3. 언어 모델

3장 내용

- **언어 모델:** 단어 시퀀스에 확률을 할당하는 것
- **N-gram 언어 모델:** 문장에서 n개의 연속된 단어에 대한 모델
- **Perplexity:** 언어 모델의 성능을 평가하기 위한 방식

언어 모델(Language model)

- 문장에서의 단어 시퀀스에 확률을 할당하는 것. 이전 단어들이 주어졌을 때 다음 단어를 예측하는데 사용할 수 있음
- 음성 인식이나 번역, 오타 교정 등에서 보다 가능성이 높은 결과를 찾기 위한 수단으로 사용될 수 있음
- **단어 시퀀스의 확률 할당 사례**
  - *P*(나는 버스를 탔다) > *P*(나는 버스를 태운다)
  - 선생님이 교실로 부리나케
    P(달려갔다) > *P*(잘려갔다)
  - *P*(나는 메롱을 먹는다) < *P*(나는 메론을 먹는다)

단어 시퀀스를 확률로 나타내기

- 단어 시퀀스의 확률: 하나의 단어를 w, 단어 시퀀스를 W라고 하면, n개의 단어가 등장하는 시퀀스 W의 확률은 다음과 같음

$$P(W) = P(w_1, w_2, \cdots, w_n)$$

- 다음 단어 등장 확률은 **조건부 확률(conditional probability)**로 표시

$$P(w_n | w_1, w_2, \cdots, w_{n-1})$$

- 전체 단어 시퀀스 W의 확률은 다음과 같이 표시할 수 있음

$$P(W) = P(w_1, w_2, \cdots, w_n) = \prod_{i=1}^{n} P(w_n | w_1, w_2, \cdots, w_{n-1})$$

통계적 언어 모델(Statistical language model)

- **통계적 언어 모델(SLM)**에서는 조건부 확률 관계식을 이용함

- 조건부 확률 A가 발생했을때 B의 확률

$$P(B|A) = \frac{P(A,B)}{P(A)}, \qquad P(A,B) = P(A)P(B|A)$$

- 조건부 확률의 연쇄 법칙(chain rule)

$$P(x_1, x_2, \cdots x_n) = P(x_1)P(x_2|x_1)\cdots P(x_n|x_1 \cdots x_{n-1})$$

문장에 대한 확률

- 문장 'An adorable little boy is spreading smiles'의 확률 $P(\text{An adorable little boy is spreading smiles})$를 식으로 표현하는 방법

$$P(\text{An adorable little boy is spreading smiles}) =$$
$$P(\text{An}) \times P(\text{adorable}|\text{An}) \times P(\text{little}|\text{An adorable}) \times P(\text{boy}|\text{An adorable little})$$
$$\times P(\text{is}|\text{An adorable little boy}) \times P(\text{spreading}|\text{An adorable little is})$$
$$\times P(\text{smiles}|\text{An adorable little boy is spreading})$$

- 문장의 확률을 구하기 위해 각 단어에 대한 예측 확률들을 곱함

한국어 문장 확률 사례

문장	확률
누명을 쓰다 누명을 당하다	0.41 0.02
선생님께는 낡은 집이 한 채 있으시다 진이에게는 존경하는 선생님이 한 분 있으시다	0.12 0.01
진이는 이 책을 세 번을 읽었다 이 책이 진이한테 세 번을 읽혔다 세 번이 진이한테 이 책을 읽혔다	0.47 0.23 0.07

Carpus

카운트 기반의 접근과 한계

- 문장의 확률을 알려면 단어에 대한 예측 확률을 알아야 함
- SLM에서는 코퍼스(corpus) 데이터를 학습하여 근사 확률을 계산함

$$P(\text{is}|\text{An adorable little boy}) = \frac{\text{count}(\text{An adorable little boy is})}{\text{count}(\text{An adorable little boy})}$$

- 어느 정도 정확한 확률을 구하려면 방대한 양의 데이터가 필요함
- **희소 문제(Sparsity problem):** 기계가 훈련한 코퍼스에 'An adorable little boy is' 시퀀스가 존재하지 않으면 이 확률은 0이 됨. 이 문제를 해결하기 위해 smoothing이나 backoff과 같은 일반화 기법을 적용함
- 이와 같은 한계로 인해 언어 모델 기법은 SLM에서 **인공 신경망 언어 모델**로 넘어가게 됨

N-gram 언어 모델

- SLM 방식의 일종인데, 앞의 n–1개까지만 고려함
- 이 방식을 이용하면 코퍼스에서 각 시퀀스를 카운트할 확률이 높아짐
- N-gram

unigrams : an, adorable, little, boy, is, spreading, smiles
bigrams : an adorable, adorable little, little boy, boy is, is spreading, spreading smiles
trigrams : an adorable little, adorable little boy, little boy is, boy is spreading, is spreading smiles
4-grams : an adorable little boy, adorable little boy is, little boy is spreading, boy is spreading smiles

