빠르게 해보는 Word Embedding
- word2vec, doc2vec with gensim

신윤식

Word Embedding의 개념

Word Embedding의 역사 및 주요 방법

Word Embedding으로 풀 수 있는 문제의 유형

주요 방법들의 동작원리와 이론

적용 가능한 API 등 소개

실제 예제적용 및 결과

.

.

15분

모든 것을 할 수는 없다.

한번쯤 거쳐가는 방법

보면서 쉽게 이해할 수 있는 주제

쉽고 빠른 구현이 가능한 API 존재

방법 - word2vec, doc2vec

예시 - 문서분류(Document Classification)

Syntactic Similarity

밥을 먹었다.

밥이 먹고 싶다.

밥에 콩을 넣었다.

"구문"의 차이

Semantic Similarity

성질이 많이 죽었다.

솜이불 숨이 죽었다.

기르던 화초가 죽었다.

"의미"의 차이

word2vec

오늘은 집에서 식사를 했다.

Skip-gram

__ 집에서 __ __

CBOW

오늘 ___ 식사를 했다.

word2vec

중심단어 주변단어

오늘은 집에서 식사를 했다.

window size =1,

오늘은 → 집에서

집에서 → 오늘은

집에서 → 식사를

식사를 → 집에서

식사를 → 했다.

했다. → 식사를

word2vec

주변단어 중심단어 주변단어

오늘은 집에서 식사를 했다.

window size =1,

오늘은 → 집에서

집에서 → 오늘은

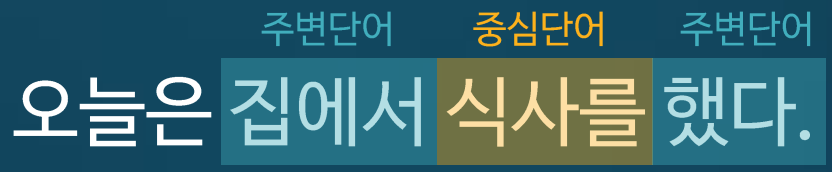
집에서 → 식사를

식사를 → 집에서

식사를 → 했다.

했다. → 식사를

word2vec



window size =1,

오늘은 → 집에서

집에서 → 오늘은

집에서 → 식사를

식사를 → 집에서

식사를 → 했다.

했다. → 식사를

word2vec

주변단어 중심단어

오늘은 집에서 식사를 했다.

