Word Embedding

Jaegul Choo (주재걸)
고려대학교 컴퓨터학과
https://sites.google.com/site/jaegulchoo/

Word Embedding

Contents

1. Word Embedding

1-1. Word Embedding이란**?**
1-2. Word2Vec 모델 및 알고리듬
1-3. Word2Vec 적용분야

01

Word Embedding 이란?

- 한 단어를 **벡터 형태**로 표현
- ‘cat’과 ‘kitty’는 **비슷한 단어**이므로 **비슷한 단어 표현형**을 가짐
- ‘hamburger’는 ‘cat’, ‘kitty’와 비슷하지 않으므로 다른 표현형 = 거리가 멀다

기존의 Word Representation 방법

- 가장 쉬운 방법으로는 “one-hot” representation
- e.g., horse = [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
- zebra = [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0]

- horse와 zebra 간의 유사도를 구하는 방법 중 하나를 생각해보면

  horse ∩ zebra = 0 (nothing)
- 그러나, 우리는 horse와 zebra가 최소한 포유동물이라는 공통점을 가진다는 것을 알고 있음.

기존의 방법으로 Word 간 유사도 구하려면?

- 전 슬라이드의 방법을 사용하여 term-document matrix 구축
- e.g., Document 1 = "John likes movies. Mary likes too."
  Document 2 = "John also likes football."
- 두 간어가 얼마나 자주 같은 문서에서 등장했나로 유사도를 측정

Vocabulary	Doc 1	Doc 2
John	1	1
likes	2	1
movies	1	0
also	0	1
football	0	1
Mary	1	0
too	1	0

Co-occurrence similarity

- John & likes : 2 (documents)
- movies & also: 0 (documents)

Word의 기존의 다른 Vector Representation 방법들

- Matrix factorization 기반 기법들
  - **Singular vector representation** (혹은 latent semantic indexing)
  - Nonnegative matrix factorization
- Probabilistic topic modeling 기반 기법들
  - Probabilistic latent semantic indexing
  - Latent Dirichlet allocation

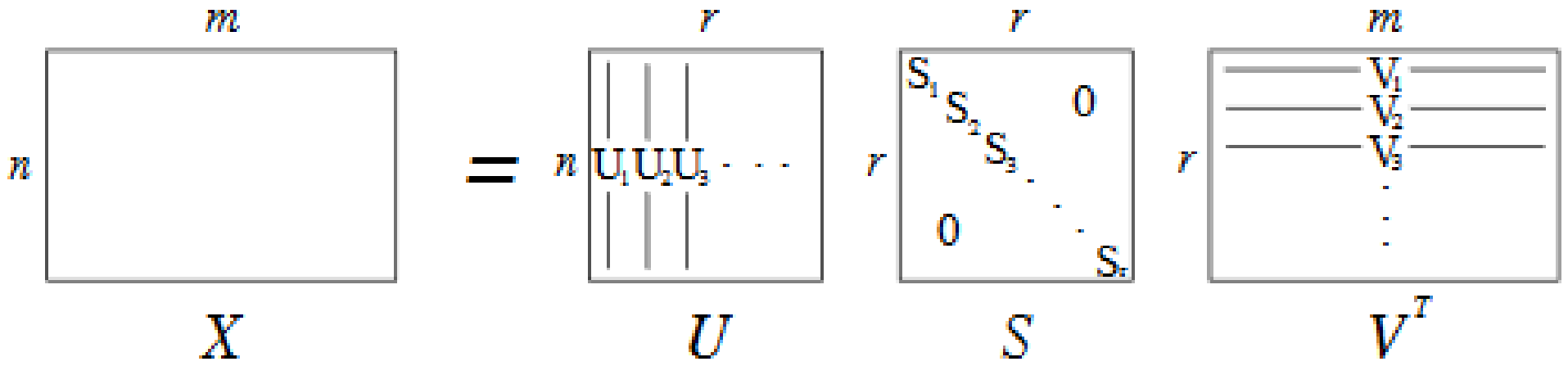
Singular Value Decomposition

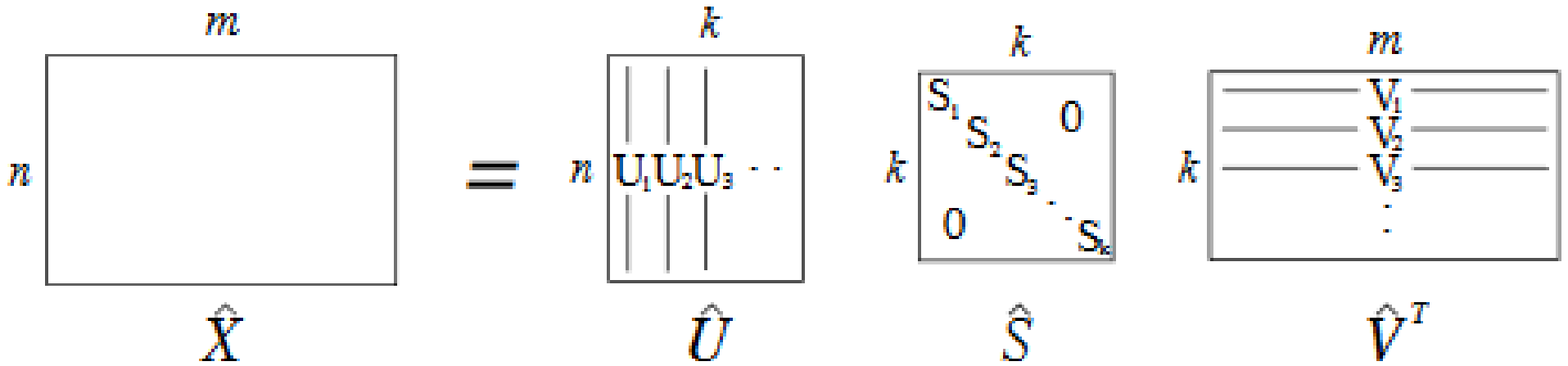
Corpus = {"I like deep learning"
"I like NLP"
"I enjoy flying"}

