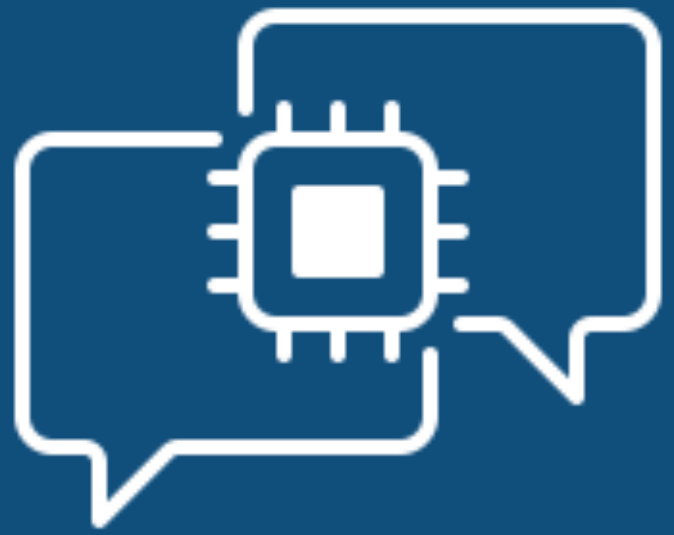


NLP – Week1

16기 허진욱

CONTENTS

ybigta

01 Language Model

Language Model

ybigta

언어 모델?

Language Model

ybigta

언어 모델?

단어 시퀀스에 확률을 할당(assign)하는 모델

다시 말해, 가장 자연스러운 단어 시퀀스(문장)을 찾아내는 모델

Language Model

ybigta

확률 할당

기계 번역

P(나는 버스를 탔다) > P(나는 버스를 태운다)

오타 교정

선생님이 교실로 부리나케

P(달려갔다) > P(잘려갔다)

음성 인식

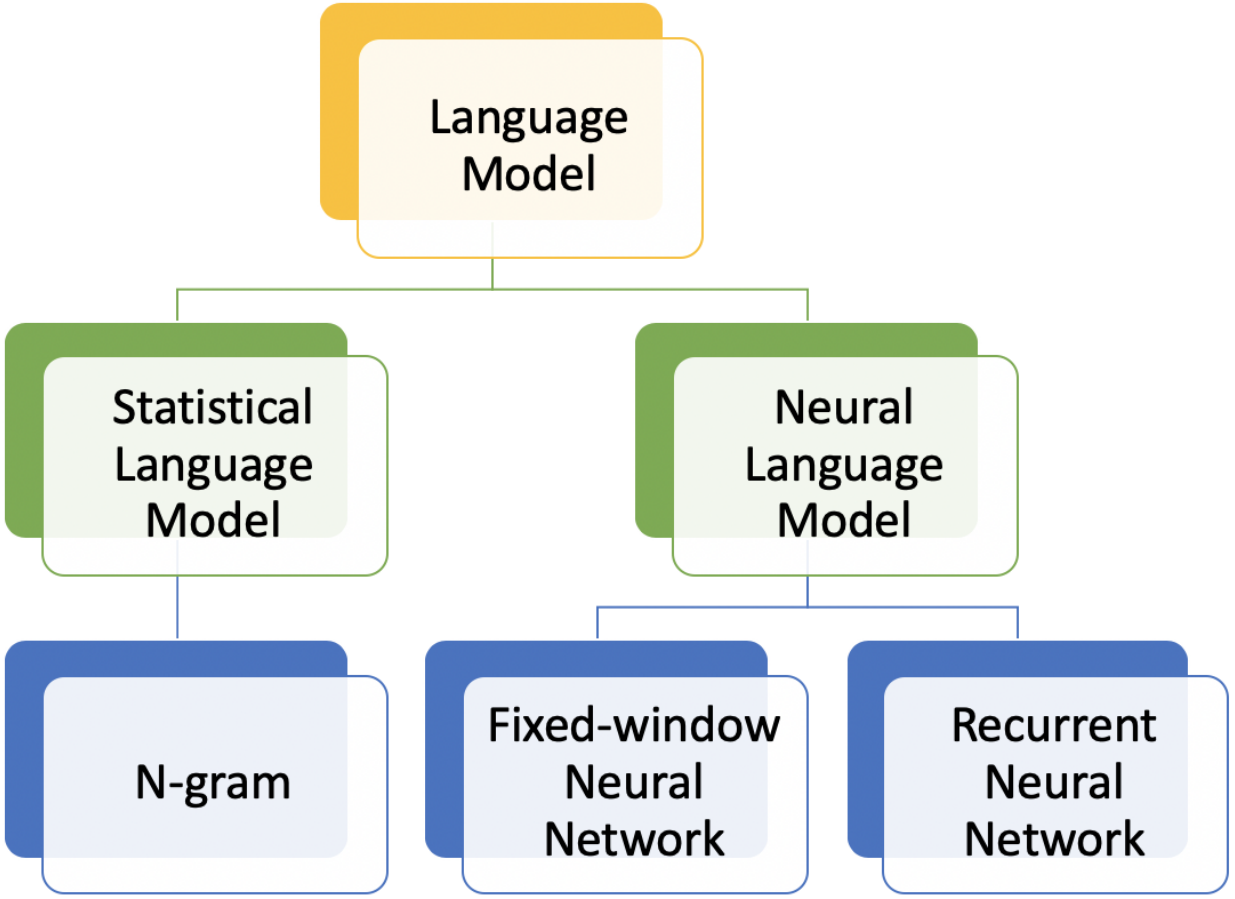
P(나는 메론을 먹는다) > P(나는 메롱을 먹는다)