window size =1,

오늘은 → 집에서

집에서 → 오늘은

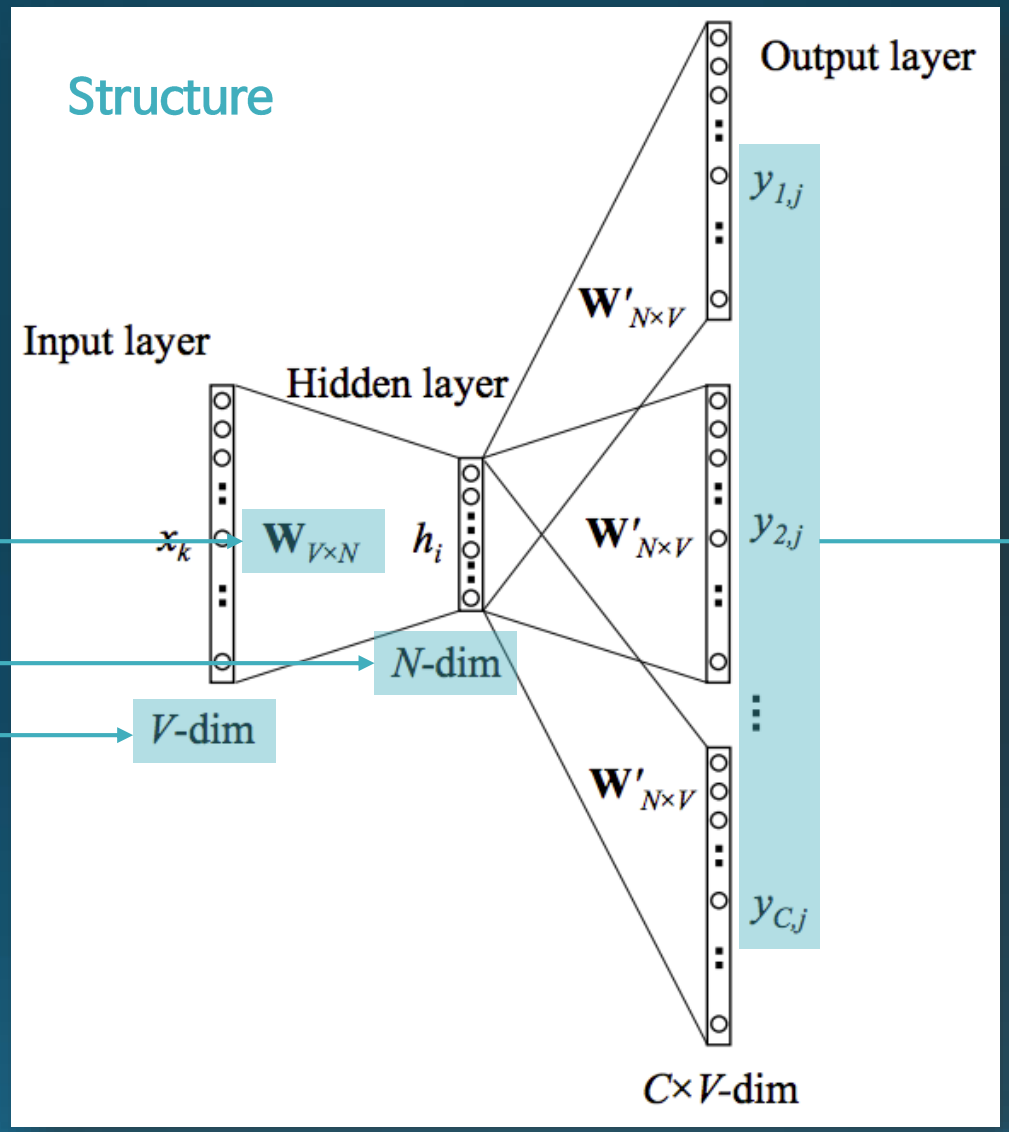
집에서 → 식사를

식사를 → 집에서

식사를 → 했다.

했다. → 식사를

Word2Vec: Skip-Gram



Embedding Matrix

Feature 수

전체 단어 수

정답 단어 벡터

(window size에 따라 가변적)

Rong, Xin. "word2vec parameter learning explained." *arXiv preprint arXiv:1411.2738* (2014).

Word2Vec: Skip-Gram

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2}$$

Cosine similarity From Wikipedia

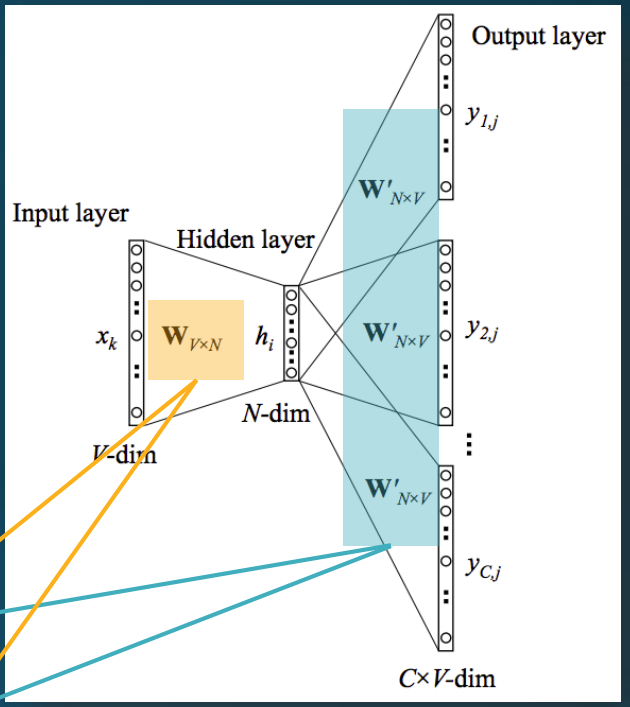
maximize ν(중심단어) x ν(주변단어)

minimize $\Sigma\nu$(중심단어) x ν(모든단어)

