Text Analytics

Ch4

서수원

Business Intelligence Lab.

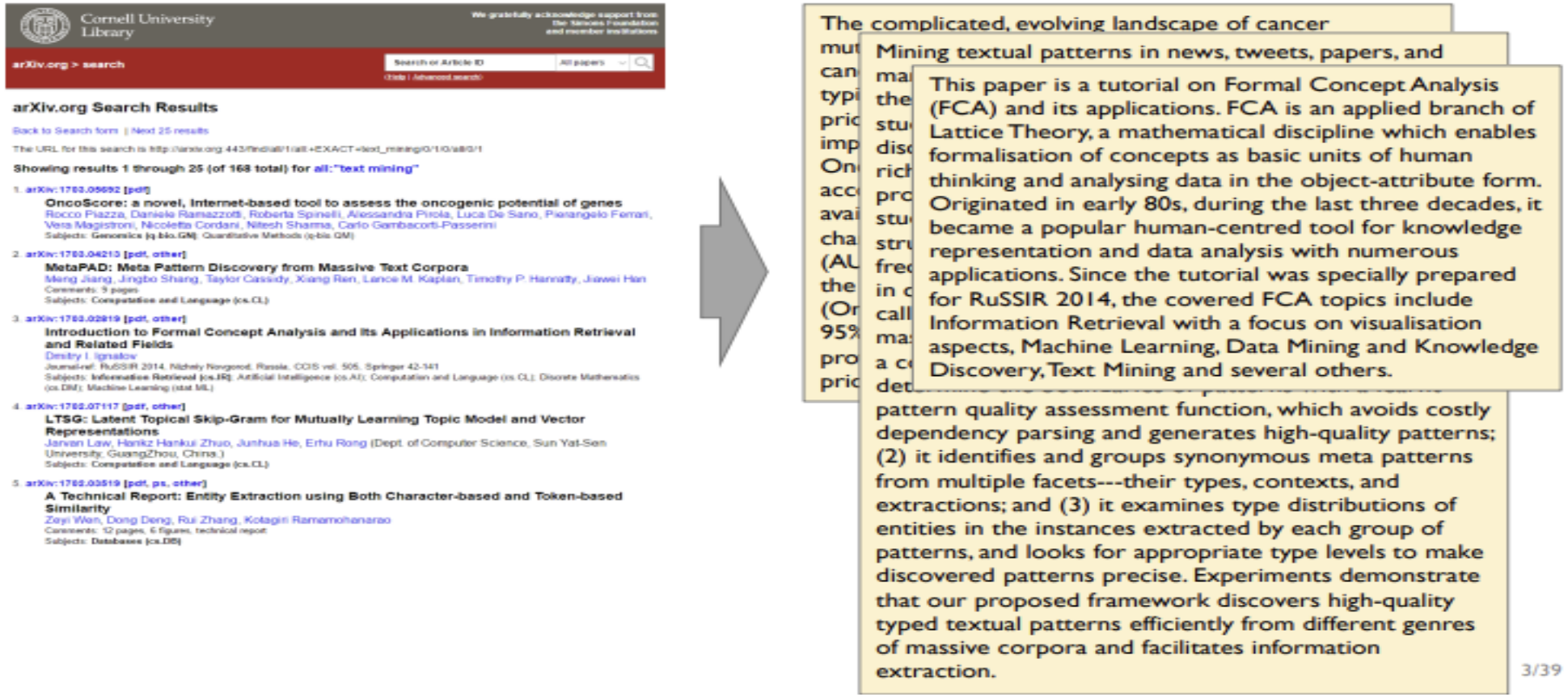
산업경영공학과, 명지대학교

01

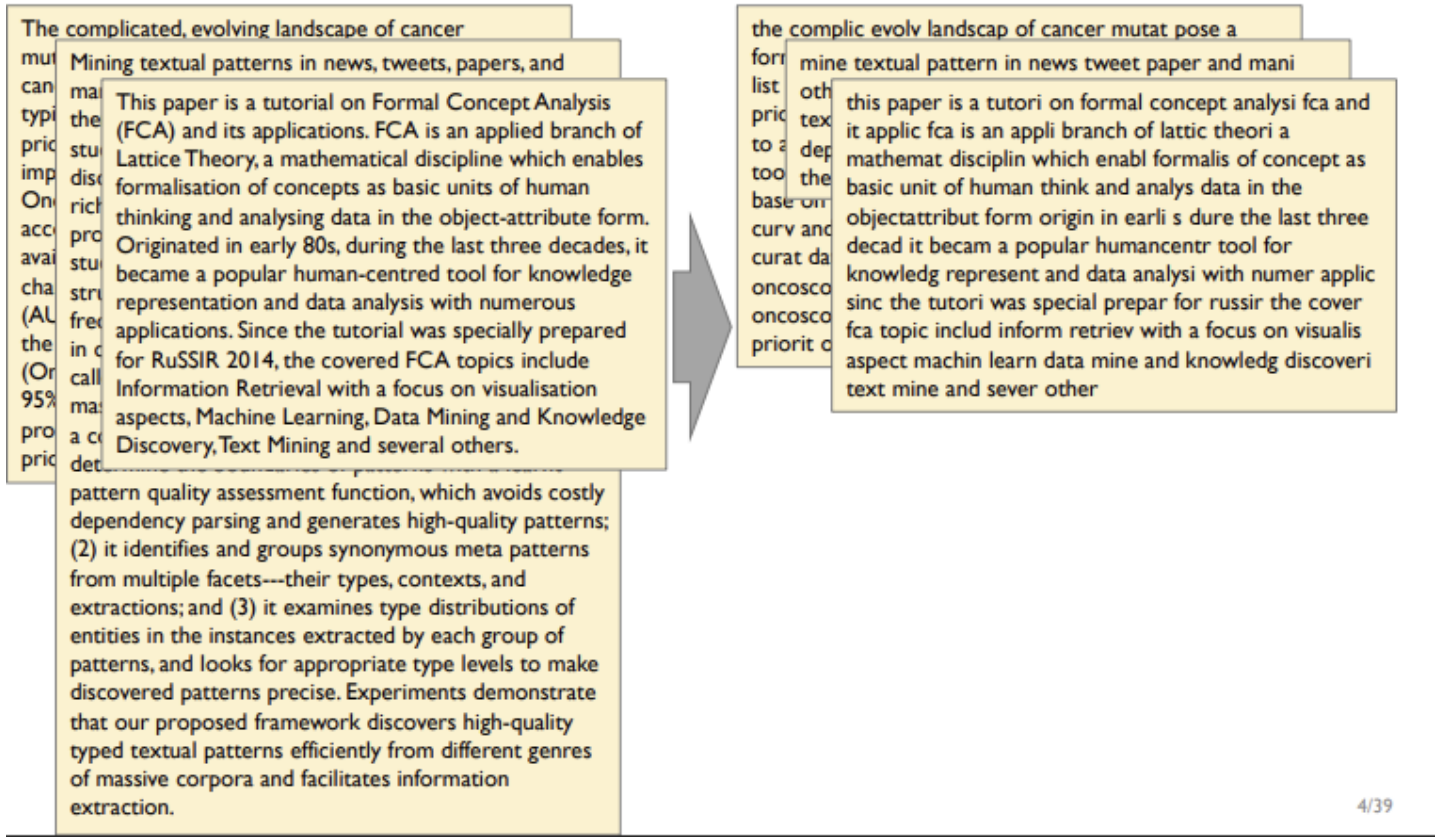
Bag of Words

Bag of words

- 텍스트 데이터를 모으면, 대부분 문서의 길이가 같지 않다.
  - 고전적 머신러닝은 기본적으로 x값인 문서의 차원수가 같아야 한다.
    - ✓ 세개의 문서 다 길이가 다름으로, 고정적 길이로 변환 해주는 과정이 필요하다.