Language Model

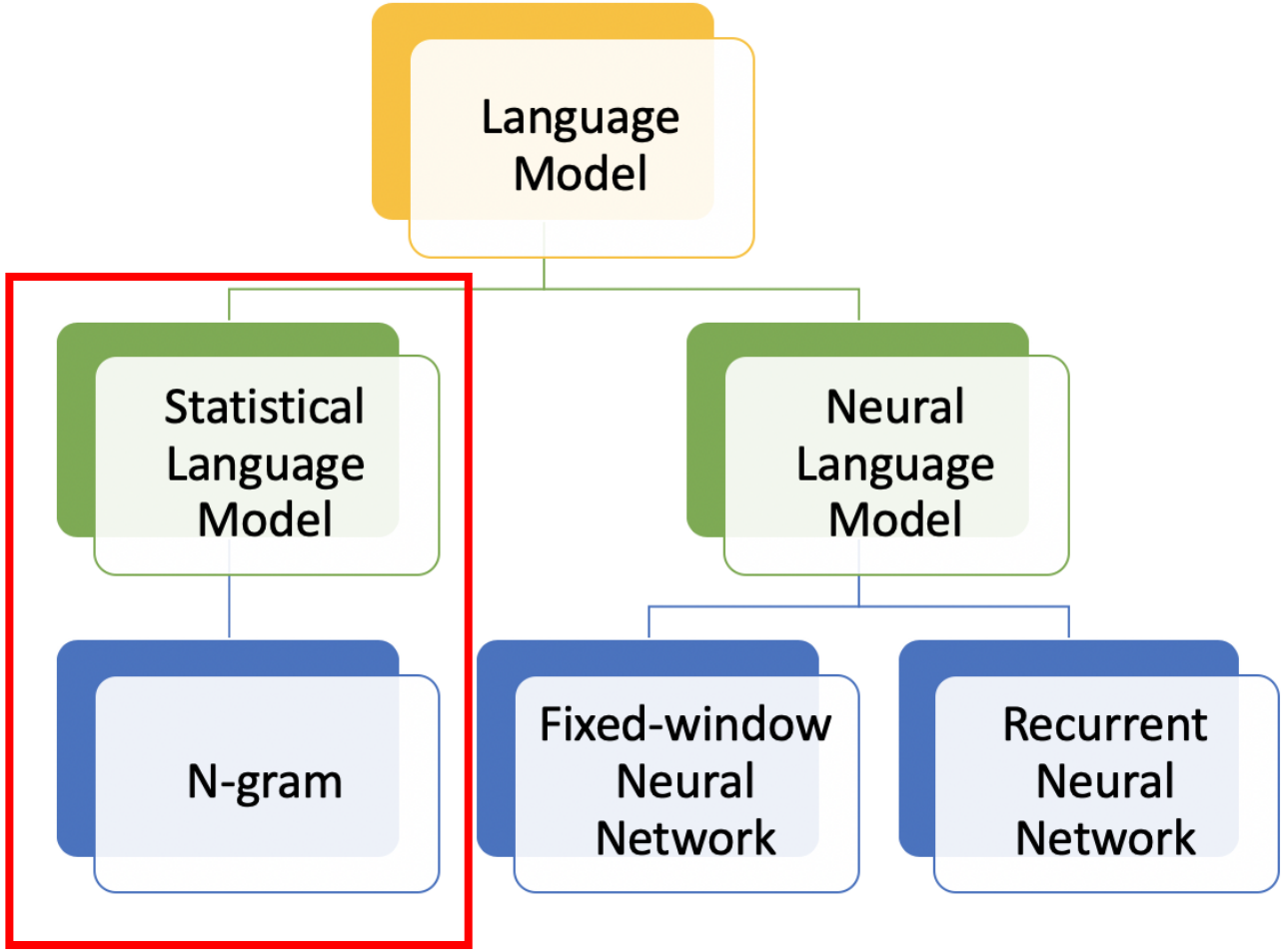
Ex)

공항에 갔는데 지각하는 바람에 비행기를 _____ .

