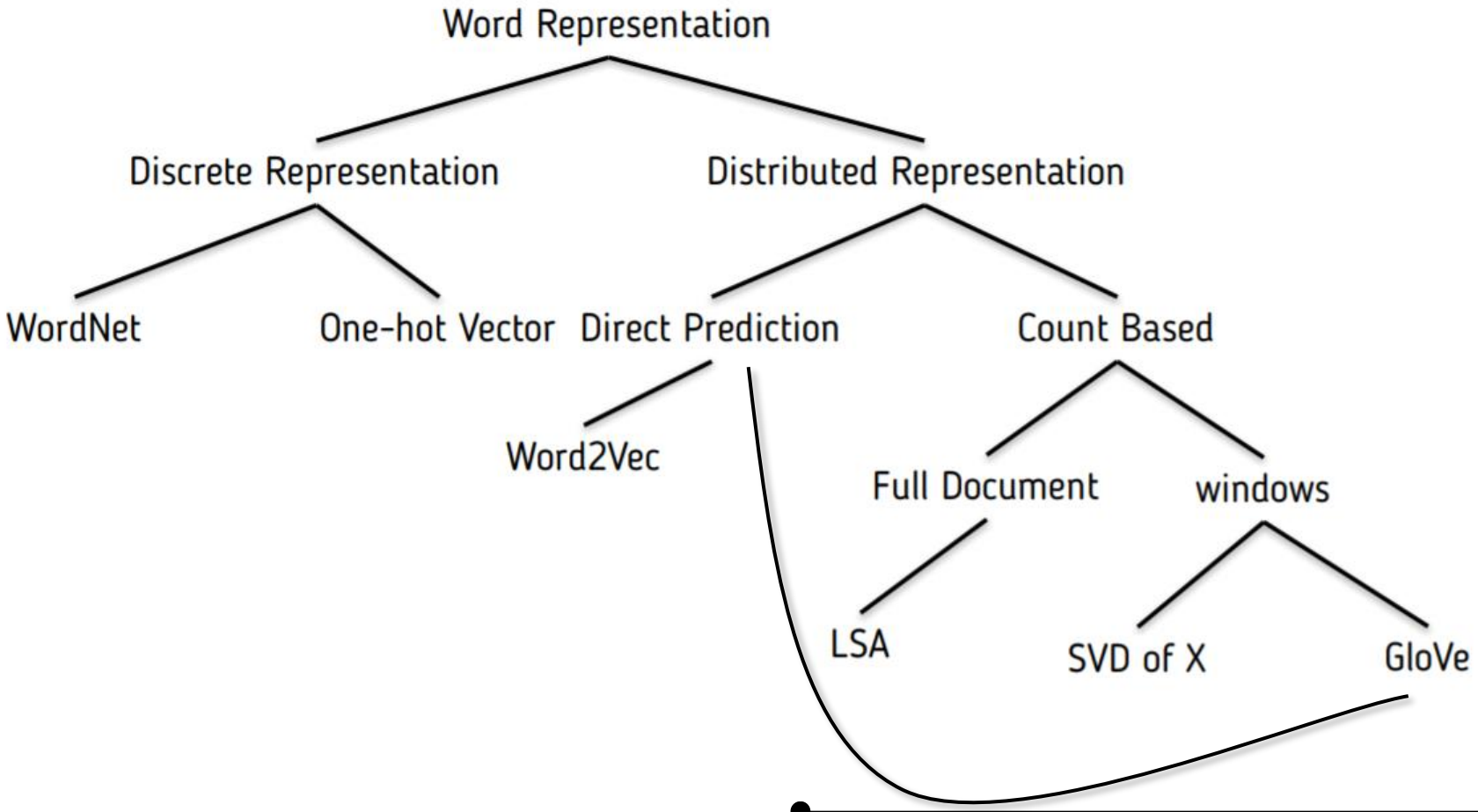
GloVe:

Global Vectors for Word Representation

ESC 2020-2

1 Background

Word Representation 분류



1 Background

Discrete Representation

Dictionary 기반 (e.g. WordNet)
혹은 One-hot Vector를 통한 Representation
(Sparse representation)

Pros:

(Dictionary 기반의 경우)
- 사람이 이해할 수 있는 형태의 Representation

(One-hot Vector의 경우)
- 비교적 간단하게 구축할 수 있다, 직관적이다.

