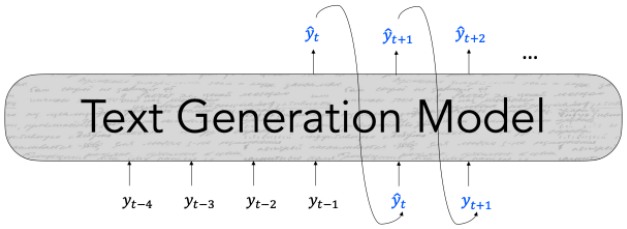
Natural Language Processing with DeepLearning
week 13

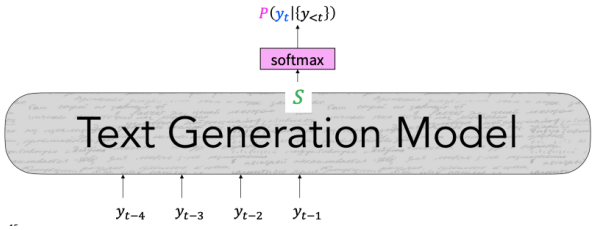
The course
- What is NLG?
- Formalizing NLG: a simple model and training algorithm
- Decoding from NLG models
- Training NLG models
- Evaluating NLG Systems
- Ethical Considerations

1. What is NLG
- Machine Translation
- Dialogue Systems
- Summarization
- Visual Description

- Autoregressive 구조
  t 시점의 단어 토큰을 예측하기 위해 이전 토큰들을 입력으로 사용



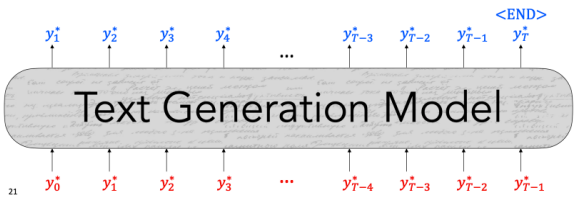
- 각 스텝에서 각 단어들에 대한 score를 계산하고 이 점수들을 softmax에 통과하여 확률분포를 계산



- negative loglikelihood를 최소화하는 방식으로 학습

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(y_t^* | \{y^*\}_{<t})$$

- teacher forcing



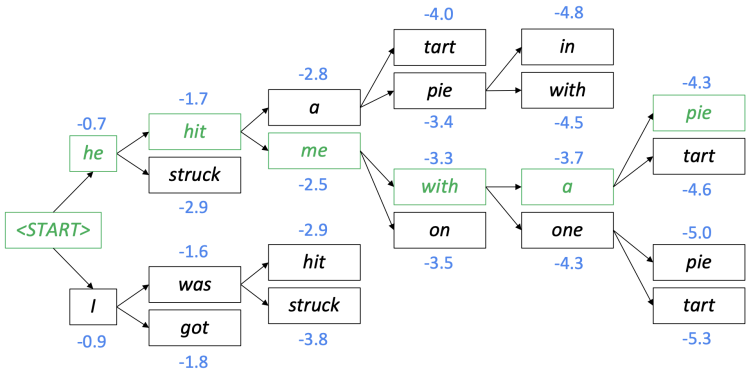
  예측한 토큰 대신 실제 문장의 토큰을 입력으로 사용하는 학습 방식이다.
  초기에 잘못 생성된 단어로 인해 계속 잘못된 단어가 생성되는 것을 방지한다.

03. Decoding from NLG models
1) Argmax Decoding

$$\hat{y}_t = \underset{w \in V}{\text{argmax}}\, P(y_t = w | y_{<t})$$

단순하게 가장 큰 likelihood를 가지는 단어 토큰을 생성 토큰으로 선정한다.

2) Beam Search



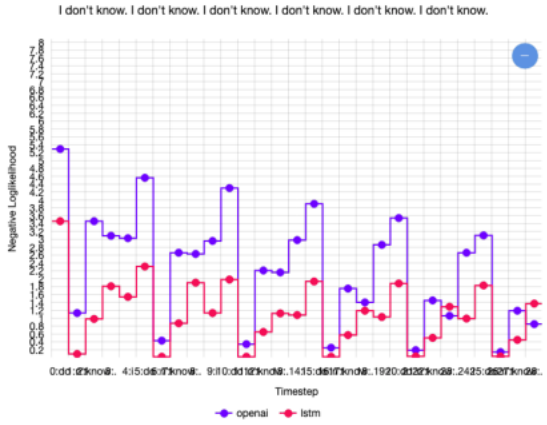
매 스텝마다 loglikelihood가 높은 k개의 후보 중 선택을 반복하여 argmax보다 자연스러운 문장을 생성한다.

