GloVe

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014.
GloVe: Global Vectors for Word Representation.

AI Robotics KR NLP Study

Presented by | Jeong Minsu

2019.08.27.

What is GloVe?

GloVe : Global Vector for Word Representation

Made in STANFORD

← Christopher D. Manning

- 단어를 벡터화하는 Word Embedding 방법론 중 하나.

What is GloVe?

GloVe : Global Vector for Word Representation

※ GloVe aims to achieve two goals

(1) Create word vectors that capture meaning in vector space.
(2) Takes advantage of global count statistics instead of only local information.

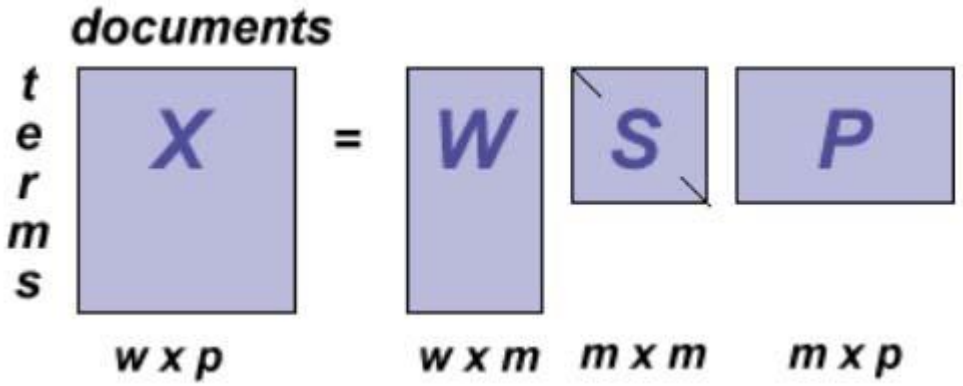
이전 워드 임베딩 방식에서의 한계를 극복!

Global Matrix Factorization Methods
(ex. Latent Semantic Analysis (LSA))

Local Context Window Methods
(ex. Word2Vec: CBOW, Skip-gram model)

Global Matrix Factorization Methods

(ex. Latent Semantic Analysis (LSA))

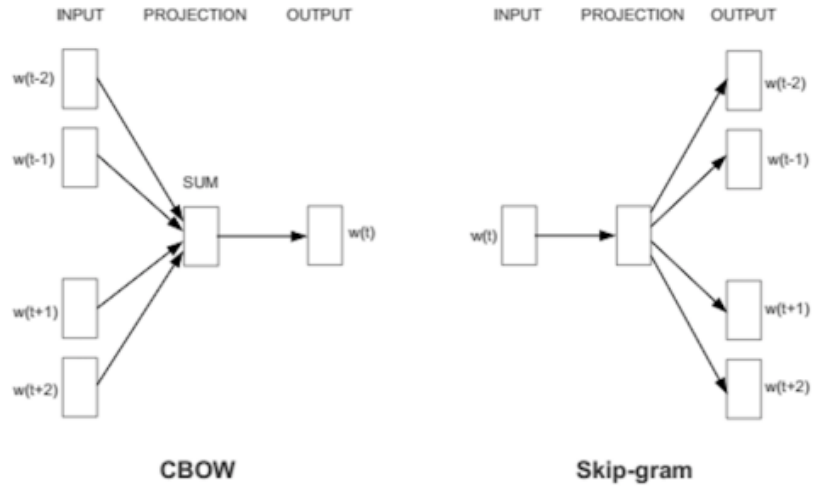


LSA (Latent Semantic Analysis, 잠재의미분석) :
TDM이나 TF-IDF 행렬과 같이 각 문서에서의 각 단어 빈도수를
카운트 한 행렬이라는 전체적인 통계 정보를 입력으로 받아
차원 축소(SVD)하여 잠재된 의미를 끌어내는 방법론.

! Problem
카운트 기반이기 때문에 전체적인 통계 정보를 고려하지만,
king : man = queen : ? 와 같은 단어 의미 반영이 미미하다.

Local Context Window Methods

(ex. Word2Vec: CBOW, Skip-gram model)



Word2Vec (CBOW, Skip-gram) :
중심 단어를 기준으로 정해진 윈도우 크기에 따라
슬라이딩 하면서 각 단어에 해당하는 벡터들의 요소 값을 학습
하는 워드 임베딩 방법론.

! Problem
임베딩 벡터가 Window 크기 내에서만 주변 단어를
고려하기 때문에 코퍼스 전체의 통계 정보 반영이 어렵다.

Co-occurrence Matrix (동시등장행렬)

"The cat sat on the mat" window size 2

	the	cat	sat	on	mat
the	2	1	2	1	1
cat	1	1	1	1	0
sat	2	1	1	1	0
on	1	1	1	1	1
mat	1	0	0	1	1

X_{ij} : 단어 j를 중심으로 단어 i가 나타난 횟수.