Bag of words

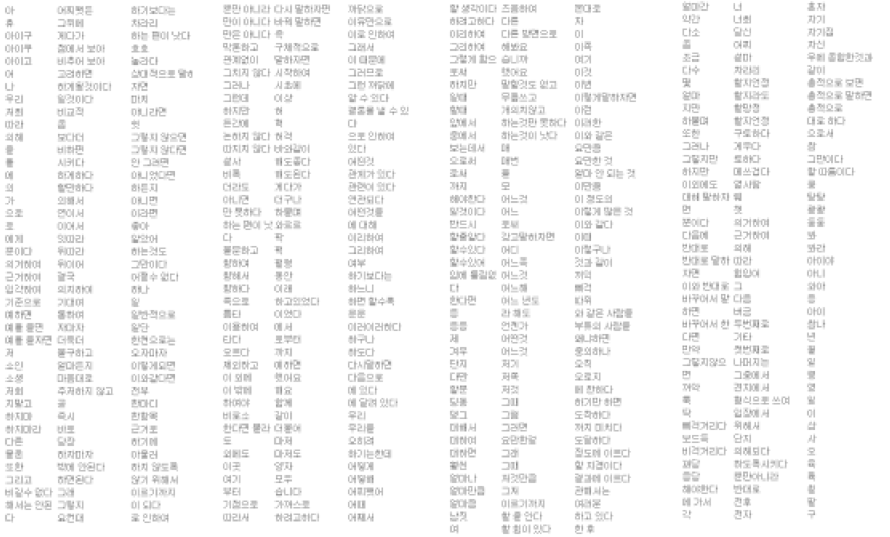


	Var 1	Var 2	…	…	Var P
Doc 1					
Doc 2					
Doc 3					
…					
…					
…					
Doc D					

Stop Words

- A total of 571 stop words

- A total of 677 stop words



...

Bag of words

- Bag of words
  - 가변길이의 문서를 고정길이의 벡터로 변환하는 방식을 의미한다.
    - ✓ 문장을 숫자로 표현하는 방식 중 하나라는 의미이다.
    - ✓ 같은 차원의 벡터로 변환 하는 것을 의미 한다. (이를 document representation)이라고 한다.
  - 단어들의 순서는 전혀 고려하지 않고 단어의 출현 빈도에만 집중하는 방식이다.(count 기반)

Bag of words

- Bag of words
  - 가변길이의 문서를 고정길이의 벡터로 변환하는 방식을 의미한다.
    - ✓ 문장을 숫자로 표현하는 방식 중 하나라는 의미이다
    - ✓ 같은 차원의 벡터로 변환 하는 것을 의미 한다. (이를 document representation)이라고 한다.
  - 단어들의 순서는 전혀 고려하지 않고 단어의 출현 빈도에만 집중하는 방식이다.

