RAG-Sequence: Top-*k* Approximation & Thorough Decoding

Wook-Shin Han

POSTECH

Contents

1. RAG-Sequence: Top-*k* Approximation

2. Thorough Decoding

1. Overview

In the Retrieval-Augmented Generation (RAG-Sequence) model, the generation of a target sequence $y = (y_1, \ldots, y_N)$ conditioned on input x involves marginalizing over a latent variable z, which denotes the retrieved document.

Exact marginal likelihood:

$$= \sum_{z} p(y, z \mid x) = \sum_{z} p(z \mid x) \cdot p(y \mid x, z)$$

$$p(y \mid x) = \sum_{z \in \mathrm{D}} p_\eta(z \mid x) \cdot p_\theta(y \mid x, z).$$

However, summing over all documents in D is intractable if D is huge (e.g., all of Wikipedia).

2. Top-k Approximation

Key Idea: Restrict the sum to the top-k most relevant documents according to $p_\eta(z \mid x)$. Then,

$$p(y \mid x) \to \sum_{z \in \text{top-}k \text{ "} \, p_\eta(\cdot|x)^{\#}} p_\eta(z \mid x) \cdot p_\theta(y \mid x, z).$$

Justification:

- Often, most probability mass of $p_\eta(z \mid x)$ lies in a small subset of documents. The retriever is
- fine-tuned to focus on these top-k.
- Empirically, it works well while remaining computationally feasible.

3. Gradient Derivation (Approx. Marginal)

The training loss is the negative log-likelihood of the *approximated* marginal probability:

$$\mathcal{L}(x,y) = \uparrow \log \Big(\sum_{z\in\mathcal{Z}} p_{\eta}(z \mid x) \cdot p_{\theta}(y \mid x,z) \Big), \quad \mathcal{Z} = \text{top-}k\big[p_{\eta}(\cdot \mid x) \big].$$

Let

$$A(z) := p_{\eta}(z \mid x) \cdot p_{\theta}(y \mid x,z).$$

Then

$$\mathcal{L}(x,y) = \uparrow \log \sum_{z\in\mathcal{Z}} A(z).$$

Gradient Derivation (continued)

Using

$$\downarrow \log \& \sum_z A(z) ' = \frac{1}{\sum_z A(z)} \sum_z \downarrow A(z),$$

we get:

$$\downarrow \mathtt{L}(x,y) = \uparrow \frac{1}{p(y \mid x)} \sum_{z \in Z} \downarrow \& p_\eta(z \mid x) \cdot p_\theta(y \mid x,z) ' .$$

By the product rule:

$$\downarrow A(z) = \downarrow \& p_\eta(z \mid x)\, p_\theta(y \mid x,z) ' = \downarrow p_\eta(z \mid x) \cdot p_\theta(y \mid x,z) + p_\eta(z \mid x) \cdot \downarrow p_\theta(y \mid x,z).$$

Thus,

$$\downarrow \mathtt{L}(x,y) = \uparrow \frac{1}{p(y \mid x)} \sum_{z \in Z} \$ \downarrow p_\eta(z \mid x) \cdot p_\theta(y \mid x,z) + p_\eta(z \mid x) \cdot \downarrow p_\theta(y \mid x,z) \% .$$

4. Remarks on Top-k

- Top-k approximation makes RAG feasible for large corpora, yet effective in practice.
- The better the retriever performance, the closer the sum over top-k is to the true marginal.
- We can train both retriever (p_η) and generator (p_θ) end-to-end.

5. RAG-Sequence Thorough Decoding

Because RAG-Sequence uses *one* document z for the entire output y, we cannot just run a single beam search that mixes documents.

Thorough Decoding Algorithm:

1. **Beam Search per Doc.** For each z_i, run beam search: $(x, z_i) \to \{y_{i,1}, y_{i,2}, \ldots\}$.
2. **Re-Evaluate.** For each candidate $y_{i,j}$, compute:

$$p(y_{i,j} \mid x) = \sum_k p_\eta(z_k \mid x) \cdot p_\theta\big(y_{i,j} \mid x, z_k\big).$$

3. **Select Best.**

$$y^* = \arg \max_{y \in \cup_i Y_{z_i}} p(y \mid x).$$

6. Thorough Decoding Example

Q: *“Who wrote The Sun Also Rises and A Farewell to Arms?”*

- z_1: Mentions only *The Sun Also Rises*. z_2: Mentions only
- *A Farewell to Arms*. z_3: Mentions both in detail.

Beam Search Results (per doc):