Language Model



Language Model



Statistical Language Model

조건부 확률

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

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

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

Chain Rule!

$$P(x_1, x_2, x_3 \dots x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2) \dots P(x_n|x_1 \dots x_n)$$

Statistical Language Model

문장에 대한 확률

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

$$P(w_1, w_2, w_3 \dots w_n) = \prod_{n=1}^{n} P(w_n | w_1 \dots w_n)$$

=

$P(\text{An}) \times P(\text{adorable}|\text{An}) \times P(\text{little}|\text{An adorable}) \dots$

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

Statistical Language Model

카운트 기반 접근

$$P(B|A)\ ?$$

Statistical Language Model

ybigta

카운트 기반 접근

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

학습한 코퍼스 데이터에서 An adorable little boy가 100번 등장했는데
그 다음에 is가 등장한 경우는 30번이라고 가정하면,

$P(\text{is} \mid \text{An adorable little boy})$는 30%가 된다

Statistical Language Model

한계

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

기계가 훈련한 코퍼스에 An adorable little boy is라는 단어 시퀀스가 없었다면?

An adorable little boy라는 단어 시퀀스가 없었다면?

Statistical Language Model

N-gram Language Model

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

확률을 예측할 때 참고하는 단어의 개수를 N개로 줄인다!

Statistical Language Model

N-gram Language Model

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

다음에 올 단어를 예측할 때 앞에 n-1개를 참고

Statistical Language Model

성능 평가 방법: Perplexity

언어 모델을 평가하기 위한 평가 지표

PPL은 모델이 헷갈려 하는 정도를 의미

수치가 낮을 수록 성능이 좋은 모델

Statistical Language Model

Perplexity

PPL은 문장의 길이를 반영하여 확률을 정규화한 값

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

문장의 확률에 체인룰(chain rule)을 적용

$$PPL(W) = \sqrt[N]{\frac{1}{P(w_1, w_2, w_3, \ldots, w_N)}} = \sqrt[N]{\frac{1}{\prod_{i=1}^{N} P(w_i | w_1, w_2, \ldots, w_{i-1})}}$$

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

Statistical Language Model

Perplexity

PPL은 모델이 특정 시점에서 평균적으로 몇 개의 선택지를 가지고 고민하고 있는지를 의미

EX) 각 시점마다 평균 10개의 단어 중에서 고민

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

실습!

02 Count Based Word Representations

Word Representations

ybigta

단어 표현

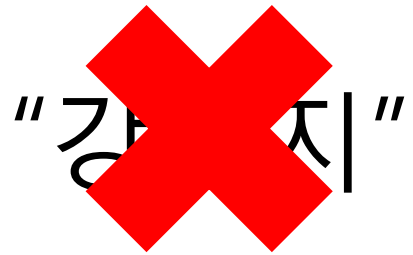
N-gram도 일종의 단어 표현 방법

그러나 머신 러닝 등의 알고리즘이 적용된 자연어 처리를 위해서는 문자의 수치화 필요

Word Representations

"강아지"

Word Representations



- 컴퓨터는 문자보다는 숫자를 더 잘 처리
- 문자의 의미를 숫자로 표현할 수 있도록 변환
- 주로 숫자로 이루어진 백터(vector)의 형태

ex) [0, 0, 0, 0, 1, 0, 0]