Context = previous word and next word

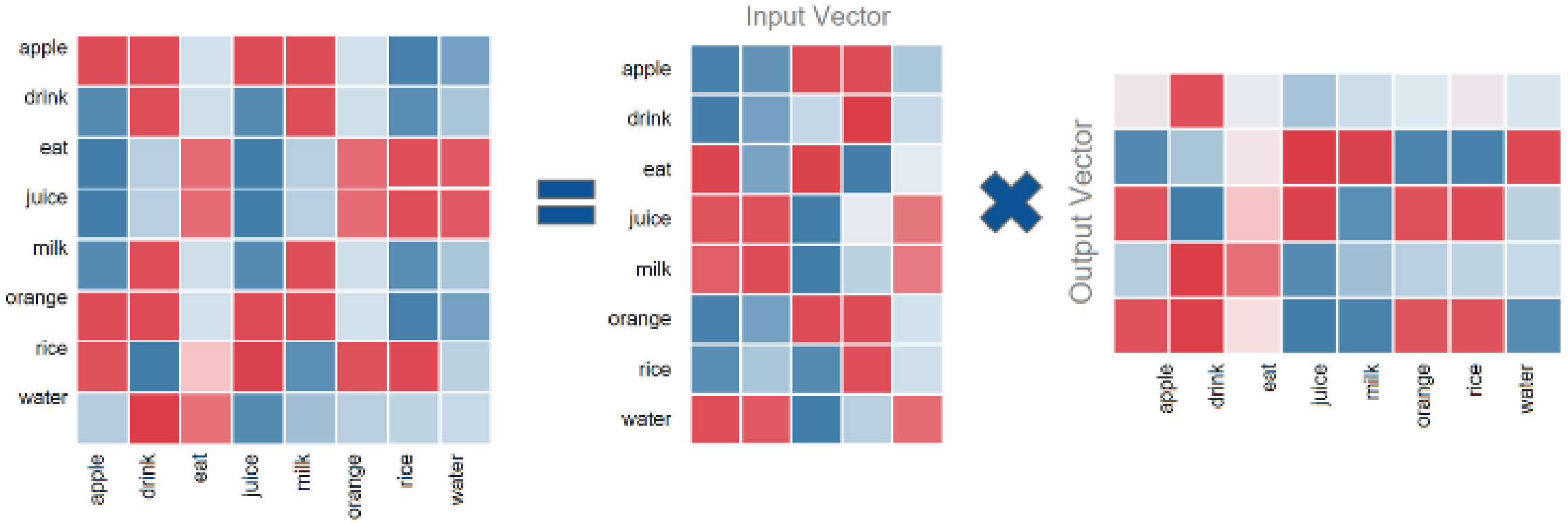
counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Dimension Reduction using Singular Value Decomposition





