R 교육 세미나

ToBig's 7기 최희정

word2vec

효율적인 단어 embedding

Contents

컴퓨터가 단어를 인지하기 위해
단어의 수치화가 필요하다!

간단한 수치화 방법

One-hot encoding: 해당 단어의 사전상 위치에는 1, 나머지 위치에는 0을 넣어 0과 1만을 원소로 가지는 vector로 단어를 벡터화하는 방법

ex) 사전: 총 5개의 단어 – [I, you, like, hate, love]
-> I = $[\,1\;0\;0\;0\;0\,]^T$, hate = $[\,0\;0\;0\;1\;0\,]^T$

-> But, 이러한 벡터화를 통해서는 단어가 본질적으로 다른 단어와 어떤 **차이점**을 가지는지 이해 불가능!!

그럼 단어의 의미 자체를 벡터화 시키자!!

어떻게?
언어학의 'Distributional Hypothesis' 가정에 입각하여!!!

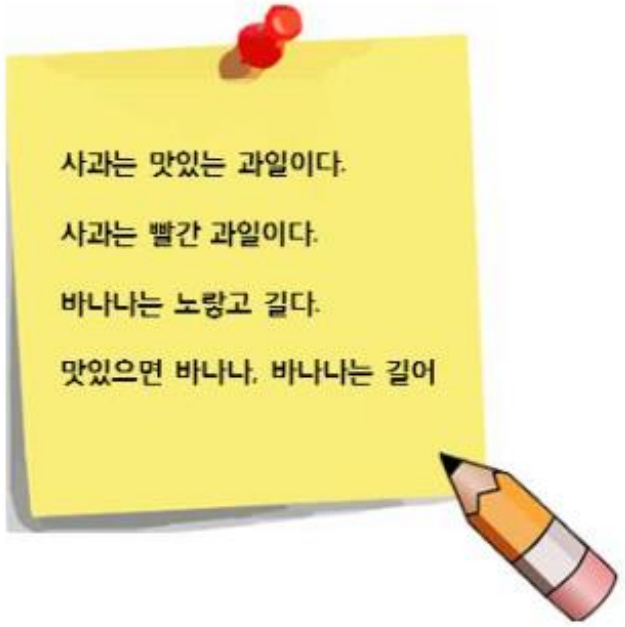
단어의 주변을 보면 그 단어를 안다.

You shall know a word by the company it keeps.

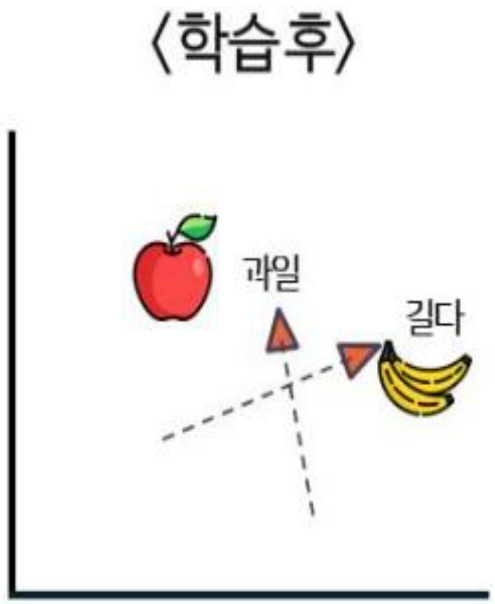
-- 언어학자 J.R. Firth (1957)

NN word embedding model

: target 단어와 주변단어를 이용한 학습을 통해 단어를 벡터화 한 model
-> similarity가 높은 단어들은 가까이 embedding



Unit 01 | Intro



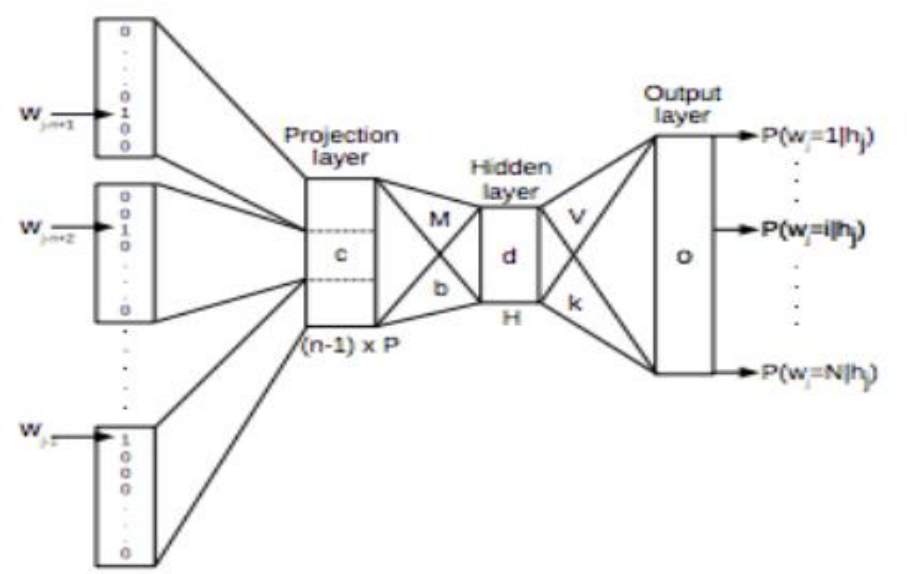
주변 단어를 이용해 학습하는 model

-> 주변에 많이 등장하는 단어가 similarity 높으므로 **함께 많이 등장**하는 단어일수록 **가까이 embedding**!!