Greedy Methods의 문제점

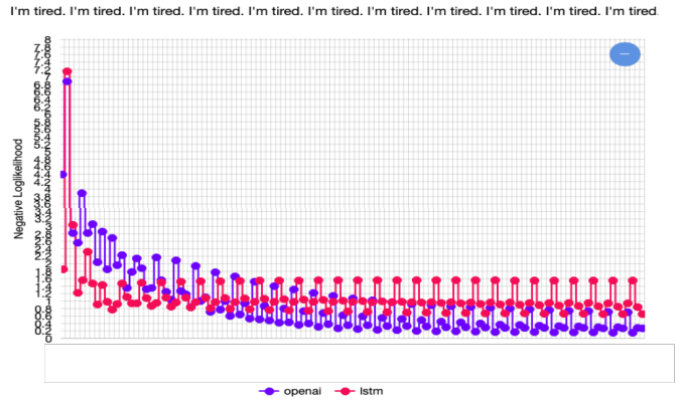
Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Continuation: 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 México...**

비슷한 단어를 반복하여 생성하는 고질적인 문제가 있다.

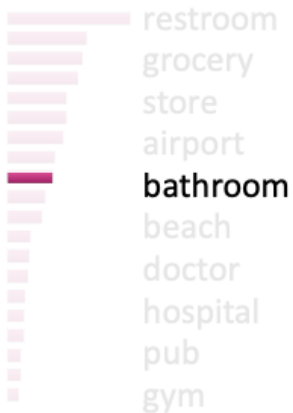


반복될수록 같은 말에 대한 negative loglikelihood가 낮아진다. => 점점 더 확신을 가지고 같은 말을 생성한다.



Transformer 모델의 negative loglikelihood가 더 급격히 떨어지는 것을 확인할 수 있다. => 같은 말에 대한 confidence를 가진다.
이는 일반적인 Decoder 구조보다 이전 step들의 embedding을 더 잘 활용할 수 있는 구조이기 때문이다.

1. Random Sampling

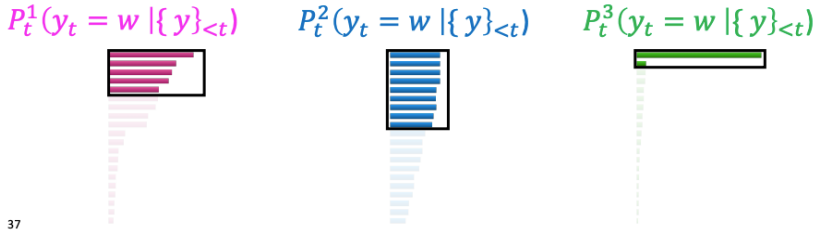


모델이 예측한 각 토큰의 확률을 가중치로 사용하여 샘플링하는 방식

2. Decoding: Top-k Sampling

- 상위 k개 중 확률값을 가중치 두고 샘플링하는 방식
- k값은 일반적으로 5, 10, 15를 사용
- k가 클수록 다양한 문장이 생성되지만 부자연스러운 문장이 생성됨.
- k가 작을수록 한정된 문장이 생성되지만 자연스러운 문장이 생성됨.

3. Decoding: Top-p Sampling



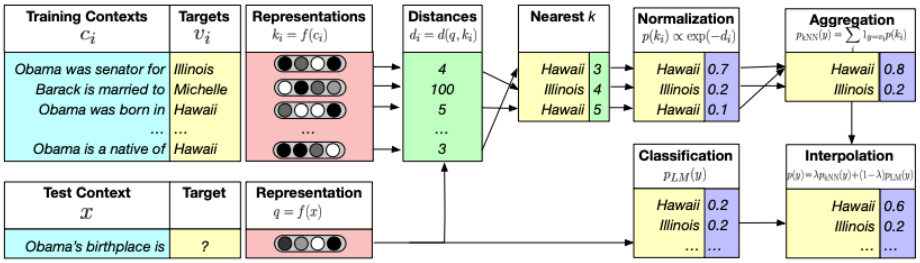
- 누적 확률 P에 속하는 상위 토큰들 중 샘플링하는 방식.
- k가 클수록 다양한 문장이 생성되지만 부자연스러운 문장이 생성됨.
- k가 작을수록 한정된 문장이 생성되지만 자연스러운 문장이 생성됨.