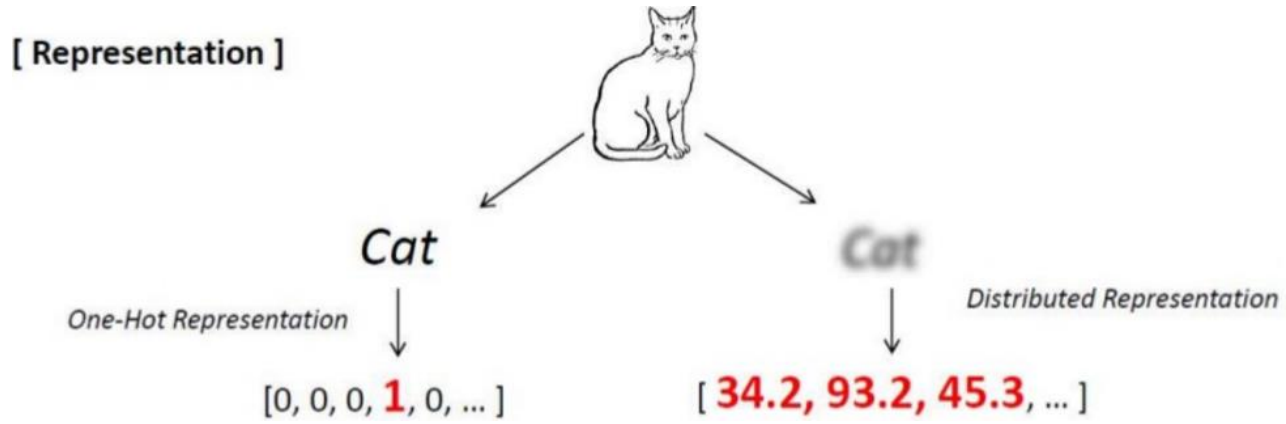
Singular Value Decomposition Example



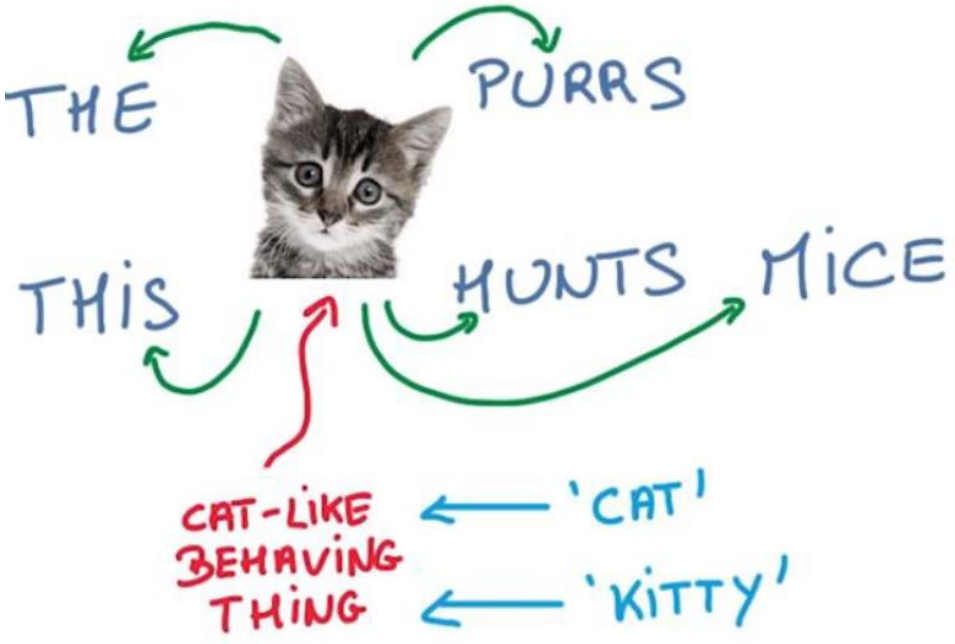
The problem with this method, is that we may end up with matrices having billions of ro ws and columns, which makes SVD computationally restrictive.

Word2Vec

- Word의 distributed vector representation 학습기법



- **주변 단어**로 해당 단어를 학습시키는 알고리즘
  - 주변에 있으면 **단어의 의미가 더 비슷**할 것이라 가정
- **예시)**
  - The **cat** purrs.
  - This **cat** hunts mice.

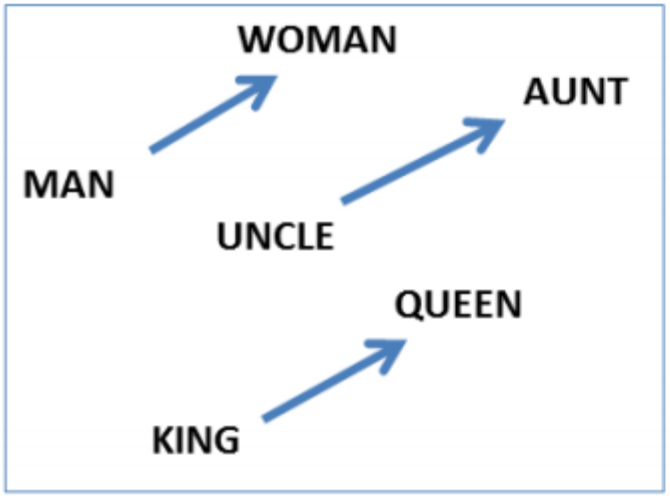


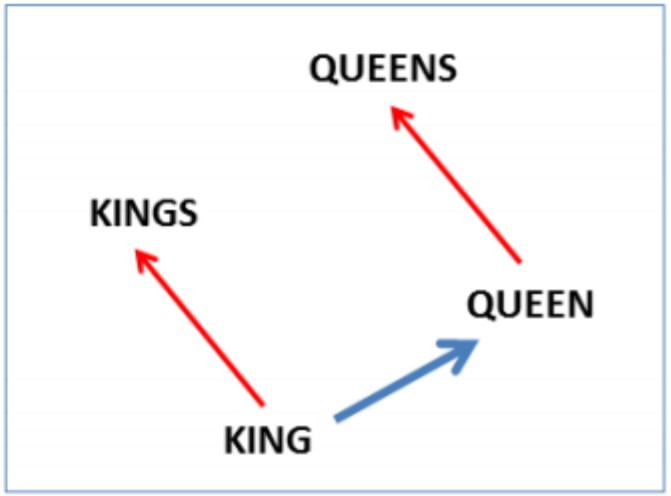
Word2Vec의 역사

- Learning representations by back-propagating errors (Rumelhart et al., 1986)
- A neural probabilistic language model (Bengio et al., 2003)
- NLP from Scratch (Collobert & Weston, 2008)
- Word2Vec (Mikolov et al., 2013):
  - Efficient Estimation of word Representation in Vector Space
  - Distributed Representations of words and phrases and their compositionality
  - 기존의 인공신경망으로 구성된 워드 임베딩 모델에 비해 **계산량/복잡도 감소** (negative sampling, hierarchical softmax)

Word2Vec의 특성

- 워드 벡터, 즉 공간에서의 벡터 포인트 간의 관계는 해당 단어들 간의 관계를 나타냄.
- 같은 관계는 같은 벡터로 나타남.





