**Sequence-to-Sequence Decoding Strategies - Detailed Notes**

**Context:**

In a Sequence-to-Sequence (Seq2Seq) model such as for Machine Translation (e.g., English to Hindi), we use an encoder-decoder setup. Given an input sequence (C) and previously generated tokens up to time step T-1 (W< T), the goal is to predict the next token WT.

This prediction is based on a probability distribution over the vocabulary generated by the decoder at each step.

**I. Types of Decoding Strategies**

**1. Greedy Decoding**

* At each time step, pick the token with the **highest probability**.
* **Pros**:
  + Simple and fast
  + Deterministic (same output every time)
  + Suitable for applications requiring **consistency** (e.g., QA)
* **Cons**:
  + **Lack of diversity**, always generates the same sequence
  + May not yield the best overall sequence

**2. Exhaustive Decoding**

* Generate **all possible sequences** up to a given length.
* **Complexity**: O(|V|n) where V = vocabulary size and n = sequence length
* **Impractical** due to high computational cost

**3. Beam Search**

* Maintain **top K most probable sequences** (called beams) at each time step
* At each step:
  + Expand all beams
  + Keep top K sequences based on cumulative probability
* **Pros**:
  + Balances between greedy and exhaustive
  + More diversity than greedy
* **Cons**:
  + Still deterministic
  + Limited diversity

**II. Deterministic vs Stochastic Strategies**

* **Deterministic**:
  + Greedy, Exhaustive, Beam Search
  + Output is consistent but lacks creativity
* **Stochastic**:
  + Introduce randomness for more **diverse** outputs
  + Examples: Top-K Sampling, Top-P (Nucleus) Sampling, Temperature Sampling

**III. Stochastic Decoding Strategies**

**1. Random Sampling**

* Sample directly from the probability distribution at each step
* **Cons**: Completely random, often leads to poor quality output

**2. Top-K Sampling**

* Keep only the **top K tokens** with highest probabilities
* Renormalize their probabilities
* Sample randomly from these K tokens

**Example**:

* Vocabulary: Cheese (0.35), Toppings (0.2), Pepperoni (0.1), ...
* Top-K (K=3): {Cheese, Toppings, Pepperoni}
* Renormalize and sample from these

**Pros**:

* Avoids very low-probability words
* Adds diversity

**Cons**:

* Fixed K may not adapt well to different distributions

**3. Top-P Sampling (Nucleus Sampling)**

* Select the **smallest set of tokens** whose cumulative probability ≥ threshold P (e.g., 0.9)
* Renormalize and sample from this subset

**Pros**:

* **Adaptive** to the distribution
* More flexible than Top-K

**Example**:

* Target cumulative probability = 0.5
* Tokens selected: Cheese (0.35) + Toppings (0.2) = 0.55

**Renormalized Probabilities** used for sampling.

**4. Temperature Sampling**

* Use a temperature hyperparameter (τ\tau) to **scale the logits** before applying softmax

P(wi)=ezi/τ∑jezj/τP(w\_i) = \frac{e^{z\_i/\tau}}{\sum\_j e^{z\_j/\tau}}

**Interpretation**:

* τ=1\tau = 1: Regular softmax
* τ<1\tau < 1: Sharper (peaky) distribution → more deterministic
* τ>1\tau > 1: Flatter distribution → more diverse, creative

**Pros**:

* Easy to tune creativity level

**Cons**:

* Needs careful tuning for specific applications

**Summary Table**

| **Strategy** | **Type** | **Diversity** | **Control** | **Suitable For** |
| --- | --- | --- | --- | --- |
| Greedy | Deterministic | Low | None | QA, Translation |
| Exhaustive | Deterministic | Very Low | Full | Not practical |
| Beam Search | Deterministic | Medium | Beam Size | MT, QA |
| Random Sampling | Stochastic | High | None | Creative tasks (poor quality) |
| Top-K Sampling | Stochastic | Medium | K | Dialogue, Poetry, Chatbots |
| Top-P Sampling | Stochastic | Medium | P | Adaptive, more robust sampling |
| Temperature | Stochastic | Flexible | Tau | Fine-grained control of diversity |

**Conclusion:**

* Choice of decoding strategy should be **application-driven**
* Deterministic for factual QA
* Stochastic for open-ended generation
* Top-P + Temperature is often a good balance for LLMs