정규세션 8주차

ToBig's 11기 김유민

Natural Language Processing Basic

자연어처리 기초

contents

✓ What is NLP?

자연어(Natural Language) = 우리가 일상생활에서 사용하는 언어

자연어처리(Natural Language Processing) = 자연어의 의미를 분석하여 컴퓨터가 처리할 수 있도록 하는 일!

-> 인공지능의 주요 연구 분야!

✓ Why NLP?

- 자연어 이해 및 자연어 처리는 인공지능 분야에 있어서 필수적
- 빅데이터에서 주목받고 있는 것은 '비정형 데이터'
- 비정형 데이터 중 상당 부분이 텍스트 데이터
- 텍스트 데이터는 인간에 대한 정보를 많이 담고 있음

✓ Where NLP is used?

✓ NLP 용어정리

Document(문서)

Corpus(말뭉치): 텍스트(문서)의 집합

Token(토큰): 단어처럼 의미를 가지는 요소

Morphemes(형태소): 의미를 가지는 언어에서 최소 단위

POS(품사): ex) Nouns, Verbs

Stopword(불용어): I, my, me, 조사, 접미사와 같이 자주 나타나지만 실제 의미에 큰 기여를 하지 못하는 단어들

Stemming(어간 추출): 어간만 추출하는 것을 의미(running, runs, run -> run)

Lemmatization(음소표기법): 앞뒤 문맥을 보고 단어를 식별하는 것

TF-IDF: 특정 단어가 문서 내에 얼마나 자주 등장하는 지를 나타내는 TF(단어 빈도)와 어떤 단어가 문서 전체 집합에서 얼마나 많이 나오는지를 나타내는 IDF(역문서 빈도)를 곱한 값

contents

Unit 02 | Process

Data Collection

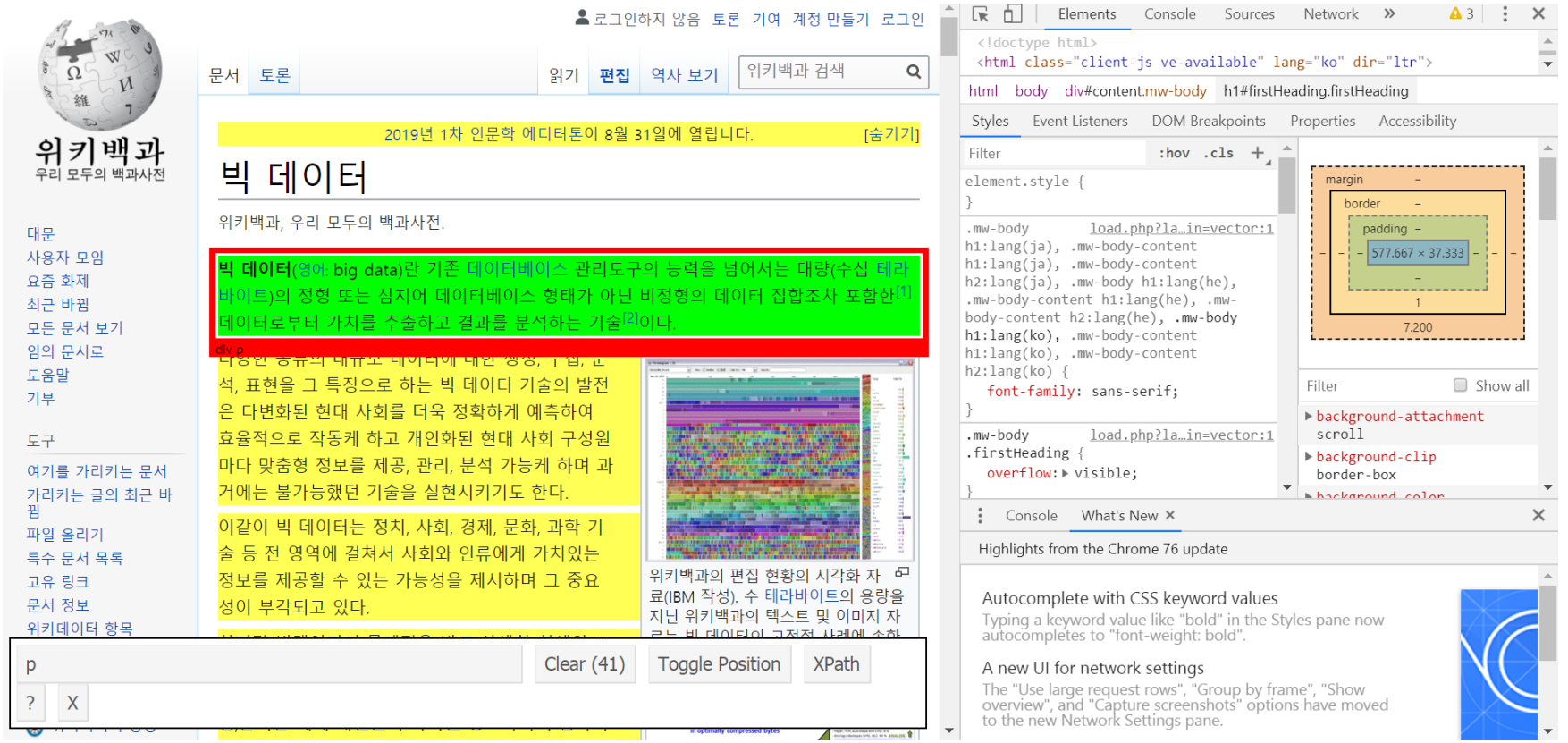
Embedding

Network

Tokenizing

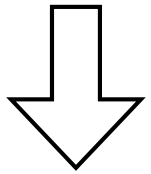
Similarity

Step 1. Data Collection



Step 2. Tokenizing

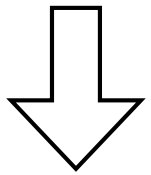
나는 그 사람이 아프다



'나', '는', '그', '사람', '이', '아프', '다'