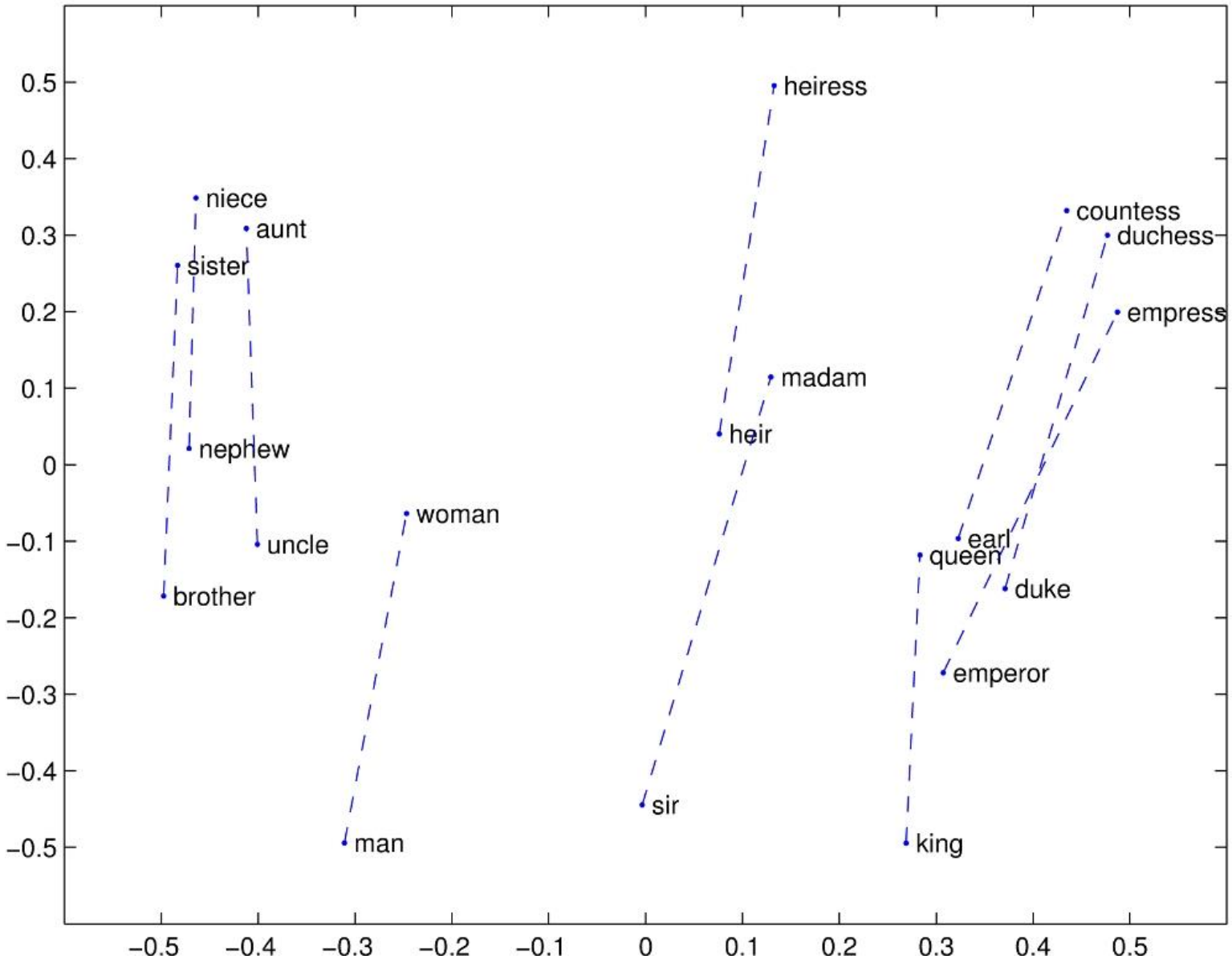
(Mikolov et al., NAACL HLT, 2013)

- e.g.,

vec[queen] - vec[king] = vec[woman] - vec[man]

