Language model · 언어모델

· 문장 자체의 출현확률을 예측
· 이전 단어들이 주어졌을 때 다음 단어를 예측하기 위한 모델

In korean
① 단어의 어순이 중요하지 않다. ex) 나는 학교에 걸어서 버스를 타고
나는 버스를타고 학교에 걸어서
버스를 타고 나는 학교에 걸어서.
(나는) 버스를 타고 학교에 걸어서.
② 단어의 생략이 가능

· 확률이 퍼지는 현상 발생
· 타고 → ., 학교에, 나는 등 하나 이상이 가능

$D = \{x^i\}_{i=1}^{N}$ ← 문장

$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log P(x^i_{1:n}; \theta)$ where $x_{1:N} = \{x_1, x_2 \cdots, x_N\}$ ← 문장

Chain rule

$$P(A,B,C,D) = P(D|A,B,C)P(A,B,C)$$
$$= P(D|A,B,C)P(C|A,B)P(A,B)$$
$$= P(D|A,B,C)P(C|A,B)P(B|A)P(A)$$

$$P(x_{1:n}) = P(x_1, \cdots, x_n)$$
$$= P(x_n|x_1, \cdots, x_{n-1}) \cdots P(x_2|x_1)P(x_1)$$
$$= \prod_{i=1}^{N} P(x_i|x_{<i})$$

$x_1 \sim x_{n-1}$까지의 단어가 주어졌을 때 x_n의 확률

$$\log P(x_{1:n}) = \sum_{i=1}^{N} \log P(x_i|x_{<i})$$

$$
\begin{aligned}
P(\text{<BOS>, I, love, to, play, <EOS>}) &= P(\text{<EOS>}|\text{<BOS>, I, love, to, play})P(\text{<BOS>, I, love, to, play}) \\
&= P(\text{<EOS>}|\text{<BOS>, I, love, to, play})P(\text{play}|\text{<BOS>, I, love, to})P(\text{<BOS>, I, love, to}) \\
&= P(\text{<EOS>}|\text{<BOS>, I, love, to, play})P(\text{play}|\text{<BOS>, I, love, to})P(\text{to}|\text{<BOS>, I, love})P(\text{<BOS>, I, love}) \\
&= P(\text{<EOS>}|\text{<BOS>, I, love, to, play})P(\text{play}|\text{<BOS>, I, love, to})P(\text{to}|\text{<BOS>, I, love})P(\text{love}|\text{<BOS>, I})P(\text{<BOS>, I}) \\
&= P(\text{<EOS>}|\text{<BOS>, I, love, to, play})P(\text{play}|\text{<BOS>, I, love, to})P(\text{to}|\text{<BOS>, I, love})P(\text{love}|\text{<BOS>, I})P(\text{I}|\text{<BOS>})P(\text{<BOS>})
\end{aligned}
$$

꽉 찬 / 수어졌을때 / 꽉 찬 / 차어졌을때

이게 최대가 되도록 파라미터 조정

Count Based Approximation

$$
P(\text{<EOS>}|\text{<BOS>, I, Love, to, Play}) \approx \frac{\text{count}(\text{<Bos, I, Love, to, Play, <EOS>})}{\text{count}(\text{<BOS>, I, Love, to, Play})}
$$

좌변: 이러한 확률을 / 우변: 등장 확률을 통해 계산

$$
P(x_n|x_{<n}) \approx \frac{\text{count}(x_1, \cdots, x_n)}{\text{count}(x_1, \cdots, x_{n-1})}
$$

★ 문제점 ★

① 해당 단어/문장이 하나도 없으면? -> $P(\text{sentence}) = 0$

↳ 이는 chain rule에 끼치는 영향이 매우 큼 한 단어만 없어도 문장의 확률은 0

↳ 이를 해결하기 위해 Markov Assumption을 Apply

Markov Assumption -> 학습 corpus에 나오지 못한 문장에 대한 확률값 도출

- 이전 단어를 통해 현재 단어를 예측할려 할때 모든 이전 단어를 보는 대신 지정한 k step 전까지의 단어만을 활용해 현재 단어 예측

$$
P(x|x_{<n}) \approx P(x_n|x_{n-1} \cdots x_{n-k}) \approx \frac{\text{count}(x_{n-k}, \cdots, x_n)}{\text{count}(x_{n-k}, \cdots, x_{n-1})}
$$

n-gram Language Model (Markov Assumption)

$n = k+1$

k	n-gram	명칭
0	1-gram	uni-gram
1	2-gram	bi-gram
2	3-gram	tri-gram

4-gram 부터는 four-gram

- n이 커질수록 오히려 확률이 정확하게 표현되는데 어려움
  n이 ↑ 단어가 없거나 외울 가능성↑ n이 너무 짧게 봐서 의미 가능성↑
  적절한 n 필요

- 보통 3-gram을 많이 사용

- corpus가 많은 경우 4-gram 사용하기도

Smoothing, Discounting (여전히 unseen word sequence에 대해 이슈)

- Markov Assumption을 도입 하였지만 unseen word sequence의 확률 = 0이라는 문제가 아직 존재

↓

그래서 0이 되는 것을 방지하기 위한 기술 등장

Add one smoothing

0이되는것을 방지

$$P(w_t \mid w_{<t}) \approx \frac{C(w_{1:t})}{C(w_{1:t-1})} \approx \frac{C(w_{1:t})+1}{C(w_{1:t-1})+|vocab\_size|}$$

↳ $$P(w \mid w_{<t}) \approx \frac{C(w_{1:t})+m \times P(w)}{C(w_{1:t-1})+m}$$ 추가적으로

kneser-Ney Discounting

핵심 아이디어

다양한 단어 뒤에 나올수록 unseen word sequence에 등장할 확률이 높지 않을까?

↳ 앞에 등장한 단어의 종류가 다양할수록 해당 확률이 높다.

Modified kneser-Ney Discounting 보통 많이 사용