-> **'사과'**는 **'과일'**이랑 **'바나나'**는 **'길다'**랑 더 가까이 embedding

기존 model

NNLM: target word 이전의 N개의 단어에 대한 one-hot vector로
target word의 조건부 확률분포를 출력하는 model



NNLM Structure

단점

1. 몇 개의 단어를 고려할지에 대한 파라미터 N 고정
2. 이전 단어들만 고려 가능하고, 앞 단어들은 고려X
3. 계산량이 많아서 학습속도 느림

NNLM의 단점을 개선한
새로운 Log-linear model을 만들자!!

word2vec

기존 model보다 **적은 계산량**을 통해
효율적으로 **단어를 벡터화한** model

그럼 word2vec은

어떤 학습을 통해 단어를 벡터화할까?

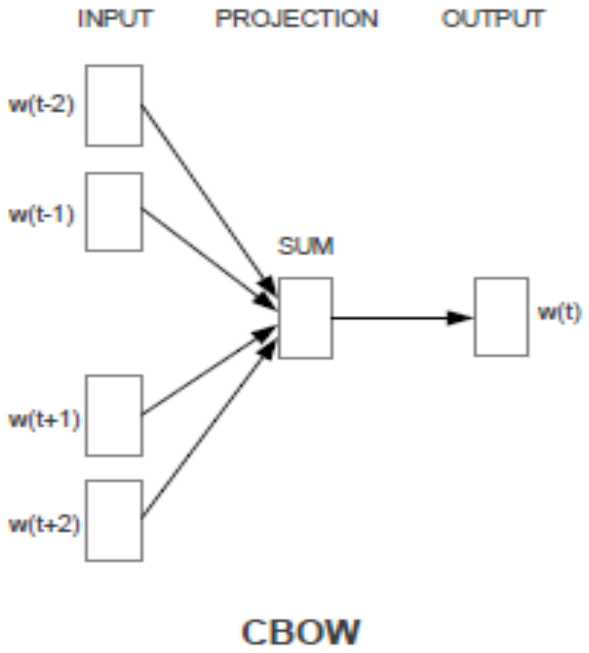
Unit 02 | word2vec – CBOW & Skip-gram

"집 앞 편의점에서 아이스크림을 사 먹었는데,
___ 시려서 너무 먹기가 힘들었다."

-> 주변단어를 이용해 target word를 예측하자!

Word2vec

CBOW: target word의 앞뒤 N/2개의 단어 즉, 총 N개의 단어를 Input으로 target word를 예측하는 model



CBOW

-> NNLM과 유사 & 단어순서의 영향 없음

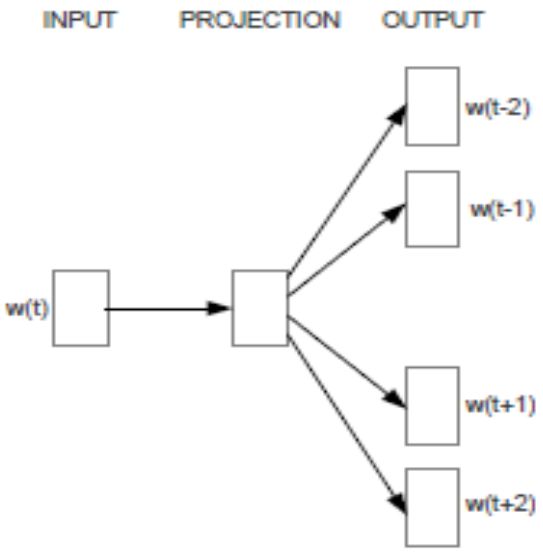
-> NNLM model 보다 **계산량 현저히 감소**

CBOW와 반대로 target word를 이용해

주변단어를 예측해보자!

Word2vec

Skip-gram: target word를 이용해 같은 문장내 특정범위내 N개의 주변 단어의 등장여부를 예측하는 model



Skip-gram

-> 주변 단어들은 target word와 가까울수록 높은 확률로 샘플링

-> NNLM model 보다 **계산량 현저히 감소**

CBOW보다 Skip-gram 성능이 더 좋아서

주로 Skip-gram model 이용!!

Word2vec

Skip-gram: 조건부 log probability를 최대화하는 weight 학습이 목표!!

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

-> 가정: 조건부독립 $$P(w_{c-m},\ldots,w_{c-1},w_{c+1},\ldots,w_{c+m}|w_C)=\prod_{j=0,\ j\neq m}^{2m}P(w_{c-m+j}|w_c)$$

Word2vec

Skip-gram의 학습특징

1. Hierarchical Softmax
   : 계산량이 많은 Softmax function 대신 빠르게 계산가능한 multinomial distribution function을 사용하는 방법

2. Negative Sampling
   : 전체 단어에 대해 계산하는 대신, 일부만 뽑아서 softmax를 계산하고 normalization 해주는 방법

3. Subsampling of Frequent Words
   : 자주 등장하는 단어들을 확률적으로 제외하고 학습하는 방법

Unit 03 | word2vec processing

Word2vec processing

1. Data 전처리
   : tokenize, 불필요한 품사 제거, stopword 제거

2. word2vec
   : 단어 embedding

   (2-1. 문서 embedding: word2vec 결과로 문서를 벡터화)

3. 분석

들어주셔서 감사합니다.