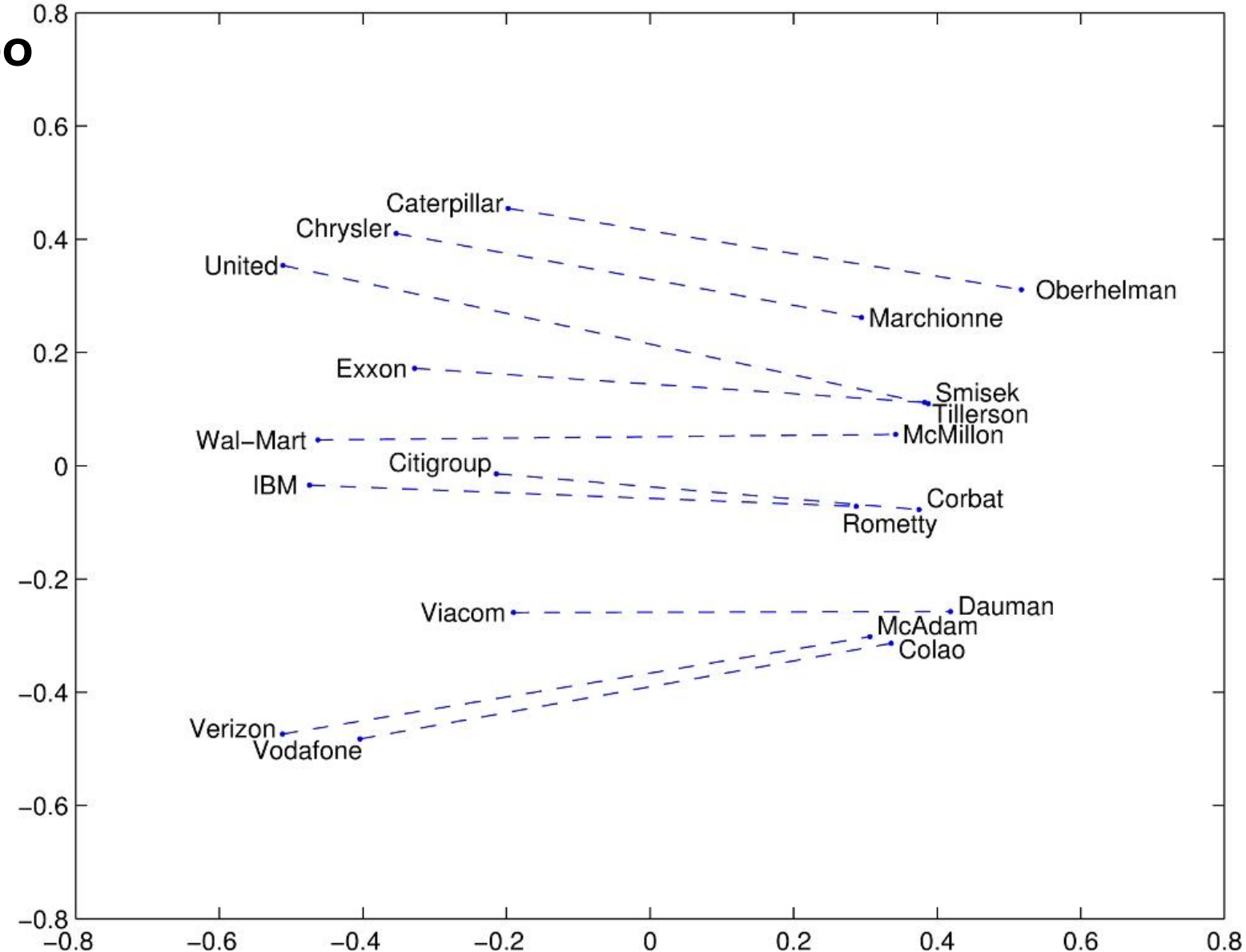
Word2Vec의 특성 – Linear Substructure

man - woman



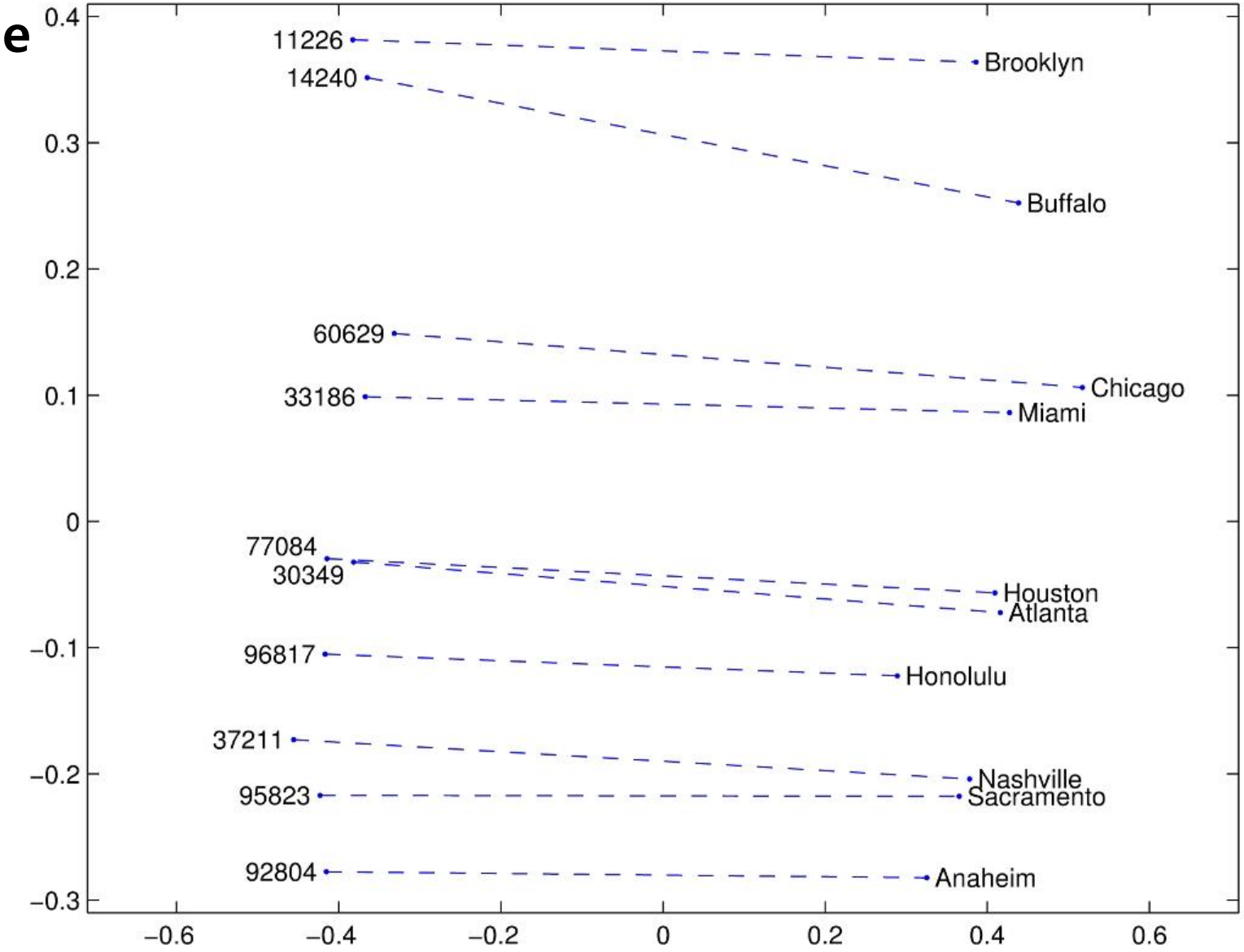
Word2Vec의 특성 – Linear Substructure

company - ceo



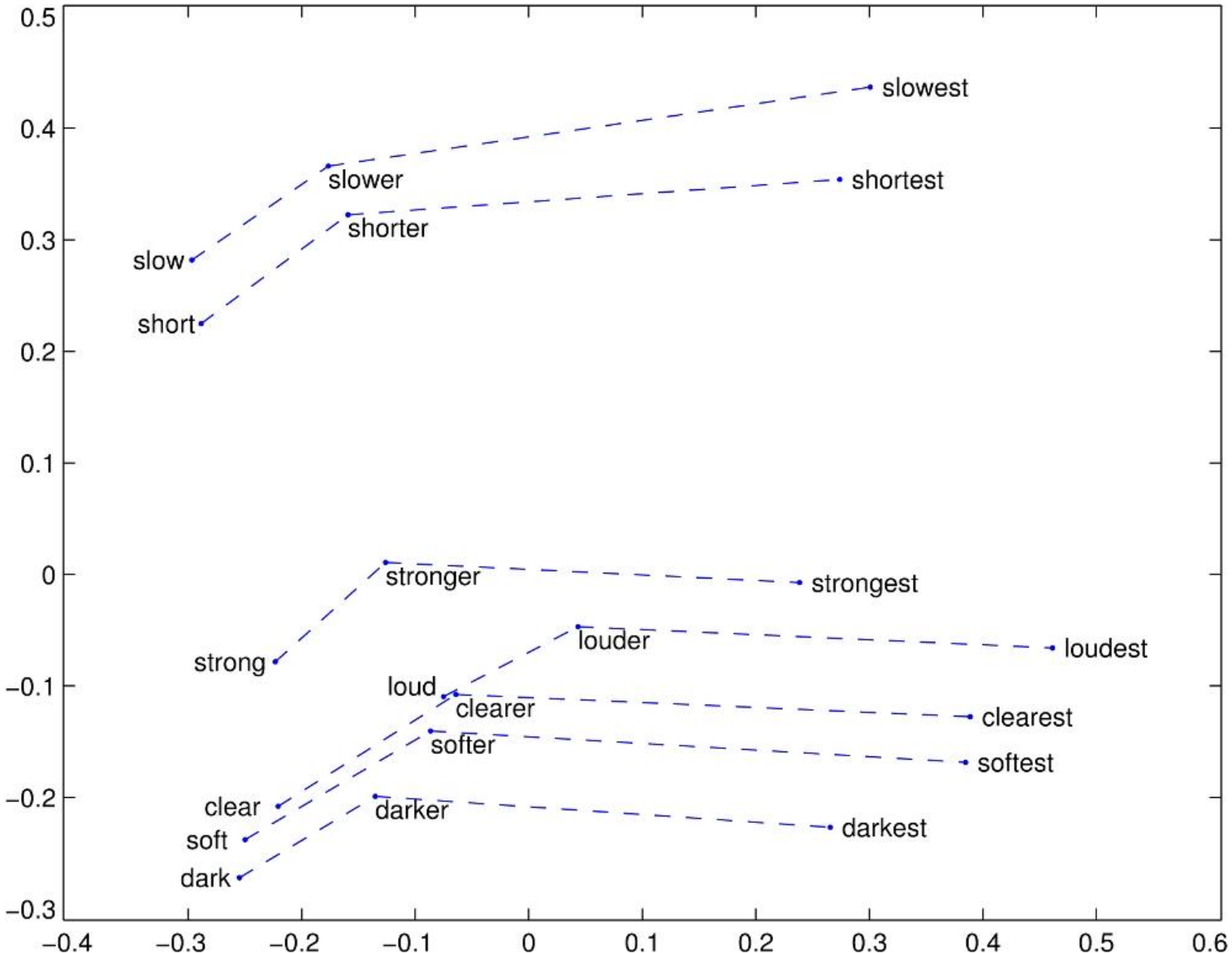
Word2Vec의 특성 – Linear Substructure

city - zip code



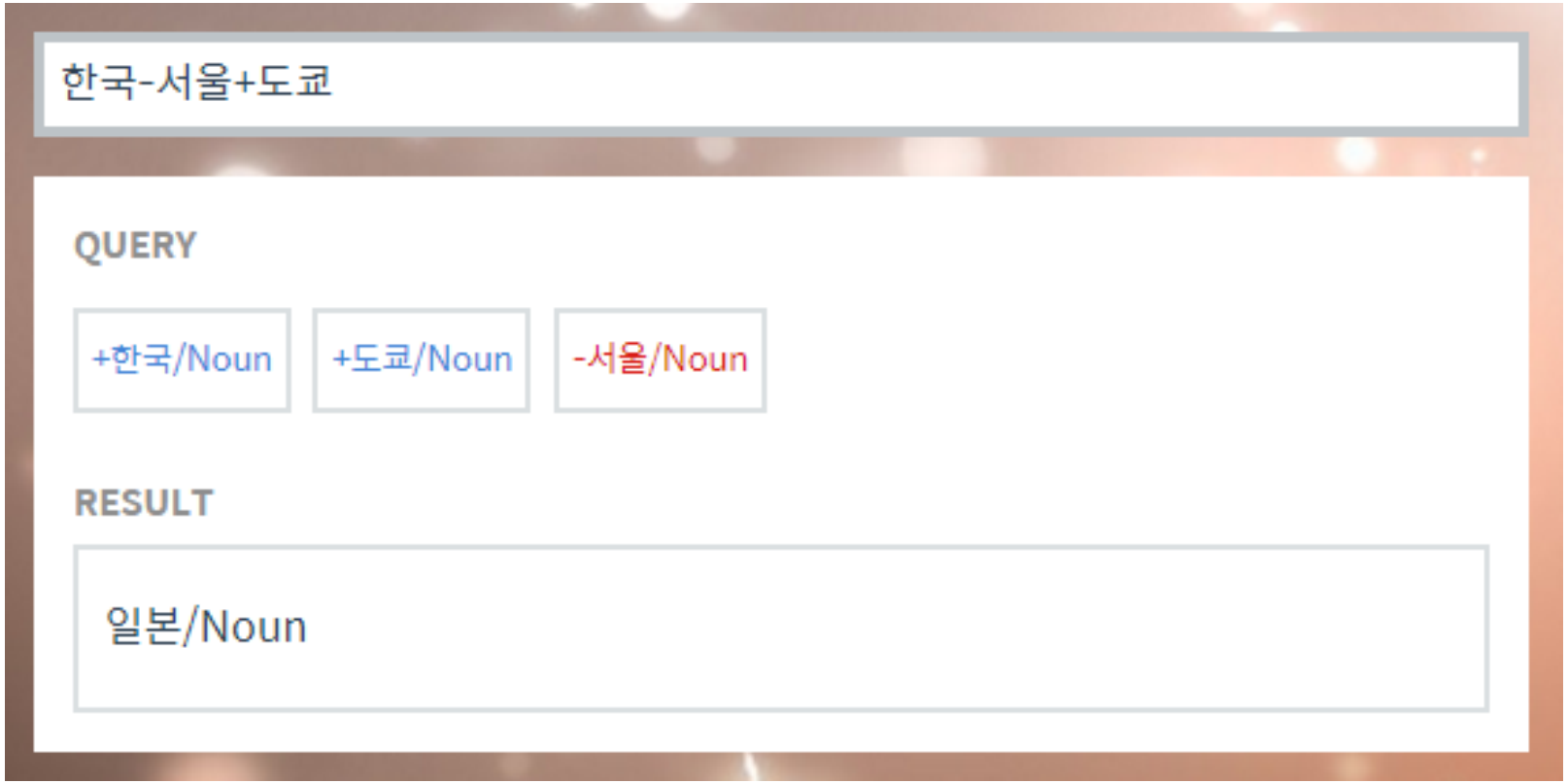
Word2Vec의 특성 – Linear Substructure

comparative
- superlative



Word2Vec의 특성 – Analogy Reasoning

