自然語言處理與應用

Natural Language Processing and Applications

Decoding Strategies

Instructor: 林英嘉 (Ying-Jia Lin)

2025/03/31

Course GitHub

Slido # NLP_0331

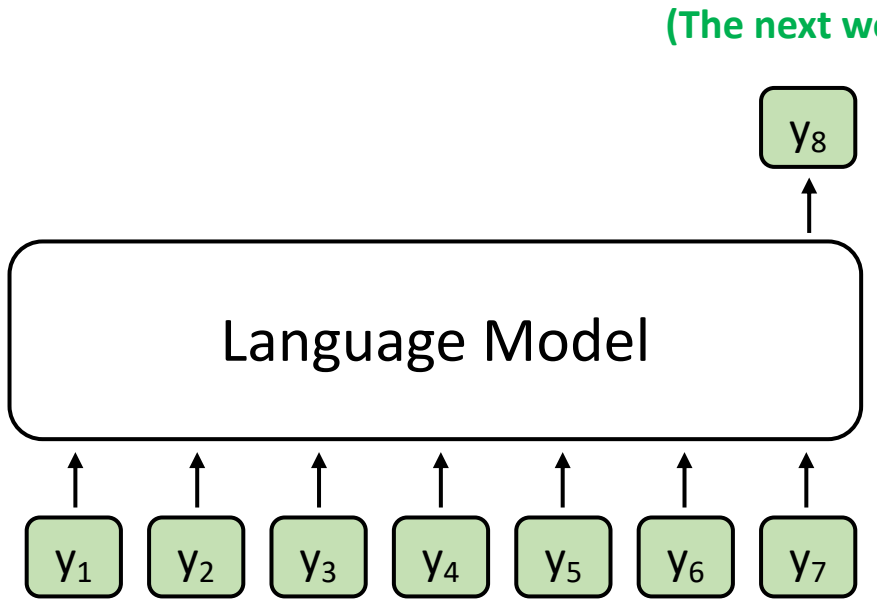
Outline

- Recap: Language Generation
- Decoding Strategies
  - Greedy Decoding
  - Beam Search
  - Top-k / Top-p Sampling

Natural Language Generation (NLG)

- Natural language generation (NLG) is a **process** that that **outputs** text.
- NLG includes a wide variety of NLP tasks.

Recap: Language Model



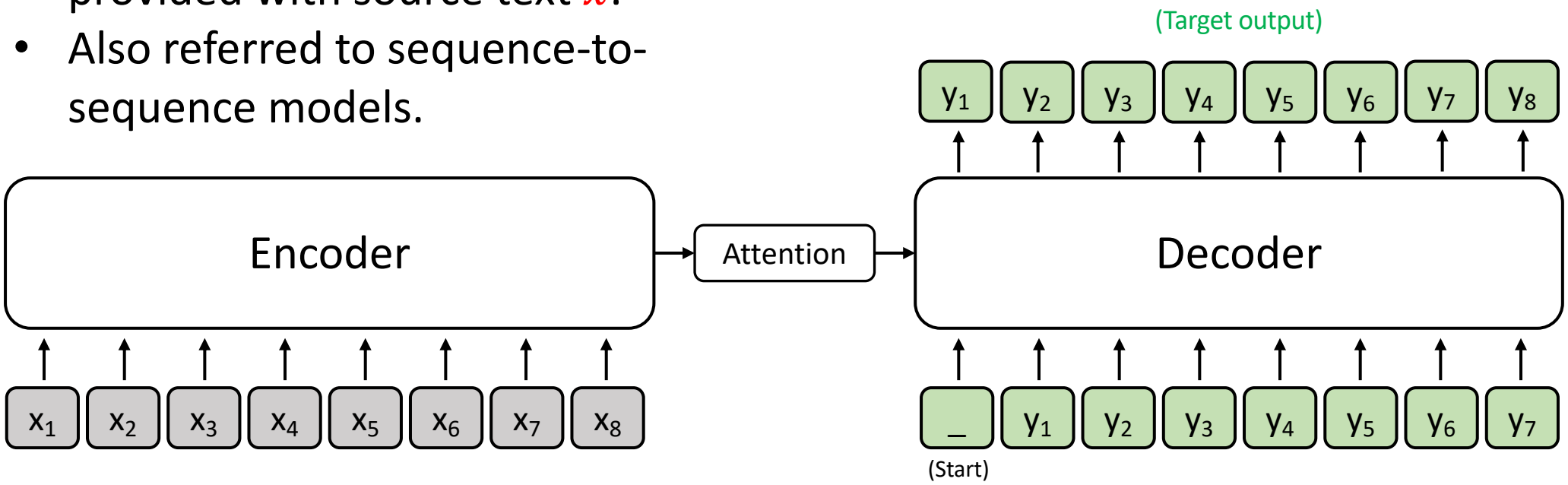
$$P(y_t|y_1, y_2, \cdots, y_{t-1})$$

- A model that assigns probabilities to upcoming words is called **a language model**.
- The task involving predictions of upcoming words is **language modeling**.

Recap: Conditional Language Model

- In addition to previous words, a conditional language model is provided with source text x.
- Also referred to sequence-to-sequence models.

$$P(y_t|y_1, y_2, \cdots, y_{t-1}, x)$$



Tasks of Conditional Language Model

- In addition to previous words (target), a conditional language model is provided with source text x.

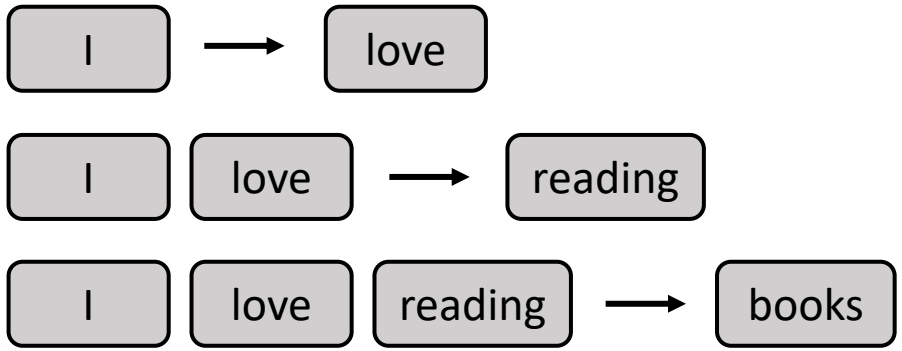
	Source	Target
Machine Translation	Language A	Language B
Summarization	Long Text	Concise Text
Dialogue Generation	User Input	Desired User Input
…		

How to train a (Conditional) Language Model?

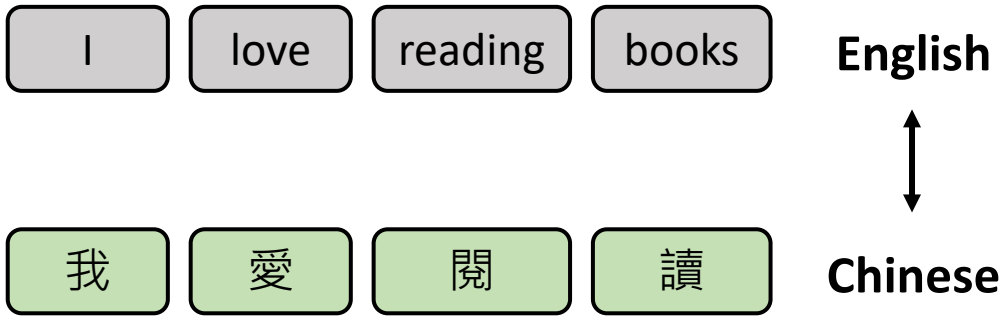
- First, you need a training corpus.

Example: I love reading books.

Language modeling (Unsupervised)

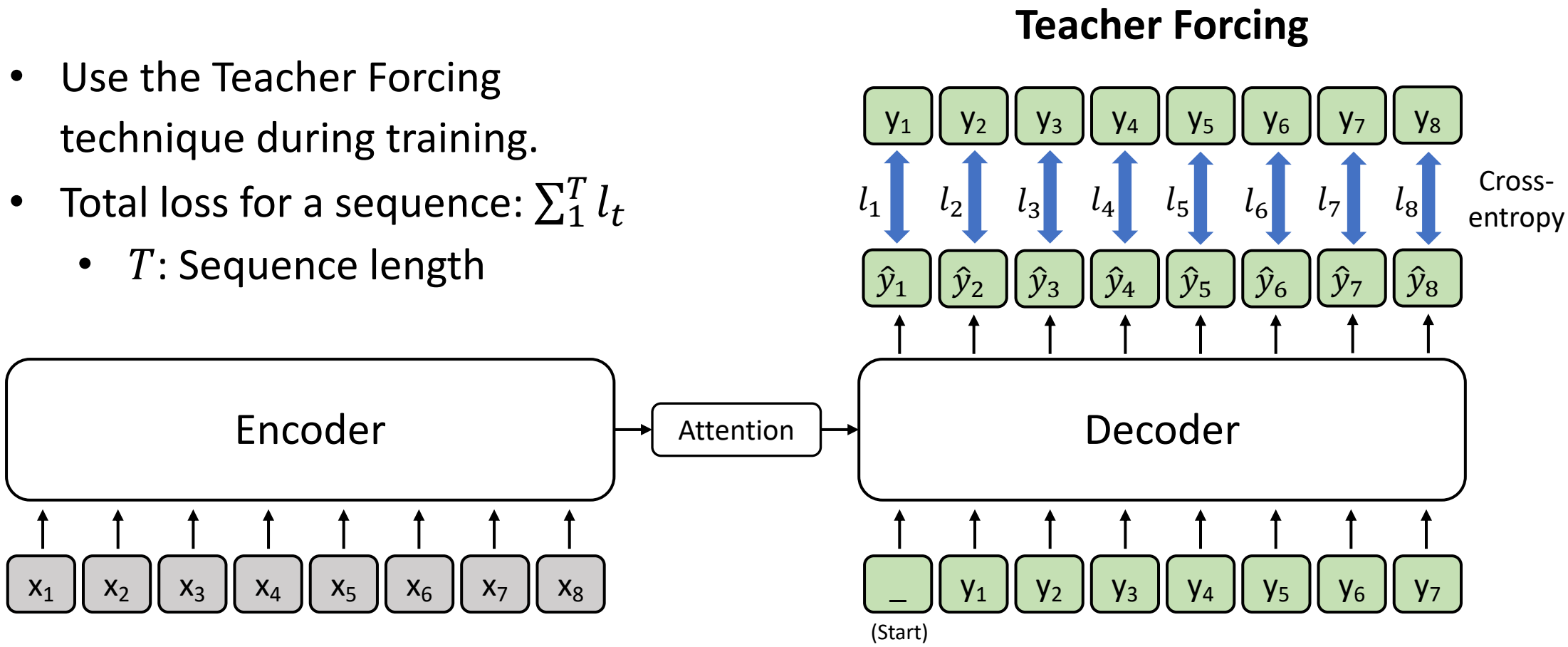


Machine Translation (Supervised)



How to train a (Conditional) Language Model?

