정규 교육 세미나

ToBig's 10기 정윤호

Nature Language Process Basic

Contents

Unit 01 | NLP Overview

Unit 02 | Process

Unit 03 | Tokenizing

Unit 04 | Embedding

Unit 05 | Similarity

Unit 06 | Assignment

Unit 01 | NLP Overview

NLP 기반 기술 이용 !

NLP가 대체 뭘까?

Natural Language Processing ?

자연어 처리 또는 **자연 언어 처리**는
인간의 언어 현상을 **컴퓨터**와 같은 기계를 이용해서 모사 할수 있도록 연구하고
이를 구현하는 **인공지능**의 주요 분야 중 하나 !

Natural Language Processing ?

NLP가 핫한 이유는?

자연어 처리 또는 자연 언어 처리는
인간의 언어 현상을 컴퓨터와 같은 기계를 이용해서 모사 할수 있도록 연구하고
이를 구현하는 인공지능의 주요 분야 중 하나 !

Unit 01 | NLP Overview

Unit 01 | NLP Overview

- 자연어 이해 및 자연어 처리는 인공지능 분야에 있어서 필수적

Unit 01 | NLP Overview

- 빅데이터에서 주목받고 있는 것은 비정형 데이터

- 비정형 데이터 중 상당 부분이 텍스트 데이터

- 또한 텍스트 데이터는 인간에 대한 정보를 많이 담고 있음

Natural Language Processing

지식 기반

통계 기반

딥러닝 기반

사례 기반

머신러닝 기반

Unit 01 | NLP Overview

Contents

Unit 01 | NLP Oveview

Unit 02 | Process

Unit 03 | Tokenizing

Unit 04 | Embedding

Unit 05 | Similarity

Unit 06 | Assignment

Unit 02 | Process

NLP는 대체적으로 어떠한 과정을 거칠까요?

Data Collection

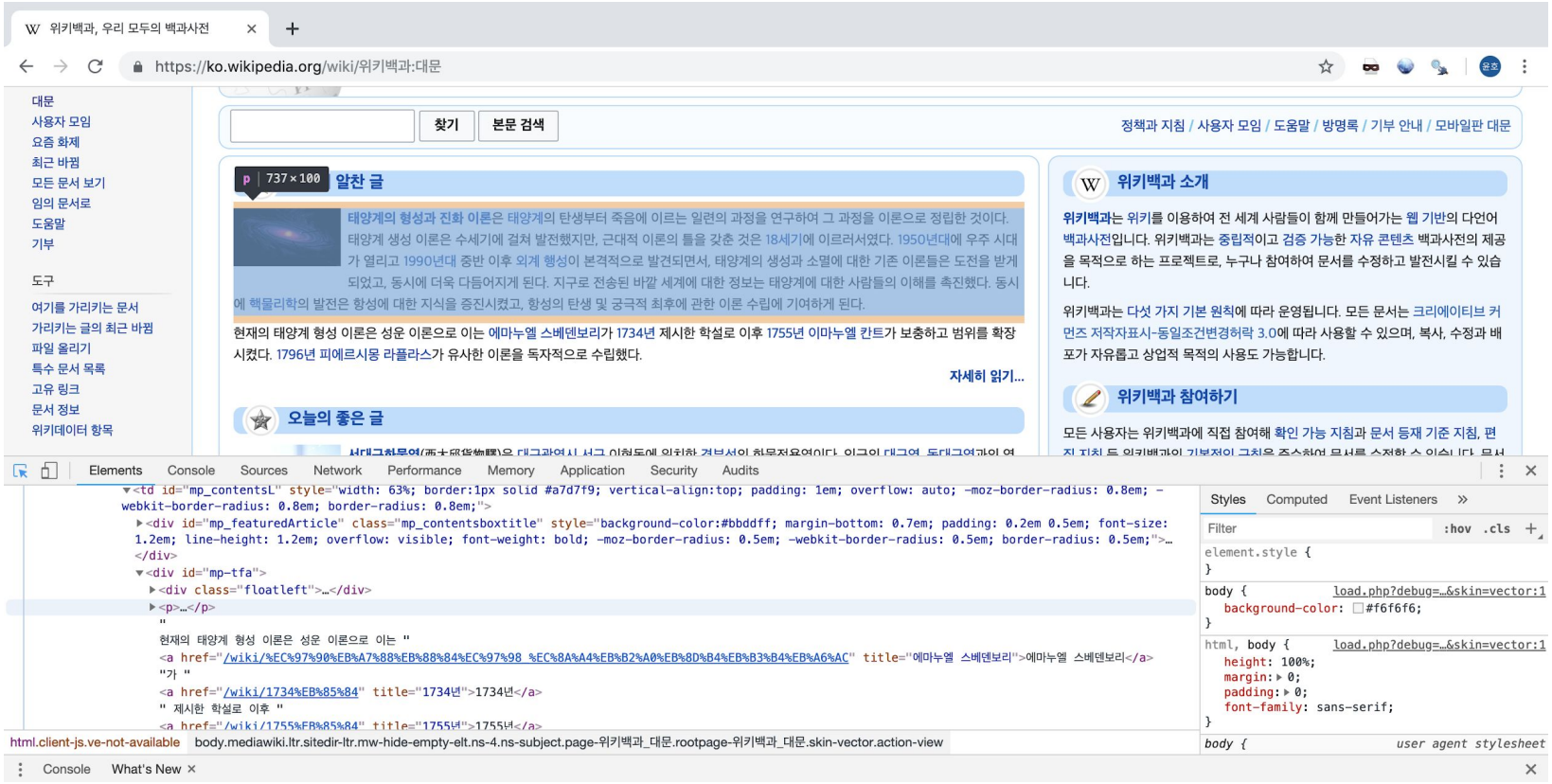
Tokenizing

Embedding

Similarity

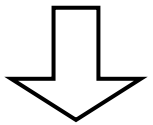
Network

Data Collection



Tokenizing

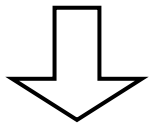
투빅스는 과제지옥입니다.



'투빅스', '는', '과제', '지옥', '입니', '다.'

Embedding

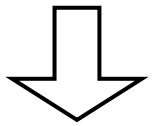
투빅스는 과제지옥입니다.



'투빅스' : [0.1234,0.1234] '는' : [0.5678,0.1234] '과제' : [0.9012,0.4321]
'지옥' : [0.3456,0.1764] '입니' : [0.7890,0.1567] '다.' : [0.1234,0399]

Similarity

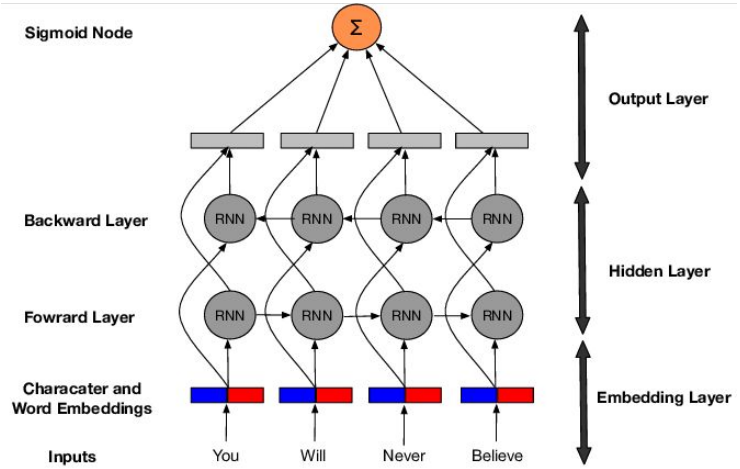
'투빅스' : [0.1234,0.1234] '는' : [0.5678,0.1234] '과제' : [0.9012,0.4321]
'지옥' : [0.3456,0.1764] '입니' : [0.7890,0.1567] '다.' : [0.1234,0399]

