

Beam Search

강사 : 백병인

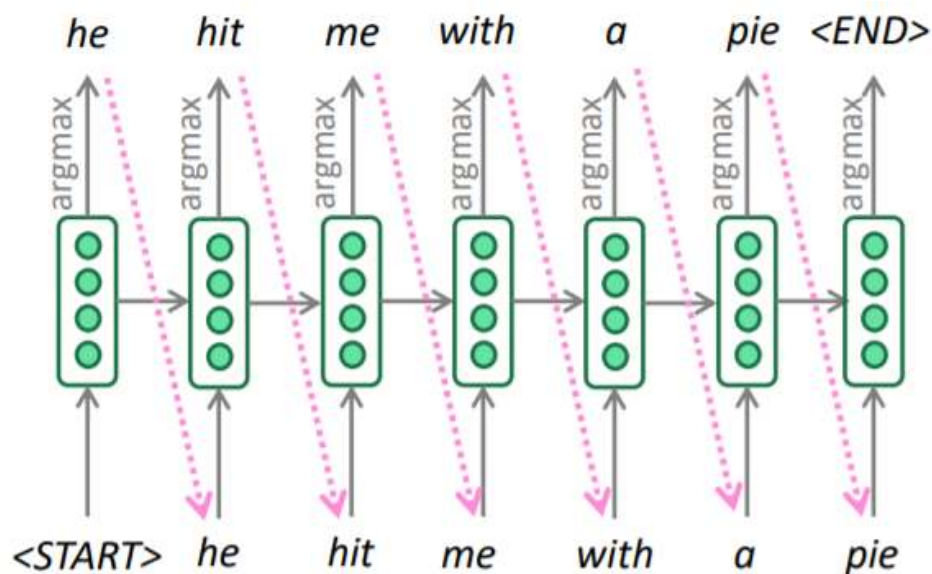
pi.paek@modulabs.co.kr

모두의연구소 Research Scientist



How to Decode to make sentences?

- Sentence Generation with Language Model



while $y_{j-1} \neq \text{"</s>"}$:
 $y_j = \operatorname{argmax} P(y_j \mid X, y_1, \dots, y_{j-1})$

매순간 그 시점의 가장 이익이 커지게끔만
선택하는 것을 탐욕(greedy) 알고리즘이라
고 합니다.

Greedy Decoding의 문제점

- 순간의 잘못된 선택을 되돌릴 방법이 없다.

- Input: *il a m'entarté* (he hit me with a pie)

- → *he* _____

- → *he hit* _____

- → *he hit a* _____ (whoops! no going back now...)



Exhaustive Search Decoding

- Greedy Decoding처럼 매 타임 스텝 확률만 보고 바로 다음 단어를 결정하지 말고,
- 만들 수 있는 모든 문장의 경우의 수를 다 만들어 보고
- 그렇게 문장이 만들어질 확률을 전부 다 계산해 보고
- 그중 확률이 제일 높은 경우를 선택하면 어떨까?

$$\begin{aligned} P(y|x) &= P(y_1|x) P(y_2|y_1, x) P(y_3|y_1, y_2, x) \dots, P(y_T|y_1, \dots, y_{T-1}, x) \\ &= \prod_{t=1}^T P(y_t|y_1, \dots, y_{t-1}, x) \end{aligned}$$

- 엄청난 계산복잡도를 감당할 수만 있다면?? $O(V^T)$ complexity



Beam Search

- 매 스텝마다, 그때 가장 확률이 높은 hypothesis를 K개만 유지하자.
- Hypothesis마다 다음 스텝에서 다시 K개의 가능한 경우를 검토해서 총 K^2 개의 경우 중 가장 확률이 높은 K개만 남기자.
- Exhaustive search처럼 best solution을 보장하지는 못하지만
- Greedy search처럼 오류수정이 안되는 문제는 극복할 수 있다.

2-Beam Search Example

Beam size = $k = 2$. Blue numbers = $\text{score}(y_1, \dots, y_t) = \sum_{i=1}^t \log P_{\text{LM}}(y_i | y_1, \dots, y_{i-1}, x)$