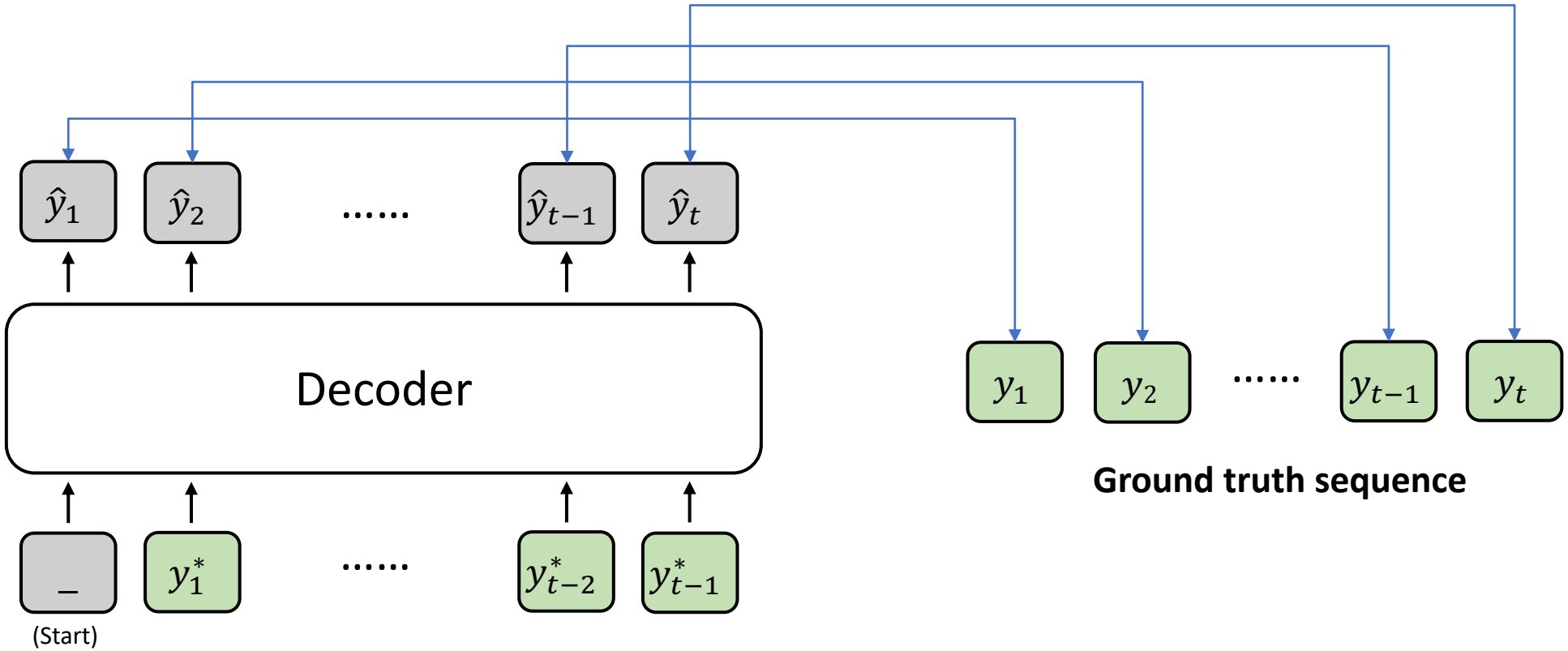
- Use the Teacher Forcing technique during training.
- Total loss for a sequence: $\sum_1^T l_t$
  - T: Sequence length



Teacher Forcing – Training Time

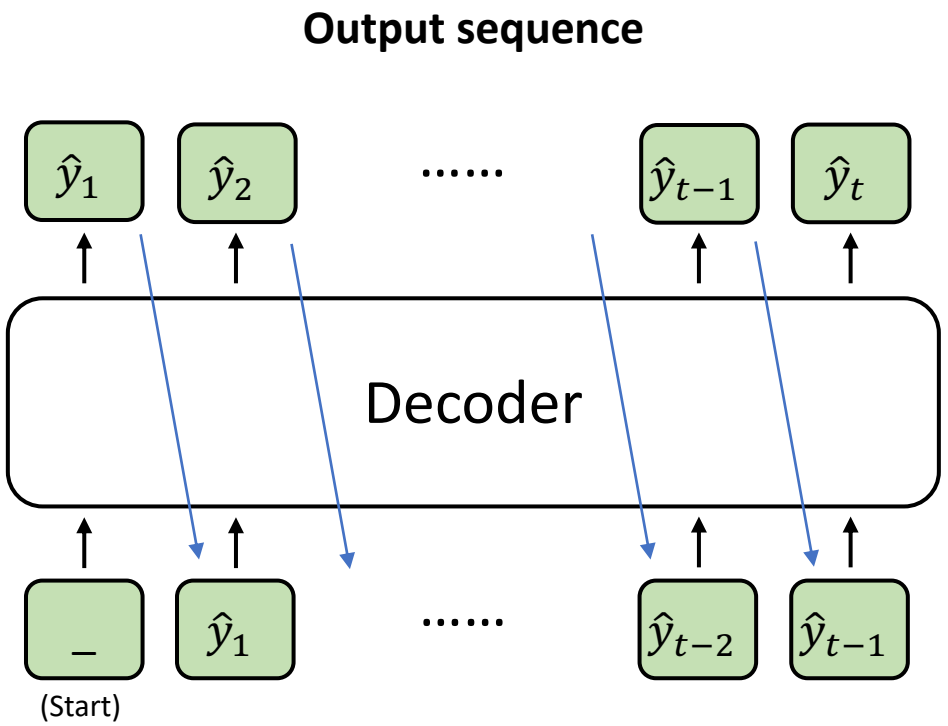
During training:

$$L_{ml} = -\sum_{t=1}^{n'} \log p(y_t | y_1, \ldots, y_{t-1}, x)$$



Teacher Forcing – Testing Time

During testing:



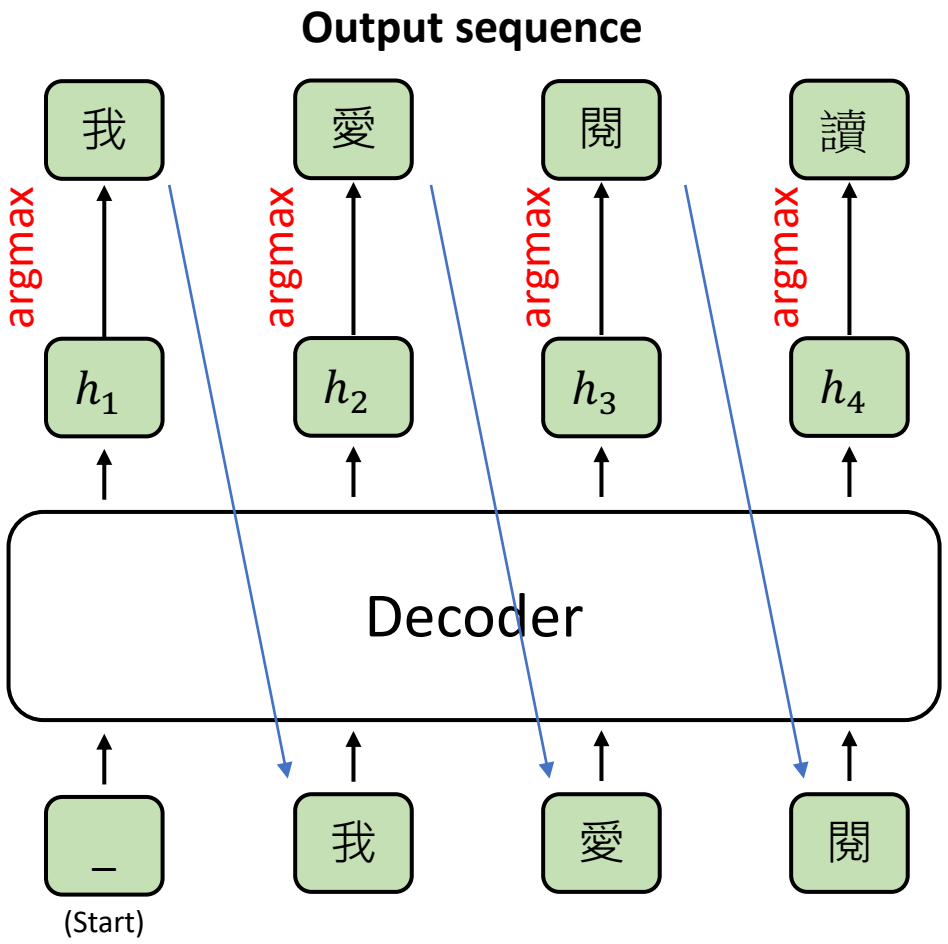
- Advantage: stabilize training and increase performance
- Question: How does the next word be determined?

Decoding Strategies

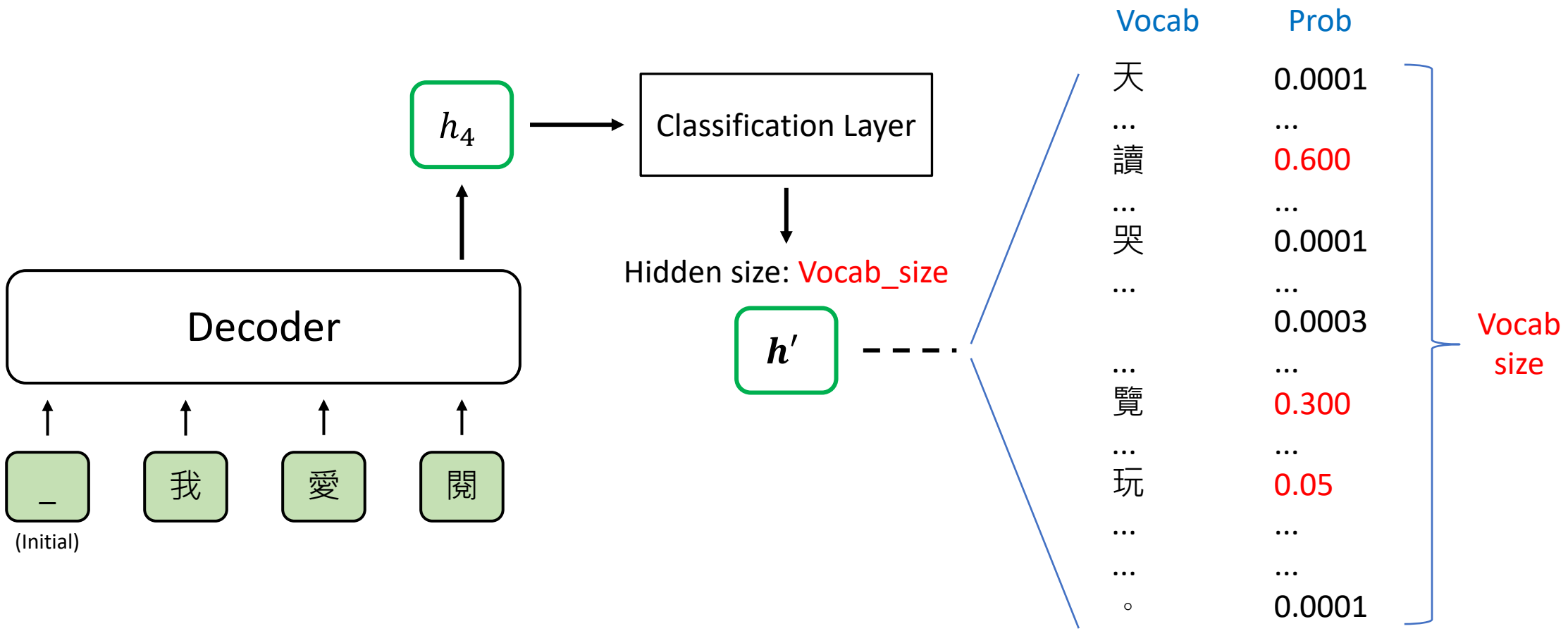
- Greedy Decoding
- Beam Search
- Top-k Sampling
- Top-p Sampling

Greedy Decoding

Example: I love reading books.

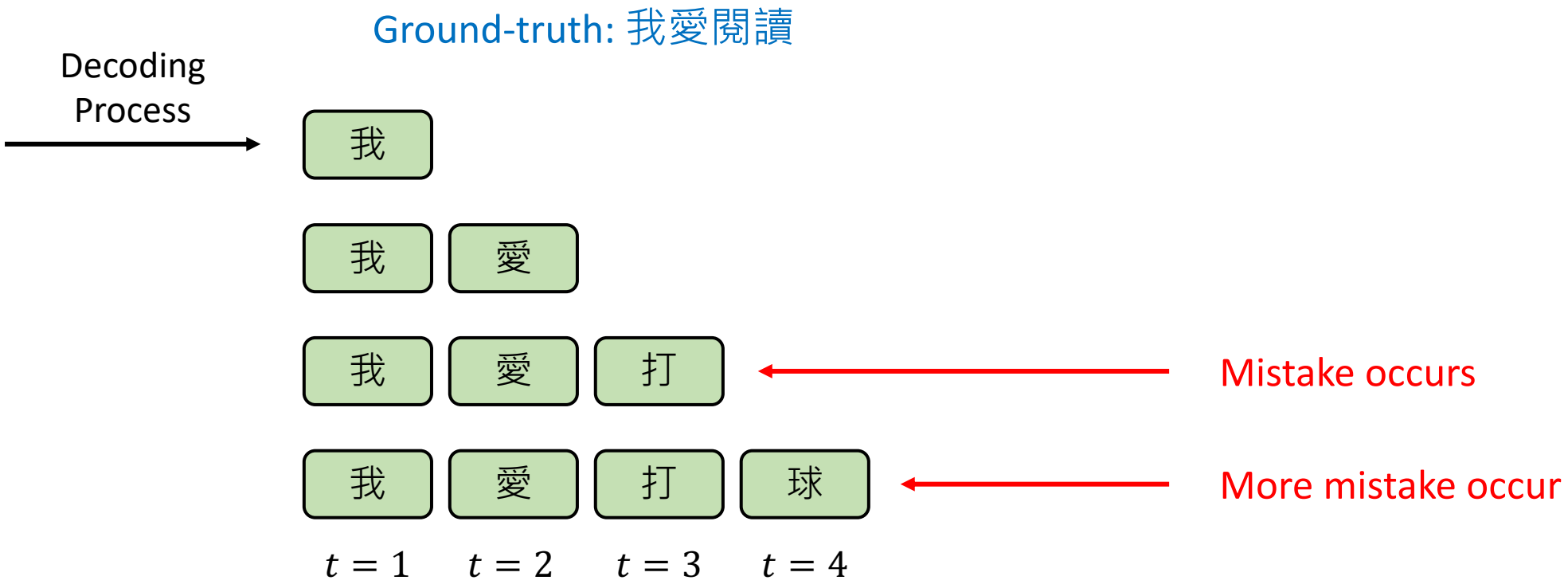


Greedy Decoding – Best Selection Process



Problem of Greedy Decoding

- Greedy decoding cannot undo!