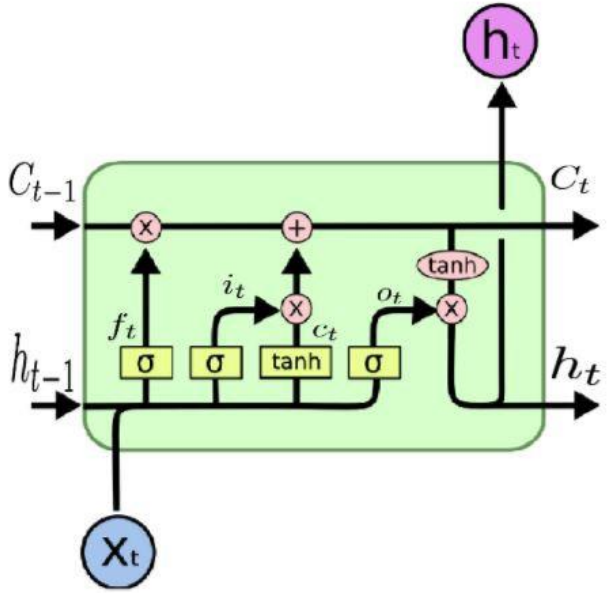


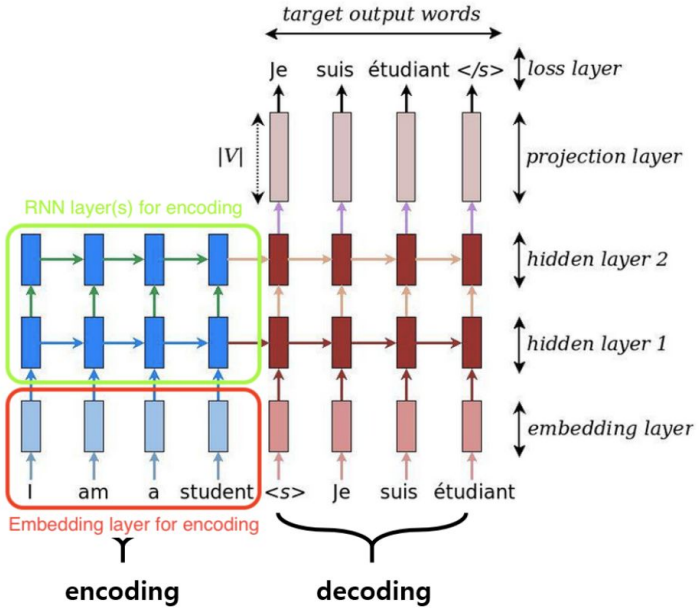
'과제' : [0.9012,0.4321], '지옥' : [0.3456,0.1764]

코사인 유사도에 따르면, 이 두 단어는 유사하다고 판단할 수 있다.

Network


Sigmoid Node
Σ
Output Layer
Backward Layer
RNN
Hidden Layer
Fowrard Layer
Characater and
Word Embeddings
Embedding Layer
Inputs
You
Will
Never
Believe


h_t
C_{t-1}
C_t
i_t
f_t
o_t
c_t
tanh
σ
h_{t-1}
h_t
X_t


target output words
Je
suis
étudiant
</s>
loss layer
|V|
projection layer
RNN layer(s) for encoding
hidden layer 2
hidden layer 1
embedding layer
I
am
a
student
<s>
Je
suis
étudiant
Embedding layer for encoding
encoding
decoding

Contents

Unit 01 | NLP Oveview

Unit 02 | Process

Unit 03 | Tokenizing

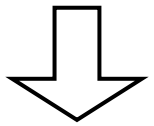
Unit 04 | Embedding

Unit 05 | Similarity

Unit 06 | Assignment

Tokenizing

투빅스는 과제지옥입니다.



'투빅스', '는', '과제', '지옥', '입니', '다.'

Tokenizing

투빅스는 과제지옥입니다

특정 기준에 의해서

Text -> Token

'투빅스', '는', '과제', '지옥', '입니', '다.'

Tokenizing

투빅스는 과제지옥입니다.

자소/음소, 형태소, 단어, 문장, 문서... etc

'투빅스', '는', '과제', '지옥', '입니', '다.'

Unit 03 | Tokenizing

English

NLTK

Korean

KONLPY

Unit 03 | Tokenizing

Kkma

Twitter(Okt)　　Komoran

Hannanum　　Mecab

Morphs

Nouns　　Pos Tagging

아버지가방에들어가신다.

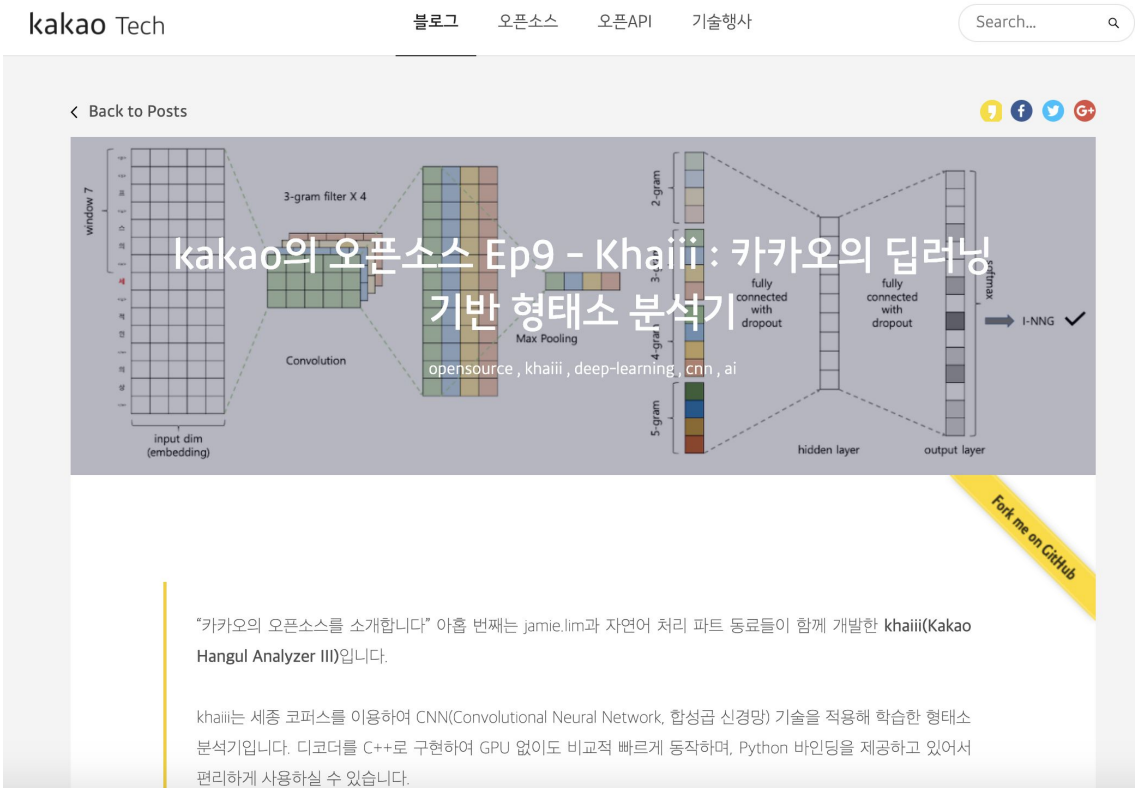
Hannanum	Kkma	Komoran	Mecab	Twitter
아버지가방에 들어가 / N	아버지 / NNG	아버지가방에 들어가신다 / NNP	아버지 / NNG	아버지 / Noun
이 / J	가방 / NNG		가 / JKS	가방 / Noun
시ㄴ다 / E	에 / JKM		방 / NNG	에 / Josa
	들어가 / VV		에 / JKB	들어가신 / Verb
	시 / EPH		들어가 / VV	다 / Eomi
	ㄴ다 / EFN		신다 / EP+EC	

아버지가방에들어가신다.

Hannanum	Kkma	Komoran	Mecab	Twitter
아버지가방에 들어가 / N	아버지 / NNG	아버지가방에 들어가신다 / NP	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 !!

