Introduction to NLP for Deep Learning

CH4. 카운트 기반의 단어 표현(Count based word Representation)

자연어 처리의 텍스트 표현

정보검색과 텍스트 마이닝 분야에 활용

DTM(Document Term Matrix) & TF-IDF(Term Frequency-Inverse Document Frequency)

텍스트를 위와 같은 방식들로 수치화를 하고나면, 통계적인 접근 방법을 통해 여러 문서로 이루어진 텍스트 데이터가 있을 때 어떤 단어가 특정 문서 내에서 얼마나 중요한 것인지를 나타내거나, 문서의 핵심어 추출, 검색 엔진에서 검색 결과의 순위 결정, 문서들 간의 유사도를 구하는 등의 용도로 사용할 수 있습니다.

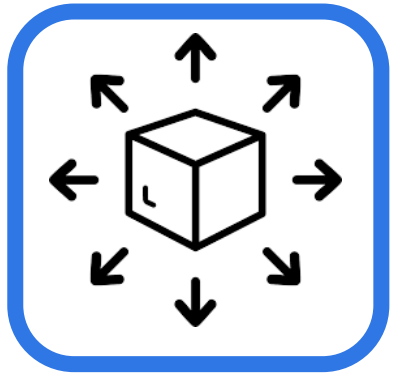
Count based word Representation 단어의 표현 방법

ch4 analysis | 3 page

국소 표현 방법

Local Representation (이산 표현)

해당 단어 그 자체만 보고, 특정값을 맵핑하여 단어를 표현하는 방법으로 단어의 의미와 뉘양스를 표현할 수 없다.

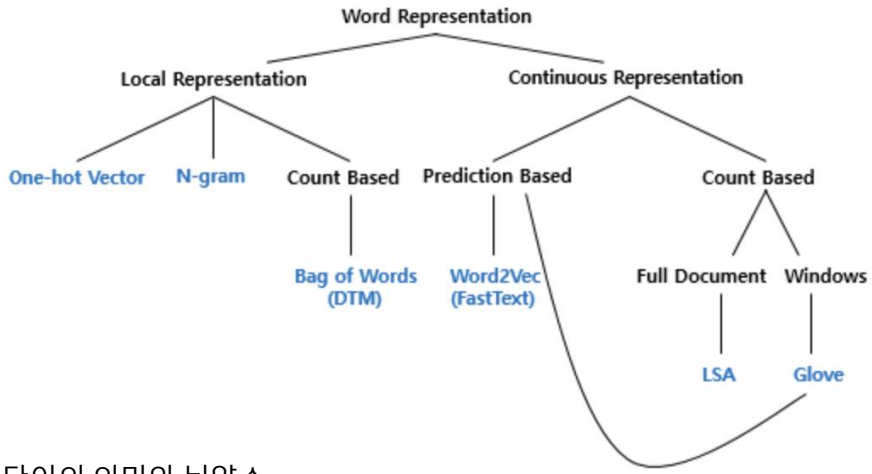


분산 표현 방법

Distributied Representation (연속 표현)

단어를 표현하고자 주변을 참고하여 단어를 표현하는 방법으로, 단어의 의미와 뉘양스를 표현할 수 있다.

Ex) puppy,lovely,cute를 맵핑할 때 함께 자주 등장함을 근거로 뉘양스를 표현



Count based word Representation *Bag of Words (BoW)*

ch4 analysis | 4 page

Bag of Words (BoW)

빈도수 기반의 국소표현

어들의 순서는 전혀 고려하지 않고, 단어들의 출현 빈도 (frequency)에만 집중하는 텍스트 데이터의 수치화 표현 방법.

BoW 생성과정

(1) 각 단어에 고유한 정수 인덱스를 부여합니다. # 단어 집합 생성.
(2) 각 인덱스의 위치에 단어 토큰의 등장 횟수를 기록한 벡터를 만듭니다.

doc1 = "정부가 발표하는 물가상승률과 소비자가 느끼는 물가상승률은 다르다."

```
vocabulary : {'정부': 0, '가': 1, '발표': 2, '하는': 3, '물가상승률': 4, '과': 5, '소비자': 6, '느끼는': 7,
'은': 8, '다르다': 9}
bag of words vector : [1, 2, 1, 1, 2, 1, 1, 1, 1, 1]
```