Re-thinking Greedy Decoding

- Greedy decoding cannot undo!
- Greedy decoding only provides one best choice at each time step.
- How about providing more than one choices at each time step?

Beam Search

Beam Search

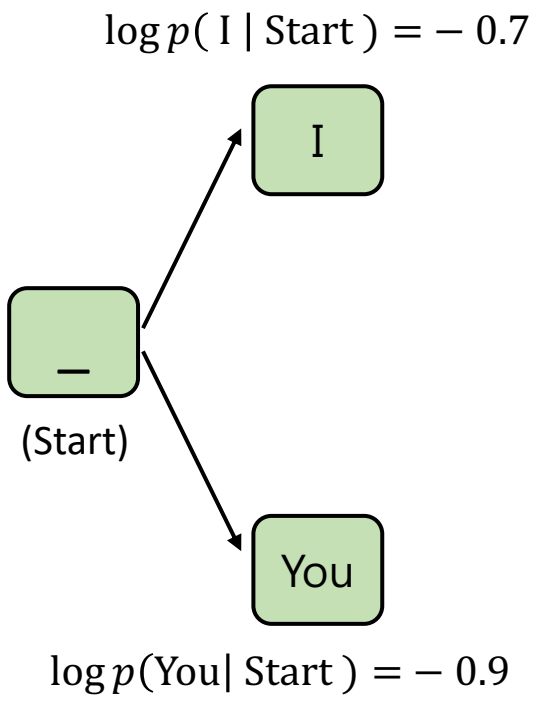
- Set the `Beam size` (or `Beam width`) = 2
  - This means that the number of candidates will be preserved at each decoding time.
  - Beam size is a hyperparameter for beam search decoding.
- At each decoding time step, a score is calculated via the following equation:

$$L_{ml} = -\sum_{t=1}^{n'} \log p(y_t^* | y_1^*, \ldots, y_{t-1}^*, x)$$

*代表 (強調) 當前時間點生成機率最大的選項

Beam Search ($t = 1$)

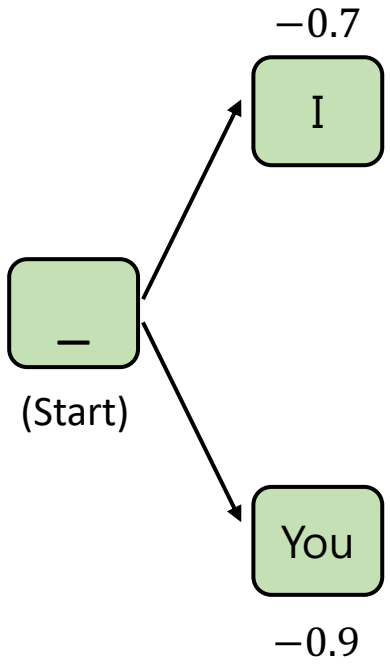
`Beam size` = 2



- At this decoding step, two choices are preserved.

Beam Search ($t = 1$)

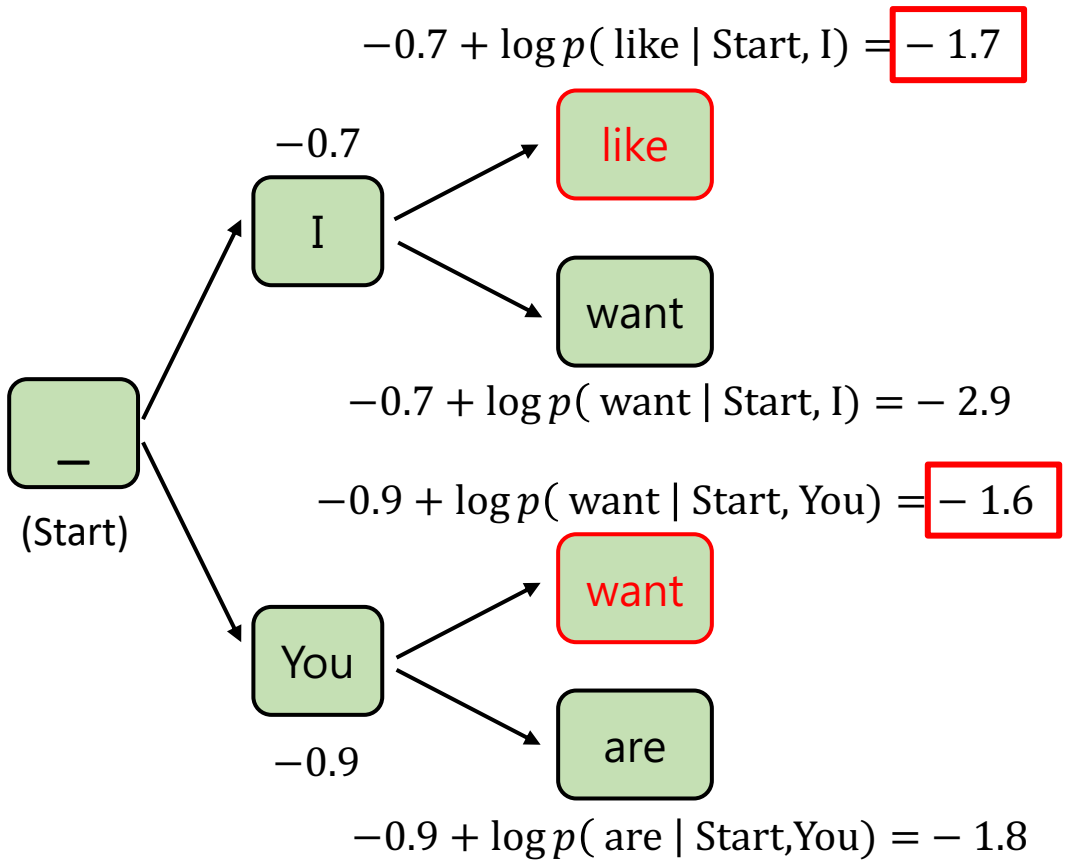
`Beam size` = 2



- At this decoding step, two choices are preserved.

Beam Search ($t = 2$)

`Beam size` = 2



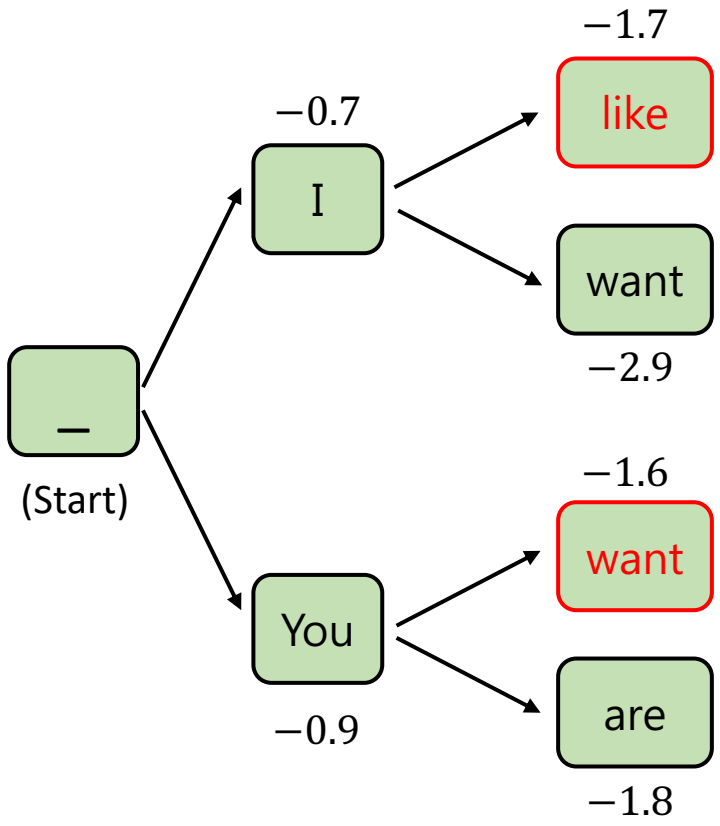
$$L_{ml} = -\sum_{t=1}^{T} \log p(y_t^* | y_1^*, \ldots, y_{t-1}^*, x)$$

Note the loglikelihood! Being close to zero is better!

- At this decoding step, two choices are preserved, and the other two are discarded.

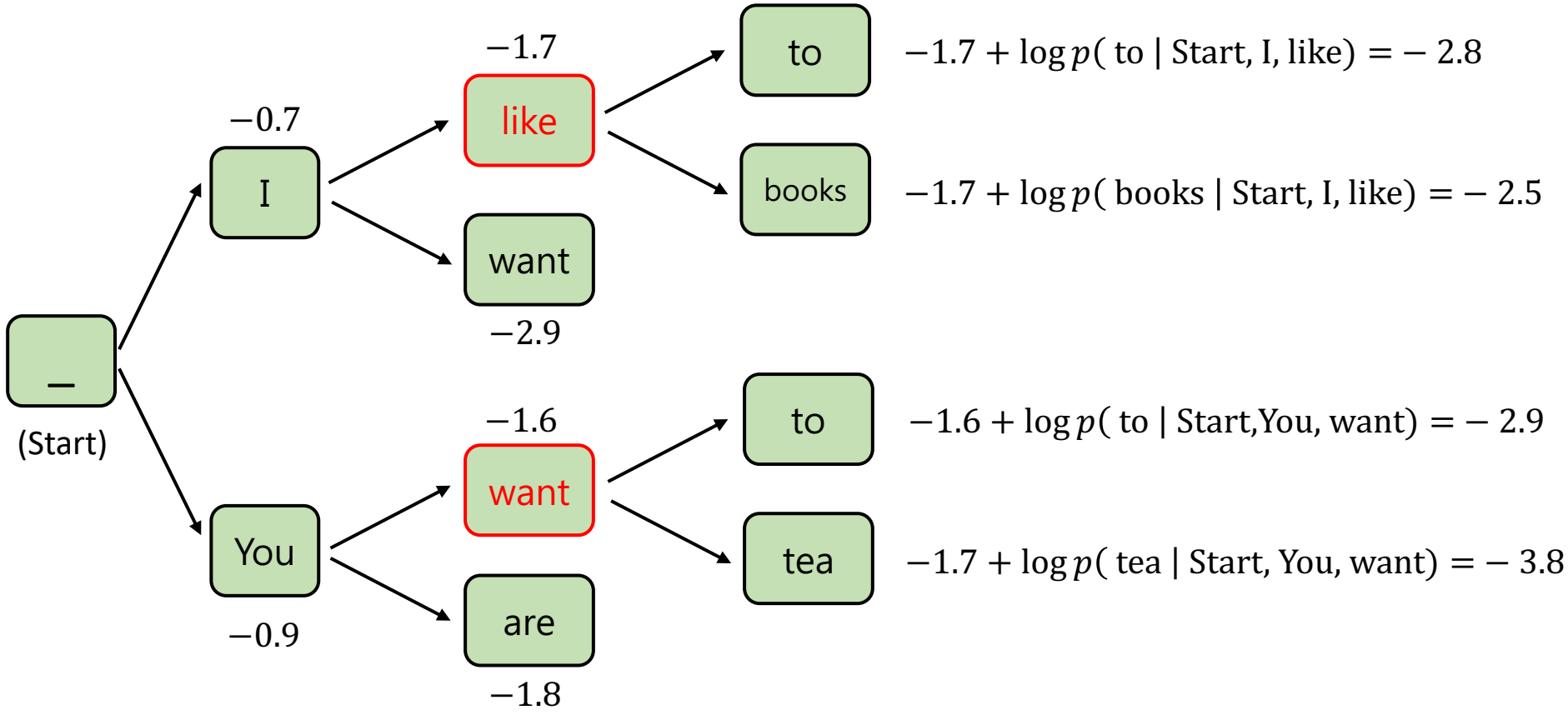
Beam Search ($t = 2$)