딥러닝 기반 형태소 분석기



https://github.com/kakao/khaiii

Contents

Unit 01 | NLP Oveview

Unit 02 | Process

Unit 03 | Tokenizing

Unit 04 | Embedding

Unit 05 | Similarity

Unit 06 | Assignment

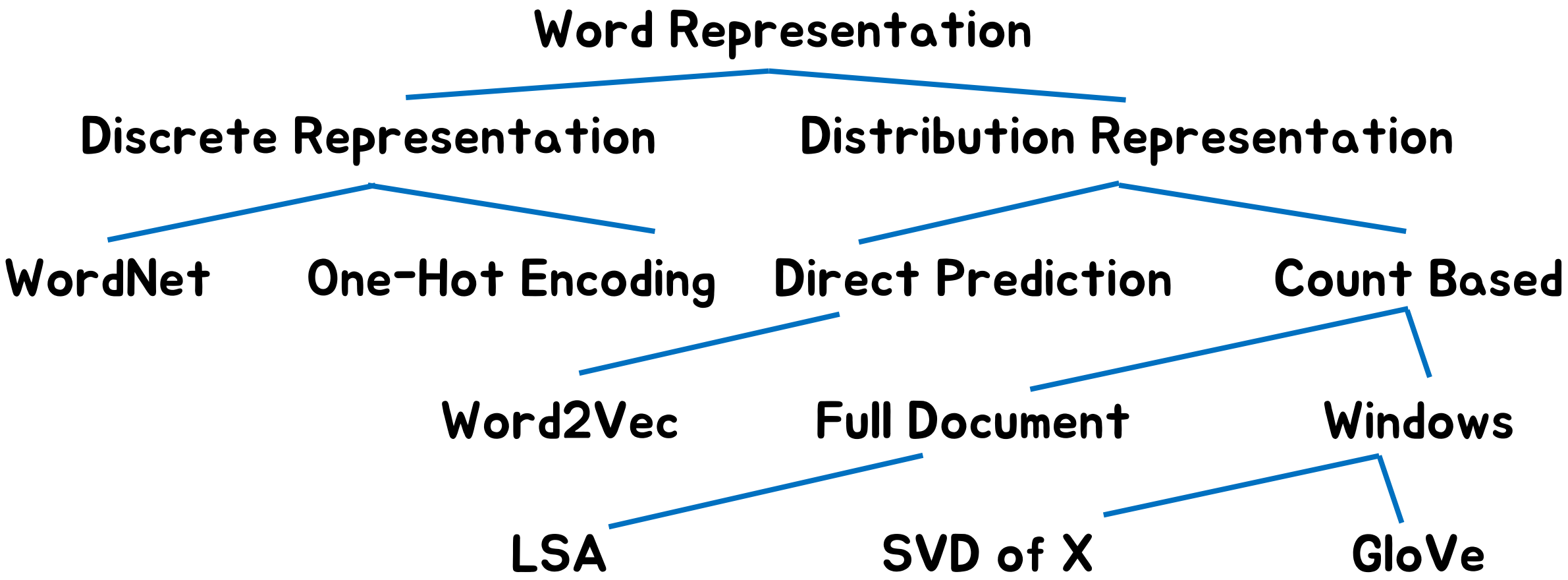
Tokenizing 왜 했지?

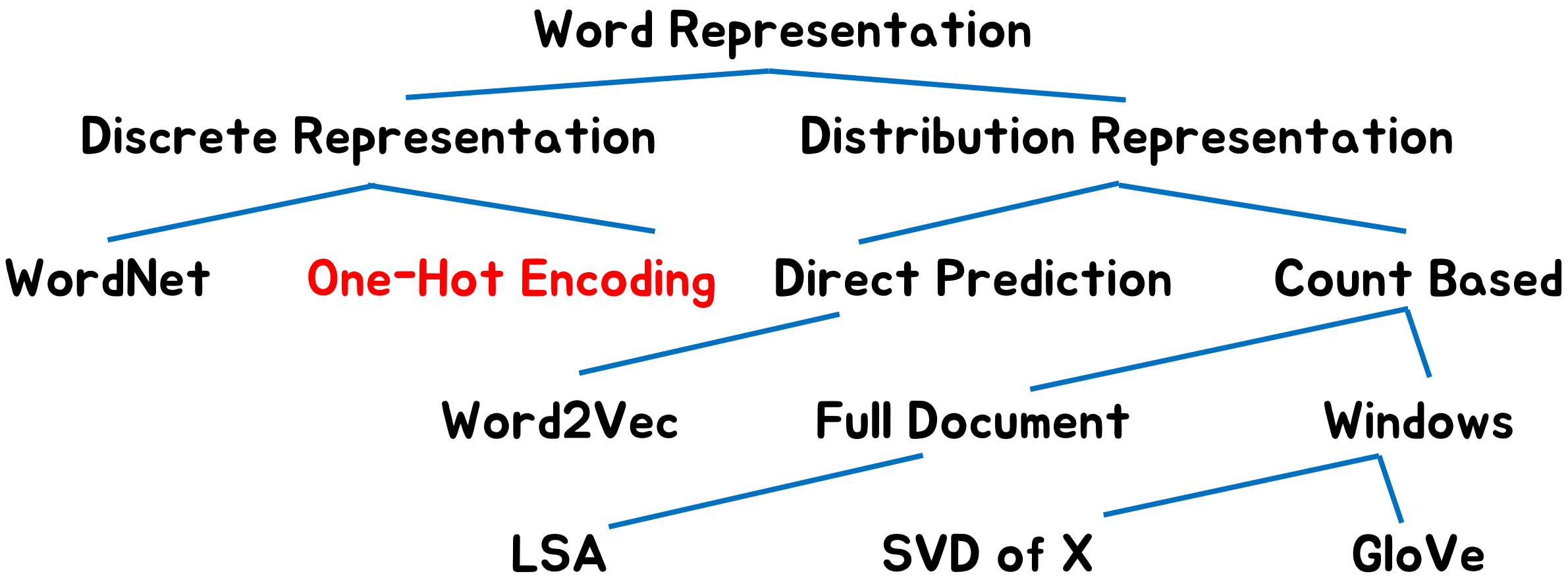
자연어 처리를 위한 의미단위를 만들기 위해 !

그런데 컴퓨터가 인간의 언어를 어떻게 이해할 수 있을까?

컴퓨터가 처리할 수 있는 것은 수치뿐

컴퓨터가 언어의 특성을 이해할 수 있도록
각 Token마다 수치를 부여


Word Representation
Discrete Representation
Distribution Representation
WordNet
One-Hot Encoding
Direct Prediction
Count Based
Word2Vec
Full Document
Windows
LSA
SVD of X
GloVe


Word Representation
Discrete Representation
Distribution Representation
WordNet
One-Hot Encoding
Direct Prediction
Count Based
Word2Vec
Full Document
Windows
LSA
SVD of X
GloVe

One-Hot Encoding의 문제점

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

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

해질 무렵 바람도 몹시 불던 날 집에 돌아오는 길 버스 창가에 앉아

['해','질','무렵','바람','도','몹시','불','던','날','집','에','돌아오는','길','버스','창가','에','앉아']

One-Hot Encoding

'해' : [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'질' : [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'무렵' : [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'바람' : [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'도' : [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'몹시' : [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'불' : [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'던' : [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'날' : [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
'집' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
'에' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
'돌아오는' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
'길' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
'버스' : [0, 0, 10 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
'창가' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
'에' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
'앉아 : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

한 문장을 표현하기 위해
17개의 feature 필요

해질 무렵 바람도 몹시 불던 날 집에 돌아오는 길 버스 창가에 앉아 불어오는 바람 어쩌지도 못한 채 난 그저 멍할 뿐이었지 난 왜 이리 바보인지 어리석은지 모진 세상이란 걸 아직 모르는 지 터지는 울음 입술 물어 삼키며 내려야지 하고 일어설 때 저 멀리 가까워 오는 정류장 앞에 희미하게 일렁이는 언제부터 기다렸는지 알 수도 없는 발만 동동 구르고 있는 그댈 봤을 때 나는 아무 말도 못하고 그댈 안고서 그냥 눈물만 흘러 자꾸 눈물이 흘러 이대로 영원히 있을 수만 있다면 오 그대여 그대여서 고마워요 낙엽이 뒹굴고 있는 정류장 앞에 희미하게 일렁이는 까치발 들고 내 얼굴 찾아 헤매는 내가 사준 옷을 또 입고 온 그댈 봤을 때 나는 아무 말도 못하고 그댈 안고서 그냥 눈물만 흘러 자꾸 눈물이 흘러 이대로 영원히 있을 수만 있다면 오 그대여 그대여서 고마워요

One-Hot Encoding

'해' : [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'질' : [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'무렵' : [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'바람' : [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'도' : [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'몹시' : [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'불' : [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'던' : [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
'날' : [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
'집' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
'에' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
'돌아오는' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
'길' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
'버스' : [0, 0, 10 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
'창가' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
'에' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
'앉아 : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

Discrete Representation
Sparse Vector
Inner Product = 0
(Independent)