- 예제: 한글 word2vec
  - http://w.elnn.kr/search/



Word2Vec의 특성 – Analogy Reasoning

- 다른 예제들: http://wonjaekim.com/archives/50

	데모	http://w.elnn.kr/
버락_오바마-미국+러시아	블라디미르/Noun_푸틴/Noun	-
버락_오바마-미국+스타워즈	아나킨/Noun_스카이워커/Noun	-
아카라카-연세대학교+고려대학교	입실렌티/Noun	입실렌티/Noun
아이폰-휴대폰+노트북	아이패드/Noun	아이패드/Noun
컴퓨터공학-자연과학+인문학	법학/Noun	게임학/Noun
플레이스테이션-소니+마이크로소프트	엑스박스/Noun_360/Number	MSX/Alpha
한국-서울+파리	프랑스/Noun	프랑스/Noun
컴퓨터-기계+인간	운영체제/Noun	일반인/Noun
게임+공부	프로그래밍/Noun	덕질/Noun
박보영-배우+가수	애프터스쿨/Noun	허각/Noun
밥+했는지	끓였/Verb	저녁밥/Noun
사랑+이별	그리움/Noun	추억/Noun
삼성-한화	노트북/Noun	후지필름/Noun
소녀시대-소녀+아줌마	아이유/Noun	에이핑크/Noun
수학-증명	경영학/Noun	이산수학/Noun
스파게티-소시지+김치	칼국수/Noun	비빔국수/Noun
아버지-남자+여자	어머니/Noun	어머니/Noun
아이유-노래+연기	송중기/Noun	송중기/Noun
안드로이드-자유	iOS/Alpha	아이폰/Noun
우주-빛	태양계/Noun_밖/Noun	NASA/Alpha
인간-직업	짐승/Noun	볼뉴르크/Noun
최현석_셰프-허세+셰프	이연/Noun_복/Noun	-
패스트푸드-체인점	영국/Noun_요리/Noun	철물/Noun