Cons:
- 단어의 관계(e.g. 유사도, 반의어, 문법 등)를 측정할 수 없다
- 사람이 직접 구축해야 한다. 주관적인 판단이 개입될 수 있다
- 새로운 단어가 나올 경우 일일이 대응해야 한
- 데이터가 많은 경우 size가 급격하게 늘어남

Distributed Representation (Dense Represenatation)

단어의 출현 빈도를 기반으로 계산한 Word Vector
-〉 하나의 정보가 여러 차원에 분산되어 표현

Pros:
- 단어의 관계를 측정, 표현 할 수 있다
- 비지도 학습!
- 새로운 단어가 나올 경우 Corpus만 제공하면 된다
- 다른 모델들과 결합해서 추가적인 정보를 제공한다

Cons:
- 성능을 측정하기가 쉽지 않다
- Train하는데 오랜 시간이 걸린다
- Non-uniform results

1 Background

Count based vs Prediction based

Count Based:
Global Matrix Factorization

LSA, HAL, COALS, Hellinger-PCA, etc.

Pros:
- 단어수가 적은 경우 학습 빠름
- Global statistics의 효율적 활용

Cons:
- 단어 유사도 파악 까지가 한계 (analogy 문제에 약하다)
- 빈도수 큰 단어들에 대해 불균형 (Disproportionate)

Prediction Based:
Local Context Window Methods

NNPM, HLBL, RNN, Skip-gram, CBOW, etc.

Pros:
- 단순 단어 유사도를 넘어선 복잡한 의미적 구조 파악 -〉 word vectors can pose semantic or syntactic relationships
- 성능 개선

Cons:
- Corpus가 큰 경우 학습량 多
- Global Statistics 활용하지 못함

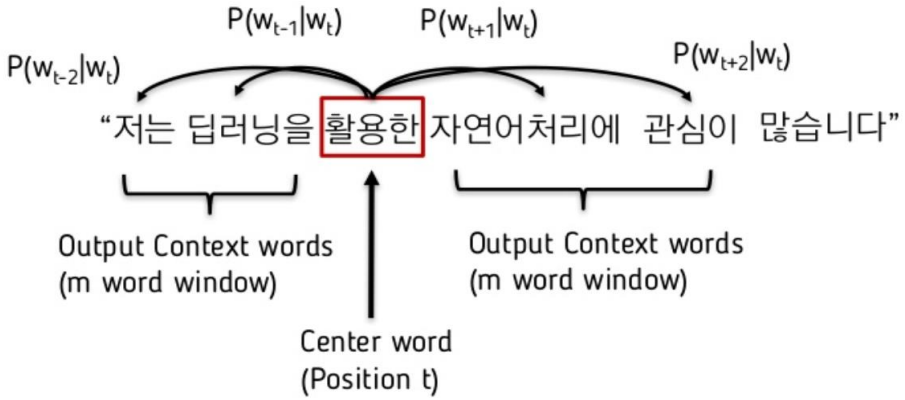
두 Approach의 장점을 살린 모델을 만들어보자!

1 Background

그런데 Word2Vec에서 window size를 corpus 전체로 하면 되지 않을까?

Word2Vec

Skip-gram, CBOW는 전체 Corpus에서 window size만큼 한 단어 한 단어 씩 훑어가며 학습한다.



하지만 Skip-Gram, CBOW의 window size를 늘리게 되면

중심단어 기준으로 대부분의 단어들을 주변 단어로 인식하기 때문에 각 단어의 벡터는 서로 변별력을 찾기 어렵고 연산이 더 많아진다.

