CAP 6640 – Natural Language Processing **Spring 2025**
Homework #6 Due date: April 4, 2025

Problem 1:

The softmax temperature is a parameter used to adjust how a language model selects words during text generation. It changes the probability distribution of possible next words, allowing us to tune the balance between randomness and predictability in the output. Given a vector of logits z_i for each possible token, the temperature scaled softmax is defined as:

$$P(w_i) = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$$

where T is the temperature, z_i is the pre-softmax score for token i, and $P(w_i)$ is the resulting probability for token i.

When the temperature is close to 0.7 or below, the model becomes more confident and focused. It strongly favors high-probability words, resulting in precise, factual, and consistent outputs. This setting is well-suited for tasks where accuracy and coherence are important, such as summarization, and QA.

In contrast, a higher temperature which can be selected as 1.2 or more makes the output distribution flatter, giving more weight to less likely words. This leads to more creative, varied, and sometimes unexpected text, which is ideal for poetry, storytelling, or brainstorming tasks. However, very high temperatures can also introduce nonsense or off-topic responses.

Problem 2:

Sampling-based decoding is a decoding strategy used in language generation where the model randomly selects the next word based on its probability, rather than always choosing the most likely one. This makes it especially suitable for creative tasks like storytelling, where unpredictability and variation are essential for producing engaging content.

Sampling-based decoding offers several benefits for open-ended text generation tasks like storytelling. By introducing randomness, sampling helps the model avoid repetitive or generic responses, creating outputs that feel more human-like and less mechanical. Furthermore, it allows for exploration of diverse narrative paths by occasionally selecting less frequent but reasonable words. Lastly, tecniques like top-n sampling where the model samples only from the top n probable words, or top-p sampling, where the model samples from the smallest set of words whose cumulative probability exceeds p, allow users to control the level of creativity and coherence of the model in the generated text.

However, sampling-based decoding also has limitations. Random selection increases the chance of the model producing disconnected or nonsensical sentences. Moreover, the model can produce very different outputs for the same set of inputs, making it harder to reproduce specific responses or maintain a consistent storytelling voice. Finally, sampling can sometimes deviate from the intended direction or tone of the story.

Problem 3:

BLEU, ROUGE, and BERTScore are three commonly used metrics for evaluating NLG, particularly in tasks like machine translation, summarization, and open-ended text generation. Each has its strengths and limitations, especially when applied to creative outputs like dialogue or storytelling.

BLEU measures the degree to which the generated text matches the reference through n-gram precision. It is particularly useful for tasks with a small set of valid outputs, such as machine translation, where phrase-level correctness is critical. However, BLEU does not account for meaning. If the model uses synonyms or rephrases the sentence, BLEU may penalize it, even if the alternative is equally valid.

ROUGE is often used in summarization tasks, and focuses on more recall than BLEU. It evaluates how much of the reference content is captured in the generated summary by comparing overlapping unigrams, bigrams, or longest common subsequences. While effective for extractive summarization,

ROUGE is less suited for abstractive summaries that creatively rephrase or reorganize content. Like BLEU, it fails to consider semantic similarity or context.

BERTScore offers a more modern and semantically aware evaluation method. It compares contextualized word embeddings between the generated and reference texts, allowing it to assess meaning alignment rather than simple word overlap. This makes it particularly effective for evaluating tasks with more expressive variability, such as dialogue generation, storytelling, or creative writing, where many phrasings may be equally valid.

In open-ended text generation like storytelling, there is hardly ever a single true, correct, and acceptable answer. Word-overlap metrics such as BLEU and ROUGE penalize valid outputs simply because they differ in wording from the reference. These tasks require understanding the intention, tone, and context of a response, which BLEU and ROUGE are not designed to measure. For example, a story may use a different reorder events creatively or analogy of wording while still conveying the same narrative. In such cases, BERTScore performs better because it can recognize that the generated output shares semantic alignment with the reference even if the wording differs.

Problem 4: Describe how hybrid QA architectures integrate both retrieval-based and generative models. Provide an example of such a system.

Problem 5: Define coreference resolution. Why is it critical for full-text understanding tasks such as QA and summarization?

Problem 6: How do end-to-end neural models approach coreference resolution differently from traditional rule-based systems?

Problem 7: How does the DecaNLP framework unify multiple NLP tasks without relying on task-specific modules or parameters?

Problem 8: Describe anti-curriculum learning. How does starting with harder tasks improve generalization in multitask training?

Problem 9: What is the MQAN architecture? List two mechanisms it uses to handle multiple tasks as QA problems.

Problem 10: What is the difference between a general language model and a conditional language model? Provide examples of tasks suitable for each.