Insight campus SeSAC

임베딩 (Embedding)

실무형 인공지능 자연어처리

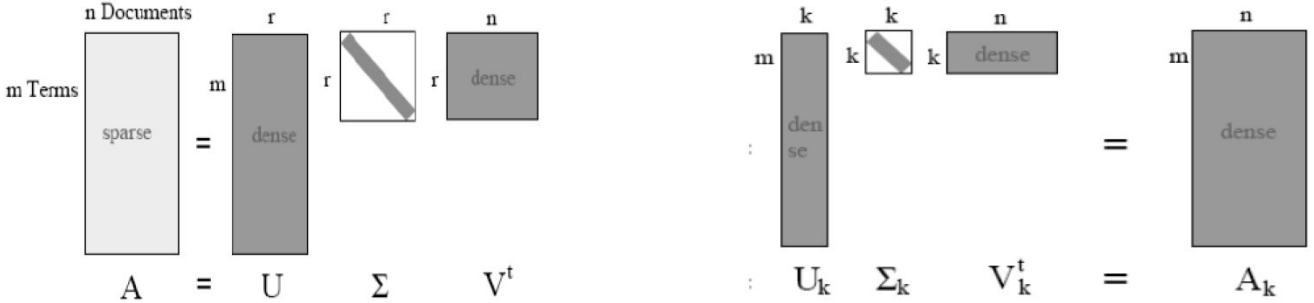
2

GloVe

Global Vectors for Word Representation

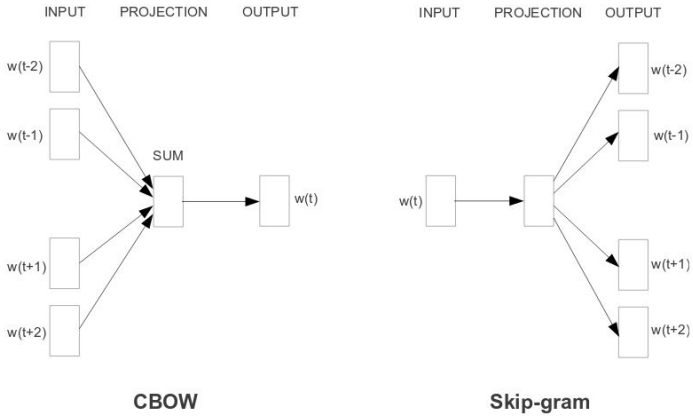
기존 임베딩의 문제점 (1)

- Global Matrix Factorization (예. LSA)
  - 단어-문서 또는 단어-단어 행렬을 분해(예. ED, SVD)하여 저차원 공간에 단어 분산 표현 (Distributed Representation)
  - 단어에 대한 **<u>전체적인 통계정보를 활용</u>**한다는 점이 강점
  - **<u>단어 유추 문제에 좋지 않은 성능</u>**을 보임
- https://nlp.stanford.edu/projects/glove/



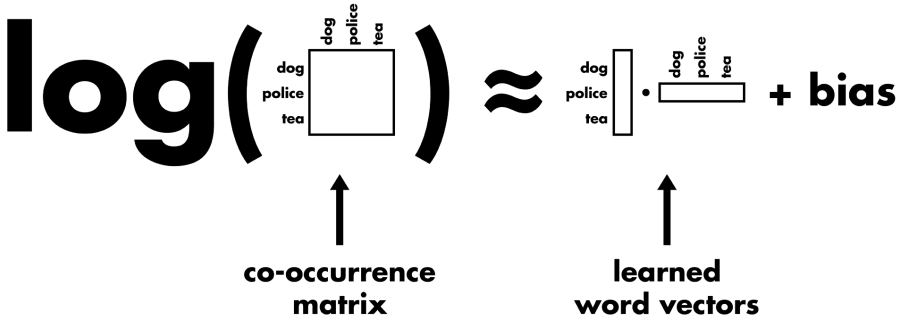
기존 임베딩의 문제점 (2)