2 Objective function

Equation (1): Starting Point

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}},$$

$$F(w_{ice}, w_{steam}, w_{solid}) = \frac{P_{ice,solid}}{P_{steam,solid}} = \frac{P(solid|ice)}{P(solid|steam)} = \frac{1.9 \times 10^{-4}}{2.2 \times 10^{-5}} = 8.9$$

$$w \in \mathbb{R}^d \quad \tilde{w} \in \mathbb{R}^d$$

word vectors　　context word vectors

Equation (2): Vector Difference

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}.$$

벡터 차

확률비

2 Objective function

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}.$$

벡터 차

확률비

Equation (3): Inner Product

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}},$$

Dot product

Equation (4): Homomorphism

A word and a context word should be an arbitrary distinction.

$w \leftrightarrow \tilde{w}$

For this to be possible, our co-occurrence matrix needs to be symmetric.

$X \leftrightarrow X^T$

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)},$$

(R,+)

(R>0, ×)

$$F((w_i^T - w_j^T)w_k') = F(w_i^T w_k' + (-w_j^T w_k')) = F(w_i^T w_k') \times F(-w_j^T w_k') = F(w_i^T)$$

$$\times F(w_j^T w_k')^{-1} = \frac{F(w_i^T w_k')}{F(w_j^T w_k')}$$

2 Objective function

Equation (5): Simplifying Equation

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}}$$

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)} = P_{ik} = \frac{X_{ik}}{X_i}$$

2 Objective function

F : exponential function

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)},$$

$$\downarrow$$

$$F(a - b) = F(a) / F(b)$$

$$\downarrow$$

$$e^{a-b} = e^a / e^b$$

2 Objective function

Equation (6): Simplifying Equation

$$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

Exp 적용 후 양변에 로그 취함

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

2 Objective function

Equation (7): Restoring Symmetry

$$w_i^T \tilde{w}_k = \log(X_{ik}) - \log(X_i)$$

Symmetric

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

Symmetric

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

Symmetric

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

학습 필요한 함수 고정값

2 Objective function

Equation (8): Cost Function

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

가중치

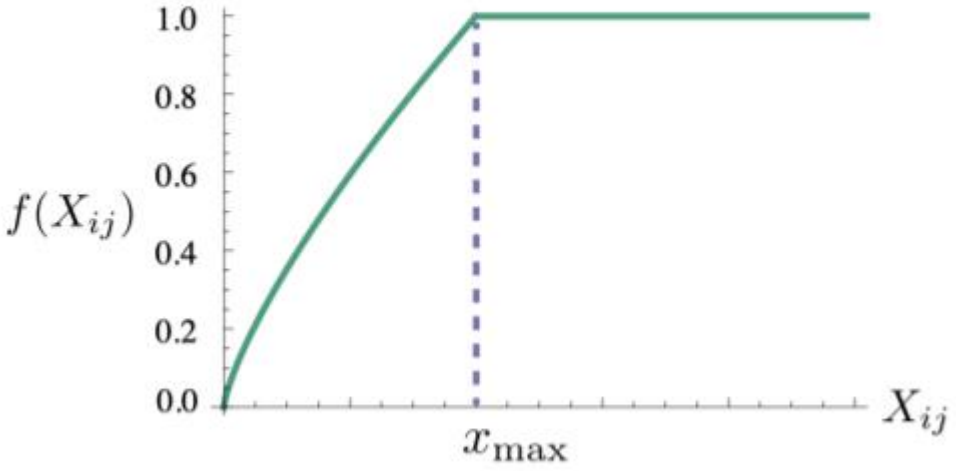
Least Squares

2 Objective function

Equation (9): Weighting Function

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

가중치 Least Squares



$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}$$

Weighting function f with $\alpha = 3/4$

fix to $x_{\max} = 100$

1. f (0) = 0. If f is viewed as a continuous function, it should vanish as x → 0 fast enough that the limx- $\lim_{x \to 0} f(x) \log^2 x$ te.
2. f (x) should be non-decreasing so that rare co-occurrences are not over-weighted.
3. f (x) should be relatively small for large values of x, so that frequent co-occurrences are not over-weighted.