Step 3. Embedding

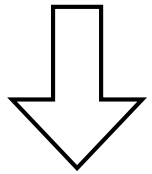
나는 그 사람이 아프다



'나': [0.1234, 0.1234] '는': [0.5678, 0.1234] '그': [0.7890, 0.1567]
'사람': [0.9021, 0.4321] '이': [0.0876, 0.3579] '아프': [0.3456, 0.1764]
'다': [0.1234, 0.0399]

Similarity

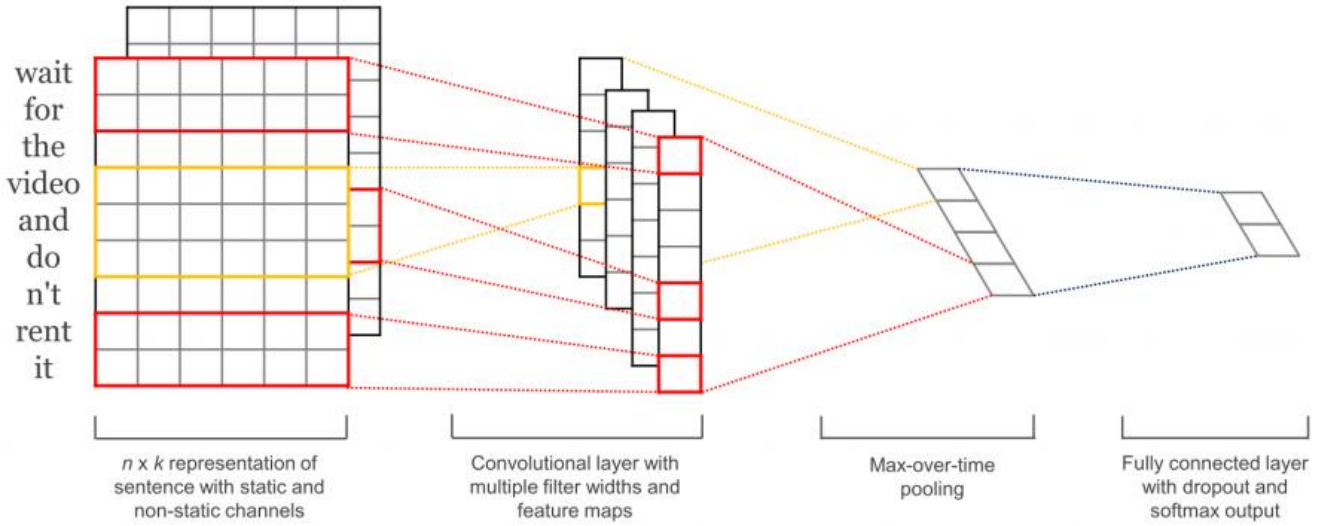
'나': [0.1234, 0.1234] '는': [0.5678, 0.1234] '그': [0.7890, 0.1567]
'사람': [0.9021, 0.4321] '이': [0.0876, 0.3579] '아프': [0.3456, 0.1764]
'다': [0.1234, 0.0399]

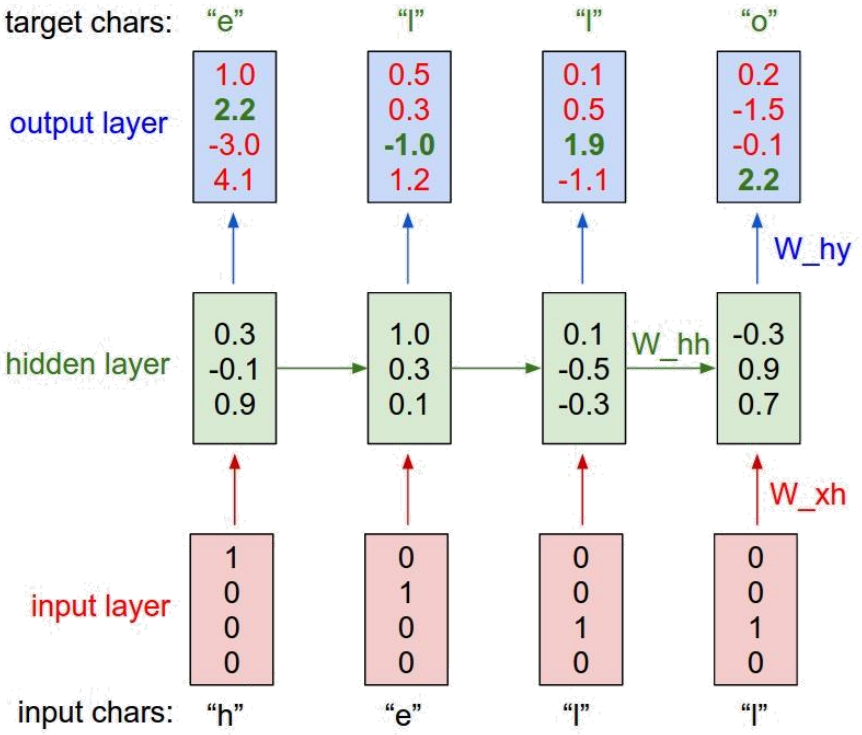


'사람': [0.9021, 0.4321] '아프': [0.3456, 0.1764]

코사인 유사도에 따르면, 이 두 단어는 유사하다고 판단할 수 있다.