`Beam size` = 2



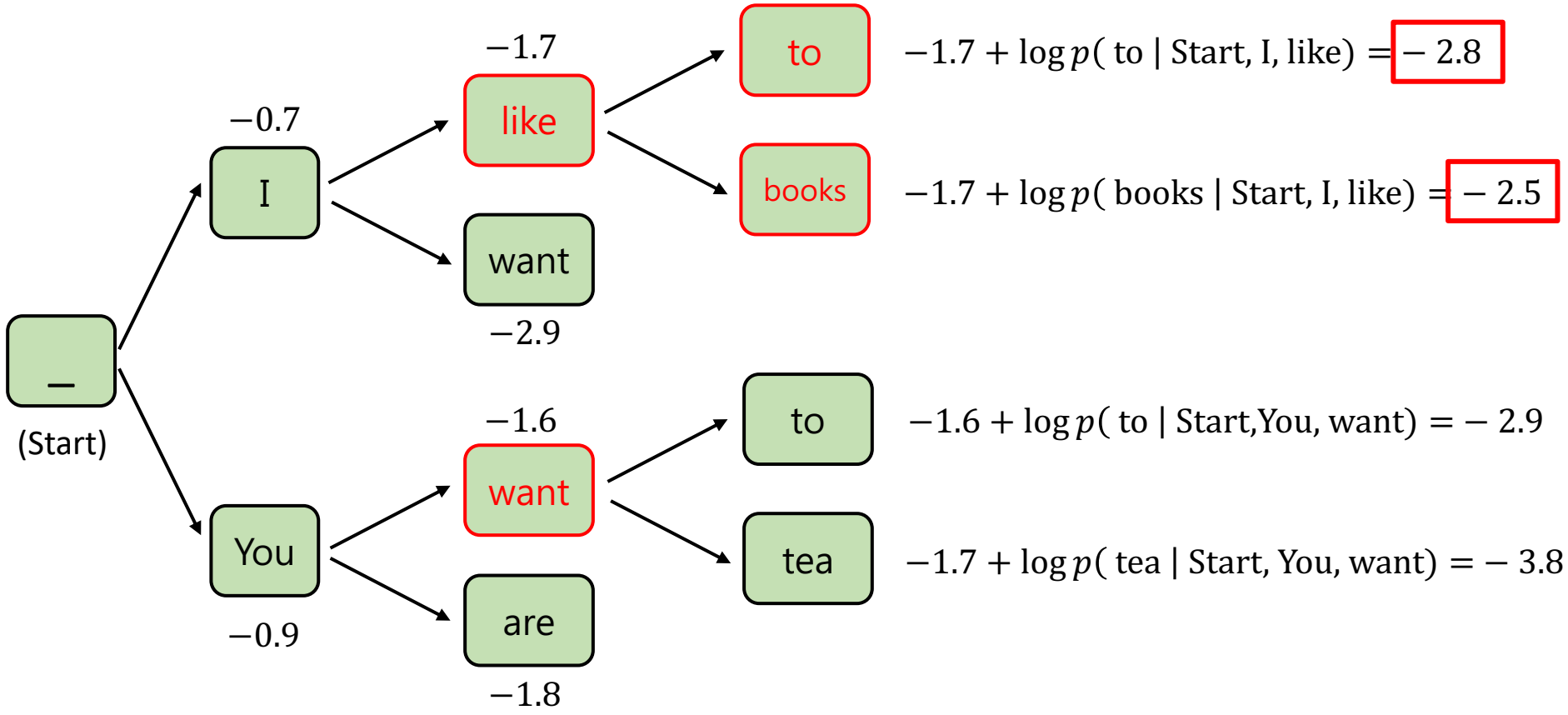
Beam Search ($t = 3$)

`Beam size` = 2



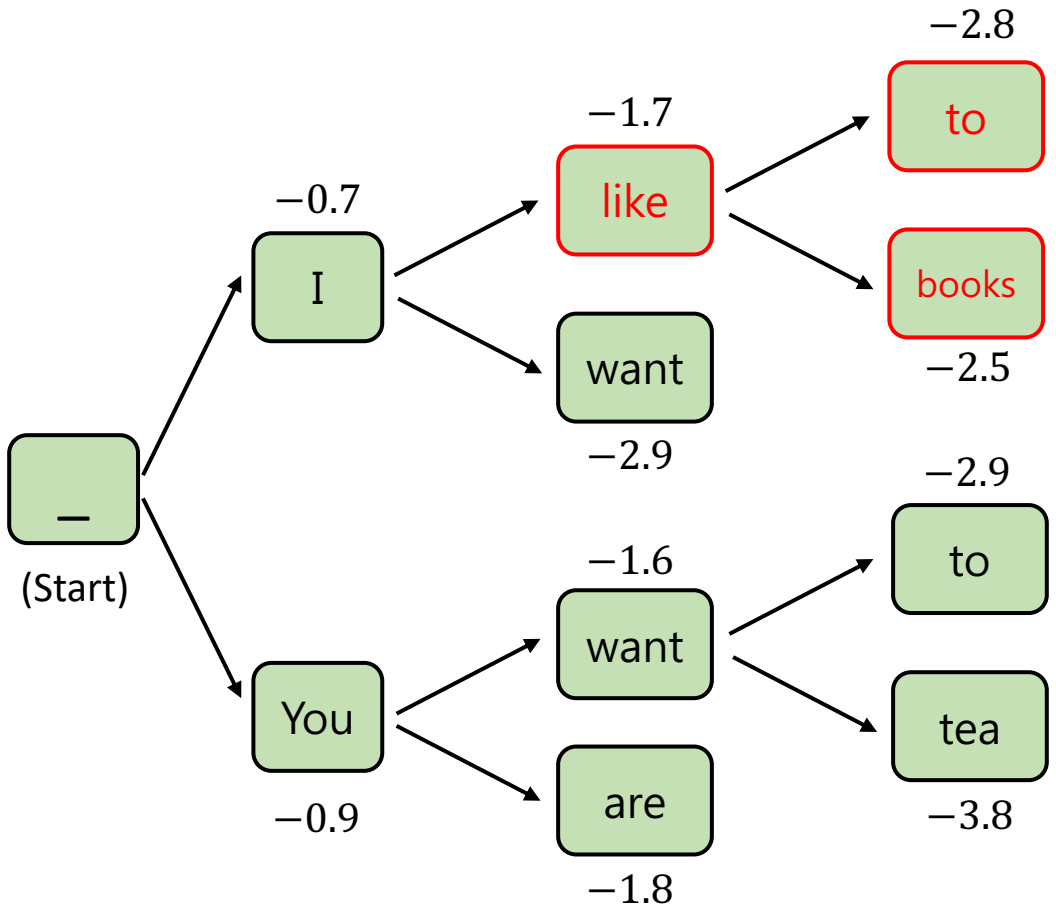
Beam Search ($t = 3$)

`Beam size` = 2



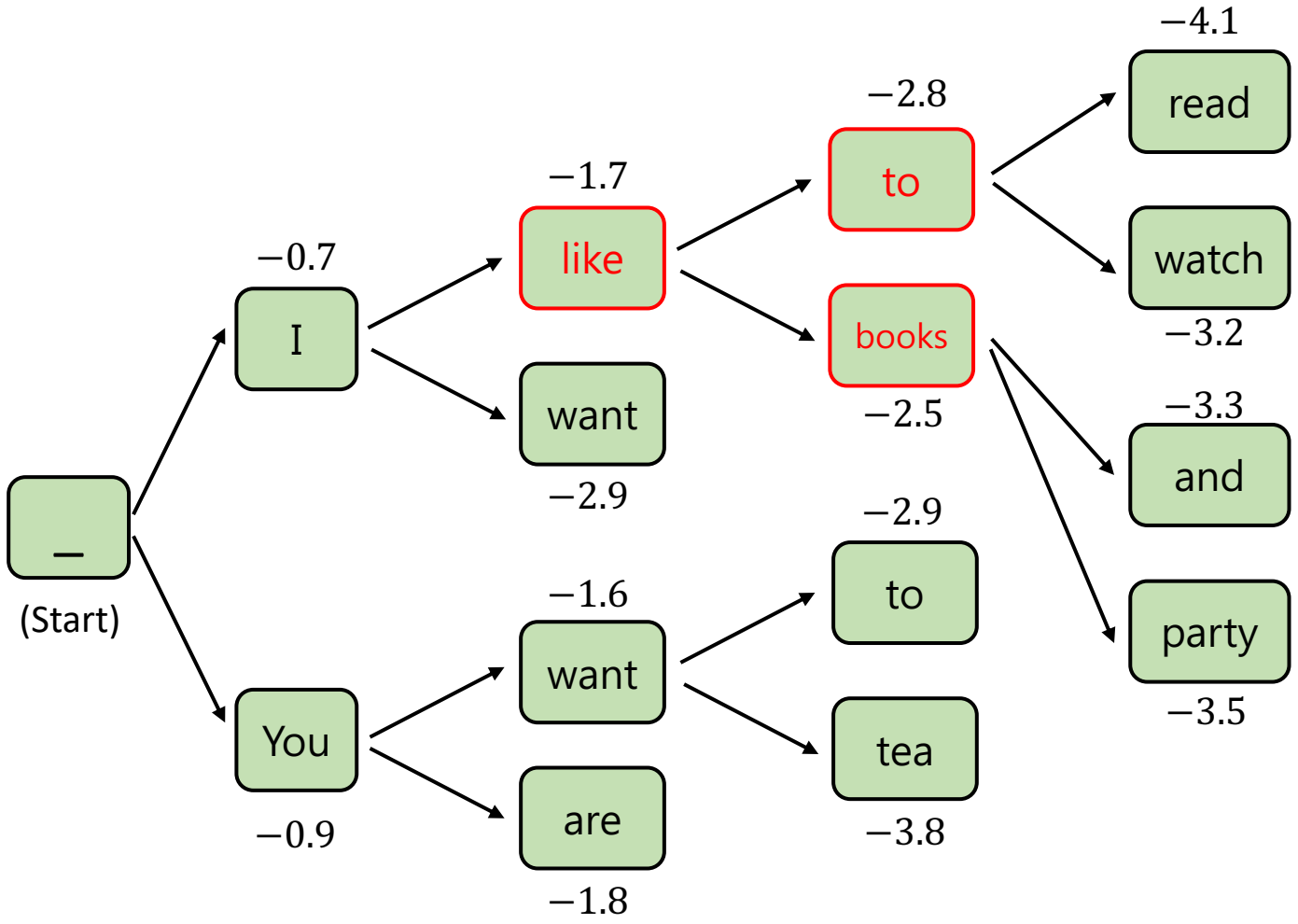
Beam Search ($t = 3$)

`Beam size` = 2



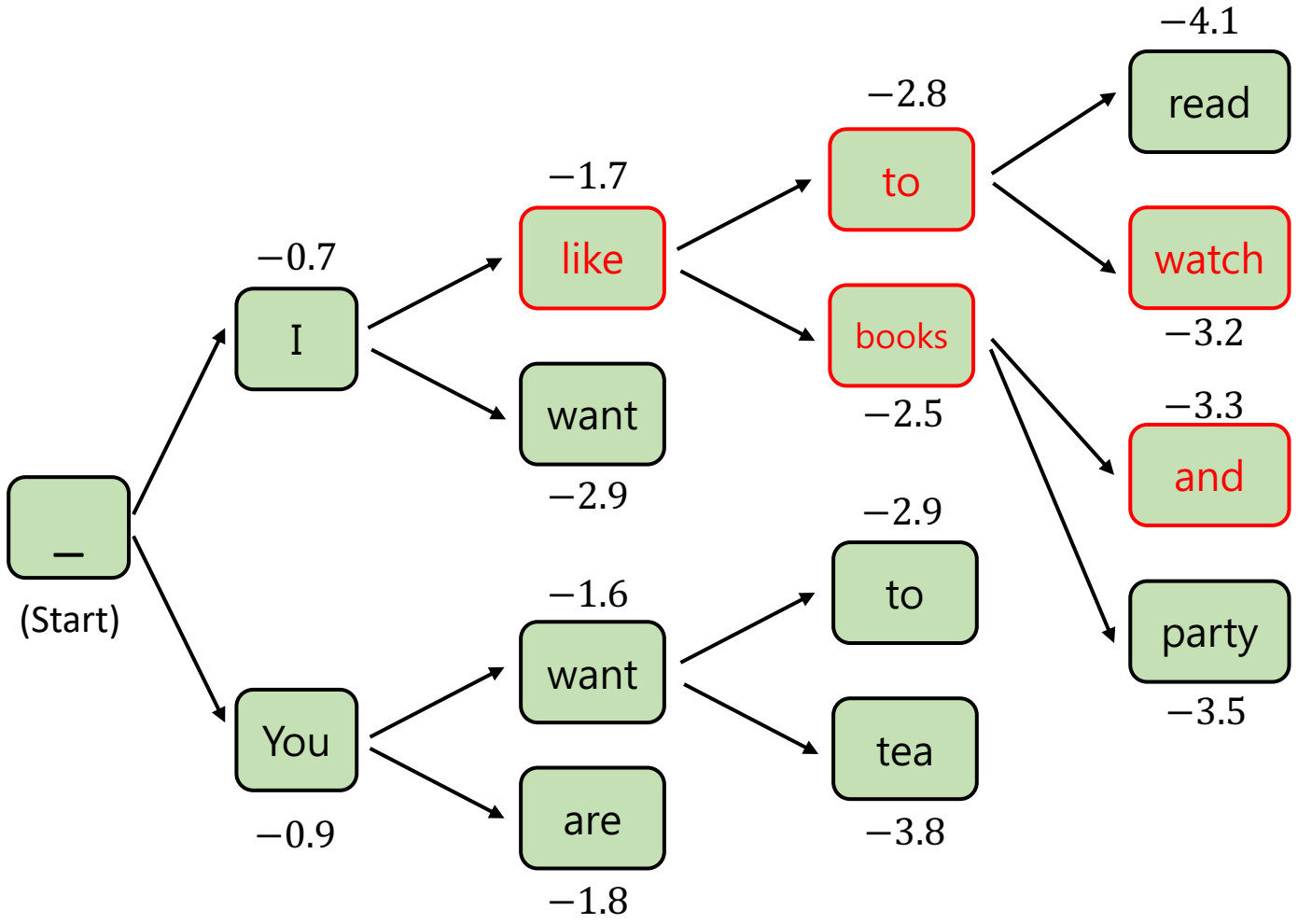
Beam Search ($t = 4$)