3 Relationship to Other Models

Model

$$Q_{ij} = \frac{\exp(w_i^T \tilde{w}_j)}{\sum_{k=1}^{V} \exp(w_i^T \tilde{w}_k)} \qquad P(o|c) = \frac{exp(u_o^T v_c)}{\sum_{w=1}^{W} exp(u_w^T v_c)}$$

Cost function

$$J(\theta) = -\frac{1}{T}\log L(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-m\le j\le m \\ j\neq 0}} \log P\left(w_{t+j} \mid w_t; \theta\right)$$

$$J = -\sum_{\substack{i\in\text{corpus} \\ j\in\text{context}(i)}} \log Q_{ij} \,.$$

$$J = -\sum_{i=1}^{V}\sum_{j=1}^{V} X_{ij} \log Q_{ij}$$

Recalling our notation for $X_i = \sum_k X_{ik}$ and $P_{ij} = X_{ij}/X_i$, we can rewrite J as,

$$J = -\sum_{i=1}^{V} X_i \sum_{j=1}^{V} P_{ij} \log Q_{ij} = \sum_{i=1}^{V} X_i H(P_i, Q_i)$$

3 Relationship to Other Models

$$J = -\sum_{i=1}^{V} X_i \sum_{j=1}^{V} P_{ij} \log Q_{ij} = \sum_{i=1}^{V} X_i H(P_i, Q_i)$$

Cross Entrophy — 두 확률분포 사이의 distance measure

↓

다른 distance measure 적용 — Least squares

$$\hat{J} = \sum_{i,j} X_i (\hat{P}_{ij} - \hat{Q}_{ij})^2$$

$$\begin{aligned} \hat{J} &= \sum_{i,j} X_i (\log \hat{P}_{ij} - \log \hat{Q}_{ij})^2 \\ &= \sum_{i,j} X_i (w_i^T \tilde{w}_j - \log X_{ij})^2 . \end{aligned}$$

where $\hat{P}_{ij} = X_{ij}$ and $\hat{Q}_{ij} = \exp(w_i^T \tilde{w}_j)$

$$\hat{J} = \sum_{i,j} f(X_{ij}) (w_i^T \tilde{w}_j - \log X_{ij})^2$$

general weighting function $f(X_{ij})$

We could also include bias terms

$$\hat{J} = \sum_{i,j} f(X_{ij}) (w_i^T \tilde{w}_j - \log X_{ij})^2$$

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

4 Conclusions

Pros:

- 전반적인 NLP task에서 보통 Word2Vec보다 성능이 좋음. -〉 ? ?
- 단어와 단어보다는 단어 쌍과 단어 쌍 사이의 관계를 고려하여 단어 벡터에 좀 더 실용적인 의미 추가.
- "the"와 같은 무의미한 stop words에 가중치를 낮게 줌

Cons:

- 계산 복잡성이 높고 메모리를 많이 필요로 함. 특히, 동시 발생 행렬과 관련된 하이퍼 파라미터를 변경하는 경우 행렬을 다시 재구성해야 하므로 시간이 많이 걸림.

- 반대 단어 쌍을 분리하는 방법. 예를 들어, "양호한"및 "나쁜"은 일반적으로 벡터 공간에서 서로 매우 가깝게 위치하므로 정서 분석과 같은 NLP 작업에서 단어 벡터의 성능이 제한(Word2Vec도 동일한 문제를 안고 있음)

5 Conclusions

- **Word2vec and Glove** word embeddings are context independent (context free)
  – these models output just one vector (embedding) for each word, combining all the different senses of the word into one vector.

- **ELMo and BERT** can generate different word embeddings for a word that captures the context of a word – that is its position in a sentence.

"*He went to the prison **cell** with his **cell** phone to extract blood **cell** samples from inmates*"

Thank you