One-Hot Encoding

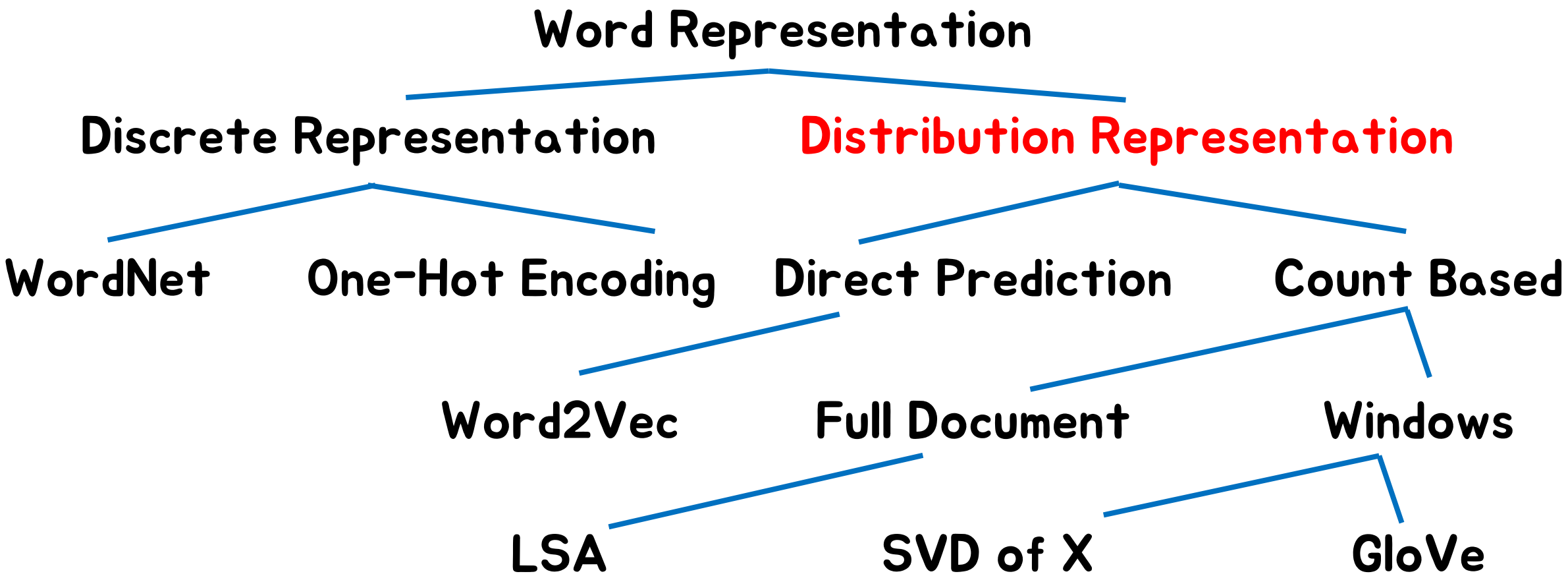
'해' : [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'질' : [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'무렵' : [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'바람' : [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'도' : [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'몹시' : [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'불' : [0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
'던' : [0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
'날' : [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
'집' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
'에' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
'돌아오는' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
'길' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
'버스' : [0, 0, 10 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
'창가' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
'에' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
'앉아 : [0, 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

효과적인 방법이 없을까?

Word Embedding이란 ?

단어를 좀 더 조밀한 차원에 벡터로 표현하는 것


Word Representation
Discrete Representation
Distribution Representation
WordNet
One-Hot Encoding
Direct Prediction
Count Based
Word2Vec
Full Document
Windows
LSA
SVD of X
GloVe

GloVe

BERT

Word2Vec

FastText

GloVe

BERT

Word2Vec

FastText

Word2Vec : 말 그대로 Word to Vector

How ?

Word2Vec : 말 그대로 Word to Vector

1. Word & Neighbor (CBOW, Skip Gram)

2. Neural Network

CBOW

Skip Gram

$$p(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}$$

EX) Skip Gram

Window Size

ex) window size = 2

원수는 외나무다리에서 만난다.

Window Size

'원수', '는', '외나무다리'
'에서', '만난다'

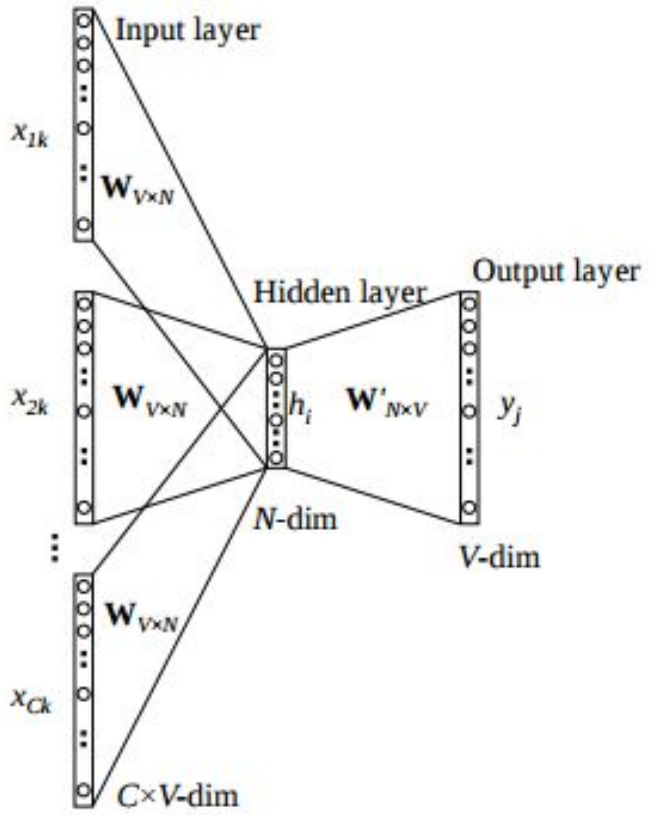
Center Word	Neighbor Words
'원수'	'는', '외나무다리'
'는'	'원수', '는', '외나무다리'
'외나무다리'	'원수', '는', '에서', '만난다'
'에서'	'는', '외나무다리', '만난다'
'만난다'	'외나무다리', '에서'

CBOW

원수는 _____에서 만난다.