Count based word Representation

Bag of Words (BoW). (2)

ch4 analysis

5 page

사용 예

각 단어가 등장한 횟수를 수치화하는 텍스트 표현 방법

-> 주로 어떤 단어가 얼마나 등장했는지를 기준으로 문서가 어떤 성격의 문서인지를 판단

즉, 분류 문제나 여러 문서 간의 유사도를 구하는데 사용됨

사이킷 런의 CountVectorizer class

```
from sklearn.feature_extraction.text
import CountVectorizer
corpus = ['you know I want your love. because I love you.']
vector = CountVectorizer()
# 코퍼스로부터 각 단어의 빈도수를 기록
print('bag of words vector :', vector.fit_transform(corpus).toarray())
# 각 단어의 인덱스가 어떻게 부여되었는지를 출력
print('vocabulary :',vector.vocabulary_)
# 단 CounterVectorizer는 기본적으로 한글자보다 큰 단어만 기록
# ex) I 등은 X
```

불용어를 제거한 Bow

```
# 사용자 정의
text = ["Family is not an important thing. It's everything."]
vect = CountVectorizer(stop_words=["the", "a", "an", "is", "not"])
print('bag of words vector :',vect.fit_transform(text).toarray())
print('vocabulary :',vect.vocabulary_)
---------------------------------------------------------------------------
# ConutVectorizer 사용
text = ["Family is not an important thing. It's everything."]
vect = CountVectorizer(stop_words="english")
print('bag of words vector :',vect.fit_transform(text).toarray())
print('vocabulary :',vect.vocabulary_)
---------------------------------------------------------------------------
#NLTK 제공
text = ["Family is not an important thing. It's everything."]
stop_words = stopwords.words("english")
vect = CountVectorizer(stop_words=stop_words)
print('bag of words vector :',vect.fit_transform(text).toarray())
print('vocabulary :',vect.vocabulary_)
```

Count based word Representation 문서단어 행렬 (DTM)

ch4 analysis | 7 page

Document-Term Matrix, DTM

문서 단어 행렬

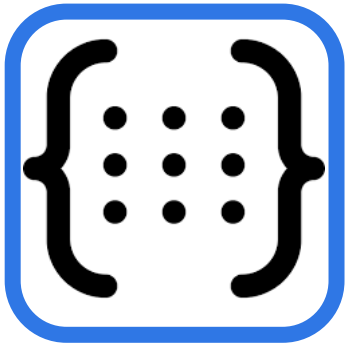
다수의 문서에서 등장하는 각 단어들의 빈도를 행렬로 표현한 것. 즉 쉽게 말하면 각 문서에 대한 BoW를 행렬로 만든 것

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

Count based word Representation *문서단어 행렬 (DTM)*

ch4 analysis | 8 page



DTM 한계(1) 희소표현

Sparse representation

원-핫 벡터는 단어 집합의 크기가 벡터의 차원이 되고 대부분의 값이 0이 되는 벡터이다. 따라서 DTM 또한 원핫 백터와 마찬가지로 대부분의 0 값을 가질 수 있고 이것은 리소스의 낭비로 이어진다. 이러한 행렬/백터들을 희소 행렬/희소 백터라한다.

--> 이러한 이유로 단어집합의 크기를 줄이는 전처리 과정이 반드시 필요하다.(불용어 제거 / 표제어 추출 등)

DTM 한계(2) 단어 빈도수 기반 접근

빈도수 기반 접근 방법 자체의 한계

중요한 단어와 불필요한 단어들이 혼재되어 있으므로 앞서 불용어(stopwords)와 같은 단어들은 빈도수가 높더라도 자연어 처리에 있어 의미를 갖지 못하는 단어가 있을 수 있다.

Ex) the 같은 단어는 의미를 갖지 않는 불용어이지만 문서에서 빈도수가 굉장히 높을 수 있다.

TF-IDF(Term Frequency-Inverse Document Frequency)

문서 단어 행렬

단어의 빈도와 역 문서 빈도(문서의 빈도에 특정 식을 취함)를 사용하여 DTM 내의 각 단어들마다 중요한 정도를 가중치로 주는 방법

$tf(d,t)$

TF(d , t)

특정 문서 d에서 특정 단어 t의 등장 횟수

DTM에서 각 단어들이 가진 값 그 자체이기도 하다

$df(t)$

DF(t)

특정 단어 t가 등장한 문서의 수

특정 단어가 각 문서에서 몇번 등장했는지는 신경쓰지 않고 오로지 몇개의 문서에서만 등장했는지 판단

$IDF(d,t)$

IDF(d , t)

Df(t)에 반비례하는 수

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

단순히 역수를 취했을 경우 기하급수적으로 커지기 때문에 역수를 취함

TF-IDF(Term Frequency-Inverse Document Frequency)

문서 단어 행렬

단어의 빈도와 역 문서 빈도(문서의 빈도에 특정 식을 취함)를 사용하여 DTM 내의 각 단어들마다 중요한 정도를 가중치로 주는 방법

	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

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