`Beam size` = 2



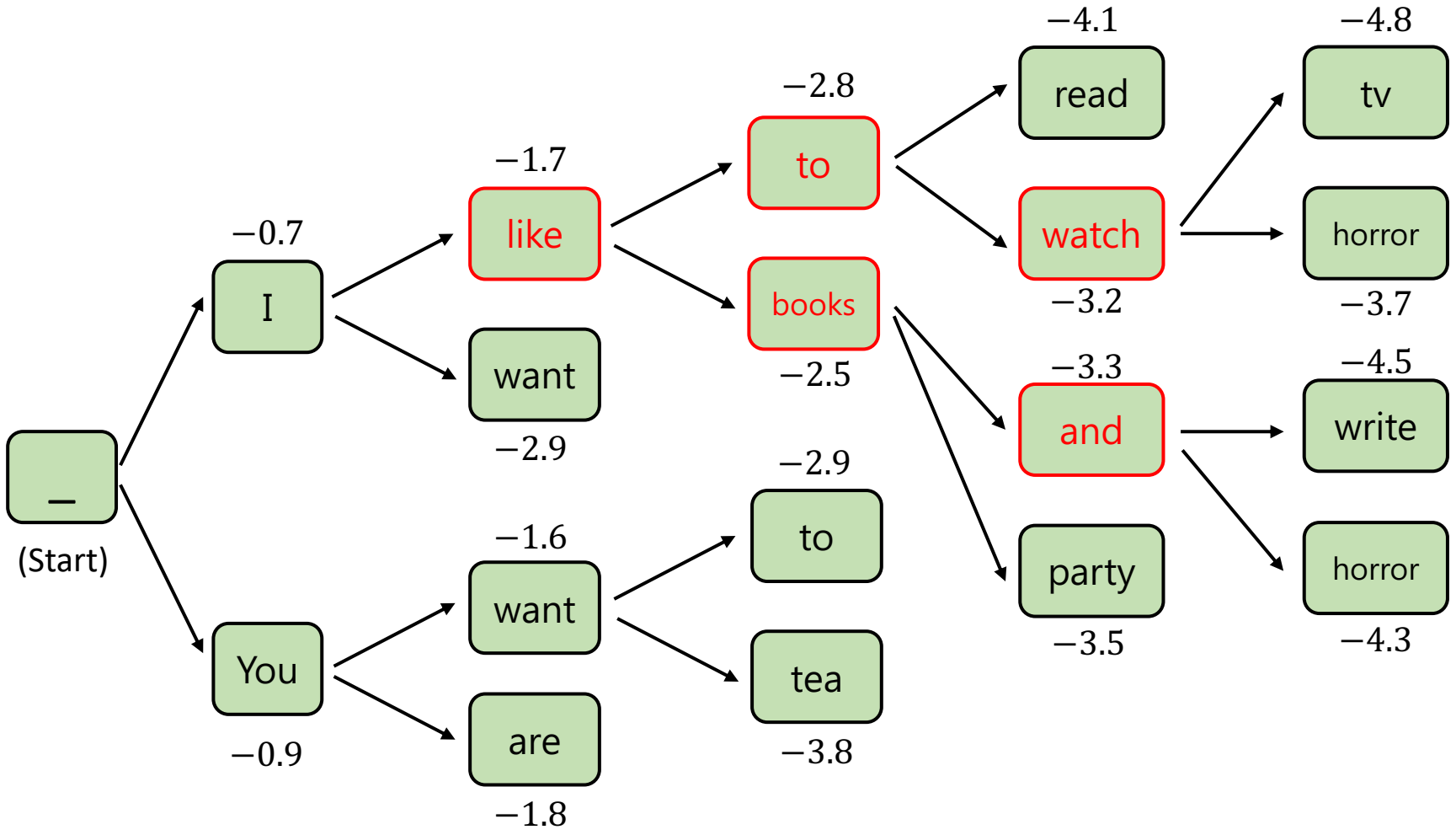
Beam Search ($t = 4$)

`Beam size` = 2



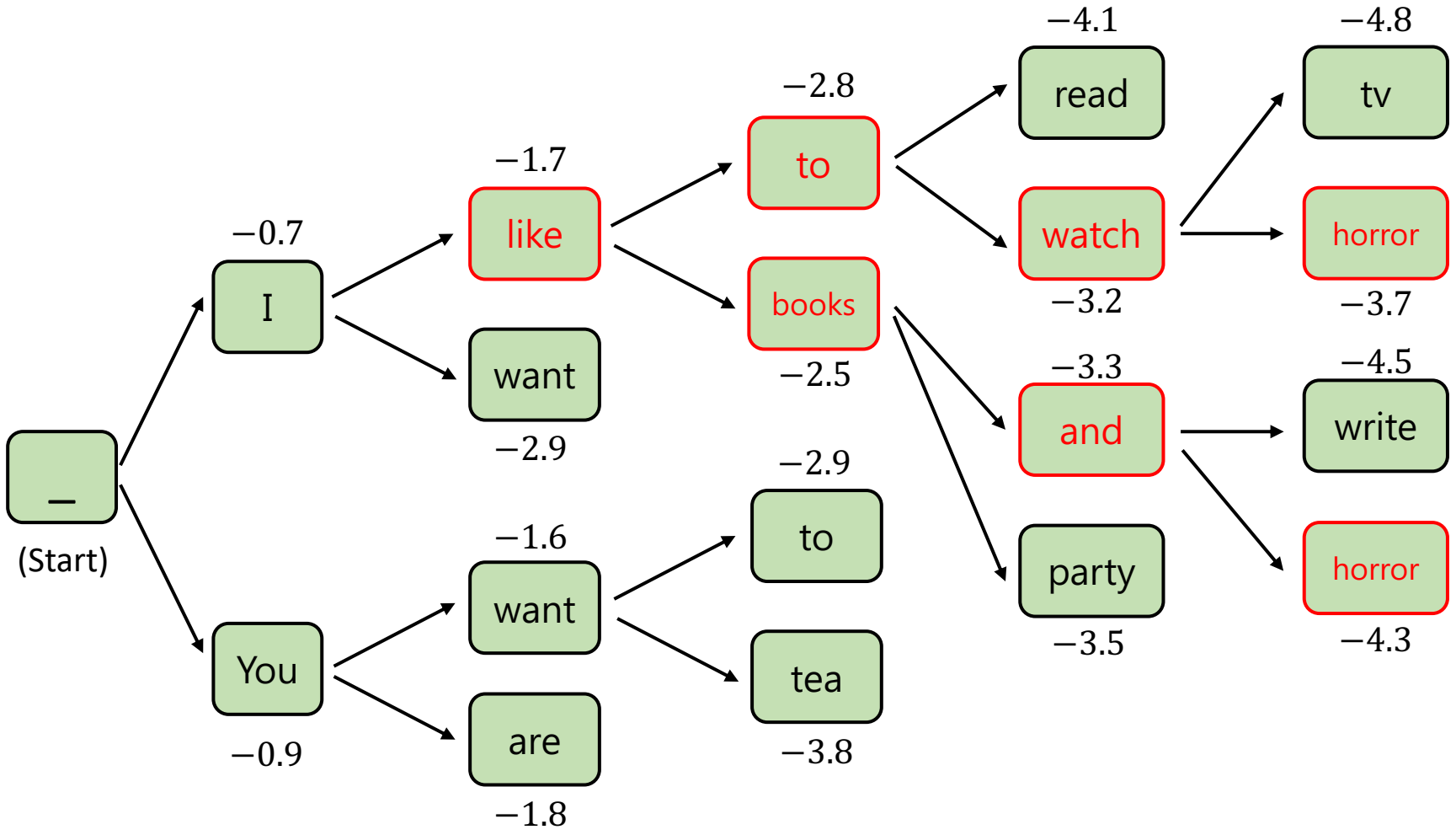
Beam Search ($t = 5$)

`Beam size` = 2



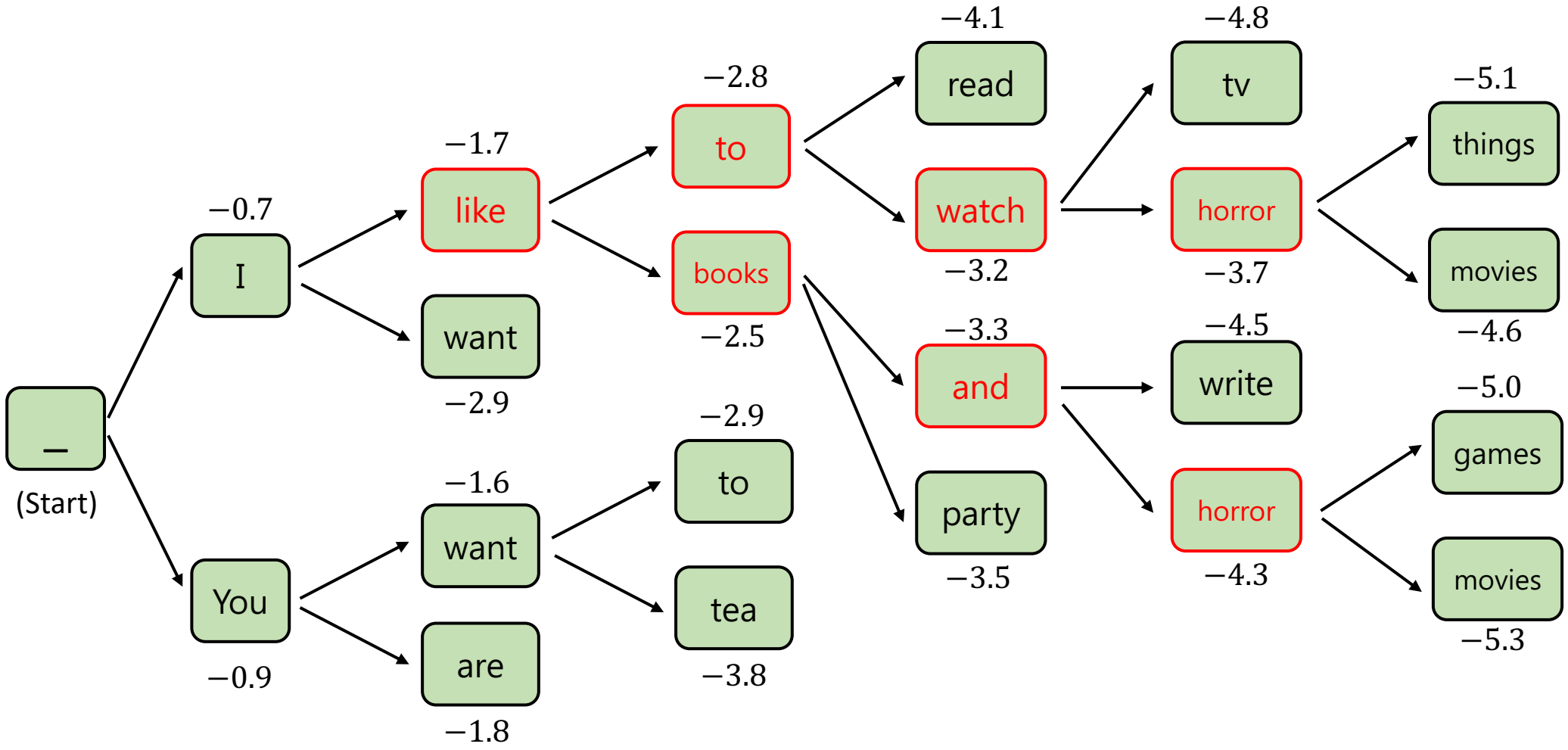
Beam Search ($t = 5$)

