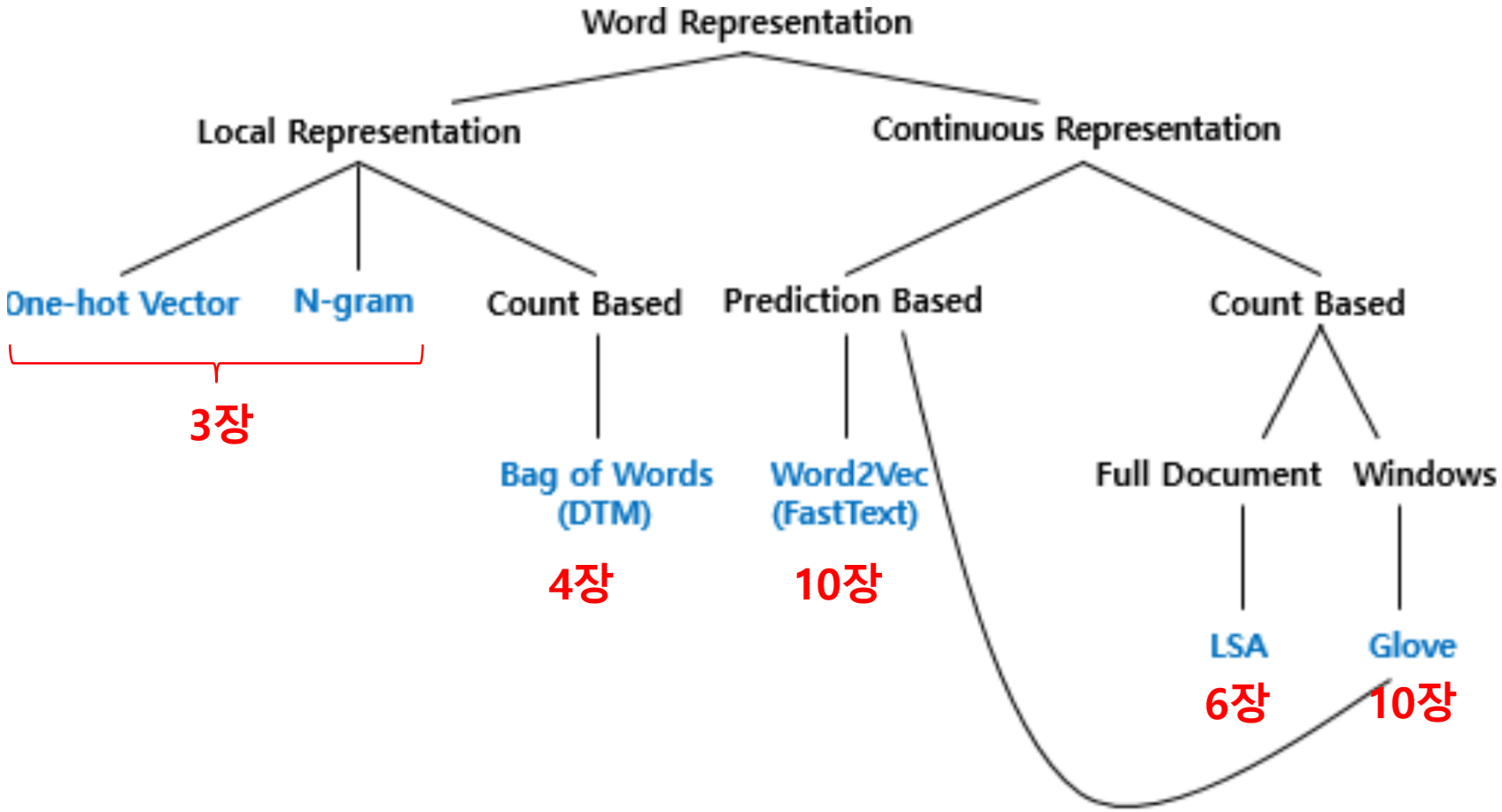
4. 카운트 기반의 단어 표현

4장 내용

- **Bag of Words(BoW):** 텍스트 데이터를 단어들의 출현 빈도로 나타내는 것
- **문서 단어 행렬(Document-Term Matrix):** 다수의 문서에 등장하는 단어들의 빈도를 행렬로 표시
- **TF-IDF(Term Frequency-Inverse Document Frequency):** DTM 내의 각 단어에 가중치를 부여

단어 표현(Word representation)

