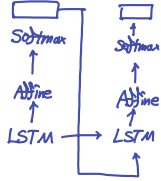


Fixer, Breaker 구조

Greedy Decoding.



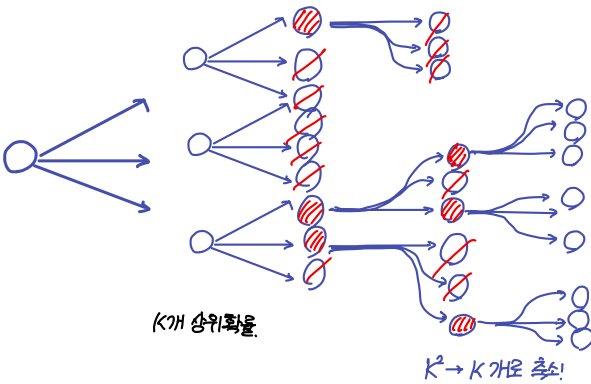
- 각 단계에서 최선의 확률은 가진 값을 사용함.

⇒ 확률이 높은 다른 조건이 존재하면, 소용확률이 아닌 값도 고려해야 한다.

Beam Search

: 해당 시점에서 유용한 길의 수를 줄여서 결정함.

K=3일때 ⇒ 한층의 최종 Output은 3개의 후보만을 가진다.



<Eos>을 만난다면,

해당 Candidates를 유지하고, <Eos>을 만난 만큼

추가로 후보를 뽑는다!

⇒ 후보가 K개가 될때까지 반복함.

이후 누적 확률이 가장 커지는 경우를 최종선택

※ 누적확률의 사용

Sequence의 길이가 길어지면, 누적확률은 당연히 작아짐!

→ Length Penalty 사용.

$$LP = \frac{(\text{minimum-length} + \text{현재 길이})^\alpha}{(\text{minimum-length} + 1)}$$

α로 계산 가능함.

∴ 최종 결과는 누적확률 / Length-Penalty 이다.

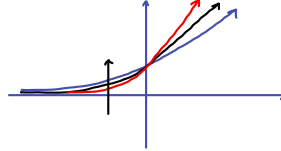
LP 주요 파라미터 α...

① α=0 : length penalty 무시함.

② α≠0 : length penalty : 길이가 커질수록 Penalty 감소?

누적확률 × (□)^{-α}에 해당함..

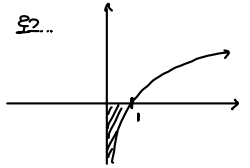
길이가 길면, □ 값이 커진다.



Sequence 길이가 길다

→ α가 음수가 아니려면

Penalty 커짐.. ↑



0.5로 가정.

$$(0.5)^{\pm} / (0.5)^{\pm}$$

들음 더 큰 값은 ↑

$$\text{Prob} = \frac{\text{Prob}}{\text{Penalty}} \quad \log \left(\frac{\text{Prob}}{\text{Penalty}} \right) \rightarrow \frac{1}{n}$$

log(P) → 확률값 0 ~ 1 사이 (-∞, 0)

P의 값이 (0,1) Penalty ↑ : log는 ↓
 $\left(\frac{m+1}{m}\right)^\alpha$ 의 경우, α>0 이라면, N이 커짐에 따라 값이 커짐

∴ Length에 따라 P ↓ 것은 맞다!