`Beam size` = 2



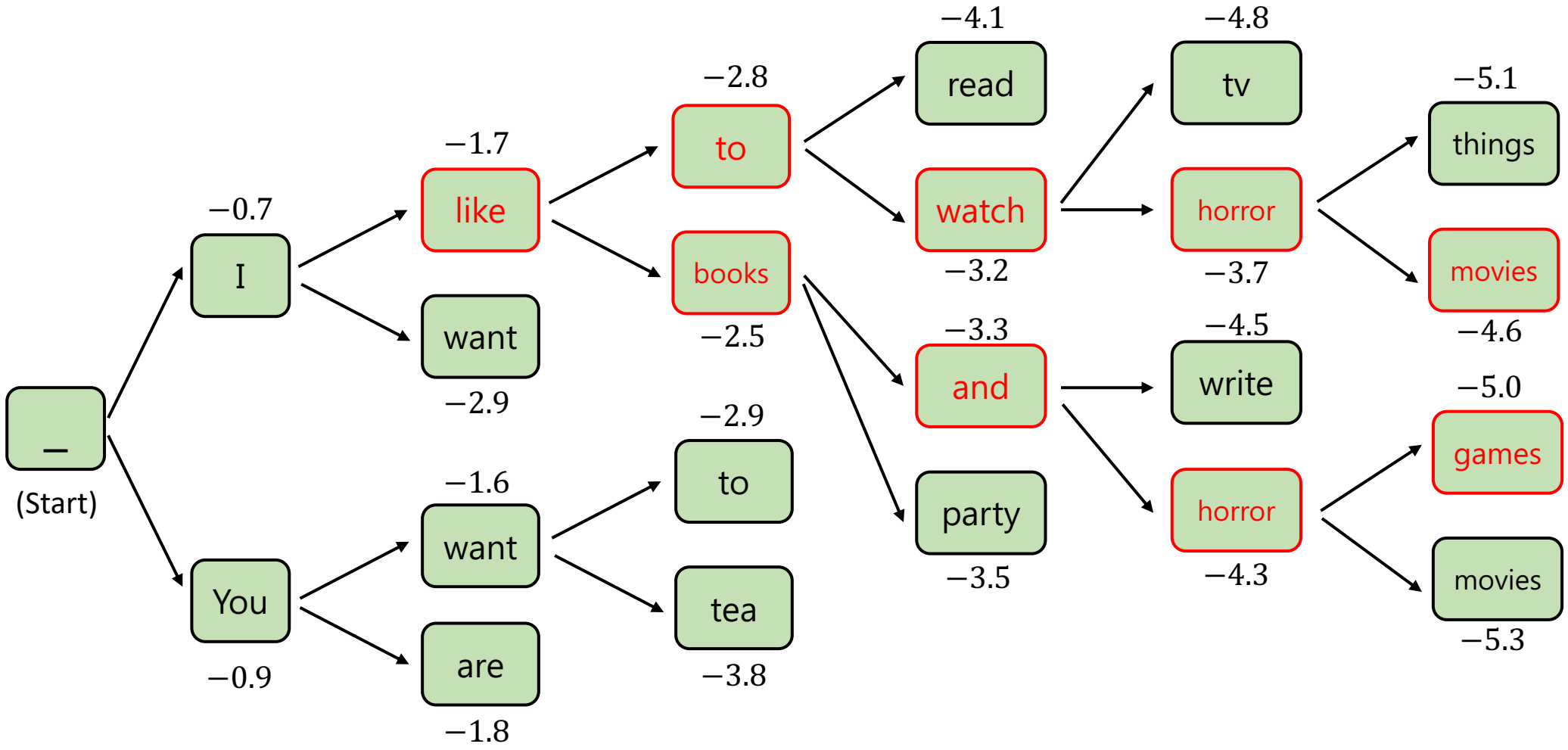
Beam Search ($t = 6$)

`Beam size` = 2



Beam Search ($t = 6$)

`Beam size` = 2



Stop Criterion (停止生成的情况)

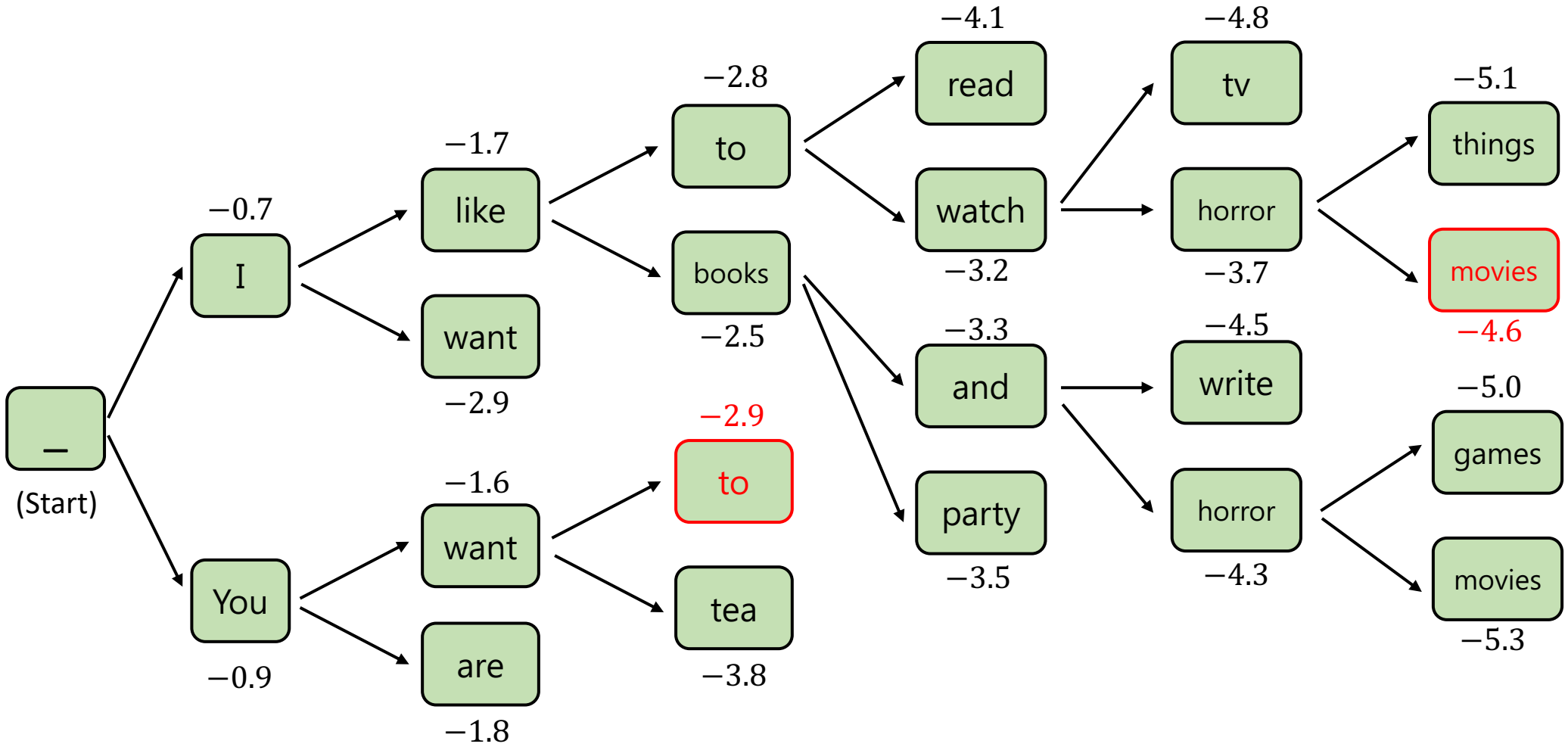
- There are two common stop criterions, either for greedy decoding or beam search decoding (or Top-p / Top-k sampling):
  - We consider a sequence of generation complete when the <EOS> token is produced by a model. *<EOS>: End of sequence
    - E.g., <Start> I like to watch horror movies <EOS>
  - A generated sequence reaches a pre-defined maximal length.

Problem of Beam Search

- Longer candidates will have lower scores.
- (Let's see again the 6th time step)

Beam Search ($t = 6$)

`Beam size` = 2



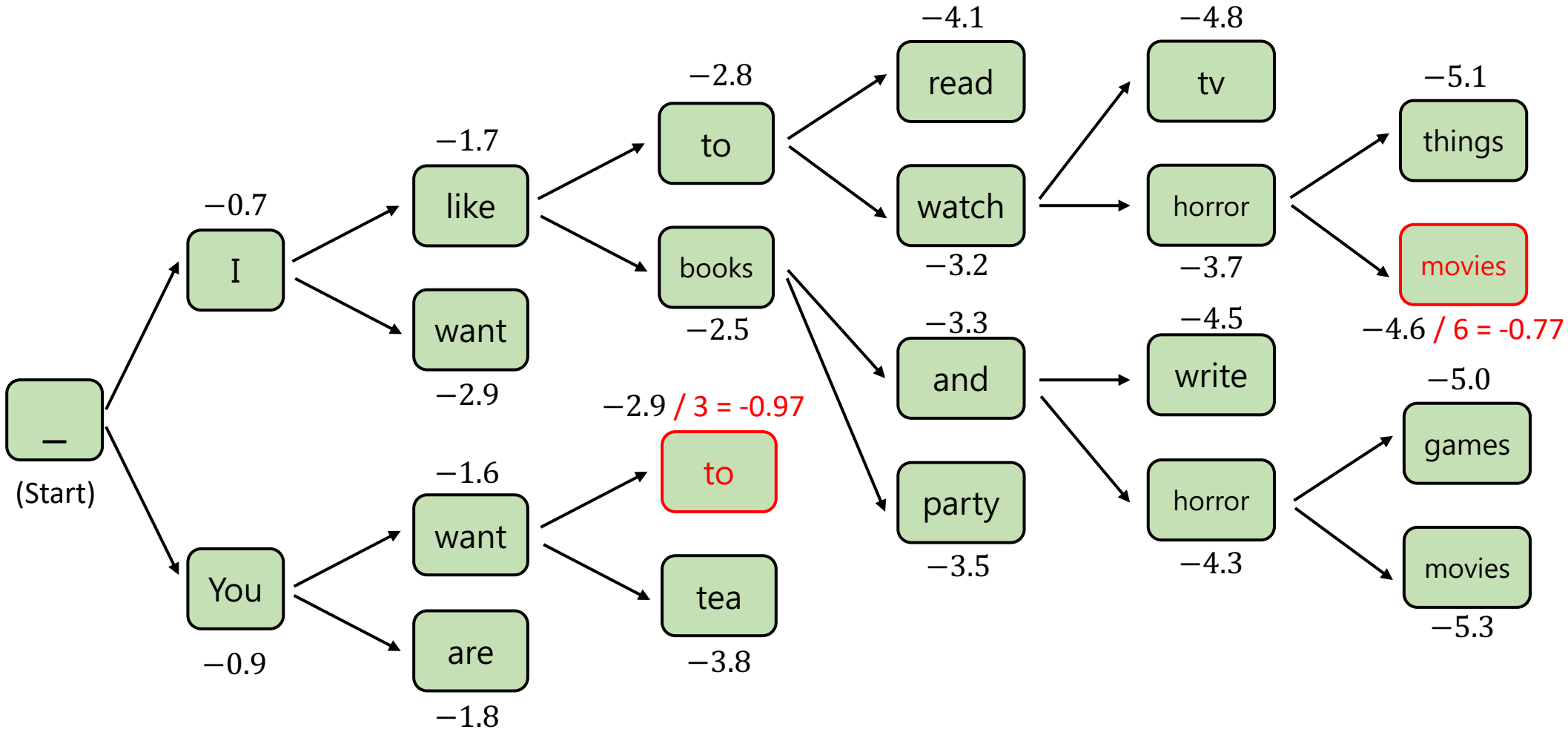
Problem of Beam Search (1)

- Longer candidates will have lower scores.
- Solution: Perform normalization to penalize on length

$$L_{ml} = -\frac{1}{T}\sum_{t=1}^{T} \log p(y_t^*|y_1^*, \ldots, y_{t-1}^*, x)$$

Beam Search ($t = 6$)

`Beam size` = 2



Problem of Beam Search (2)

- Decoding strategies that optimize for output with high probability, such as beam search, lead to text that is incredibly degenerate (e.g., repetitive words), even when using state-of-the-art models such as GPT-2 Large (in 2020).

Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations (ICLR). 2020.