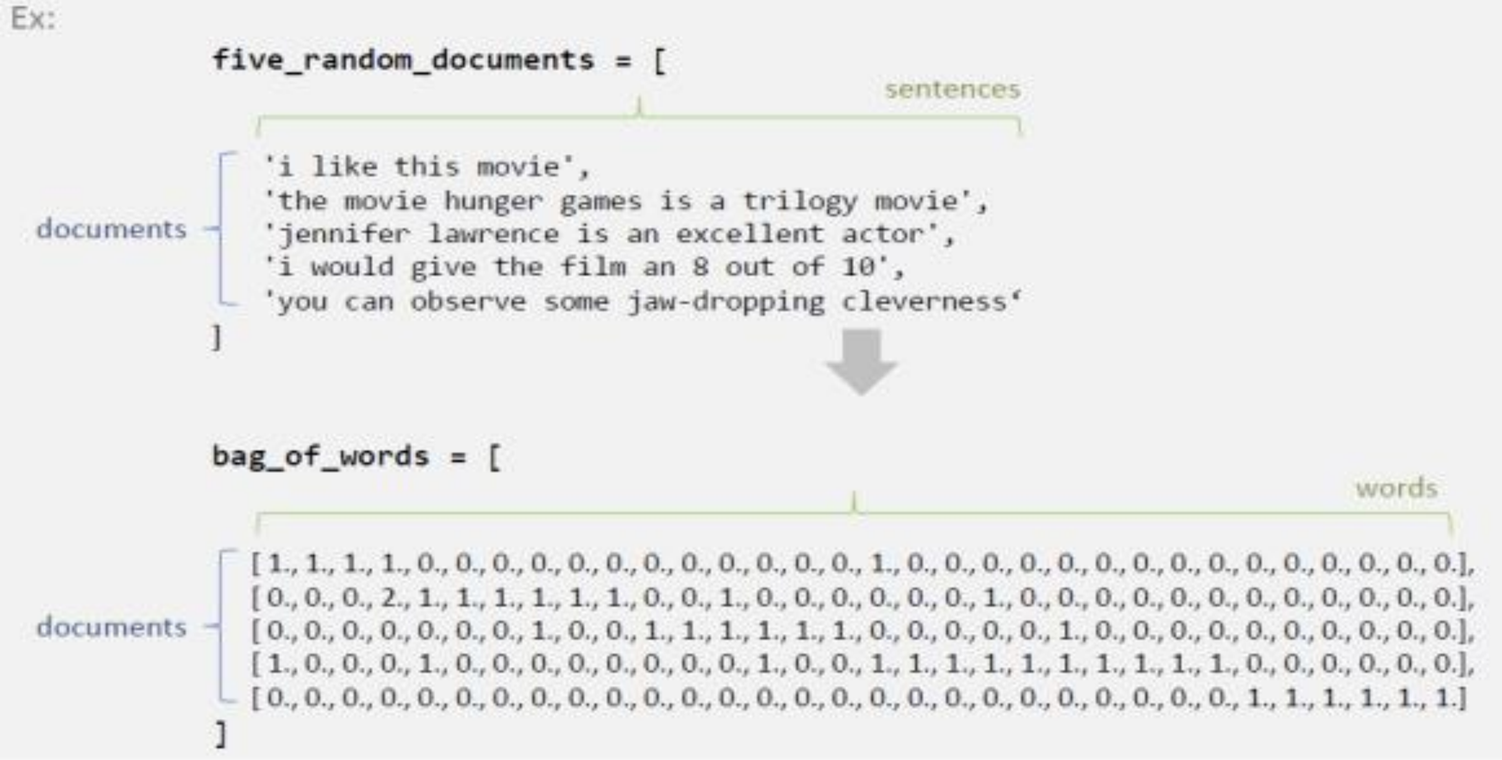
txt1 : I don't love dog. I love cat not dog.

txt2 : I love dog. dog love me too.

	I	love	dog	don't	cat	me	too	not
txt1	2	2	2	1	1	0	0	1
txt2	1	2	2	0	0	1	1	0

Bag of words

- Bag of words
  - 가변길이의 문서를 고정길이의 벡터로 변환하는 방식을 의미한다.
    - ✓ 문장을 숫자로 표현하는 방식 중 하나라는 의미이다
    - ✓ 같은 차원의 벡터로 변환 하는 것을 의미 한다. (이를 document representation)이라고 한다.
  - 단어들의 순서는 전혀 고려하지 않고 단어의 출현 빈도에만 집중하는 방식이다.



Bag of words

- Bag of words : Term – Document Matrix
  - 두가지 표현 방식이 존재 한다.
    - ✓ Binary representation : 문서에 해당 단어가 등장 했는지 안 했는지 유무만 판별, 등장하면 1 아니면 0 으로 표현한다.
    - ✓ Frequency representation : 등장빈도를 기록한다.

S1: John likes to watch movies. Mary likes too.

S2: John also likes to watch football game.

