Word Embedding

/. 자연어를 좌표평면 위에 표기 하기 위한 방법.

V one-hot encoding

단어를 하나의 숫자 형태로 표현.

표현된 단어는 해당 위치의 Index에만 1을 두고 나머지는 0으로 표현

ex) I am a student => [0 / 2 3]

=> [[1 0 0 0],
[0 1 0 0],
⋮
[0 0 0 1]]

· 단어의 표현(representation)이 너무 sparse 해짐.

=> 불필요한 저장 공간 확보 → Data는 크지만 정작 제공하는 정보 ↓

차원의 저주 (Curse of Dimension)

=> 단어의 의미와 관련된 정보를 제공 하는것이 불가능.

-> 단어간 유사도 확인 역시 불가능.

✓ Word2Vec

· 단어의 의미를 벡터공간에 임베딩.

· 단어의 주변 단어를 활용해 의미를 파악.

ex) 나는 (소고기를) 좋아해. 소고기와 돼지고기는 비슷한 의미라는 것을

나는 (돼지고기를) 좋아해. 유추 가능.

· Word embedding의 성능 검증.

① WordSim 353 (사람이 직접 만든 Dataset)

단어 사이의 코사인 유사도를 사용하여 단어간 유사도 점수를

비교한다. (0.7 이상인 경우 잘 학습된 임베딩 모델)

② Semantic & Syntactic analogy
의미적 · 구문적 분석을 통한 확인.

· 학습되지 않은 단어에 대해 OOV 문제가 생긴다.

✓ FastText

· FastText의 학습 방법은 기존의 Word2Vec과 같다.

· OOV 문제 해결을 위해 단어를 n-gram 단위로 분할 하여 학습

ex) <orange> "임의"

→ (<o, or, ra, an ... , <ora, ... , orange >

↑ 마지막은 전체의 단어

orange의 단어 벡터는 n-gram으로 분할된 토큰들의 합이 된다.
따라서 oranges라는 새로운 단어가 들어오더라도 비슷한 의미로 해석이 가능!!

· 한국어의 경우 자소단위를 사용하는 경우가 많다.