4. Scaling randomness: Softmax temperature

$$P_t(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{w' \in V} \exp(S_{w'}/\tau)}$$

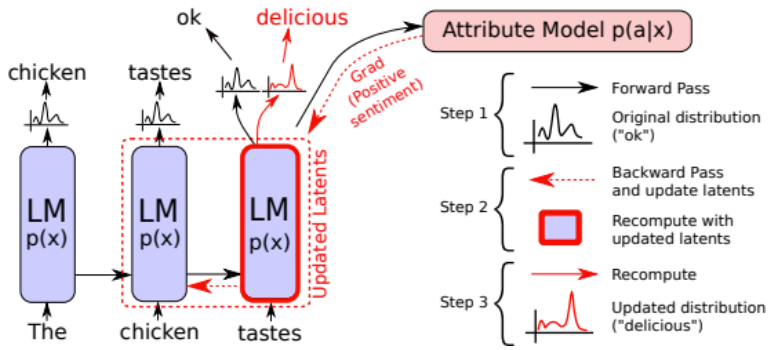
- Logit(S)에 일정한 상수 temperature(t)를 나눠 distribution을 scaling하는 방법.
- t가 클수록 분포가 고르게 되어 다양한 문장을 생성할 수 있다.
- t가 작을수록 분포의 차이가 커 일반적인 문장을 생성할 수 있다.

5. Improving decoding: re-balancing distributions



- training contexts와 target data를 학습하면서 training contexts의 representations를 저장해 두었다가 inference 시에 모델 예측값을 보정할 목적으로 사용한다.
- 과정
  - Test Context의 reprezentation vector와 저장된 Training context의 representation vector 간 거리를 구한다. 그 중 가장 거리가 가까운 k개의 training contexts를 선정한다.
  - 이 샘플의 target에 해당하는 probability를 negative distance의 exponential 값으로 구한다.
  - 중복되는 class는 서로 값을 더한다.
  - 일반적인 예측 class에 이렇게 구한 값을 interpolation하여 최종 예측 분포를 구한다.
- 기존 모델의 decoding 방법만 수정하여 간단히 적용 가능하고 inference 시간을 단축할 수 있다.

6. Backpropagation-based distribution re-balancing



- Plug & Play Language Model의 약어로, PPLM이라고 한다.
- 기존 모델의 구조 변경 없이 별도의 모델을 사용하여 성능을 개선한다.
- 과정
    - 기존의 학습된 모델의 생성 토큰을 별도의 모델에 전달한다.
    - 별도의 모델에서 계산된 결과에 대한 그레디언트는 기존 모델의 latent vector를 업데이트한다.
    - 업데이트된 레이턴트 벡터를 통해 새로운 단어 토큰을 생성할 수 있게 된다.

04. Training NLG models

Unlikelihood Training

$$\mathcal{L}_{UL}^{t} = -\sum_{y_{neg} \in \mathcal{C}} \log(1 - P(y_{neg} \mid \{y^*\}_{<t}))$$

$$\mathcal{L}_{ULE}^{t} = \mathcal{L}_{MLE}^{t} + \alpha \mathcal{L}_{UL}^{t}$$

- 새롭게 생성된 토큰이 t 시점 이전까지 등장했던 토큰들을 중에 있다면 loss가 커지도록 하는 loglikelihood loss를 기존 mle loss에 추가로 사용하는 방식
- 텍스트를 다양하게 생성할 수 있다.

Exposure Bias Solutions

- 테스트 시에는 적용될 수 없기 때문에 teacher forcing이 Bias를 야기하는 문제에 대한 해결 방법

Scheduled sampling

- 특정한 시점에 teacher forcing을 사용할 확률인 p를 적용하는 방법.
- p값은 학습이 진행될수록 모델의 성능이 올라가기 때문에 점점 더 작은 값을 사용한다. -> 점점 모델의 예측 토큰을 다음 스텝의 입력으로 사용하여 테스트 시와 동일하게 만들어주어 바이어스르 낮춘다.

Sequence re-writing

- train dataset에서 추출한 프로토타입을 변형하여 문장을 생성하는 방법.
- 에디트 벡터는 샘플링된 프로토 타입을 adding, removing, modifying tokens 방법을 사용해서 변형한다.

Reinforcement Learning

$$\mathcal{L}_{RL} = -\sum_{t=1}^{T} r(\hat{y}_t) \log P(\hat{y}_t | \{y^*\}; \{\hat{y}_t\}_{<t})$$

- BLEU, ROUGE 스코어와 같은 metric을 reward로 사용할 수 있다.

- 하지만 의도하지 않은 shortcut을 학습하지 않도록 reward function을 잘 정리해야 한다는 문제점이 있다.