Binary representation

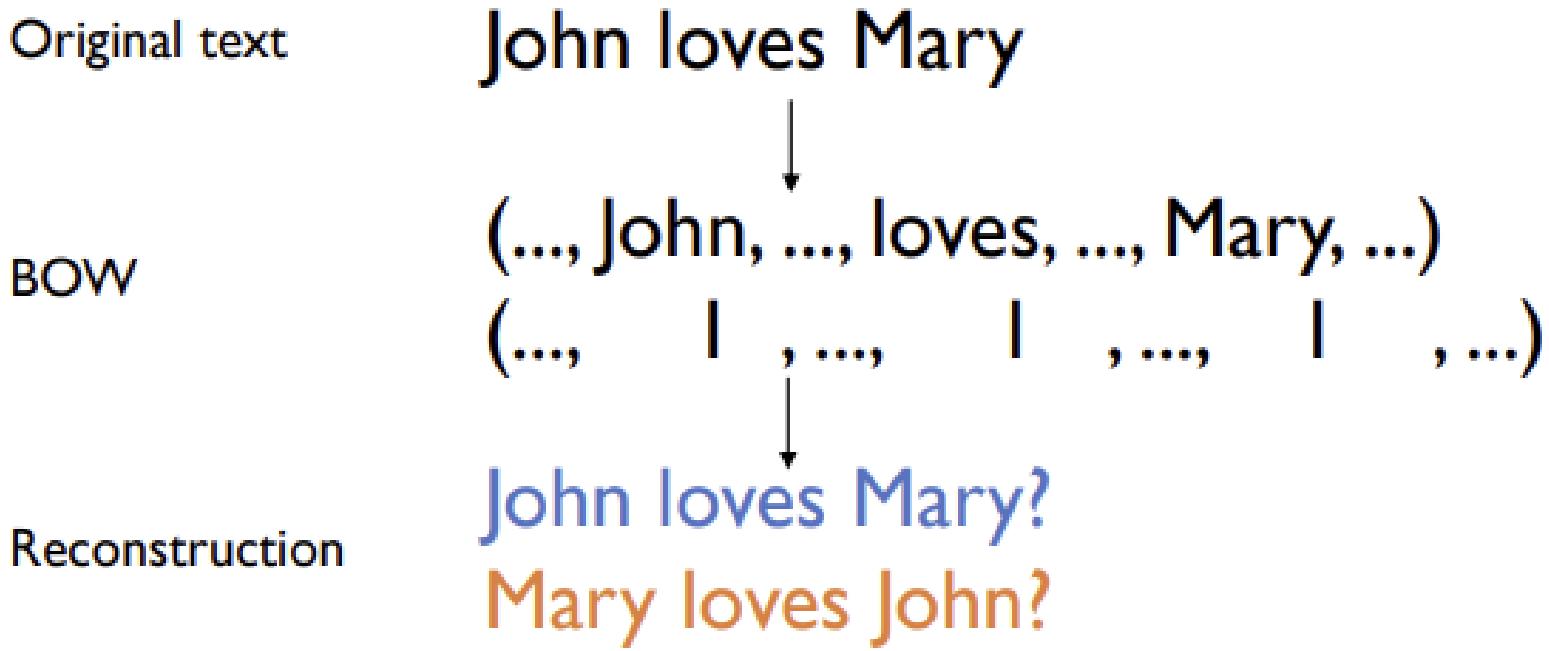
Word	S1	S2
John	1	1
Likes	1	1
To	1	1
Watch	1	1
Movies	1	0
Also	0	1
Football	0	1
Games	0	1
Mary	1	0
too	1	0

Frequency representation

Word	S1	S2
John	1	1
Likes	2	1
To	1	1
Watch	1	1
Movies	1	0
Also	0	1
Football	0	1
Games	0	1
Mary	1	0
too	1	0

Bag of words

- Bag of words : 한계점
  - 벡터 표현은 문서에서 단어의 순서를 고려하지 않는다.
    - ✓ 수원이는 커피를 마시고, 지훈이는 라떼를 마신다. = 지훈이는 커피를 마시고, 수원이는 라떼를 마신다.
  - 벡터로 표현하게 되면 다시 원 문장으로 돌릴 수 없다.