Problem of Beam Search (2) – Continued.

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations (ICLR). 2020.

Why is Beam Search so weak?

- 現代語言模型通常使用 maximum likelihood 的方式 (language modeling) 進行訓練，這會導致模型過度偏向常見或高頻 tokens
- 當模型在 early steps 中對某些 tokens 給予極高機率時，這些 tokens 所在的路徑就會大幅壓制了其他 candidate tokens，導致生成缺少多樣性，甚至進入重複生成的loop (例如 I don’t know I don’t know I don’t know …)

Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations (ICLR). 2020.

[Summary] Strategy for MLE decoding

Maximum Likelihood Estimation (MLE): greedy decoding, beam search

既然機率值最高的不是我們想要的字詞

Strategy: Add more randomness!

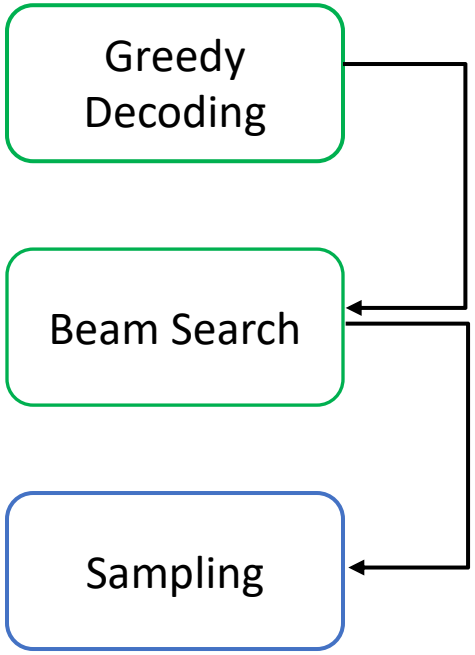
Strategy for MLE decoding: 改用 Sampling

- Sampling 就是「讓模型隨機選下一個詞，而不是每次都選最有可能的那個詞」。
- Sampling 是根據模型對每個詞給出的 機率分布 來進行「隨機抽樣」

Token	Probability (p)
cat	0.5
dog	0.3
car	0.15
book	0.05

Sampling 不是亂抽，而是根據機率 (模型的信心) 來產生下一次生成

Summary and the Thinking Route



Greedy decoding always selects the best choice at each time step and becomes problematic for early mistakes.

Models can fall into repetitive loops with beam search, especially for longer text.

Problem of Pure Sampling

獨角獸

Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

山谷

英語

如果完全 random sampling

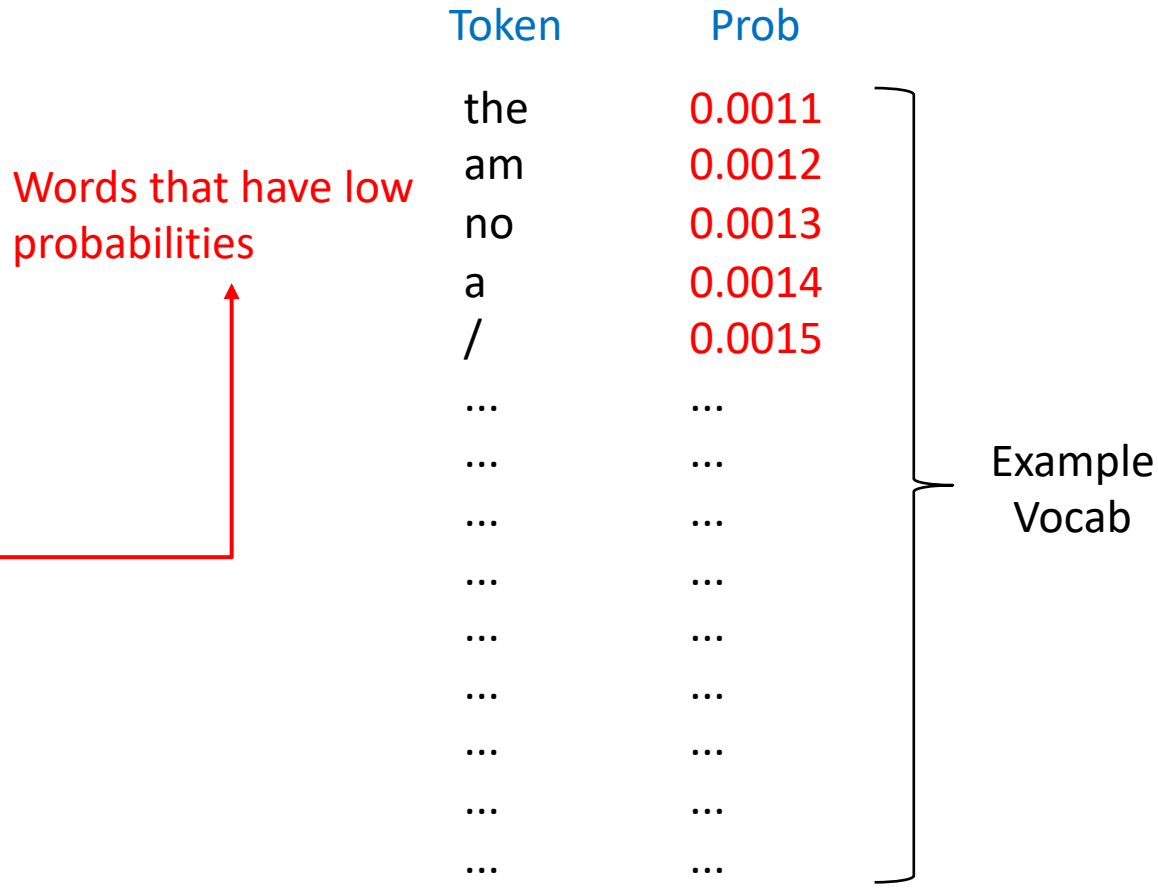
Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

它們是被稱為「玻利維亞騎士」的牛；他們住在遠離城鎮的偏遠沙漠，他們說話很大聲，美麗、天堂般的玻利維亞語言。他們說，「吃午餐了，瑪姬。」他們沒有告訴午餐是什麼，」導演丘佩拉斯·奧姆威爾教授告訴天空新聞。「他們只是和科學家交談，就像接受電視採訪一樣記者。我們甚至沒有留下來接受採訪電視記者。也許這就是他們發現他們扮演玻利維亞騎士。」

Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations (ICLR). 2020.

Why is Pure Sampling so weak?

- Pure Sampling does not show repetitive loop, but the result becomes incoherent and almost unrelated to the context
- Why? -> Unreliable tail



[Summary] Strategy for Sampling

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

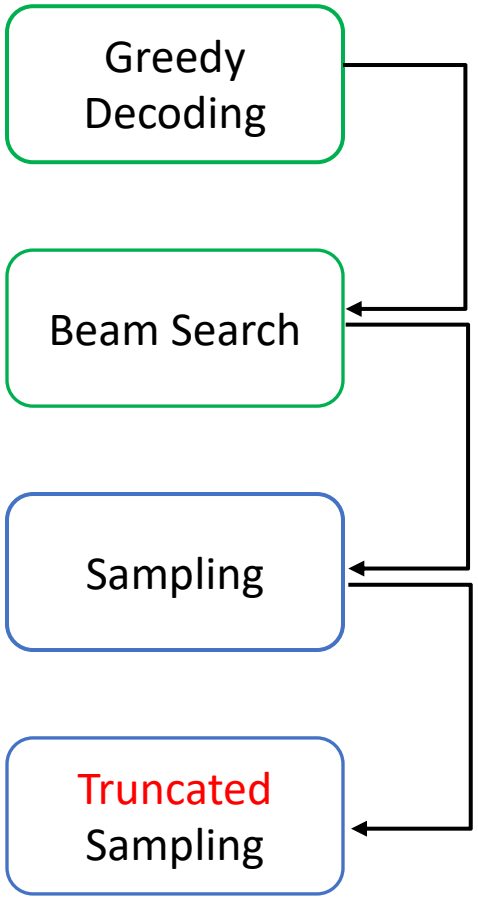
Let's truncate the vocabulary!

Token	Prob
Bolivian	0.0011
Chinese	0.0012
Egyptian	0.0013
…	…
…	…
…	…
…	…
…	…
…	…
…	…
…	…
…	…
…	…
Scottish	0.09
English	0.10

Bad candidates
可能有很多字
但機率總和才10%

Good candidates
可能沒有很多字
但機率總和有90%

Summary and the Thinking Route



Greedy decoding always selects the best choice at each time step and becomes problematic for early mistakes.

Models can fall into repetitive loops with beam search, especially for longer text.

Random sampling takes words with low probabilities into considerations.

Top-p Sampling

Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations. 2020.

- 又稱作 Nucleus Sampling
- Core idea: truncate the vocabulary based on probability mass
- Steps:
  1. Define a value p as the probability threshold.
  2. Truncate the vocabulary whose sum of probabilities is greater than p :

$$\sum_{x \in V^{(p)}} P(x|x_{1:x-1}) \geq p$$

where $V^{(p)} \subset V$ is the smallest set that fulfills the equation.
Now you get a new truncated vocabulary $V^{(p)}$.

Top-p Sampling

- Core idea: truncate the vocabulary based on probability mass
- Steps:

  3. Re-build the probability distribution based on the following normalization:

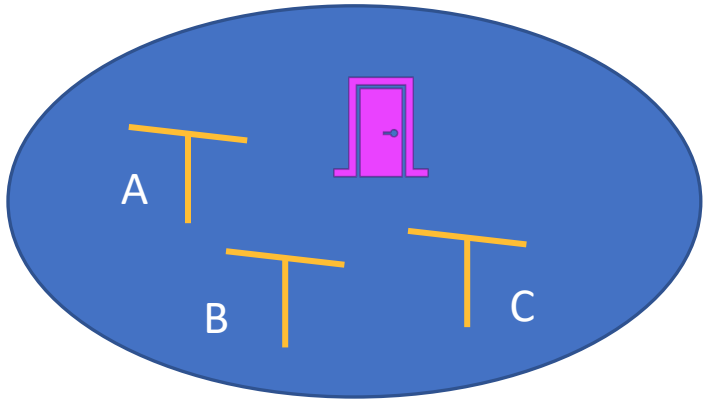
     3-1. 把 $V^{(p)}$ 的機率值加總

     3-2. $V^{(p)}$ 內的每個機率值/加總

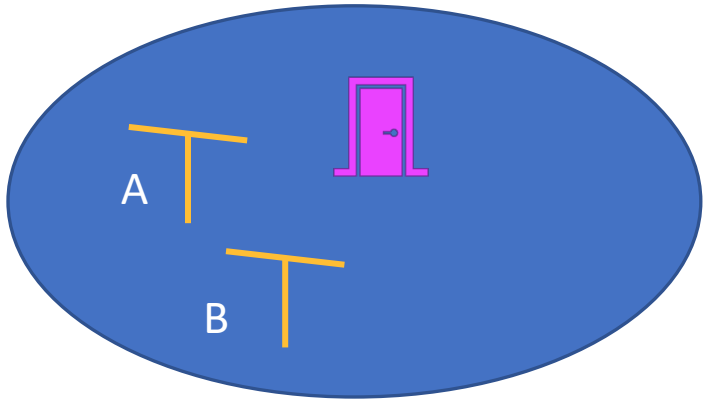
$$p' = \sum_{x \in V^{(p)}} P(x|x_{1:x-1})$$

$$\mathrm{P}'(\mathrm{x}|x_{1:x-1}) = \begin{cases} P(x|x_{1:x-1}/p') & \text{if } x \in V^{(p)} \\ 0 & \text{otherwise} \end{cases}$$

機率重新調整 toy example



任意門的機率：1/4



任意門的機率：0.25 / (0.25+0.25+0.25) = 1/3

竹蜻蜓A機率：0.25 / (0.25+0.25+0.25) = 1/3
竹蜻蜓 B 機率：0.25 / (0.25+0.25+0.25) = 1/3

Top-p Sampling example

Token	Probability (p)
cat	0.5
dog	0.3
car	0.15
book	0.05

- 設 p 為 0.8
  - 那 $V^{(p)}$ 中只會有 cat 和 dog 兩個詞

Top-k Sampling

- Core idea: truncate the vocabulary with the most probable words
- Steps:
  1. Define a value k as the size of truncated vocabulary.
  2. Leave the k words with the highest probabilities. Now you get a new truncated vocabulary $V^{(k)}$. (假設k=40，那$V^{(k)}$就只剩40個 tokens)
  3. Re-build the probability distribution based on the following normalization:

     3-1. 把 $V^{(k)}$ 的機率值加總

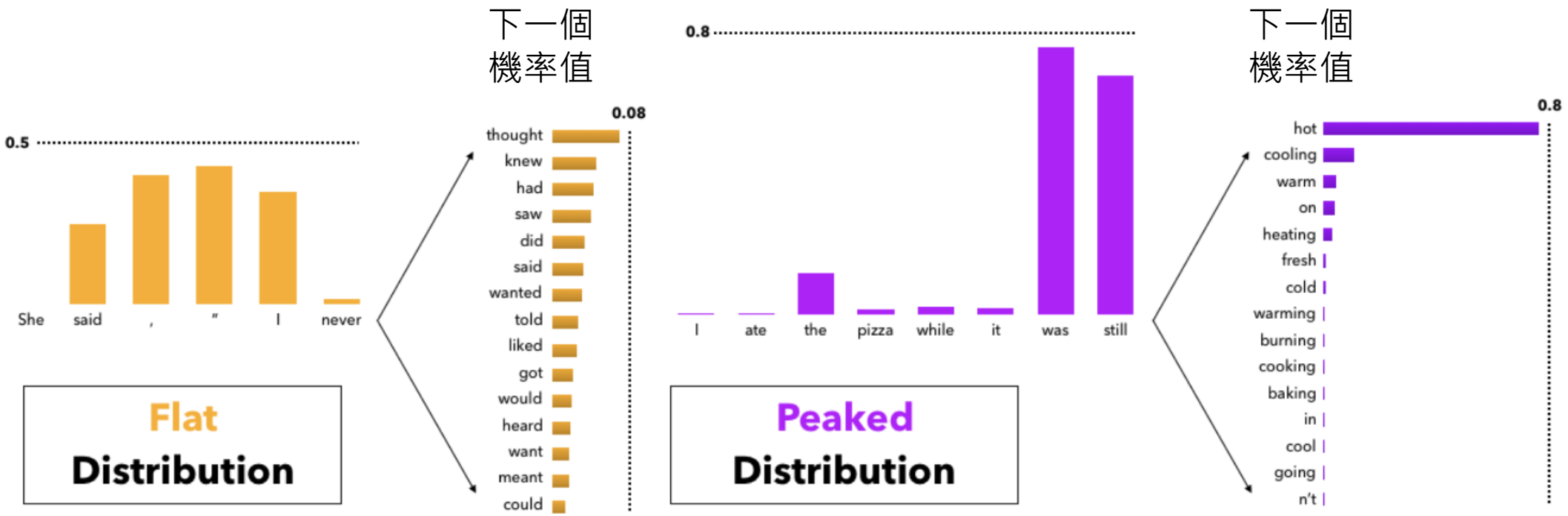
     3-2. $V^{(k)}$ 內的每個機率值/加總

$$p' = \sum_{x \in V^{(k)}} P(x|x_{1:x-1})$$

$$\mathrm{P}'(\mathrm{x}|x_{1:x-1}) = \begin{cases} P(x|x_{1:x-1})/p' & \text{if } x \in V^{(k)} \\ 0 & \text{otherwise} \end{cases}$$

Problem of Top-k Sampling

- Top-k sampling 的 k 值需根據情況調整

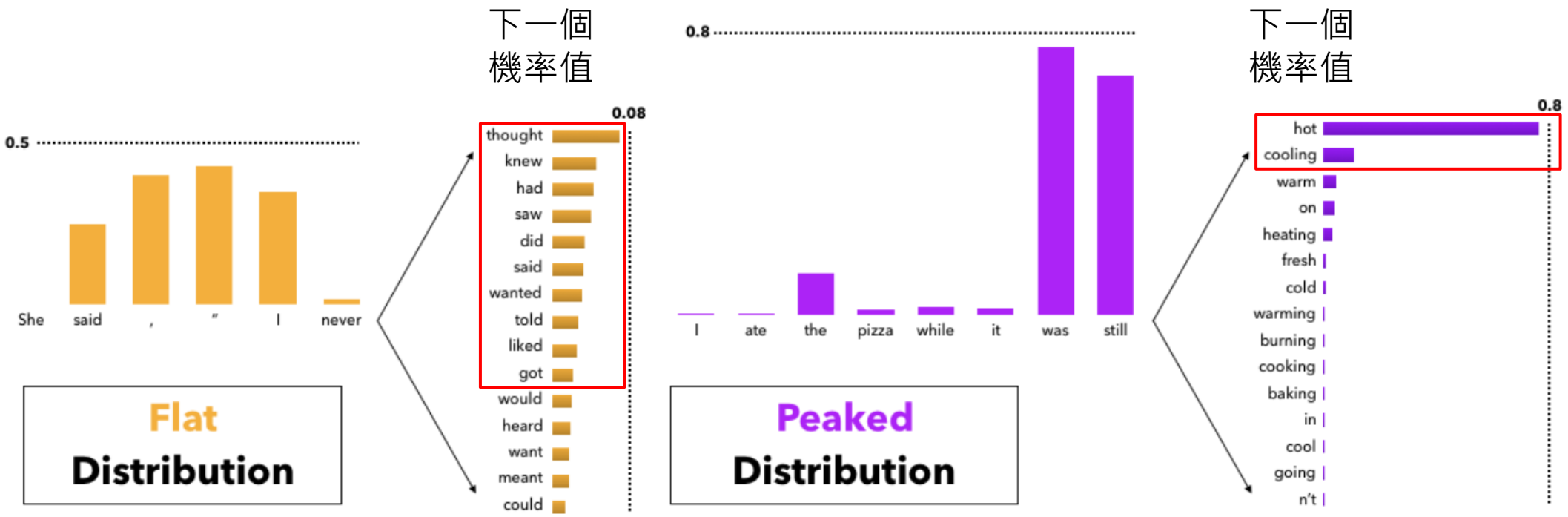


此時 k 如果太小的話，容易錯過最正確的字 -> k 要設大一點

此時 k 如果太大的話，容易取到與上下文無關的字 -> k 要設小一點

Problem of Top-k Sampling (看看 Top-p)

- 假設 p = 0.9



此時 k 如果太小的話，容易錯過最正確的字 -> k 要設大一點

此時 k 如果太大的話，容易取到與上下文無關的字 -> k 要設小一點

Softmax

- When generating the next word, `softmax` is performed to get the probabilities among the words in the vocabulary.
- Softmax formula:

$$\frac{\exp(u_l/\mathrm{t})}{\sum_{l'}^{|V|}\exp\left(u_{l'}/t\right)}$$

u_l: logits (model outputs before softmax)
$|V|$: size of the vocabulary
t: softmax temperature

Softmax

	Word	Logits		Probability
Example Vocab	the	0.0011	→	0.78
	am	0.0012	→	0.11
	no	0.0013	→	0.03
	a	0.0014	→	0.02
	/	0.0015	→	0.06

- Note that softmax is required for every decoding strategies since we need to find out the next word from a vocabulary.

Properties of Softmax Temperature

u_l: logits (model outputs before softmax)
$|V|$: size of the vocabulary
t: softmax temperature

- Softmax formula:

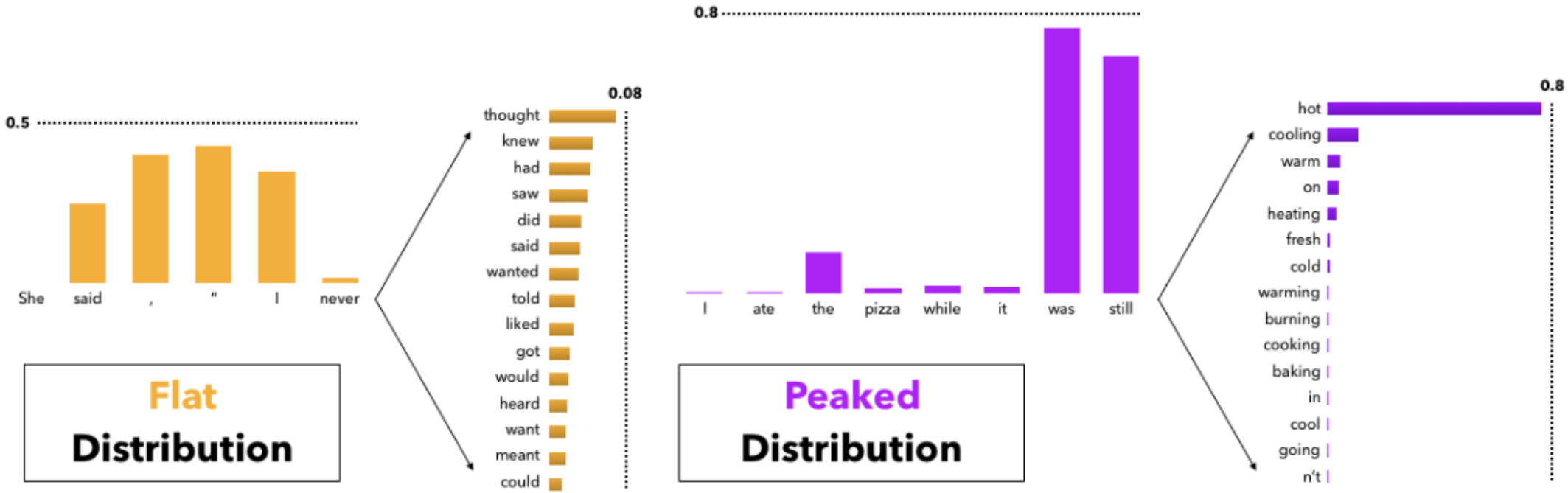
$$\frac{\exp(u_l/\mathrm{t})}{\sum_{l'}^{|V|}\exp\left(u_{l'}/t\right)}$$

- Larger t -> Lower probability value -> Smaller ranges of probability distribution -> More diverse Outputs
- Smaller t -> Higher probability value -> Greater ranges of probability distribution -> Less diverse Outputs

Softmax Temperature

Temperature 高的時候 (more diverse)

Temperature 低的時候 (less diverse)



Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations (ICLR). 2020.

Top-k and Top-p Sampling Improve Repetition

Method	Perplexity	Self-BLEU4	Zipf Coefficient	Repetition %	HUSE
Human	12.38	0.31	0.93	0.28	-
Greedy	1.50	0.50	1.00	73.66	-
Beam, b=16	1.48	0.44	0.94	28.94	-
Stochastic Beam, b=16	19.20	0.28	0.91	0.32	-
Pure Sampling	22.73	0.28	**0.93**	0.22	0.67
Sampling, t=0.9	10.25	0.35	0.96	0.66	0.79
Top-k=40	6.88	0.39	0.96	0.78	0.19
Top-k=640	13.82	**0.32**	0.96	**0.28**	0.94
Top-k=40, t=0.7	3.48	0.44	1.00	8.86	0.08
Nucleus p=0.95	**13.13**	**0.32**	0.95	0.36	**0.97**

Table 1: Main results for comparing all decoding methods with selected parameters of each method. The numbers *closest to human scores* are in **bold** except for HUSE (Hashimoto et al., 2019), a combined human and statistical evaluation, where the highest (best) value is **bolded**. For Top-k and Nucleus Sampling, HUSE is computed with interpolation rather than truncation (see §6.1).

Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations. 2019.

The k value should be carefully picked

Method	Perplexity	Self-BLEU4	Zipf Coefficient	Repetition %	HUSE
Human	12.38	0.31	0.93	0.28	-
Greedy	1.50	0.50	1.00	73.66	-
Beam, b=16	1.48	0.44	0.94	28.94	-
Stochastic Beam, b=16	19.20	0.28	0.91	0.32	-
Pure Sampling	22.73	0.28	**0.93**	0.22	0.67
Sampling, t=0.9	10.25	0.35	0.96	0.66	0.79
Top-k=40	6.88	0.39	0.96	0.78	0.19
Top-k=640	13.82	**0.32**	0.96	**0.28**	0.94
Top-k=40, t=0.7	3.48	0.44	1.00	8.86	0.08
Nucleus p=0.95	**13.13**	**0.32**	0.95	0.36	**0.97**

Table 1: Main results for comparing all decoding methods with selected parameters of each method. The numbers *closest to human scores* are in **bold** except for HUSE (Hashimoto et al., 2019), a combined human and statistical evaluation, where the highest (best) value is **bolded**. For Top-k and Nucleus Sampling, HUSE is computed with interpolation rather than truncation (see §6.1).

Holtzman, Ari, et al. "The Curious Case of Neural Text Degeneration." International Conference on Learning Representations. 2019.

Comparison for Top-k and Top-p Sampling

	Top-k Sampling	Top-p Sampling
Hyperparameter	k: top-k words are preserved	p: sum of the minimum set of words exceeds the value of p
Performance (Who is better?)	By cases (these two are both widely used)	
Hyperparameter Tuning	Harder	Easier
Common Hyperparameter Value	k=40	p=0.95

Thank you!

Instructor: 林英嘉

✉ yjlin@cgu.edu.tw