CBOW

Input : Neighbor Words



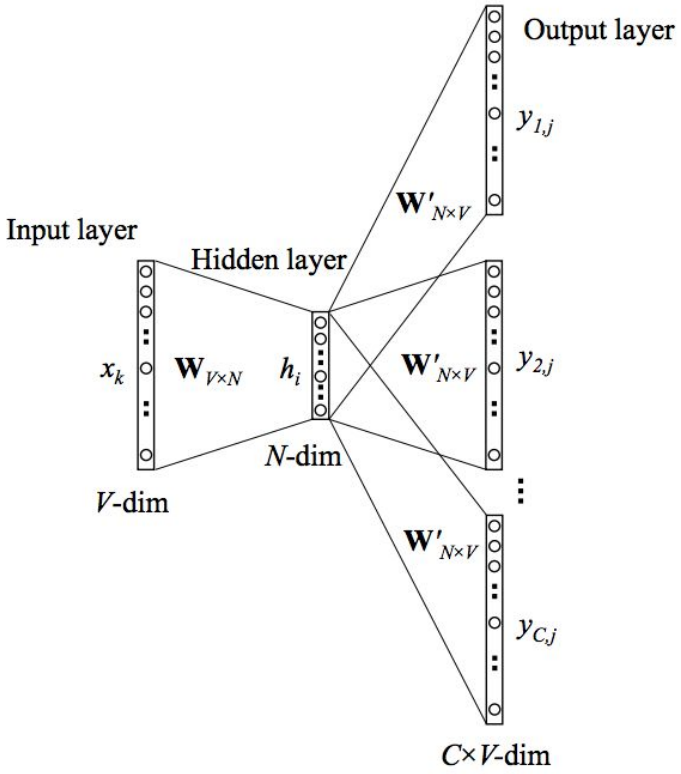
Target : Center Word

Skip Gram

___ 외나무다리__ ___.

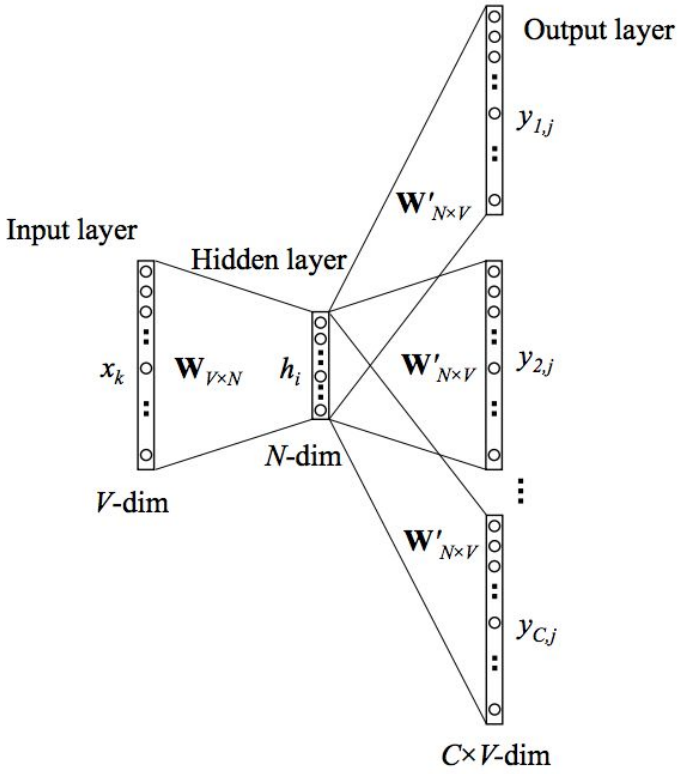
Skip Gram

Input : Center Word

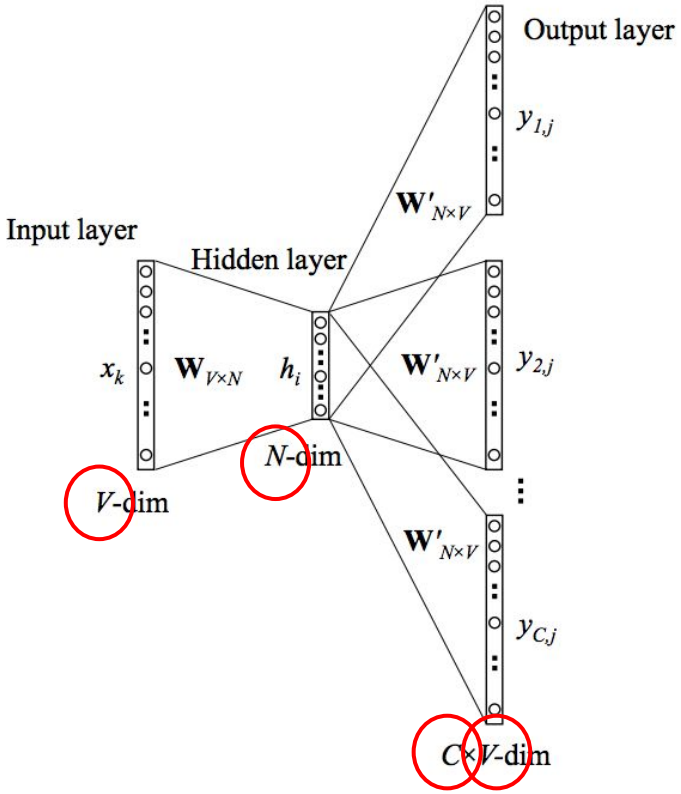


Target : Neighbor Words

Neural Network

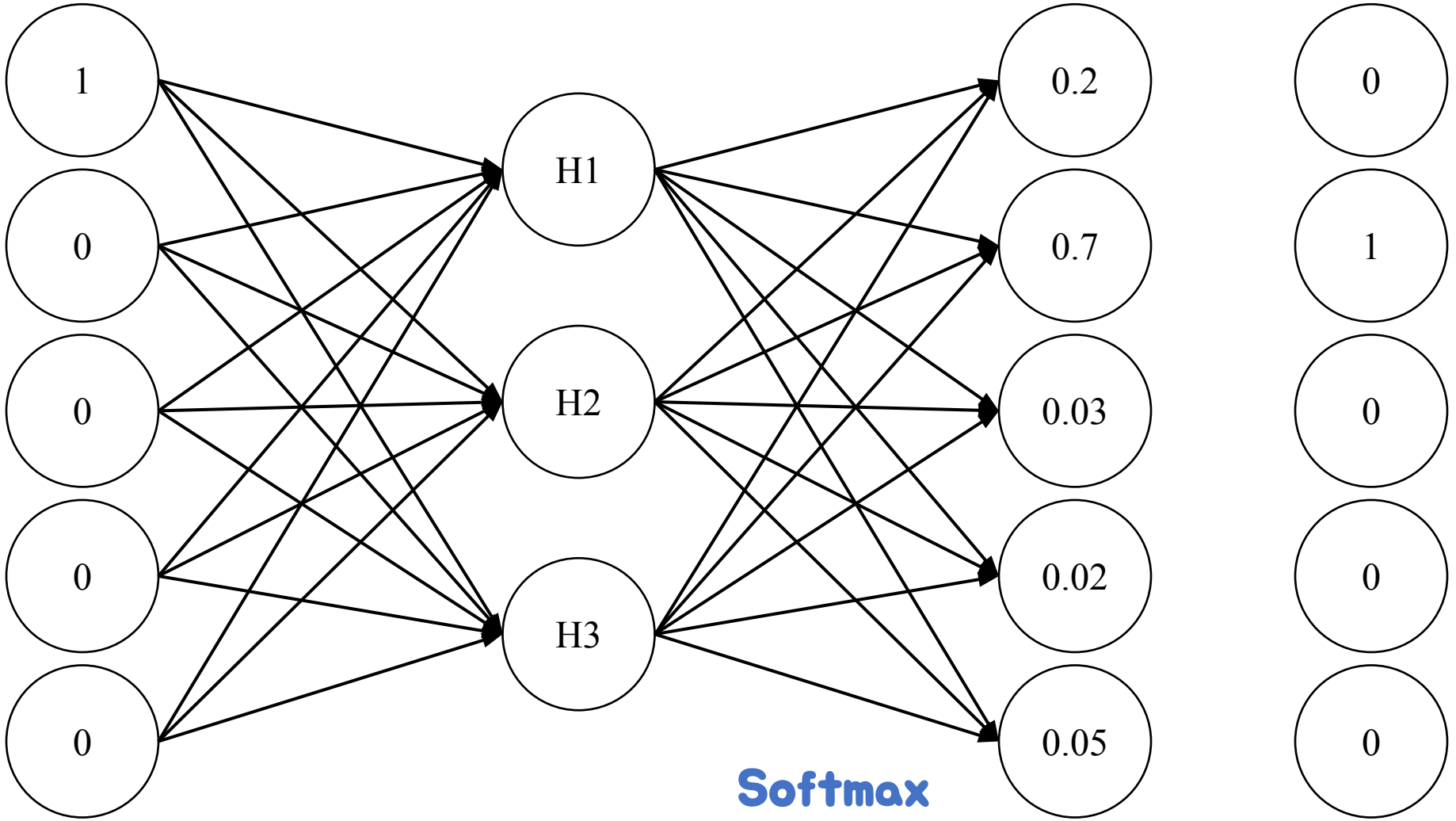


Neural Network



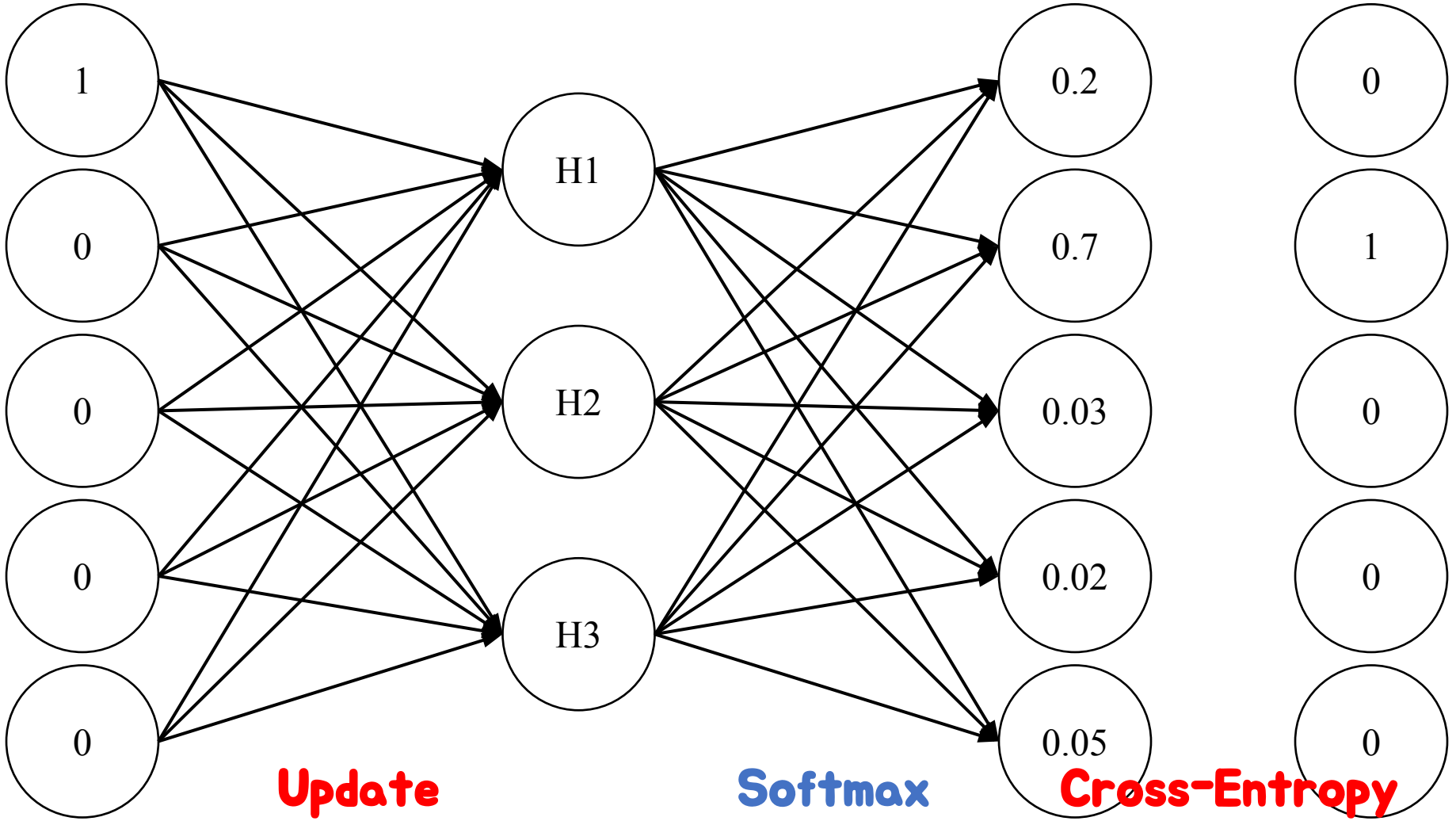
Unit 04 | Embedding

One-Hot Encoding

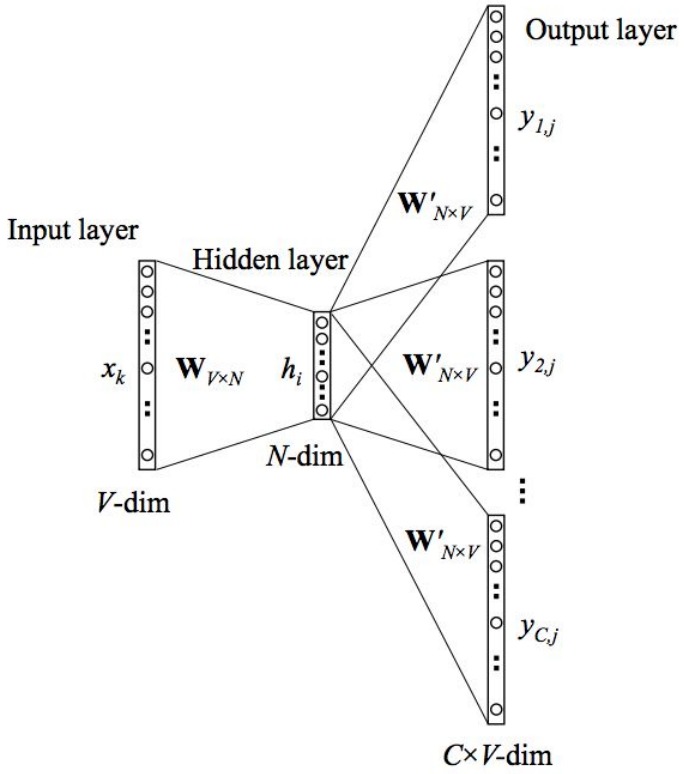


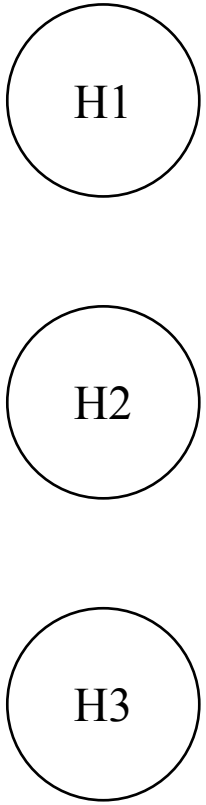
Unit 04 | Embedding

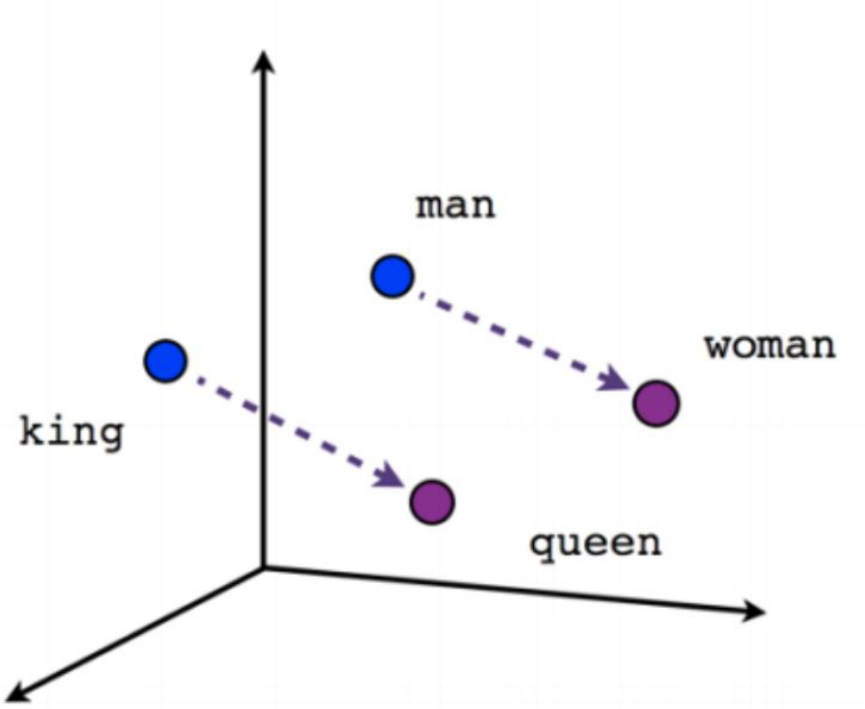
