

# Ch3. 언어 모델

Language Model

# Contents

## 01. 언어모델

(1) 언어모델

(2) 언어 모델 만드는 방법

## 02. 통계적 언어모델

(1) 조건부확률

(2) 희소문제

(3) n-gram

## 03. 펄플렉서티 (Perplexity)



# 01. 언어모델

---



딥 러닝을 이용한 |

딥 러닝을 이용한 부동산가격지수 예측  
딥 러닝을 이용한 자연어 처리 입문  
딥 러닝을 이용한 한국어 의존 구문 분석  
딥 러닝을 이용한 개체명 인식  
딥 러닝을 이용한 차량 번호판 검출  
딥 러닝을 이용한 한국어 의미역 결정  
딥 러닝을 이용한 한국어 형태소의 원형 복원 오류 수정

- 언어모델 (Language Model) : 단어 시퀀스에 **확률 할당**하는 일을 하는 모델
- 언어모델링 (Language Modeling) : 언어 모델이 주어진 단어들로부터 다음 단어 예측하는 작업

## (1) 언어모델

- 단어 시퀀스에 **확률 할당** --> 확률을 통해 보다 적절한 문장(단어) 판단

ex)

P(나는 버스를 탔다)    >    P(나는 버스를 태운다)

P(나는 메롱을 먹는다)    <    P(나는 메론을 먹는다)

## (1) 언어모델

---

- 가장 보편적 방법 : 이전 단어 → 다음 단어 예측

ex) 어제 배가 고파서 카레를 [?]

## (2) 언어모델 만드는 방법

---

- 언어모델 (Language Model) : 단어 시퀀스에 확률 할당하는 일을 하는 모델

1. 통계를 이용한 방법

2. 인공 신경망 이용한 방법



## 02. 통계적 언어모델 (Statistical Language Model, SLM)

---

- 언어 모델의 전통적 접근 방법
- 카운트에 기반하여 확률 계산
- 조건부 확률 → 문장의 확률

## (1) 조건부확률

---

- 조건부 확률의 연쇄 법칙

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

$$P(x_1, x_2, x_3 \dots x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \dots P(x_n|x_1 \dots x_{n-1})$$



n-1개 단어 나열된 상태에서 n번째 단어의 확률

## (1) 조건부확률

- 문장의 확률 = 이전 단어가 주어졌을 때 다음 단어가 등장할 확률의 곱

$$P(W) = P(w_1, w_2, w_3, w_4, w_5, \dots, w_n) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{i-1})$$

ex) An adorable little boy is spreading smiles.

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

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

## (1) 조건부확률


- SLM : 카운트에 기반한 확률 계산

ex) An adorable little boy is spreading smiles.

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

## (2) 희소문제 (Sparsity Problem)

---

- 충분한 데이터 관측하지 못하여 정확히 모델링 X
  - 완화 방법 : n-gram -- 근본 해결책 X
  - 인공 신경망 언어 모델 트렌드
- 


### (3) n-gram

---

- 카운트에 기반한 SLM
- 모든 단어가 아닌 일부 단어만 고려 : 카운트 가능성 ↑

$$P(\text{is}|\text{An adorable little boy}) \approx P(\text{is}|\text{boy})$$

$$P(\text{is}|\text{An adorable little boy}) \approx P(\text{is}|\text{little boy})$$



일부 단어 (임의의 개수)가 N

## (3) n-gram

- n-gram : n개의 연속적 단어 나열

: n-1개 단어로 다음 나올 단어 예측

ex) An adorable little boy is spreading smiles.

n = 1 ) unigrams : an, adorable, little, boy, is, spreading, smiles

n = 2 ) bigrams : an adorable, adorable little, little boy, boy is, is spreading, spreading smiles

n = 3 ) trigrams : an adorable little, adorable little boy, little boy is, boy is spreading, is spreading smiles

n = 4 ) 4-grams : an adorable little boy, adorable little boy is, little boy is spreading, boy is spreading smiles

### (3) (예시) 4-gram

ex) 'An adorable little boy is spreading' [?]

~~An adorable little~~ boy is spreading ?  
무시됨! ↘  
n-1개의 단어

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

## (3) n-gram

---

- n-gram의 한계

- :수식어에 따른 문맥 고려 부족

- :여전히 희소문제 존재

- :적절한  $n$  선택 필요



## 03. 펄플렉서티 (Perplexity, PPL)

---

### <모델 성능 비교>

#### 1. 외부 평가 (extrinsic evaluation)

: 모델에 실제 작업 후 정확도 비교  
비교하는 모델 수 많을 때 많은 시간 필요

#### 2. 내부 평가 (Intrinsic evaluation)

: 모델 내 자신의 성능 수치화 결과 내놓음  
부정확할 수는 있어도 빠르게 식으로 계산



# PPL

---

- 정규화(normalization)된 테스트 데이터에 대한 확률의 역수

$$PPL(W) = P(w_1, w_2, w_3, \dots, w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, w_3, \dots, w_N)}}$$

- 특정 시점에서 평균적으로 몇 개의 선택지를 가지고 고민?

ex. PPL = 10 : 다음 단어 예측하는 모든 시점마다 평균적으로 10개의 단어로 정답 고민

## PPL 최소화 = 문장의 확률 최대화

---

$$PPL(W) = P(w_1, w_2, w_3, \dots, w_N)^{-\frac{1}{N}} = \left(\frac{1}{10}\right)^{-\frac{1}{N}} = \frac{1}{10}^{-1} = 10$$

ex. PPL = 10 : 다음 단어 예측하는 모든 시점마다 평균적으로 10개의 단어로 정답 고민

## 참고)

인공신경망 이용한 언어모델의 PPL < n-gram 이용한 언어모델의 PPL

Model	Perplexity
Interpolated Kneser-Ney 5-gram (Chelba et al., 2013)	67.6
RNN-1024 + MaxEnt 9-gram (Chelba et al., 2013)	51.3
RNN-2048 + BlackOut sampling (Ji et al., 2015)	68.3
Sparse Non-negative Matrix factorization (Shazeer et al., 2015)	52.9
LSTM-2048 (Jozefowicz et al., 2016)	43.7
2-layer LSTM-8192 (Jozefowicz et al., 2016)	30
Ours small (LSTM-2048)	43.9
Ours large (2-layer LSTM-2048)	39.8