$\rightarrow X_{the\ sat} = 2$

$X_i = \sum_k X_{ik}$: 단어 i를 중심으로 한 모든 단어 출현 횟수.

$P_{ij} = P(i|j) = X_{ij}/X_i$

Glove의 아이디어: Context내에서 두 단어의 동시 등장 비율은 두 단어의 의미와 관련이 깊다!

Co-occurrence Probabilities (동시등장확률)

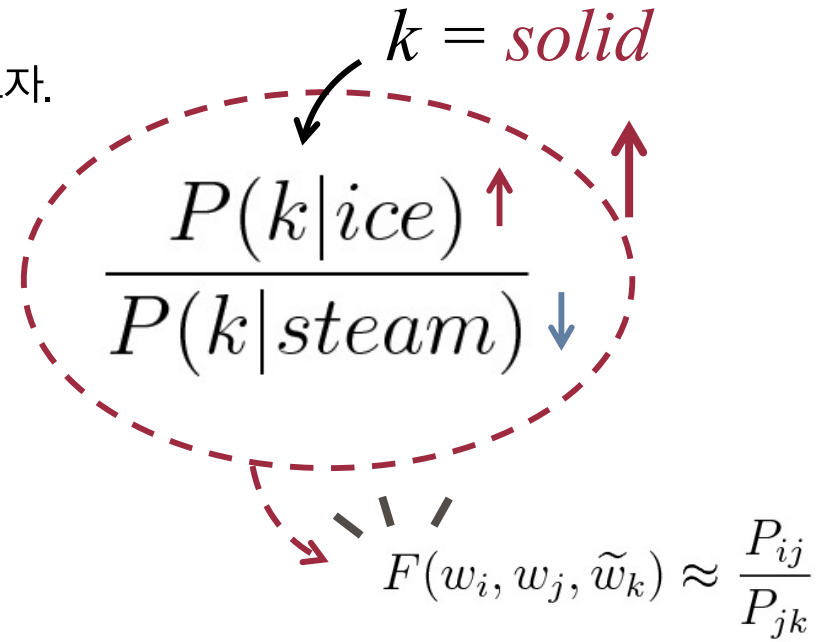
$$P_{ij} = P(i|j) = X_{ij}/X_i$$

※ $i = ice$

$j = steam$

$k = solid, gas, water, fashion$ 일 때, 동시등장확률이 어떻게 바뀌는지 보자.

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



Loss Function (1)

(1) $$F(w_i, w_j, \widetilde{w}_k) \approx \frac{P_{ik}}{P_{jk}}$$

w_i : input word i's embedding vector
w_j : input word j's embedding vector
$\widetilde{w}_k$: output word k's embedding vector

↓ vector 공간은 선형이고, 벡터 간의 연산이 의미를 가져야 하므로
단어 차이를 벡터의 차이로 input 할 수 있음.

(2) $$F(w_i - w_j, \widetilde{w}_k) \approx \frac{P_{ik}}{P_{jk}}$$

↓ 좌변은 벡터, 우변은 스칼라이기 때문에 형태를 통일하기 위해,
좌변의 인풋 벡터를 내적한다.

(3) $$F((w_i - w_j)^T \widetilde{w}_k) \approx \frac{P_{ik}}{P_{jk}}$$

Loss Function (2)

중심 단어와 주변 단어는 무작위 선택이므로, 둘 관계는 자유롭게 교환 가능.
→ F 가 실수의 덧셈과 양수의 곱셈에 대해서 준동형(Homomorphism)을 만족시켜야 함. (뺄셈은 나눗셈)

$F(a+b) = F(a)F(b), \forall a, b \in R$

$$F(v_1^T v_2 + v_3^T v_4) = F(v_1^T v_2)F(v_3^T v_4),\ \forall v_1,\ v_2,\ v_3,\ v_4 \in V$$

$$F(v_1^T v_2 - v_3^T v_4) = \frac{F(v_1^T v_2)}{F(v_3^T v_4)},\ \forall v_1,\ v_2,\ v_3,\ v_4 \in V$$

(4) $$F((w_i - w_j)^T \widetilde{w}_k) = \frac{F(w_i^T \widetilde{w}_k)}{F(w_j^T \widetilde{w}_k)}$$

F 는 e^x 형태이기 때문에 log로 분리할 수 있다.

(5) $$w_i^T \widetilde{w}_k = log(P_{ik}) = log(X_{ik}) - log(X_i)$$ $F(w_i^T \widetilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$ 단일 항목에 대한 방정식

w_i 와 $\widetilde{w}_k$ 두 값의 위치를 바꿔도 식이 성립해야 함. 하지만, $log(X_i)$ 때문에 불가능하다.
따라서 w_i 에 대한 편향 b_i 으로 대체함.