Objective Function

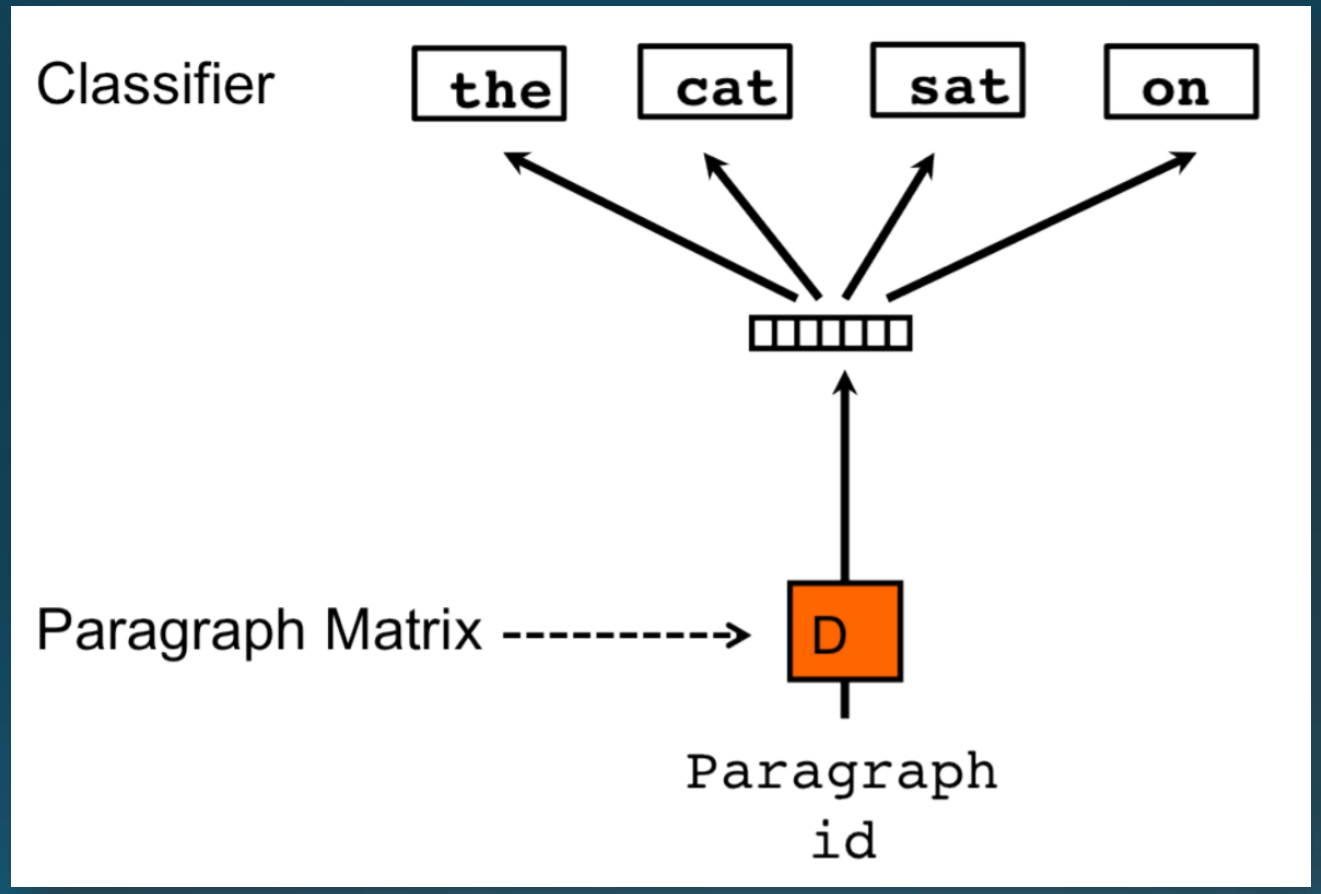
$$p(w_O|w_I) = \frac{\exp\left(v'_{w_O}{}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left(v'_{w}{}^{\top} v_{w_I}\right)}$$

Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.



Rong, Xin. "word2vec parameter learning explained." *arXiv preprint arXiv:1411.2738* (2014).

Doc2Vec: DBOW



Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. 2014.

Gensim

Learning rate decay

Initial Learning rate

the end point Learning rate

threading
(intel CPU는 cores/2)

embedding dimension
(100~300, 500)

문서 길이에 따라 가변적

문서 및 말뭉치 규모에 따라 가변적
(3~10)

```
class gensim.models.word2vec.Word2Vec(sentences=None, size=100, alpha=0.025, window=5, min_count=5, max_vocab_size=None,
sample=0.001, seed=1, workers=3, min_alpha=0.0001, sg=0, hs=0, negative=5, cbow_mean=1, hashfxn=<built-in function hash>, iter=5,
null_word=0, trim_rule=None, sorted_vocab=1, batch_words=10000, compute_loss=False)
```

Gensim Word2vec API, https://radimrehurek.com/gensim/models/word2vec.html

고빈도 단어를 학습에서
확률적으로 제외
(0~0.00001)

Skip-gram 사용여부
(1 = 사용)

Hierarchical softmax 사용여부
(0 = 미사용)

Negative sampling 사용여부
(5~25)

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}}$$

대상 단어빈도 / 전체 단어 수

window에 없는 단어

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^{n} \left(f(w_j)^{3/4}\right)}$$

모든 단어

Chris McCormick, "*Word2Vec Tutorial*", http://mccormickml.com/. (2017)

Q&A

E.O.D