02

Word weighting

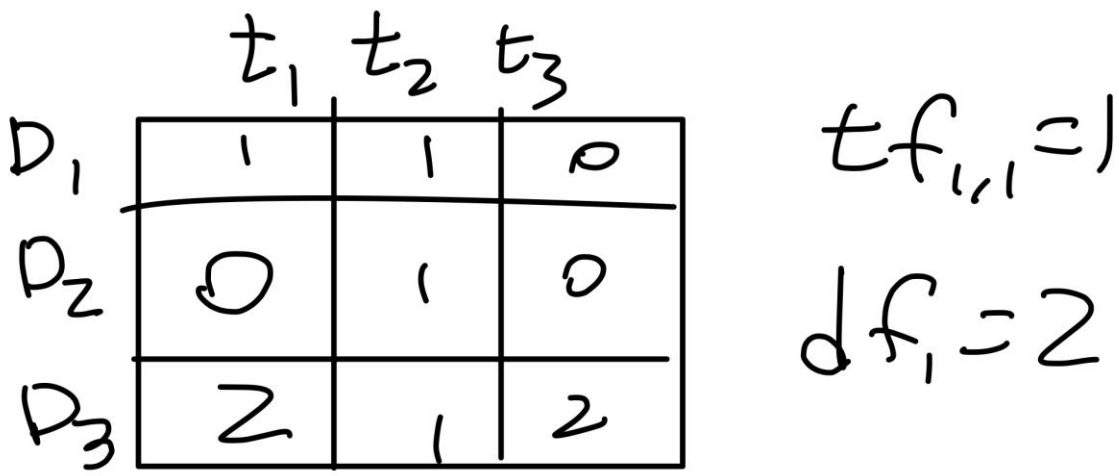
Word Weighting : Term-Frequency

- Word Weighting : Term-Frequency
  - 어떤 문서에서 어떤 단어가 중요한지 파악하기 위함이다.
  - 문서와 단어의 출현 빈도수를 확인한다.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

Word Weighting : Document Frequency

- Word Weighting : Document - Frequency
  - 어떤 코퍼스에서 어떤 단어가 중요한지 파악하기 위함이다.
  - 특정 단어가 특정 문서에 몇 번 등장했는지 보다, 등장한 문서의 수 에만 관심을 갖는다.
  - 많이 출현할수록 덜 중요한 단어일 확률이 높다.
    - ✓ the, is,…



Word Weighting : Inverse Document Frequency

- Word Weighting : Inverse Document - Frequency
  - ✓ $idf_t = \log_{10}(N/df_t)$
  - N은 코퍼스 문서의 개수를 의미한다.
  - Log를 취하는 이유는 값이 기하급수적으로 증가하는 경향이 있기 때문이다.
  - 밑 표는 단어가 100만개가 들어있는 코퍼스의 기준으로 예시를 들은 것이다.

term	df_t	idf_t
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

Word Weighting : TF – IDF

- Word Weighting : TF – IDF
  - TF – IDF 값이 낮으면 중요도가 낮고, 크면 중요도가 크다.

$$TF - IDF(w) = tf(w) \times \log\left(\frac{N}{df(w)}\right)$$

Word Weighting : TF – IDF

- Word Weighting : TF – IDF
  - TF – IDF 값이 낮으면 중요도가 낮고, 크면 중요도가 크다.
    - ✓ 너무 고차원이 된다는 단점이 있다.
      - ❖ V-dimensional vector space(v= vocabulary)
      - ❖ 해당 문서가 쓰인 언어에서의 총 단어의 수를 의미한다.
        - » 한국어의 경우 100만개의 단어가 표현된다.
    - ✓ 희소성의 문제가 발생한다.(거의 모든 셀의 값이 0이 된다.)

.

Word Weighting : TF – IDF

- Word Weighting : TF – IDF
  - TF – IDF 값이 낮으면 중요도가 낮고, 크면 중요도가 크다.
    - ✓ 너무 고차원이 된다는 단점이 있다.

	Doc1	Doc2	Doc3
Term1	5	0	0
Term2	1	0	0
Term3	5	5	5
Term4	3	3	3
Term5	3	0	1

Doc1	TF	DF	IDF	TF-IDF
Term1	5	1	Log3	5log3
Term2	1	1	Log3	1log3
Term3	5	3	Log1	0
Term4	3	3	Log1	0
Term5	3	2	Log(3/2)	3log(3/2)

Word weighting: Term 1 > Term 5 > Term 2 > Term 3 = Term 4

Word Weighting : TF Variants

- Word Weighting : TF Variants
  - TF값의 변형을 할 수 있다.
    - ✓ 한 논문에만 6가지가 넘고, d나 k의 값에 따라서도 다르다.

