relatively lightweight model, top-p sampling and beam search decoding generated more polished sentences compared to the more complex methods of contrastive search and DoLa decoding. As interest grows in using LLMs on devices like mobile phones and given that users on these platforms are likely to request simpler tasks, it is worth exploring methods that perform well in these specialized environments.

In comparing different prompting methods with the base model, the chain-of-thought approach, especially in the one-shot setting, produced the most accurate answers. However, since providing examples of logical reasoning is often challenging in real-world settings, it can be concluded that expecting appropriate responses to logical questions from the base model is difficult. When comparing the base model and the instruct model, the latter demonstrated significantly superior performance across all prompt types. This shows that instruction fine-tuning plays a major role in dramatically enhancing language models' capabilities on complex tasks.

References

- [1] Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." *arXiv preprint arXiv:2302.13971* (2023).
- [2] Holtzman, Ari, et al. "The curious case of neural text degeneration." *arXiv preprint arXiv:1904.09751* (2019).
- [3] Su, Yixuan, et al. "A contrastive framework for neural text generation." *Advances in Neural Information Processing Systems* 35 (2022): 21548-21561.
- [4] Chuang, Yung-Sung, et al. "Dola: Decoding by contrasting layers improves factuality in large language models." *arXiv preprint arXiv:2309.03883* (2023).
- [5] Radford, Alec, et al. "Language models are unsupervised multitask learners." *OpenAI blog* 1.8 (2019): 9.