Network

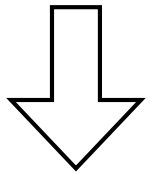

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output


target chars:
"e"
"l"
"l"
"o"
output layer
1.0
2.2
-3.0
4.1
0.5
0.3
-1.0
1.2
0.1
0.5
1.9
-1.1
0.2
-1.5
-0.1
2.2
W_hy
hidden layer
0.3
-0.1
0.9
1.0
0.3
0.1
0.1
-0.5
-0.3
W_hh
-0.3
0.9
0.7
W_xh
input layer
1
0
0
0
0
1
0
0
0
0
1
0
0
0
1
0
input chars:
"h"
"e"
"l"
"l"

contents

Unit 03 | Tokenizing

나는 그 사람이 아프다



'나', '는', '그', '사람', '이', '아프', '다'

나는 그 사람이 아프다

특정 기준에 의해

Text -> Token

‘나’, ‘는’, ‘그’, ‘사람’, ‘이’, ‘아프’, ‘다’

Q. What is "token"?

A. 의미를 가지는 요소!

(ex) 자소/음소, 형태소, 단어, 문장, 문서… etc

Unit 03 | Tokenizing

English

NLTK

Korean

KONLPY

Unit 03 | Tokenizing

Kkma

Komoran

Twitter(Okt)

Hannaum

Mecab

Morphs

Pos Tagging

Nouns

아버지가방에들어가신다

Hannanum	Kkma	Komoran	Mecab	Twitter
아버지가방에 들어가 / N	아버지 / NNG	아버지가방에 들어가신다 / NNP	아버지 / NNG	아버지 / Noun
이 / J	가방 / NNG		가 / JKS	가방 / Noun
시ㄴ다 / E	에 / JKM		방 / NNG	에 / Josa
	들어가 / VV		에 / JKB	들어가신 / Verb
	시 / EPH		들어가 / VV	다 / Eomi
	ㄴ다 / EFN		신다 / EP+EC	

아버지가방에들어가신다

Hannanum	Kkma	Komoran	Mecab	Twitter
아버지가방에 들어가 / N	아버지 / NNG	아버지가방에 들어가신다 / NNP	아버지 / NNG	아버지 / Noun
이 / J	가방 / NNG		가 / JKS	가방 / Noun
시ㄴ다 / E	에 / JKM		방 / NNG	에 / Josa
	들어가 / VV		에 / JKB	들어가신 / Verb
	시 / EPH		들어가 / VV	다 / Eomi
	ㄴ다 / EFN		신다 / EP+EC	

실습 코드!

contents

Q. Tokenizing, 왜 하나요?

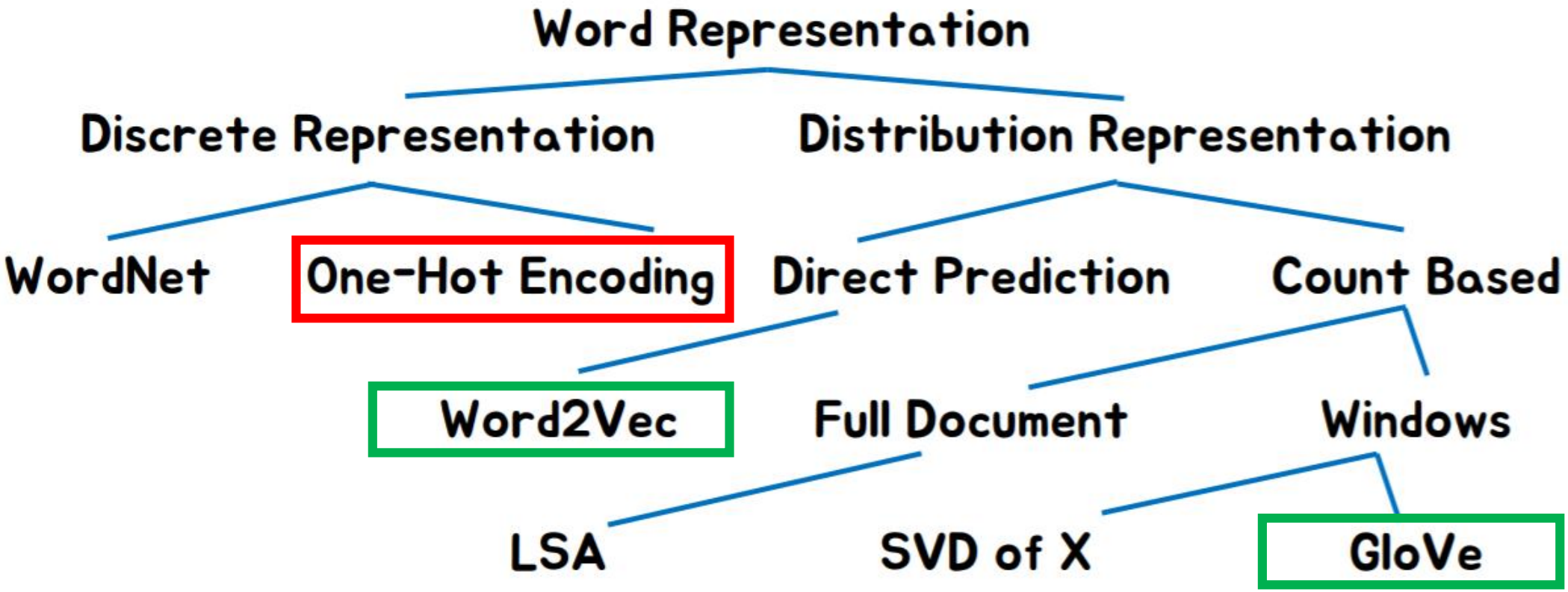
A. 자연어 처리를 위한 의미단위를 만들기 위해!

Q. 컴퓨터가 인간의 언어를 어떻게 이해할 수 있을까?

-> 컴퓨터가 처리할 수 있는 것은 수치 뿐

-> 컴퓨터가 언어의 특성을 이해할 수 있도록 각 token마다 수치를 부여!

Unit 04 | Embedding



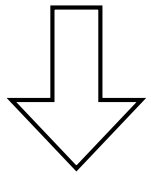
Q. One-Hot Encoding의 문제점?

1. n개 token -> n개 feature: 불필요한 계산이 너무 많다.

2. 유사도 측정이 어려워 유의어, 반의어 등의 언어적 특성을 고려하기 힘들다.