Definition 2.1 TF Variants: $\mathrm{TF}(t,d)$. $\mathrm{TF}(t,d)$ is a quantification of the within-document term frequency, tf_d. The main variants are:

$$\mathrm{tf}_d := \mathrm{TF}_{\mathrm{total}}(t,d) := \mathrm{lf}_{\mathrm{total}}(t,d) := n_L(t,d) \tag{2.1}$$

$$\mathrm{TF}_{\mathrm{sum}}(t,d) := \mathrm{lf}_{\mathrm{sum}}(t,d) := \frac{n_L(t,d)}{N_L(d)} \left(= \frac{\mathrm{tf}_d}{\mathrm{dl}}\right) \tag{2.2}$$

$$\mathrm{TF}_{\max}(t,d) := \mathrm{lf}_{\max}(t,d) := \frac{n_L(t,d)}{n_L(t_{\max},d)} \tag{2.3}$$

$$\mathrm{TF}_{\log}(t,d) := \mathrm{lf}_{\log}(t,d) := \log(1+n_L(t,d)) \left(= \log(1+\mathrm{tf}_d)\right) \tag{2.4}$$

$$\mathrm{TF}_{\mathrm{frac},K}(t,d) := \mathrm{lf}_{\mathrm{frac},K}(t,d) := \frac{n_L(t,d)}{n_L(t,d)+K_d} \left(= \frac{\mathrm{tf}_d}{\mathrm{tf}_d+K_d}\right) \tag{2.5}$$

$$\mathrm{TF}_{\mathrm{BM25},k_1,b}(t,d) := \mathrm{lf}_{\mathrm{BM25},k_1,b}(t,d) := \frac{n_L(t,d)}{n_L(t,d)+k_1\cdot(b\cdot \mathrm{pivdl}(d,c)+(1-b))} \tag{2.6}$$

Word Weighting : DF Variants

- Word Weighting : DF Variants
  - DF가 0 일 경우 문제가 발생하는 경우가 있다.
    - ✓ 스무딩을 추가한다.

Definition 2.3 DF Variants. $\mathrm{DF}(t,c)$ is a quantification of the document frequency, $\mathrm{df}(t,c)$. The main variants are:

$$\mathrm{df}(t,c) := \mathrm{df}_{\mathrm{total}}(t,c) := n_D(t,c) \tag{2.18}$$

$$\mathrm{df}_{\mathrm{sum}}(t,c) := \frac{n_D(t,c)}{N_D(c)} \left(= \frac{\mathrm{df}(t,c)}{N_D(c)}\right) \tag{2.19}$$

$$\mathrm{df}_{\mathrm{sum,smooth}}(t,c) := \frac{n_D(t,c)+0.5}{N_D(c)+1} \tag{2.20}$$

$$\mathrm{df}_{\mathrm{BIR}}(t,c) := \frac{n_D(t,c)}{N_D(c)-n_D(t,c)} \tag{2.21}$$

$$\mathrm{df}_{\mathrm{BIR,smooth}}(t,c) := \frac{n_D(t,c)+0.5}{N_D(c)-n_D(t,c)+0.5} \tag{2.22}$$

Definition 2.4 IDF Variants. $\mathrm{IDF}(t,c)$ is the negative logarithm of a DF quantification. The main variants are:

$$\mathrm{idf}_{\mathrm{total}}(t,c) := -\log \mathrm{df}_{\mathrm{total}}(t,c) \tag{2.23}$$

$$\mathrm{idf}(t,c) := \mathrm{idf}_{\mathrm{sum}}(t,c) := -\log \mathrm{df}_{\mathrm{sum}}(t,c) \tag{2.24}$$

$$\mathrm{idf}_{\mathrm{sum,smooth}}(t,c) := -\log \mathrm{df}_{\mathrm{sum,smooth}}(t,c) \tag{2.25}$$

$$\mathrm{idf}_{\mathrm{BIR}}(t,c) := -\log \mathrm{df}_{\mathrm{BIR}}(t,c) \tag{2.26}$$

$$\mathrm{idf}_{\mathrm{BIR,smooth}}(t,c) := -\log \mathrm{df}_{\mathrm{BIR,smooth}}(t,c) \tag{2.27}$$