- Shallow Window-Based Method (예. Word2Vec)
  - 지역적인 문맥(local context) 정보를 한정적으로 사용하여 단어를 vector로 표현
  - **단어 유추 문제에서는 비교적 좋은 성능**을 보임
  - 학습 데이터(corpus)에서 관찰되는 **단어사용 통계정보를 활용하지 않는 다**는 점에서 한정적
  - 지역적 문맥에 대한 학습은 가능, 학습 데이터(corpus)에서 관찰되는 서로 다른 두 단어의 동시발생 횟수 (co-occurrence)에 기반한 학습은 할 수 없음



GloVe (GloVe: Global Vectors for Word Representation)

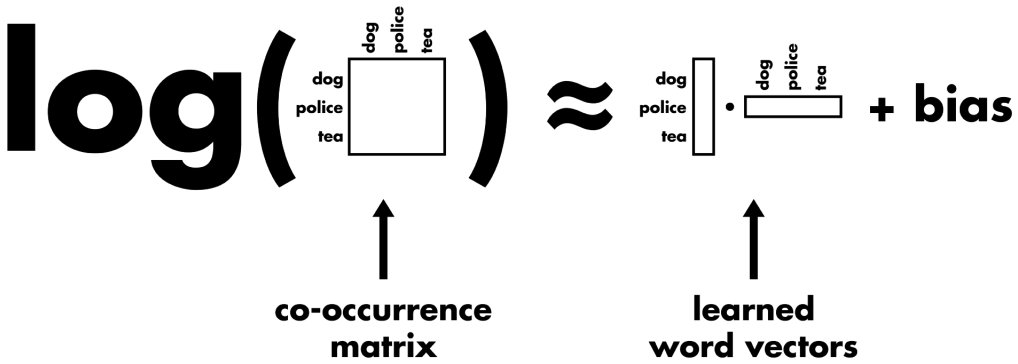
- GloVe는 2013년 구글에서 개발한 Word2Vec의 단점을 보완
- “임베딩된 단어 벡터간 유추문제에 좋은 성능을 보이면서(word2vec의 장점) 말뭉치 전체의 통계 정보를 반영(LSA의 장점)”이 GloVe핵심 목표
- 임베딩된 두 단어벡터의 내적이 말뭉치 전체에서의 동시 등장확률 로그값이 되도록 목적함수를 정의 (their dot product equals the logarithm of the words’ probability of co-occurrence)
- https://nlp.stanford.edu/projects/glove/



[그림]. https://towardsdatascience.com/emnlp-what-is-glove-part-v-fa888272c290

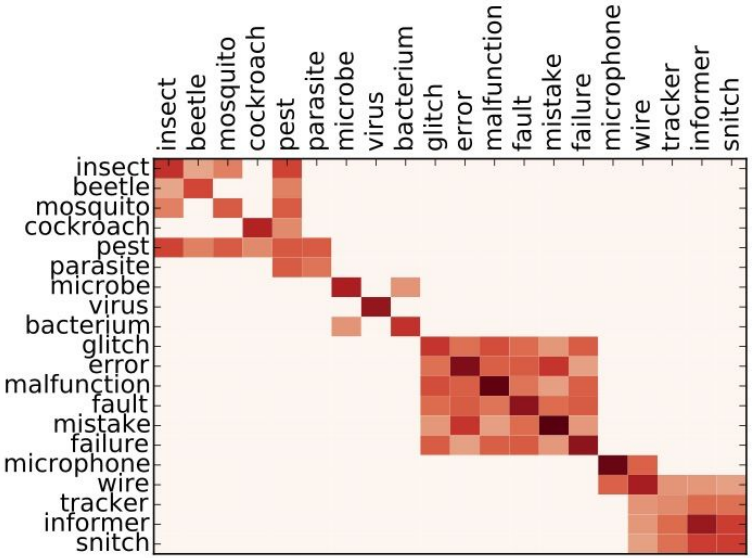
GloVe의 목적함수

- 임베딩된 두 단어벡터의 내적이 말뭉치 전체에서의 동시 등장확률 로그값이 되도록 목적함수를 정의 (their dot product equals the logarithm of the words' probability of co-occurrence)
- 특정 단어 k가 주어졌을 때 임베딩된 두 단어벡터의 내적이 두 단어의 동시등장확률 간 비율이 되도록 임베딩
  - solid라는 단어가 주어졌을 때 ice와 steam 벡터 사이의 내적값이 8.9가 되도록
  - gas가 주어졌을 때 ice와 steam 벡터 사이의 내적값이 0.0085가 되도록