Q. 그냥 One-Hot Encoding 하면 안되나요?

지금은 새벽 3시야 나는 강의를 준비하고 있지 자고 싶다



['지금', '은', '새벽', '3시', '야', '나', '는', '강의',
'를', '준비', '하', '고', '있지', '자', '고', '싶다']

Unit 04 | Embedding

Q. 그냥 One-Hot Encoding 하면 안되나요?

‘지금’: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
‘은’: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
‘새벽’: [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
‘3시’: [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
‘야’: [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
‘나’: [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
‘는’: [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
‘강의’: [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
‘를’: [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
‘준비’: [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
‘하’: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
‘고’: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
‘있지’: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
‘자’: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
‘고’: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
’싶다’: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

- 차원이 너무 커지고 불필요한 계산이 많아짐
- 유사도 측정이 어려워 유의어, 반의어 등의 언어적 특성을 고려하지 못함

Q. 효과적인 방법이 없을까?

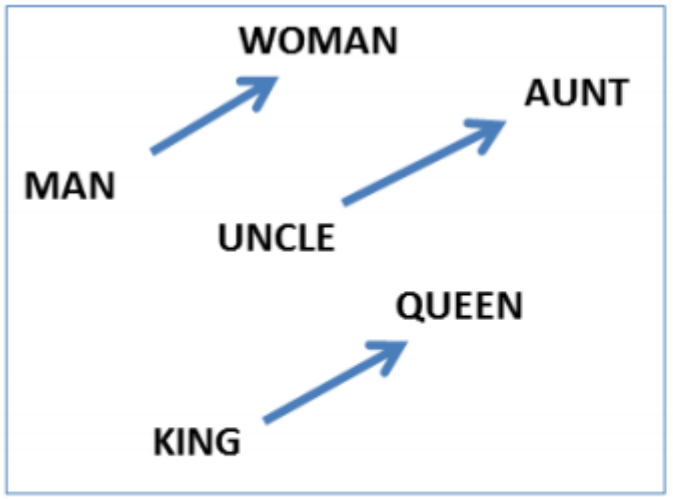
A. 단어를 좀 더 조밀한 차원에 벡터로 표현해보자!

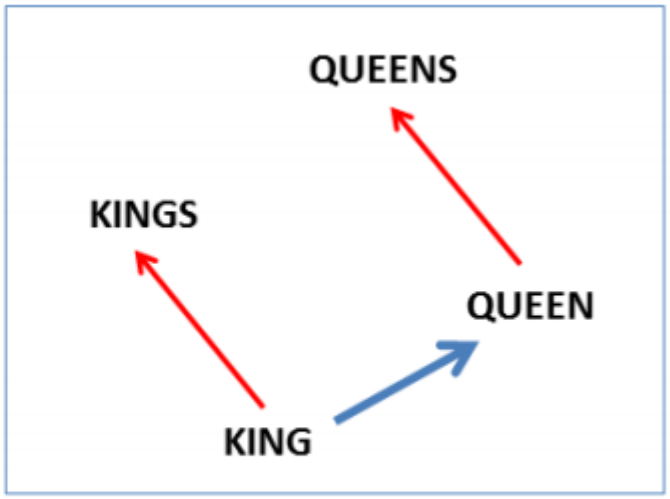
(ex) Word2Vec, Glove, BERT, FastText

[임베딩 모델 1: w2v]

✓ Word2Vec: 말 그대로 word to vector!

1) CBOW
2) Skip-gram
3) Neural Net





(Mikolov et al., NAACL HLT, 2013)

[임베딩 모델 1: w2v]

1) CBOW

내가 어떻게 해야 그대를 잊을 수 있을까

cf) window size?

[임베딩 모델 1: w2v]

1) CBOW

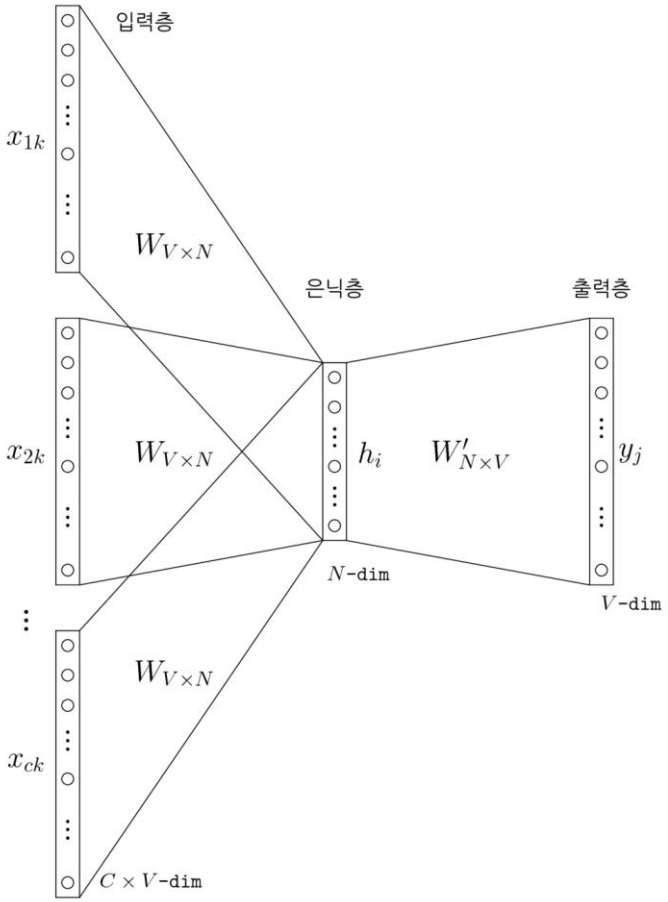
‘내’, ‘가’, ‘어떻게,
‘해야’, ‘그대’, ‘를‘,
‘잊을‘, ‘수’ ‘있을’,
‘까 ’

Center Word	Neighbor Words
‘내’	‘가’, ‘어떻게’
‘가’	‘내‘, ‘어떻게’, ‘해야’
‘어떻게’	‘가’, ‘해야‘, ‘그대’
‘해야’	‘어떻게’, ‘그대‘, ‘를’
‘그대’	‘해야’, ‘를‘, ‘잊을’
‘를’	‘그대’, ‘잊을‘, ‘수’
‘잊을’	‘를’, ‘수’, ‘있을’
‘수’	‘잊을’, ‘있을’, ‘까’
‘있을 ’	‘잊을’, ‘수’, ‘까’
‘까’	‘수’, ‘있을’