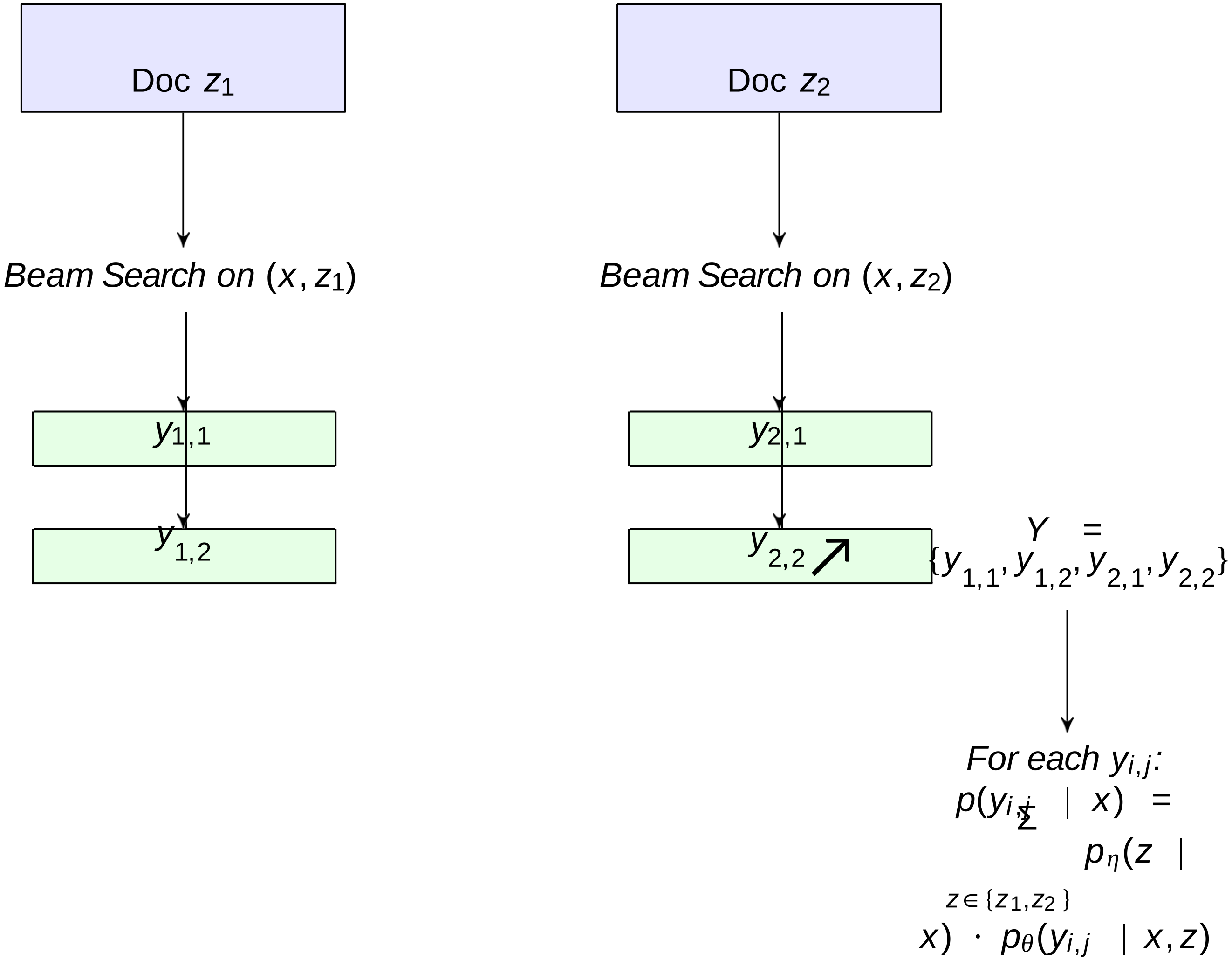
$$Y_{z_1} = \{\text{“Ernest Hemingway”, “Hemingway”}\}, \quad Y_{z_2} = \{\text{“Hemingway”, “Ernest Miller Hemingway”}\}, \ldots$$

Marginalize:

$$p(y \mid x) = \sum_{i=1}^{3} p_\eta(z_i \mid x) \cdot p_\theta(y \mid x, z_i).$$

Choose: The y with highest likelihood.

7. Visualization: Thorough Decoding



Doc z_1

Beam Search on (x, z_1)

$y_{1,1}$

$y_{1,2}$

Doc z_2

Beam Search on (x, z_2)

$y_{2,1}$

$y_{2,2}$ ↗

$Y = \{y_{1,1}, y_{1,2}, y_{2,1}, y_{2,2}\}$

For each $y_{i,j}$:

$$p(y_{i,j} \mid x) = \sum_{z\in\{z_1,z_2\}} p_\eta(z \mid x) \cdot p_\theta(y_{i,j} \mid x, z)$$

8. Observations

- Complexity (rough): Potentially $O(k^2 \searrow \text{beam\_size})$ forward passes.
- Pros: More accurate because it computes a *true* marginal (across top-k docs).
- Fast Decoding: A simpler method that avoids re-scoring sequences that never appear in each doc's beam.
- Trade-off: Thorough decoding can be slower but often yields better results; fast decoding is more scalable.

9. Why $O(k^2 \searrow \text{beam\_size})$?

Step 1: Beam Search per Document

- We have k documents.
- Each beam search produces beam_size candidate sequences.
- Total candidate sequences = $k \searrow \text{beam\_size}$.

Step 2: Re-evaluation (Marginalization)

- Each candidate $y_{i,j}$ must be re-scored under *all* k documents:

$$p(y_{i,j} \mid x) = \sum_{m=1}^{k} p_\eta(z_m \mid x)\, p_\theta\left(y_{i,j} \mid x, z_m\right).$$

- Hence, $\left(k \searrow \text{beam\_size}\right) \searrow k$ forward passes $= k^2 \searrow \text{beam\_size}$.

Overall Cost:

$$O\left(\underbrace{k \searrow (\text{beam search cost})}_{\text{Step 1}} + \underbrace{k^2 \searrow \text{beam size}}_{\text{Step 2}}\right).$$

In practice, the $k^2 \searrow \text{beam\_size}$ re-scoring dominates.