One-Hot Encoding



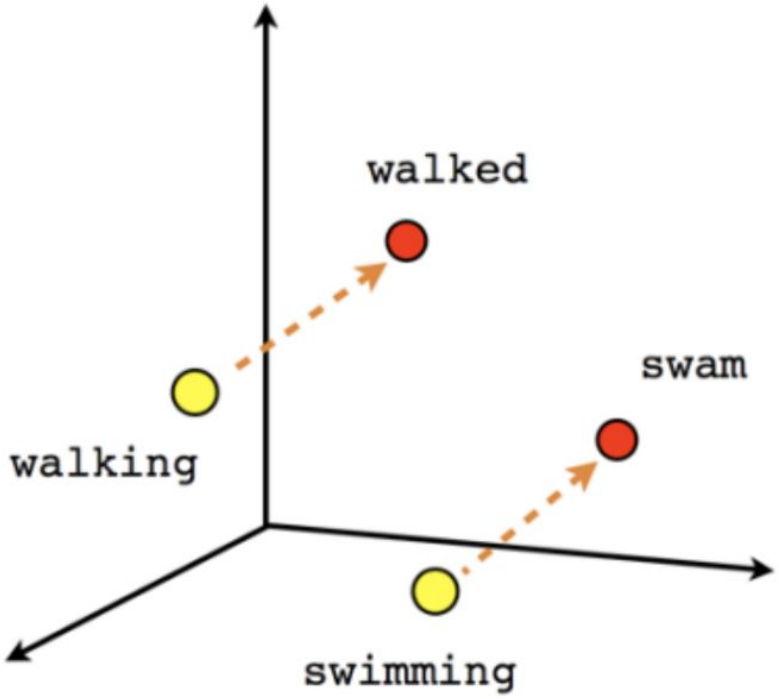
Word Vector


Output layer
$y_{1,j}$
$\mathbf{W}'_{N\times V}$
Input layer
Hidden layer
x_k
$\mathbf{W}_{V\times N}$
h_i
$\mathbf{W}'_{N\times V}$
$y_{2,j}$
N-dim
V-dim
$\mathbf{W}'_{N\times V}$
$y_{C,j}$
$C\times V$-dim


H1
H2
H3



Male-Female



Verb tense

Word2Vec의 문제점 ?

1. 한번에 하나의 동시 출현만 계산해서 업데이트 -> 전체적인 통계(빈도 수 등) 정보 이용x -> 비효율적인 면이 있지 않은가?

2. train corpus에 존재하지 않았던 단어의 벡터를 만들어낼 수 없다.

GloVe

전체적인 **통계 정보**를 이용해보자 !
학습 말뭉치에서 동시에 **같이 등장한 단어의 빈도**를 각각 세어서
전체 말뭉치의 단어 개수로 나눠준 **동시등장확률**을 고려하자 !