Word Representations

ybigta

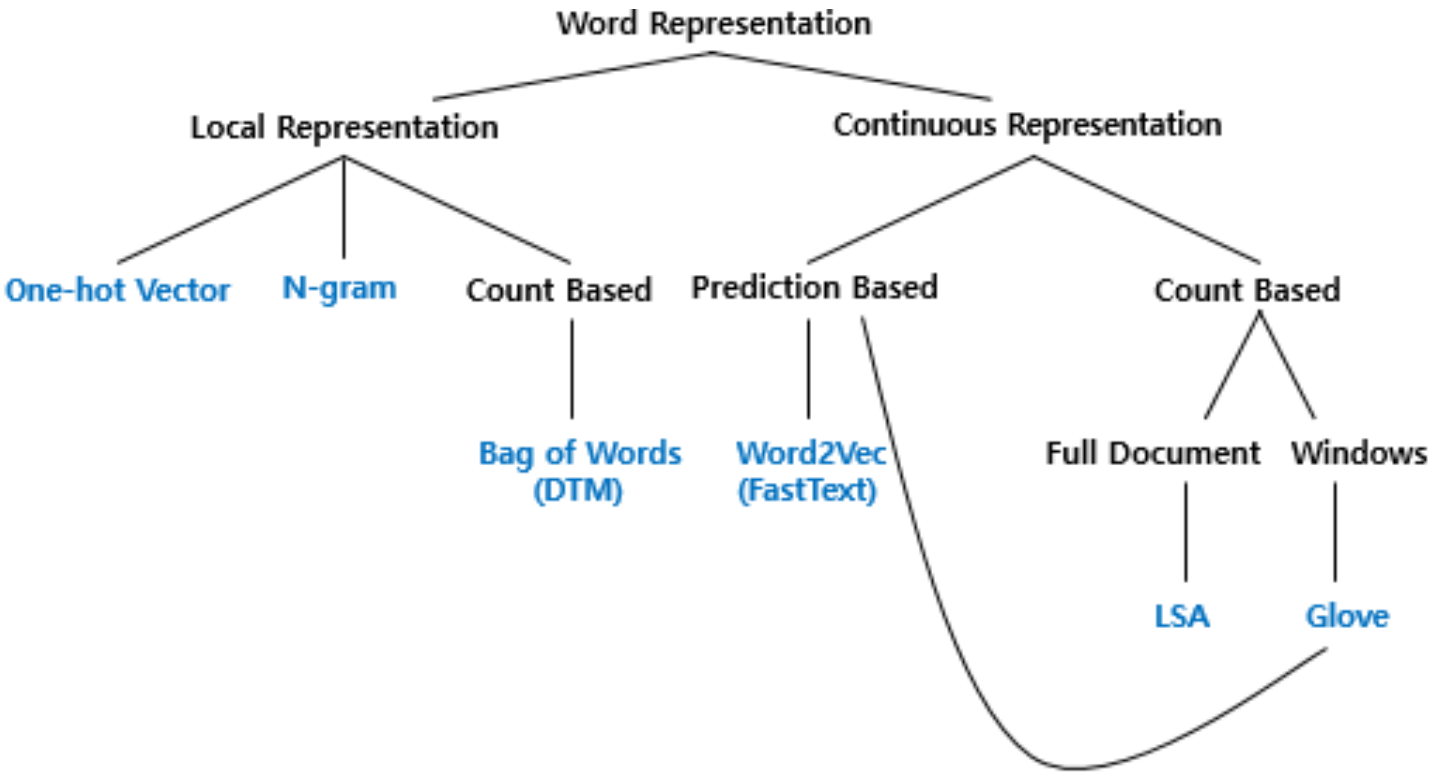
국소 표현 (Local Representation)

- 인덱싱
- 각 단어에 1번, 2번, 3번 등과 같은 숫자를 부여
- Bag of Words

분산 표현 (Distributed Representation)

- 특정 단어를 주변 단어들을 이용하여 표현
- 단어의 의미 표현 가능
- Word2Vec

Word Representations



Word Representations

ybigta

BoW (Bag of Words)

- 문서 내에서 단어들의 출현 빈도에 따라 수치화
- 각 단어에 고유한 정수 인덱스를 부여
- 각 인덱스의 위치에 단어 토큰의 등장 횟수를 기록한 벡터 생성

Word Representations

Ex)

정부가 발표하는 물가상승률과 소비자가 느끼는 물가상승률은 다르다.

인덱스 부여

('정부': 0, '가': 1, '발표': 2, '하는': 3, '물가상승률': 4, '과': 5, '소비자': 6, '느끼는': 7, '은': 8, '다르다': 9)

결과

[1, 2, 1, 1, 2, 1, 1, 1, 1, 1]

Word Representations

ybigta

불용어 (Stopwords)

- 갖고 있는 데이터에서 유의미한 단어 토큰만을 선별하기 위한 작업
- 큰 의미가 없는 단어 토큰을 제거하는데 이를 불용어라고 한다.
- Ex) I, my, me, 나, -이다, -해서, 조사, 접미사 등등
- 직접 정의 할 수도 있고 패키지 내에서 미리 정의된 불용어를 사용해도 된다.