(6) $$w_i^T \widetilde{w}_k + b_i + \widetilde{b}_k = log(X_{ik})$$

$log(X_i)$ 는 bias에 의해 흡수

Loss Function (3)

학습을 통해 바뀌는 값

(7) $$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = log(X_{ik})$$

↓ 일반화

(8) $$J = \sum_{i,j=1}^{V} (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - log(X_{ij}))^2$$ (V = 단어 집합의 크기)

↓ 가중치 함수 $f(x) = min(1, (x/x_{max})^{3/4})$ 추가

(9) $$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - log(X_{ij}))^2$$

! Problem $log(X_{ij})$가 0이 될 수 있다.

Co-occurrence Matrix는 많은 값이 0이거나, 동시 등장 빈도가 적어서 작은 수치를 가지는 경우가 많음.

따라서, 가중치 함수를 손실 함수에 도입.

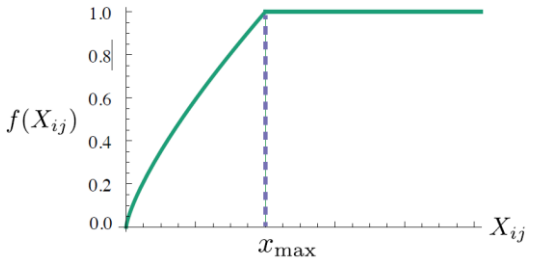
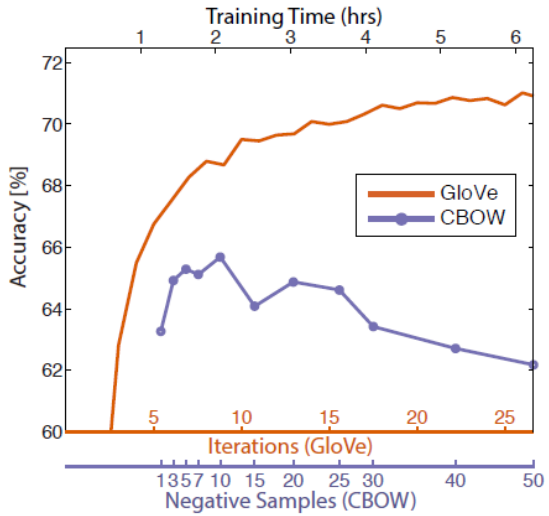


Figure 1: Weighting function f with $\alpha = 3/4$.

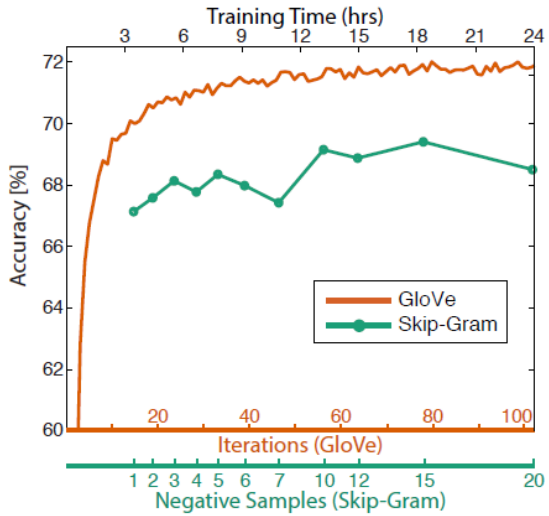
X_{ij}값에 따라서 상대적으로 함수의 값을 다르게 결정함. X_{ij}값이 작으면 함수 값은 작게, 크면 함수의 값이 크게. (하지만 최대값 1을 넘을 수 없음.)

$$f(x) = min(1, (x/x_{max})^{3/4})$$

결과 : word2vec과 비교



(a) GloVe vs CBOW



(b) GloVe vs Skip-Gram

- GloVe가 word2vec보다 일관된 학습을 보여준다.
- 이후 논문들 → GloVe와 word2vec은 수학적으로 동일한 모델

GloVe 한계

출처 : https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/03/11/embedding/

- 나는 ___에 간다.
  └ 회사, 집, 슈퍼스타트 등 모두 들어갈 수 있음.
- 회사, 집, 슈퍼스타트 등은 명백히 다른 단어임에도 임베딩 시, 유사도가 매우 높게 나타남.
- GloVe가 '단어 동시 등장 정보 ' 를 보존하는 특성 때문.
- 학습 말뭉치가 충분히 크다면 단어들의 사용 사례가 다양해지기 때문에 이런 문제가 해소될 수 있음.
- 하지만 말뭉치가 크지 않을 경우, 이런 문제가 발생할 수 있음.

→ ELMo 의 등장

감사합니다.