Assignment 3 Report
CS-726: Advanced Machine Learning

Deeptanshu Malu Deevyanshu Malu Neel Rambhia

1 Task 0

The model used is `meta-llama/Llama-2-7b-hf`.

Decoding Strategy	**BLEU** (↑)	**ROUGE-1** (↑)	**ROUGE-2** (↑)	**ROUGE-LCS** (↑)
Greedy	**0.3097**	**0.3538**	**0.1297**	**0.2704**
Random (temp=0.5)	0.2856	0.2929	0.1113	0.2387
Random (temp=0.9)	0.1996	0.1791	0.0550	0.1477
Top-k (k=5)	0.2388	0.2230	0.0607	0.1715
Top-k (k=10)	0.2427	0.2491	0.0799	0.2004
Nucleus Sampling (p=0.5)	0.2706	0.2554	0.0905	0.1973
Nucleus Sampling (p=0.9)	0.1957	0.1883	0.0469	0.1329

Table 1: Evaluation metrics for different decoding strategies.

Greedy decoding performs best across all metrics. On adding more randomness (higher temperature or larger sampling pools), the performance gets worse. Among sampling methods, Random with temperature=0.5 and Nucleus Sampling with p=0.5 work better than others. This tells us that some randomness can help, but too much hurts performance.

2 Task 1

The model used is `meta-llama/Llama-2-7b-chat-hf`.

We implemented constrained decoding using a trie data structure to limit the model's output to a predefined vocabulary. The trie is created by first tokenizing the target vocabulary and then inserting each token into the trie. We have also added the end-of-sequence (EOS) token to the root of the trie. The algorithm selects the highest probability token that is valid according to the trie at each step.

We tested two trie traversal strategies: resetting the trie traversal when reaching an end node (is_end) and resetting when a node has no children. The second strategy performed slightly better, as shown in the results table below.

Decoding Strategy	**BLEU** (↑)	**ROUGE-1** (↑)	**ROUGE-2** (↑)	**ROUGE-LCS** (↑)
Regular greedy	0.2781	0.3351	0.1230	0.2640
Reset when is_end	0.5111	0.5270	0.3189	**0.4694**
Reset when no children	**0.5116**	**0.5307**	**0.3205**	0.4675

Table 2: Evaluation metrics for constrained decoding strategies.

Algorithm 1 Constrained Decoding using Trie

```
Build trie from target vocabulary (word list)
Add EOS token to trie
node ← trie.root
for i = 1 to max_output_length do
    Compute logits for the last token in the sequence
    Convert logits to probabilities using softmax
    Sort probabilities in descending order
    for each token t in sorted order do
        if t is a child of current node then
            Select t as the next token
            node ← node.children[t]
            if node has no children then          ▷ Can be replaced with node.is_end condition
                node ← trie.root                  ▷ Reset trie state
            end if
            break
        end if
    end for
    Append selected token to the output sequence
    if selected token is EOS then
        break
    end if
end for
return generated sequence
```

3 Task 2

The model used is `FasterDecoding/medusa-v1.0-vicuna-7b-v1.5`.

3.1 Single-head decoding

For single-head Medusa decoding, we evaluated the model using greedy decoding strategy:

BLEU	**ROUGE-1**	**ROUGE-2**	**ROUGE-LCS**	**RTF**
0.2921	0.3963	0.1483	0.3177	0.0573

Table 3: Evaluation metrics for single-head Medusa decoding.

3.2 Multi-head decoding

When the EOS token appears at the end of a candidate sequence then we don't append any more tokens to it. The results for different beam widths (W) and numbers of Medusa heads (S) are shown below:

W	**S**	**BLEU** (↑)	**ROUGE-1** (↑)	**ROUGE-2** (↑)	**ROUGE-LCS** (↑)	**RTF** (↓)
2	2	0.1595	0.1700	0.0324	0.1429	0.0581
5	2	0.1849	0.1767	0.0433	0.1382	0.1288
10	2	**0.2788**	**0.2375**	**0.0695**	**0.1822**	0.2396
2	5	0.0760	0.0965	0.0066	0.0827	**0.0345**

5	5	0.0425	0.0799	0.0138	0.0682	0.0714
10	5	0.0077	0.0250	0.0017	0.0219	0.1691

Table 4: Evaluation metrics for multi-head Medusa decoding with different beam widths and number of heads.

Key observations from the multi-head decoding results:

- The best performance was achieved with W=10 and S=2.
- Increasing the number of heads (S) did not improve performance, and in fact, hurt it.
- When the number of heads was 2, the BLEU and ROUGE scores improved with increasing beam width.
- When the number of heads was increased to 5, the BLEU and ROUGE scores dropped with increasing beam width.
- With constant number of Medusa heads, increasing beam width increased the RTF.
- With constant beam width, increasing the number of Medusa heads decreased the RTF.

Contributions

- **Deeptanshu Malu**:
  1. Formulated the trie data structure design for Task 1.
  2. Coded the multi head decoding algorithm for Task 2.
- **Deevyanshu Malu**:
  1. Coded the trie data structure and the constrained decoding algorithm for Task 1.
  2. Coded the multi head decoding algorithm for Task 2.
- **Neel Rambhia**:
  1. Completed all decoding algorithms in Task 0.
  2. Coded the single head decoding algorithm for Task 2.

Acknowledgements

- We have also used Copilot for faster coding and not for direct logic.
- Used ChatGPT to generate the trie data structure but filled in the logic ourselves.