GloVe

- Example corpus:
  - I like deep learning.
  - I like NLP.
  - I enjoy flying.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

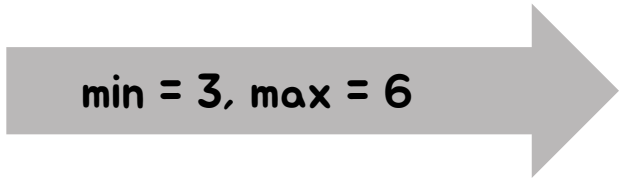
Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

FastText

단어가 아닌 단어내부의 n-gram을 최소단위로 !

FastText

'apple'



'<ap', 'app', 'appl', 'apple', 'apple>', 'ppl', 'pple', 'pple>', 'ple', 'ple>', 'le>'

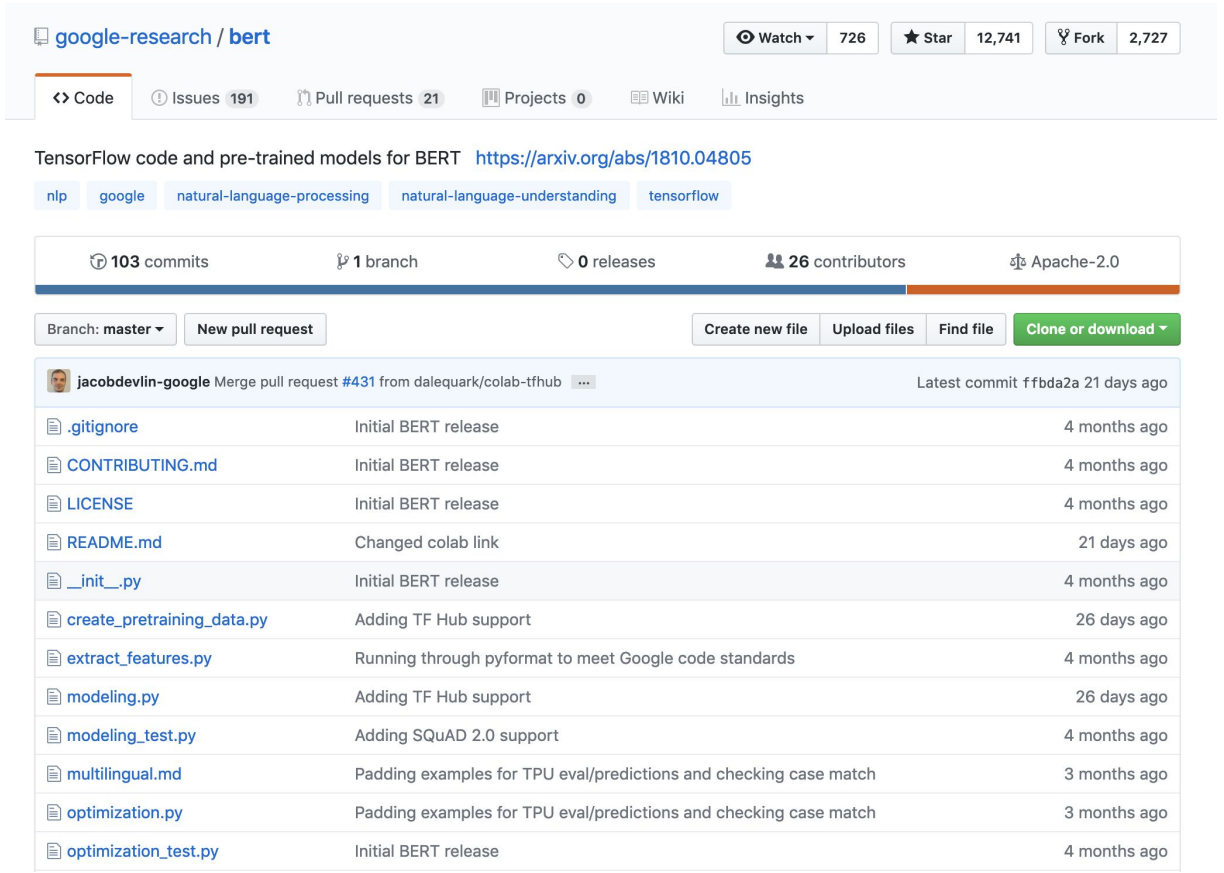
FastText

1. train corpus에 존재하지 않았던 단어의 벡터를 만들어낼 수 있다.(word2vec,glove에서는 불가능)

2. 희소한 단어에 대해 더 좋은 word embedding이 가능하다.