- N-gram 모델에서는 앞의 n–1 개의 단어만 고려함
- n=4의 경우

~~**An adorable little**~~ **boy is spreading** **?** → w 값 찾을 때

n–1개

$$P(w|\text{boy is spreading}) = \frac{\text{count}(\text{boy is spreading } w)}{\text{count}(\text{boy is spreading})}$$

N-gram 언어 모델의 한계

- **희소 문제:** 작은 시퀀스에 대한 등장 확률이 높아지지만 여전히 등장하지 않을 가능성이 있음
- n을 선택하는 것은 trade-off 문제: n을 크게 선택하면 희소문제가 심각해짐. **n은 최대 5를 넘게 잡아서는 안된다고 권장함**

한국어에서의 언어 모델

- 한국어 자연어 처리는 영어보다 훨씬 어려움

1. **한국어는 어순이 중요하지 않다:** 확률 기반 모델이 다음 단어를 예측하기 어려움
   - ① 나는 운동을 합니다 체육관에서.
   - ② 나는 체육관에서 운동을 합니다.
   - ③ 체육관에서 운동을 합니다.
   - ④ 나는 운동을 체육관에서 합니다.
2. **한국어는 교착어이다:** 조사가 붙으므로 이를 분리하는 것이 중요함

   '학생' => 학생이, 학생을, 학생과, 학생에게, 학생처럼, 학생으로
3. **한국어는 띄어쓰기가 제대로 지켜지지 않는다:** 토큰이 제대로 분리되지 않으면 언어모델이 제대로 동작하지 않음

네이버 영화 말뭉치의 표현별 등장 횟수

표현	빈도
내	1309
마음	172
속에	155
영원히	104
기억될	29
최고의	3503
명작이다	298
내 마음	93
속에 영원히	7
기억될 최고의	1
최고의 명작이다	23
영원히 기억될 최고의 명작이다	1
내 마음 속에 영원히 기억될 최고의 명작이다	0

한국어 문장의 발생 확률

- 말뭉치에 있는 문장은 등장 횟수를 이용하여 확률을 구할 수 있음

$$P(\text{명작이다} \mid \text{최고의}) = \frac{Count(\text{최고의, 명작이다})}{Count(\text{최고의})} = \frac{23}{3503}$$

- 말뭉치에 없는 문장은 발생 확률이 0임

$$P(\text{명작이다} \mid \text{내, 마음, 속에, 영원히, 기억될, 최고의})$$

$$= \frac{Count(\text{내,마음,속에,영원히,기억될,최고의,명작이다})}{Count(\text{내,마음,속에,영원히,기억될,최고의})} = \frac{0}{A}$$

발생 확률의 근사적 처리

- 말뭉치에 없는 문장에 대해서 작은 확률값을 배정하기 위해 Back-off와 Smoothing 방식을 사용
- **Back-off 기법**

$$Count(\text{내, 마음, 속에, 영원히, 기억될, 최고의, 명작이다}) \rightarrow 0$$

$$\approx \alpha Count(\text{영원히, 기억될, 최고의, 명작이다}) + \beta$$

→ 작은확률을 assign 하는 방법.
확률은 0으로 안드는것보다는

- **Smoothing 기법:** 등장 빈도 표에 모두 k를 더함

Perplexity

- 언어 모델의 성능을 비교하기 위해 테스트 데이터를 이용하여 평가하는 방식으로 **perplexity(PPL)**가 있음
- 테스트 데이터에서 확률의 역수로 정의되는데, PPL이 낮을수록 언어 모델의 성능이 좋은 것임
- 문장 W의 길이가 N일 때 PPL은 다음과 같이 정의됨

$$PPL(W) = P(w_1, w_2, \cdots w_n)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, \cdots w_n)}}$$

  - 체인 룰을 적용하면 다음과 같이 됨

$$PPL(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, \cdots w_n)}} = \sqrt[N]{\frac{1}{\prod_{i=1}^{N} P(w_i | w_1, w_2, \cdots w_{i-1})}}$$

Perplexity 분기 계수(branching factor)

- PPL은 현재 위치에서 선택할 수 있는 가지의 개수를 의미하므로, 이 숫자가 크면 불확실성이 높다고 할 수 있음 다음 단어가 나타날 확률
- **사례 1:** 주사위를 던질 때 나오는 수열의 PPL은

$$PPL(x) = \left(\frac{1}{6}\right)^{N(-\frac{1}{N})} = 6$$

- **사례 2:** 20,000개의 어휘로 이루어진 기사에서 단어의 출현 확률이 모두 같다면 PPL은 20,000이 됨(불확실성이 높음). 3-gram을 사용한 언어 모델을 적용했을 때 PPL이 30이면 불확실성이 매우 줄어든 것임
- 월스트리트 저널에서 3,800만 개의 단어 토큰에 대해 n-gram 모델의 PPL은 다음과 같이 나왔다고 함

	1-gram	2-gram	3-gram
Perplexity	962	170	109