단어-문맥 행렬(Term-Context matrix)

- 단어-문맥 간의 동시등장(co-occurrence) 행렬
- 문맥은 사용자가 설정한 window의 크기로 결정
- 문맥 내 등장하는 단어의 빈도를 표기



Co-occurrence probability

- 단어간 co-occurrence 행렬을 생성
- ice와 steam의 단어가 있음
  - ice가 사용된 문맥에서, 단어 k도 사용되었을 확률
  - steam이 사용된 문맥에서, 단어 k도 사용되었을 확률
  - P(k | ice)/P(k | steam) 상대 비율. 둘 중 상대적으로 더 많이 사용된 곳을 구분
- water와 fashion은 P(k | ice)/P(k | steam) 비율이 1에 가까워 steam과 ice를 구분에 비적합
- solid와 gas의 경우 P(k | ice)/P(k | steam) 값이 1보다 월등히 크거나 작음, 따라서 구분에 적합

Probability and Ratio	*k = solid*	*k = gas*	*k = water*	*k = fashion*
P(k | ice)	1.9 x 10^-4	6.6 x 10^-5	3.0 x 10^-3	1.7 x 10^-4
P(k | steam)	2.2 x 10^-5	7.8 x 10^-4	2.2 x 10^-3	1.8 x 10 ^-5
P(k | ice)/P(k | steam)	8.9	8.5 x 10 ^-2	1.36	0.96

GloVe의 목적함수

- P_{ik}를 P(k | i)로 정의
  - i 번째 단어 주변(윈도우 크기는 사용자 지정)에 k번째 단어가 등장할 조건부확률
  - 빈도수(X_{ik})'를 '$X_i=\Sigma_k X_{ik}$'로 나눠준 값입니다. 위 표 기준으로 예를 들면 P(solid | ice) 정도의 의미가 되겠네요.
- P_{ik}/P_{jk}의 의미 : P(solid | ice)/P(solid | steam) = 8.9
- d차원 벡터공간에 임베딩된 ice, steam, solid 벡터를 넣으면 8.9를 반환하는 F(목적함수)를 찾는 최적화 문제

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

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

Weighted error

- 목적함수에 아래와 같은 모양의 f(X)를 추가. x_{ij}가 특정 값 이상 빈도가 큰 경우 가중치를 조정.
- 범위 초과($x_{ij} > x_{max}$) 빈도를 가지는 단어들은 error를 그대로 학습에 사용
- 범위 이내($x_{ij} < x_{max}$) 빈도를 가지는 단어들(=infrequent word)은 error의 중요도를 낮춰 학습에 사용

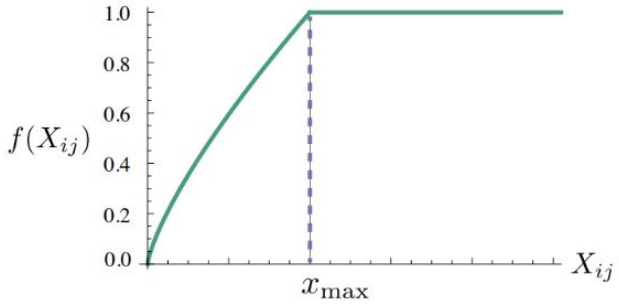


Figure 1: Weighting function f with $\alpha = 3/4$.