Unit 04 | Embedding

BERT



https://github.com/google-research/bert

Contents

Unit 01 | NLP Oveview

Unit 02 | Process

Unit 03 | Tokenizing

Unit 04 | Embedding

Unit 05 | Similarity

Unit 06 | Assignment

Similarity Analysis

Distance

Cosine Similarity

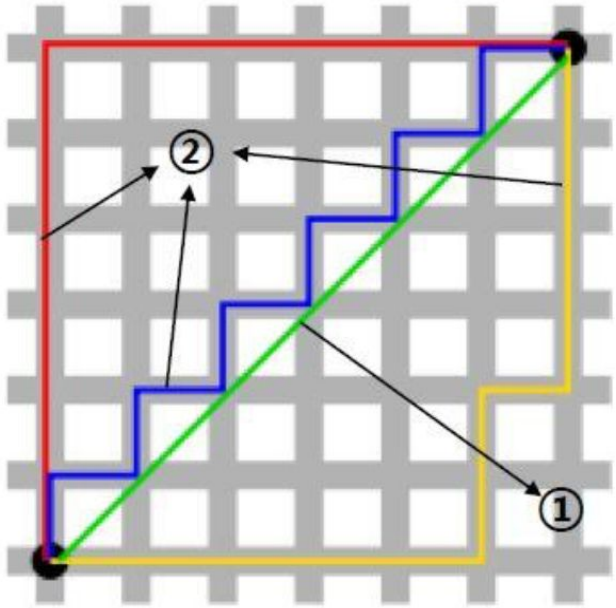
Distance

$$d(x,y) = \left(\sum_{i=1}^{p}(x_i - y_i)^2\right)^{1/2}$$

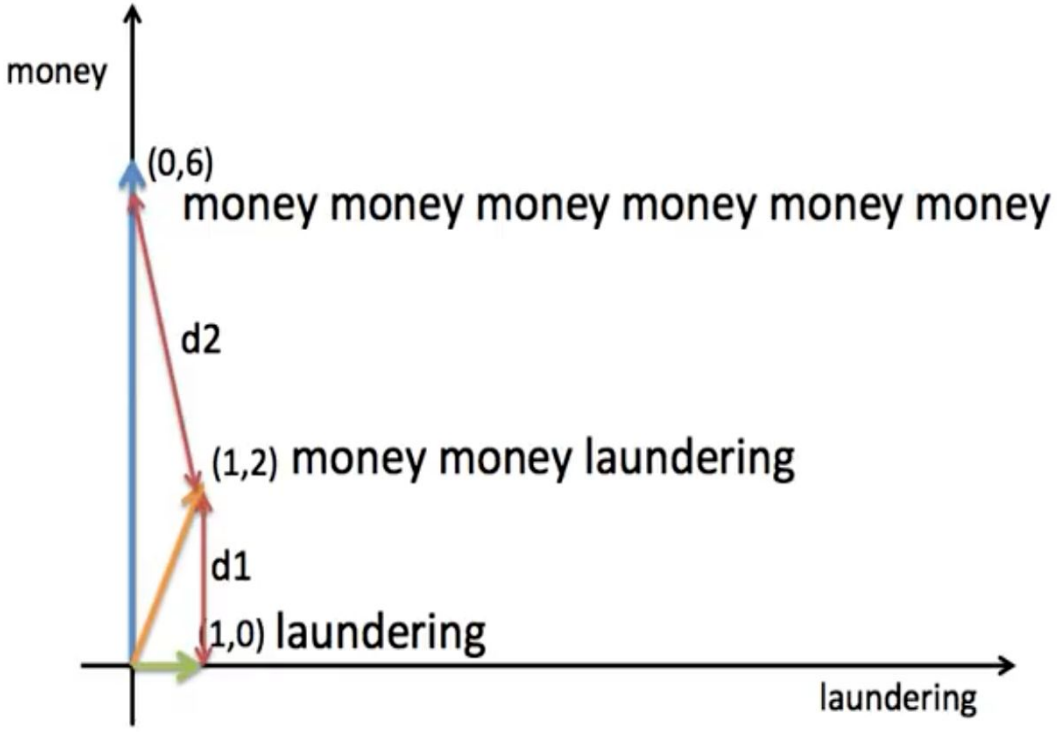
$$d(x,y) = \sum_{i=1}^{p}|x_i - y_i|$$

$$d(x,y) = \left(\sum_{i=1}^{p}(x_i - y_i)^2/s_i^2\right)^{1/2}$$

$$d(x,y) = \left(\sum_{i=1}^{p}(x_i - y_i)^m\right)^{1/m}$$

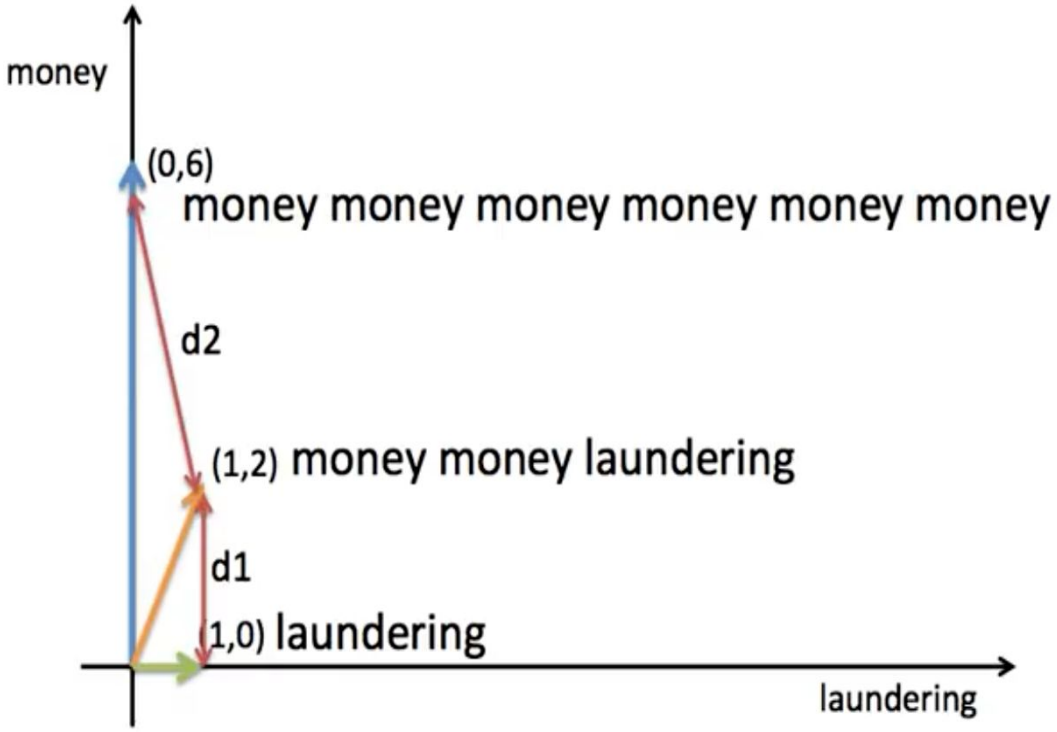


Cosine Similarity

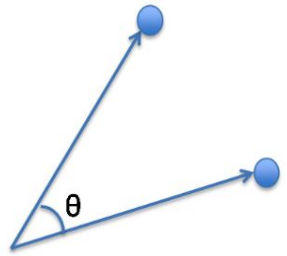


d2>d1
하지만 money money laundering은
money money money money money와
더 유사하지 않을까?

Cosine Similarity

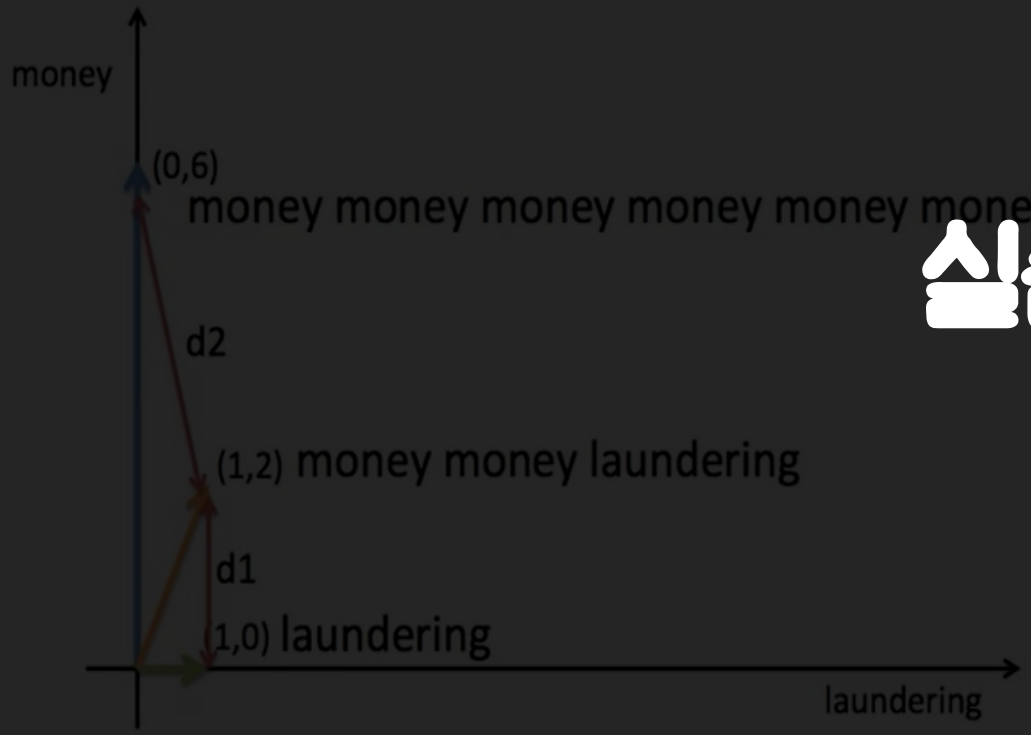


$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



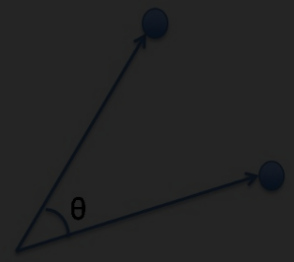
벡터의 크기를 무시하고 방향성만 고려하여 계산
두 벡터가 이루고 있는 각도를 활용하여
작을수록 유사하다고 판단

Cosine Similarity



실습 코드 !!

$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



벡터의 크기를 무시하고 방향성만 고려하여 계산
두 벡터가 이루고 있는 각도를 활용하여,
작을수록 유사하다고 판단

Contents

Unit 01 | NLP Overview

Unit 02 | Process

Unit 03 | Tokenizing

Unit 04 | Embedding

Unit 05 | Similarity

Unit 06 | Assignment

Unit 06 | Assignment

< 과제1 >

Word2Vec을 구현한 TensorFlow 코드에서, # 부분에 있는 코드의 의미에 대해 주석달아보기 ! 그리고 하이퍼 파라미터 조정해서 코드 다시 실행해보기 !

< 과제 2>

프로젝트 주제 혹은 관심분야를 정해서 해당 주제에 대한 텍스트를 크롤링하고, 전처리와 임베딩을 하고 난 후 그래프 그리기 !(우수과제는 크롤링이나 전처리 과정에서의 센스, 혹은 임베딩 결과를 어떻게 이용하고 분석하느냐에 달려있음. 이 부분은 선택적)

Unit 06 | Assignment

〈Reference〉

- http://www.datamarket.kr/xe/(투빅스 9기 안상준 NLP Basic)
- https://www.youtube.com/watch?v=uZ2GtEe-50E
- https://ratsgo.github.io/blog/categories/
- http://web.stanford.edu/class/cs224n/
- https://www.youtube.com/watch?v=sY4YyacSsLc
- https://wikidocs.net/22660
- https://datascienceschool.net/view-notebook/6927b0906f884a67b0da9310d3a581ee/

Unit 06 | Assignment

〈IMG〉

- http://techm.kr/bbs/board.php?bo_table=article&wr_id=4051
- http://www.ndsl.kr/ndsl/issueNdsl/detail.do?techSq=50
- https://towardsdatascience.com/seq2seq-model-in-tensorflow-ec0c557e560f
- https://hackernoon.com/understanding-architecture-of-lstm-cell-from-scratch-with-code-8da40f0b71f4
- https://www.researchgate.net/figure/BiDirectional-RNN-architecture-for-detecting-clickbaits_fig1_311430194
- https://imgur.com/TupGxMl
- https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/
- https://www.tensorflow.org/images/linear-relationships.png
- http://web.stanford.edu/class/cs224n/

들어주셔서 감사합니다.