[임베딩 모델 1: w2v]

1) CBOW

<Input>
Neighbor Words



<Target>
Center Word

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^{\top} v_{w_I}\right)}$$

- W_O : output word
- W_I : context words

Unit 04 | Embedding

[임베딩 모델 1: w2v]

1) CBOW

center word	context words
[1,0,0,0,0,0,0]	[0,1,0,0,0,0,0] [0,0,1,0,0,0,0]
[0,1,0,0,0,0,0]	[1,0,0,0,0,0,0] [0,0,1,0,0,0,0] [0,0,0,1,0,0,0]
[0,0,1,0,0,0,0]	[1,0,0,0,0,0,0] [0,1,0,0,0,0,0] [0,0,0,1,0,0,0] [0,0,0,0,1,0,0]
[0,0,0,1,0,0,0]	[0,1,0,0,0,0,0] [0,0,1,0,0,0,0] [0,0,0,0,1,0,0] [0,0,0,0,0,1,0]
[0,0,0,0,1,0,0]	[0,0,1,0,0,0,0] [0,0,0,1,0,0,0] [0,0,0,0,0,1,0] [0,0,0,0,0,0,1]
[0,0,0,0,0,1,0]	[1,0,0,1,0,0,0] [0,0,0,0,1,0,0] [0,0,0,0,0,0,1]
[0,0,0,0,0,0,1]	[0,0,0,0,1,0,0] [0,0,0,0,0,1,0]

2) Skip-gram

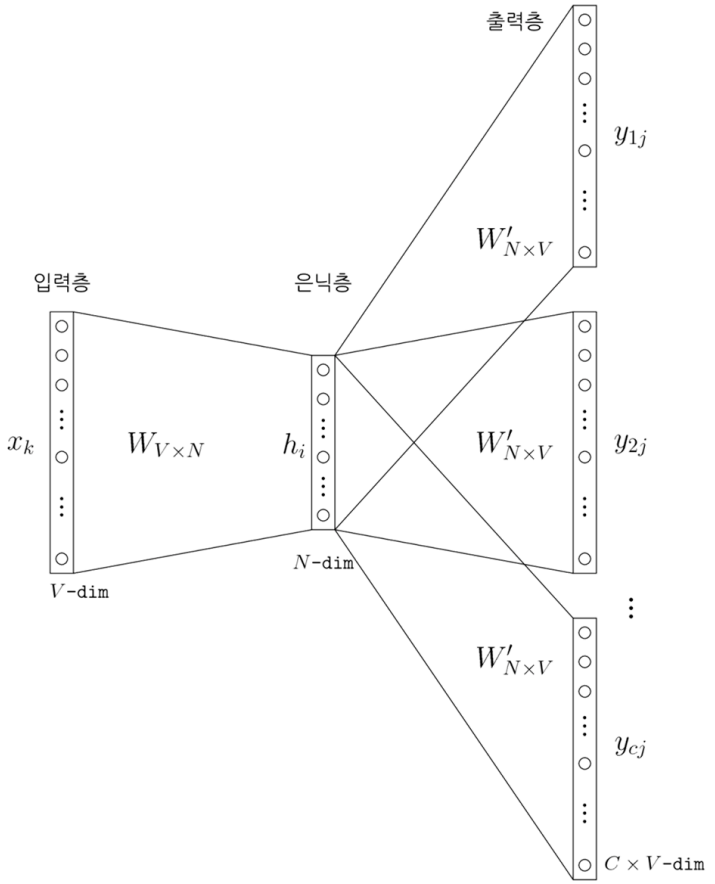
내가 어떻게 해야 **그대를** 잊을 수 있을까

cf) window size?

[임베딩 모델 1: w2v]

2) Skip-gram

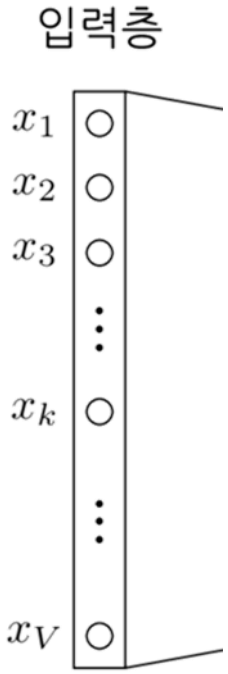
<Input>
Target Word



<Target>
Neighbor Words

[임베딩 모델 1: w2v]

3) Neural Net



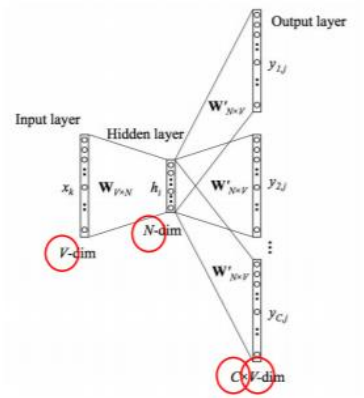
➢ Bow(Bag of Words)