- **단어 표현:** 각 단어를 숫자로 나타내는 방식



단어 표현 방식 분류

- **국소 표현(Local representation):** 각 단어에 대해 특정값을 부여하는 방식. 이산(Discrete) 표현이라고도 함
  - One-hot, n-gram, BoW 방식 등이 있음
  - 단어의 의미나 뉘앙스를 표현할 수 없음
- **분산 표현(Distributed representation):** 해당 단어를 표현하기 위해 주변 단어를 참고. 연속(Continuous) 표현이라고도 함
  - 사례: puppy(강아지)를 'cute, lovely한 느낌이다'라고 정의
  - 단어의 뉘앙스를 표현할 수 있음

Bag of Words(BoW)

- 텍스트 데이터에 각 단어들이 나온 횟수를 기록하는 것. 단어의 순서는 고려하지 않음
- BoW를 만드는 절차
  1) 각 단어에 고유한 정수 인덱스를 부여
  2) 각 단어 토큰의 등장 횟수를 기록한 벡터를 관리

Bag of Words(BoW) 사례

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

한국어 BoW 사례

- **문장:** 정부가 발표하는 물가상승률과 소비자가 느끼는 물가상승률은 다르다.
- **할당 인덱스:** `('정부': 0, '가': 1, '발표': 2, '하는': 3, '물가상승률': 4, '과': 5, '소비자': 6, '느끼는': 7, '은': 8, '다르다': 9)`
- `BoW: [1, 2, 1, 1, 2, 1, 1, 1, 1, 1]`
- 수행 프로그램은 교재 참고

불용어를 제거한 BoW 만들기

- 불용어는 자연어 처리에서 별로 의미가 없으므로 BoW의 정확도를 높히려면 이들을 제거해야 함
- 영어의 경우 사이킷 런 패키지의 CountVectorizer를 이용하면 불용어를 처리할 수 있음
  - 수행 프로그램은 교재 참고

문서 단어 행렬(Document-Term Matrix: DTM)

- 서로 다른 문서들의 BoW를 결합하여 만든 행렬
- 다수의 문서에서 등장하는 각 단어들의 빈도를 행렬로 표현
- **사례:**

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

문서 단어 행렬의 한계

- **희소 표현(sparse representation)**
  - 벡터의 크기가 단어 집합에서의 단어 수이므로 크기가 커질 수 있음
  - 많은 단어에서 벡터값이 0이 될 수 있음
- **단순 빈도 수 기반 한계**
  - 중요한 단어와 불필요한 단어들이 혼재되어 있을 수 있음
  - 두 문서가 유사한지 비교할 때 한계가 있음

TF-IDF(Term frequency-Inverse document frequency)

- DTM 내에 있는 각 단어들의 중요도를 계산하려고 함
- 문서들의 유사성을 비교할 때 사용할 수 있음
- DTM을 구한 후 TF-IDF 가중치를 부여

TF-IDF 계산 절차

- TF-IDF는 모든 문서에서 자주 등장하는 단어보다 특정 문서에서 자주 등장하는 단어가 중요도가 높다고 판단
- 문서를 d, 단어를 t, 문서의 총 개수를 n이라 가정
  - **tf(d,t)**: 문서 d에서 단어 t가 등장한 횟수
  - **df(t)**: 특정 단어 t가 등장한 문서의 수. 단어의 등장 횟수는 고려하지 않음
  - **idf(d,t)**: df(t)에 반비례하는 수. df(t)가 작을수록 idf는 커짐

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

  - **tf-idf**: tf * idf를 계산하여 얻음

$$tf * idf = tf(d,t) * idf(d,t)$$

TF-IDF 계산 사례

- 문서 단어 행렬에서 DTM을 수정

-	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

- 각 단어에 대한 idf는 다음과 같음

-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
df	1	1	1	2	2	1	2	1	1
idf	ln(4/(1+1)) = 0.693	ln(4/(1+1)) = 0.693	ln(4/(1+1)) = 0.693	ln(4/(2+1)) = 0.288	ln(4/(2+1)) = 0.288	ln(4/(1+1)) = 0.693	ln(4/(2+1)) = 0.288	ln(4/(1+1)) = 0.693	ln(4/(1+1)) = 0.693

TF-IDF 계산 사례(계속)

- **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**

-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
idf	0.693	0.693	0.693	0.288	0.288	0.693	0.288	0.693	0.693

- **tf-idf**

-	과일이	길고	노란	먹고	바나나	사과	싶은	저는	좋아요
문서1	0	0	0	0.288	0	0.693	0.288	0	0
문서2	0	0	0	0.288	0.288	0	0.288	0	0
문서3	0	0.693	0.693	0	0.576	0	0	0	0
문서4	0.693	0	0	0	0	0	0	0.693	0.693

DTM과 tf-idf 구하기

- 교재를 보면 tf-idf를 구현하는 절차를 볼 수 있음
- pandas를 이용하여 DTM과 tf-idf를 구할 수 있음