Word Weighting : DF Variants

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha < 1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t \in d}(\mathrm{tf}_{t,d}))}$				

03

N-grams

N-Grams

- N-Gram-based Language Models in NLP
  - 앞의 n-1개의 단어로 다음 단어를 예측한다.

$$P(w_n | w_{n-1}, w_{n-2}, \cdots, w_1) = \frac{P(w_n, w_{n-1}, w_{n-2}, \cdots, w_1)}{P(w_{n-1}, w_{n-2}, \cdots, w_1)}$$

  - Q) One of the hottest topics in artificial intelligence is deep _____?

- N-Gram in Text Mining
  - 토큰화가 쉽다.
    - ✓ Six sigma, supply chain management

N-Grams

- Bigram example
  - ✓ Total counts in a corpus

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

N-Grams

- N-Gram Research
  - 20카테고리의 2만개의 신문기사를 활용하여 N-Gram을 적용 했을 때의 결과 값을 나타낸 논문을 본다.
    - ✓ Gram을 늘리면 error rate가 내려가긴 하지만, 늘린 만큼의 수고만큼 올라가지 않는다.

Pruning	#-grams	Error rate	CPU secs.	No. Features
set-of-words		47.07 ± 0.92	n.a.	71,731
	1	46.18 ± 0.94	12686.12	36,534
DF: 3	2	45.28 ± 0.51	15288.32	113,716
TF: 5	3	45.05 ± 1.22	15253.27	155,184
	4	45.18 ± 1.17	14951.17	189,933
	1	45.51 ± 0.83	12948.31	22,573
DF: 5	2	45.34 ± 0.68	13280.73	44,893
TF: 10	3	46.11 ± 0.73	12995.66	53,238
	4	46.11 ± 0.72	13063.68	59,455
	1	45.88 ± 0.89	10627.10	13,805
DF: 10	2	45.53 ± 0.86	13080.32	20,295
TF: 20	3	45.58 ± 0.87	11640.18	22,214
	4	45.74 ± 0.62	11505.92	23,565
	1	48.23 ± 0.69	10676.43	n.a.
DF: 25	2	48.97 ± 1.15	8870.05	n.a.
TF: 50	3	48.69 ± 1.04	10141.25	n.a.
	4	48.36 ± 1.01	10436.58	n.a.
	5	48.36 ± 1.01	10462.65	n.a.
	1	51.54 ± 0.60	8547.43	n.a.
DF: 50	2	49.71 ± 0.53	8164.27	n.a.
TF: 100	3	51.21 ± 1.26	8079.59	n.a.
	4	51.21 ± 1.26	8078.55	n.a.
	5	51.21 ± 1.26	8147.75	n.a.
	1	52.59 ± 0.71	6609.05	n.a.
DF: 75	2	52.83 ± 0.25	6532.80	n.a.
TF: 150	3	52.36 ± 0.48	6128.49	n.a.
	4	52.36 ± 0.48	6128.49	n.a.
	5	52.36 ± 0.48	6119.27	n.a.

N-Grams

- N-Gram Research

.

Pruning	n-grams	Recall	Precision	F1	Accuracy	No. Features
set-of-words		76.71	83.42	79.92	99.5140	n.a.
DF: 3 TF: 5	1	77.22	83.55	80.26	99.5211	9,673
	2	80.34	82.03	81.18	99.5302	28,045
	3	77.56	82.74	80.07	99.5130	38,646
	4	78.18	82.31	80.19	99.5130	45,876
DF: 5 TF: 10	1	77.19	83.65	80.29	99.5221	6,332
	2	80.05	82.06	81.04	99.5278	13,598
	3	77.96	82.29	80.07	99.5106	17,708
	4	78.21	82.13	80.12	99.5106	20,468
DF: 10 TF: 20	1	76.92	83.99	80.30	99.5241	4,068
	2	79.06	82.04	80.52	99.5177	7,067
	3	77.32	82.67	79.91	99.5096	8,759
	4	76.98	82.91	79.84	99.5096	9,907