'I': 0
'am': 1
'a': 2
'boy': 3
'girl': 4

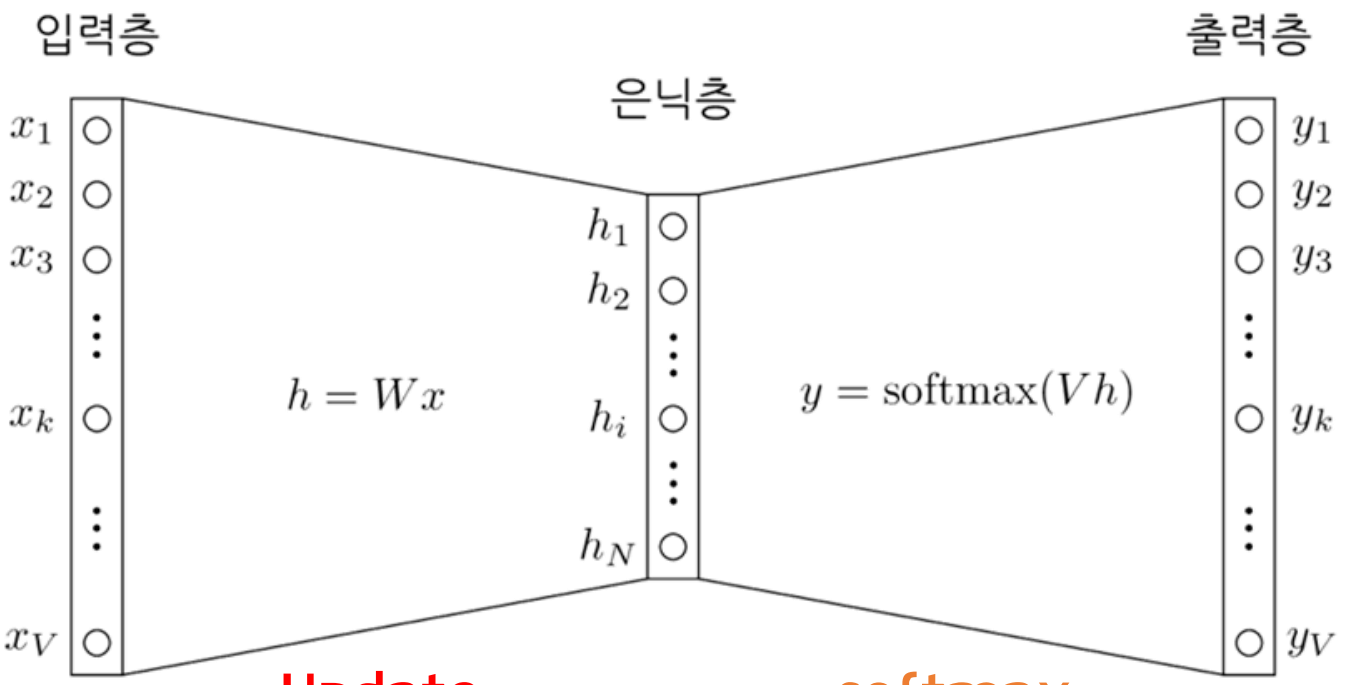
"I am a girl" => [1 1 1 0 1]

[임베딩 모델 1: w2v]

3) Neural Net



BOW Encoding



Target

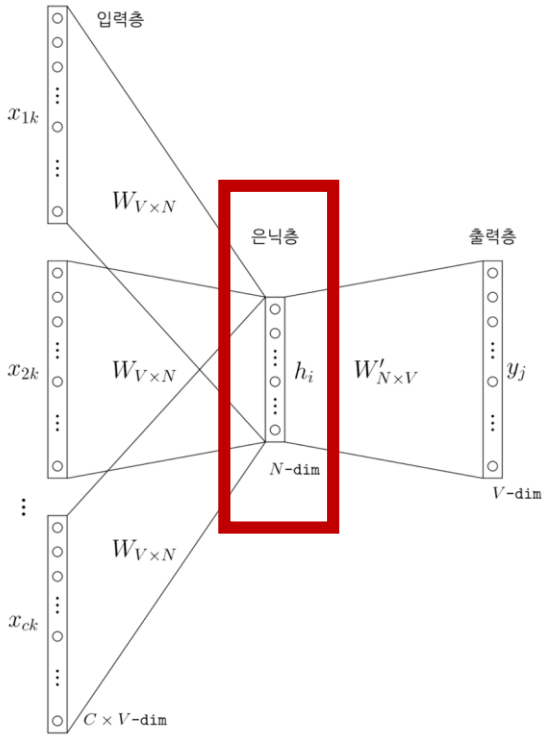
Update

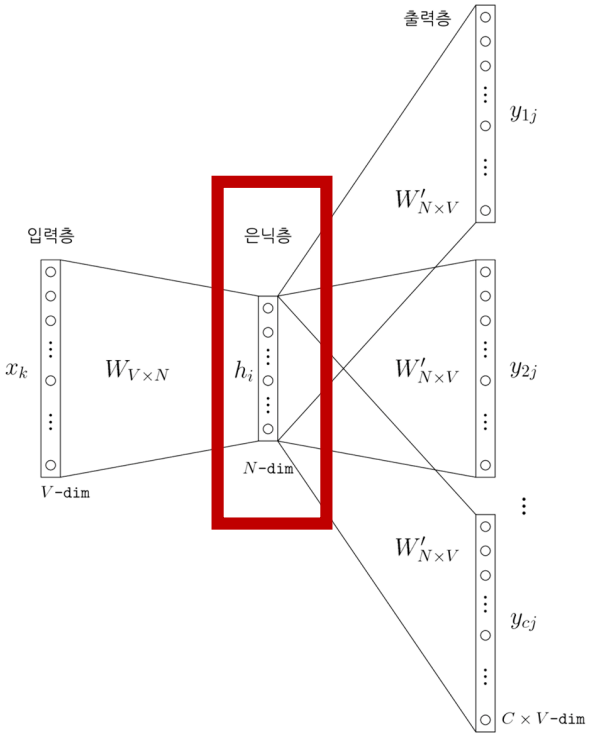
softmax

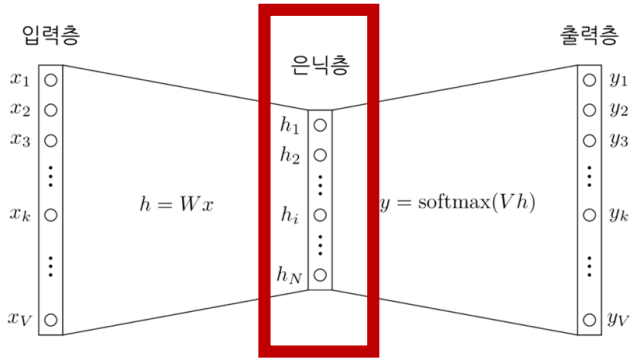
Cross-Entropy

[임베딩 모델 1: w2v]

Word Vector







[임베딩 모델 1: w2v]

✓ Word2Vec의 문제점

① 한번에 하나의 동시 출현만 고려 -> 전체적인 정보 이용 x -> 비효율성&부정확성

② train corpus에 존재하지 않았던 단어의 벡터를 만들어낼 수 없음

[임베딩 모델 2: GloVe]

✓ GloVe

전체 텍스트의 정보를 이용해보자!
train corpus에서 동시에 같이 등장한 단어의 빈도를 세어서
corpus의 단어 개수로 나눠준 **"동시 등장 비율"**을 고려하자!