Word Representations

문서-단어 행렬 (DTM)

문서1 : 먹고 싶은 사과

문서2 : 먹고 싶은 바나나

문서3 : 길고 노란 바나나 바나나

문서4 : 저는 과일이 좋아요

-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	1	0	1	1	0	0
문서2	0	0	0	1	1	0	1	0	0
문서3	0	1	1	0	2	0	0	0	0
문서4	1	0	0	0	0	0	0	1	1

문서1 = [0, 0, 0, 1, 0, 1, 1, 0, 0]

Word Representations

ybigta

문서-단어 행렬 (DTM)

- 단어 집합의 크기가 벡터의 차원
- 대부분의 값이 0이 됨
- 공간, 계산 리소스 낭비
- 희소 행렬 (Sparse Matrix)
- 단어의 중요도를 알 수 없다

Word Representations

ybigta

TF-IDF

- 단어의 빈도와 역 문서 빈도를 사용
- DTM에서 단어 별 중요도를 계산하여 가중치 부여
- 문서 유사도, 검색 결과 중요도, 문서내 단어 중요도 구하는 작업에 사용

Word Representations

ybigta

TF-IDF

d: 문서
t: 단어
n: 문서의 총 개수

(1) tf(d, t): 특정 문서 d에서 특정 단어 t의 등장 횟수

- DTM과 동일

(2) df(t): 특정 단어 t가 등장한 문서의 수

- 각 문서에서 등장한 횟수는 중요하지 않음
- 오직 특정 단어 t가 등장한 문서의 수만 사용

Word Representations

ybigta

TF-IDF

d: 문서
t: 단어
n: 문서의 총 개수

(3) idf(d, t): df(t)에 반비례하는 수

$$idf(d,t) = log(\frac{n}{1+df(t)})$$

- df의 역수에 log를 취한 값
- 특정 단어 t의 등장한 문서의 개수가 적을수록 idf 값이 높다
- log를 취한 이유는 문서의 개수 n이 커질수록 idf의 값이 기하급수적으로 커지는 것을 막기 위해

Word Representations

TF-IDF

- 모든 문서에서 자주 등장하는 단어는 중요도 낮고
- 자주 등장하지 않는 단어의 중요도가 높다고 판단
- TF에 IDF 가중치를 곱하여 최종 행렬 구함

d: 문서
t: 단어
n: 문서의 총 개수

Word Representations

TF-IDF

- TF

-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	1	0	1	1	0	0
문서2	0	0	0	1	1	0	1	0	0
문서3	0	1	1	0	2	0	0	0	0
문서4	1	0	0	0	0	0	0	1	1

×

- IDF

단어	IDF(역 문서 빈도)
과일이	ln(4/(1+1)) = 0.693147
길고	ln(4/(1+1)) = 0.693147
노란	ln(4/(1+1)) = 0.693147
먹고	ln(4/(2+1)) = 0.287682
바나나	ln(4/(2+1)) = 0.287682
사과	ln(4/(1+1)) = 0.693147
싶은	ln(4/(2+1)) = 0.287682
저는	ln(4/(1+1)) = 0.693147
좋아요	ln(4/(1+1)) = 0.693147

Word Representations

ybigta

TF-IDF

-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	0.287682	0	0.693147	0.287682	0	0
문서2	0	0	0	0.287682	0.287682	0	0.287682	0	0
문서3	0	0.693147	0.693147	0	0.575364	0	0	0	0
문서4	0.693147	0	0	0	0	0	0	0.693147	0.693147

- '과일'이라는 단어는 문서4에서만 등장 했으므로 가중치가 높다
- '바나나'는 두개의 문서에 등장 했기 때문에 가중치가 낮다

03 Document Similarity

Document Similarity

코사인 유사도

- 두 벡터 간 코사인 각도를 이용한 방법
- 두 벡터의 방향이 완전히 동일하면 1, 90도면 0, 180도면 -1을 갖는다
- 두 벡터의 방향이 같을수록 유사도가 높다고 판단

Document Similarity

ybigta

코사인 유사도

$$similarity = cos(\Theta) = \frac{A \cdot B}{||A||\,||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

- 내적의 결과를 총 벡터 크기로 정규화
- L2 Normalization

Document Similarity

ybigta

자카드 유사도

- 두 집합 A와 B가 있을 때 A와 B의 합집합에서 교집합의 비율을 통해 유사도 측정
- 두 집합이 동일하다면 1, 공통된 원소가 없다면 0의 값을 가진다
- 문서간 유사도를 구할 때는 문서에 속한 단어를 원소로 한다

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$J(doc_1, doc_2) = \frac{doc_1 \cap doc_2}{doc_1 \cup doc_2}$$

실습!

과제

수고하셨습니다!