Error

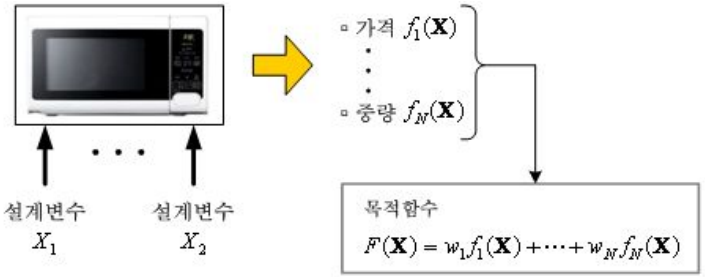
$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

$$where \quad f(x) = \begin{cases} \left(\frac{x}{x_{max}}\right)^{\alpha} & if \quad x < x_{max} \\ 1 & otherwise \end{cases}$$

최적화 문제

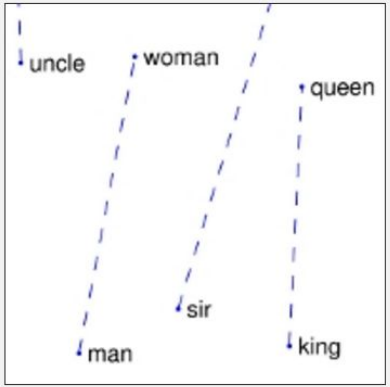
편리한 인간생활을 추구하기 위하여 개발된 각종 제품들은 각기 고유한 성능을 제공하기 위하여 설계되었다. 이러한 성능들을 가장 잘 만족시키는 제품을 설계하는 일을 **최적설계(optimum design)**라고 부르고, 가장 최적으로 만족시키고자 설계한 성능을 특별히 **목적함수**로 정의하고 있다. 해당 설계업무 시 고려의 대상이 되는 성능만이 목적함수에 해당된다. 따라서 해당 제품의 개발 목표에 따라 목적함수가 달라지게 되며, 각 목적함수 내에 포함되어 있는 세부 성능들의 상대적인 중요도도 달라질 수 있다.

하나 이상의 세부성능들로 구성된 목적함수를 특별히 다목적 함수(multiobjective function)라고 부르며, 일반적으로 각 세부성능에 가중치(weighting factor)를 곱하여 대수적으로 합한 것으로 정의된다.

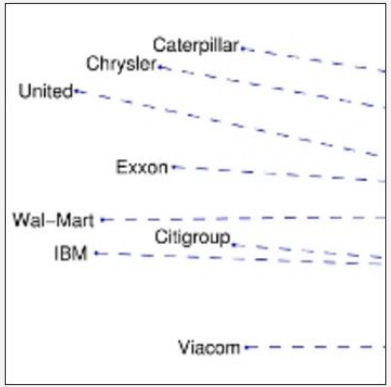


https://kor.midasuser.com/nfx/techpaper/keyword_view.asp?idx=223

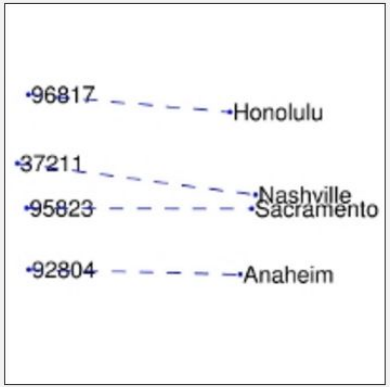
GloVe의 결과



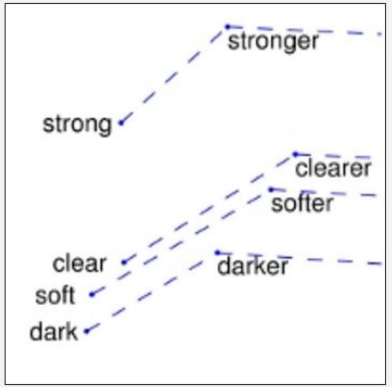
man - woman



company - ceo



city - zip code



comparative - superlative

감사합니다.

Insight campus SeSAC

daniel@fins.ai