[임베딩 모델 2: GloVe]

✓ GloVe

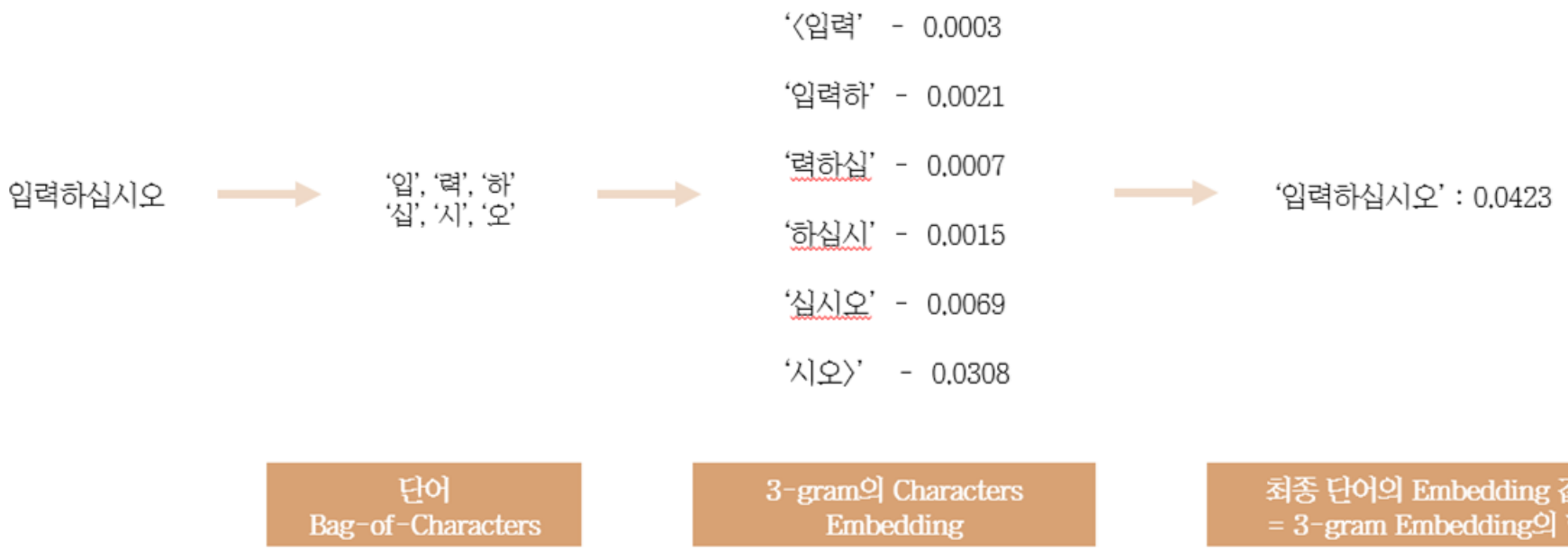
Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

“solid”?
“ice” vs “steam”

[임베딩 모델 3: FastText]

✓ FastText

단어가 아닌 단어 내부의 n-gram이 최소 단위!



[임베딩 모델 3: FastText]

✓ FastText

① train corpus에 존재하지 않았던 단어의 embedding이 가능함 (ex) 'disaster'/'disastrous'

② 희소한 단어에 대해 더 좋은 embedding이 가능함

contents

✓ Similarity

: 유사도를 구해 의미론적 해석을 이끌어낸다!

1) Euclidean Similarity

2) Cosine Similarity

3) Jaccard Similarity

[유사도 1: 유클리디안]

1) Euclidean Similarity

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

-	바나나	사과	저는	좋아요
문서1	2	3	0	1
문서2	1	2	3	1
문서3	2	1	2	2

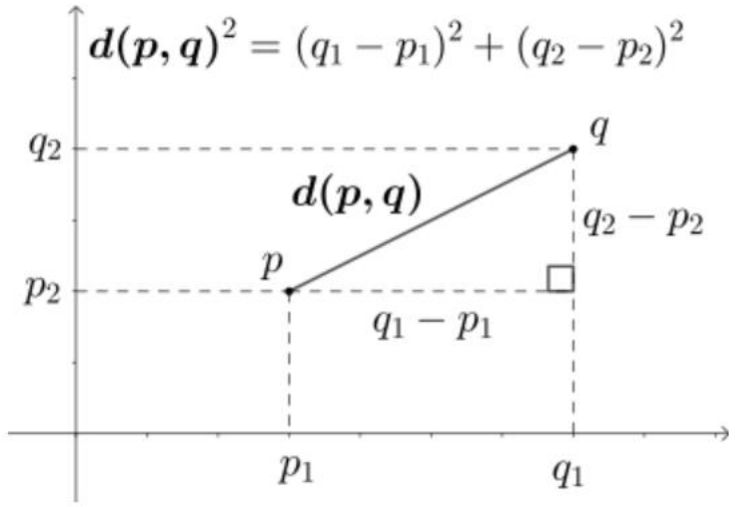
÷

-	바나나	사과	저는	좋아요
문서Q	1	1	0	1

[유사도 1: 유클리디안]

1) Euclidean Similarity

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$



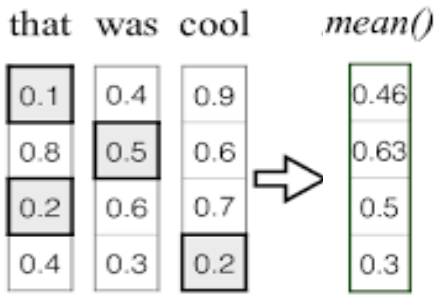

```
import numpy as np
def dist(x, y):
    return np.sqrt(np.sum((x-y)**2))

doc1 = np.array((2,3,0,1))
doc2 = np.array((1,2,3,1))
doc3 = np.array((2,1,2,2))
docQ = np.array((1,1,0,1))

print(dist(doc1, docQ))
print(dist(doc2, docQ))
print(dist(doc3, docQ))

2.23606797749979
3.1622776601683795
2.449489742783178
```

[유사도 2: 코사인]

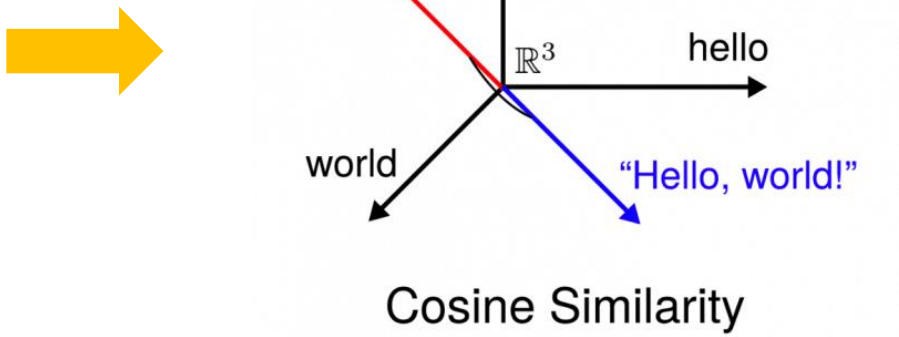
2) Cosine Similarity

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

	I	love	dogs	hate	and	knitting	is	my	hobby	passion
Doc 1	0.18	**0.48**	0.18							
Doc 2	0.18		0.18	**0.48**	0.18	0.18				
Doc 3					0.18	0.18	**0.48**	**0.95**	**0.48**	**0.48**



-	바나나	사과	저는	좋아요
문서1	2	3	0	1
문서2	1	2	3	1
문서3	2	1	2	2



[유사도 3: 자카드]

3) Jaccard Similarity

- 두 집합의 **교집합의 크기를 합집합의 크기로 나눈 값**으로 두 문서(집합)의 유사도를 측정
- 0에서 1사이의 값을 가지며 두 집합 사이에 교집합이 없으면 0, 두 집합이 동일하면 1의 값을 가짐

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

문서 A: 그대 내품에 안겨 눈을 감아요
문서 B: 그대 내품에 안겨 사랑의 꿈 나눠요

	그대	내품에	안겨	눈을	감아요	사랑의	꿈	나눠요
문서 A	O	O	O	O	O	X	X	X
문서 B	O	O	O	X	X	O	O	O

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{3}{8}$$

[유사도 3: 자카드]

3) Jaccard Similarity

- 두 집합의 **교집합의 크기를 합집합의 크기로 나눈 값**으로 두 문서(집합)의 유사도를 측정
- 0에서 1사이의 값을 가지며 두 집합 사이에 교집합이 없으면 0, 두 집합이 동일하면 1의 값을 가짐

실습 코드!

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

문서 A: 그대 내품에 안겨 눈을 감아요
문서 B: 그대 내품에 안겨 사랑의 꿈 나눠요

	그대	내품에	안겨	눈을	감아요	사랑의	꿈	나눠요
문서 A	O	O	O	O	O	X	X	X
문서 B	O	O	O	X	X	O	O	O

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{3}{8}$$

contents

Unit 06 | Assignment

<과제> “NLP 제대로 맛보기”

Step1. 관심 주제 관련 텍스트 데이터 크롤링

Step2. 전처리 (ex) 불용어 처리, 특수 문자 제거 등

Step3. 임베딩

Step4. 인사이트 도출 (ex) 유사도, 그래프 해석, 요약 알고리즘 등

Unit 06 | Assignment

[주의사항]

1. 파일로 제공되는 정형 데이터가 아닌, 지난 시간에 배운 '크롤링'으로 데이터를 수집해주세요.
2. 임베딩 모델을 2개 이상 적용해본 후, Step4의 결과에 따라 가장 좋은 모델을 선택해주세요.
   (ex) CBOW, Skip-gram, GloVe, NN, FastText 등
3. Step 4의 인사이트가 핵심입니다. 크롤링한 데이터에서 유의미한 인사이트를 도출해주세요.
   (ex) 그래프 하나 보여주고 한 문장으로 인사이트 끝? – BYE BYE
4. 이론적 궁금증 해결이나 참고를 위한 구글링은 OK, but 데이터 및 인사이트 그대로? NO!
5. 모델 선택 과정이나 인사이트 해석은 주석or워드 파일로 설명 부탁드립니다.

[주의사항]을 하나라도 준수하지 않은 경우, 고민 없이 돌려보내겠습니다.

[우수과제 선정 기준]

① 임베딩 모델을 선정한 판단 근거가 명확한가 (파라미터 포함)

② NLP에 대해 스스로 공부하고 고민한 흔적이 보이는가

③ 주제 및 인사이트 해석의 창의성

④ 전처리를 얼마나 꼼꼼히 진행하였는가

⑤ 정성이 담긴 과제 (김유민을 이해시켜라)

NLP에 대한 모든 연락은 환영입니다 ☺

들어주셔서 감사합니다.

Reference

[자료 참고]

ToBig's 11기 정규세션 NLP 기초 강의(정윤호님)

ToBig's 제 8회 컨퍼런스 프로젝트: 가사도우미(SeqGAN과 RNN-LM을 통한 노래가사 생성)

[정보 참고]

ToBig's 11기 정규세션 NLP 기초 강의(정윤호님)

https://datascienceschool.net/view-notebook/6927b0906f884a67b0da9310d3a581ee/

http://hero4earth.com/blog/learning/2018/01/17/NLP_Basics_01/