Word2Vec의 특성 – Semantic Similarity

- 예제: https://github.com/dhammack/Word2VecExample

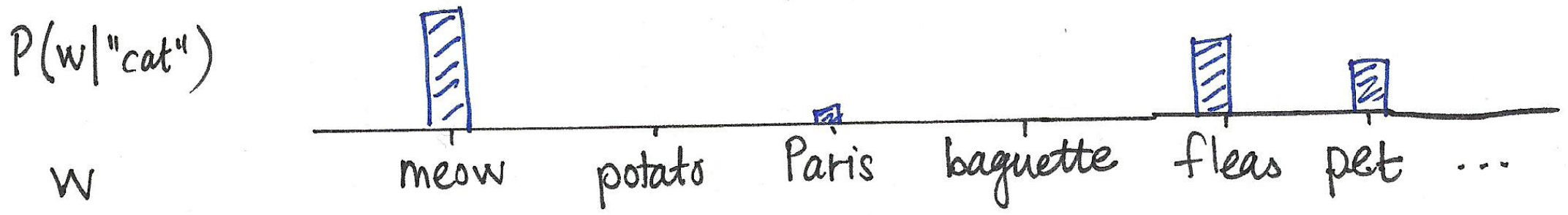
- Word intrusion detection
  - staple hammer saw drill
  - math shopping reading science
  - rain snow sleet sun
  - eight six seven five three owe nine
  - breakfast cereal dinner lunch
  - england spain france italy greece germany portugal australia

Word Embedding 모델의 기본 아이디어

- "You shall know a word by the company it keeps"

  - J. R. Firth 1957

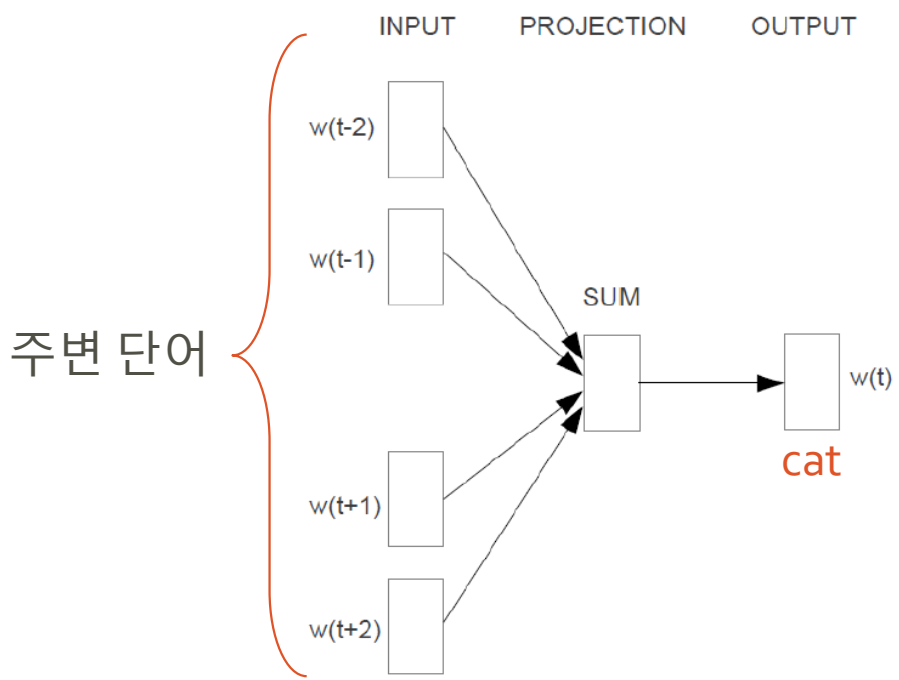
- Suppose we read the word "cat". What is the probability P(w|cat) that we'll read the word w nearby?



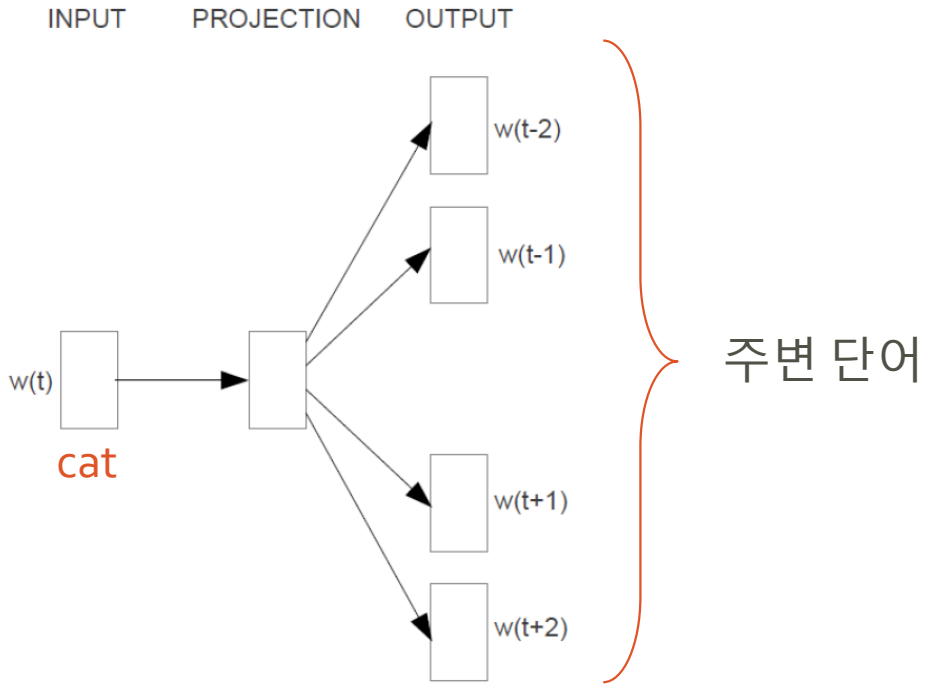
- Distributional Hypothesis: The meaning of "cat" is captured by the probability distribution P(w|cat).

Two Models of Word2Vec

Continuous Bag-Of-Words (CBOW)

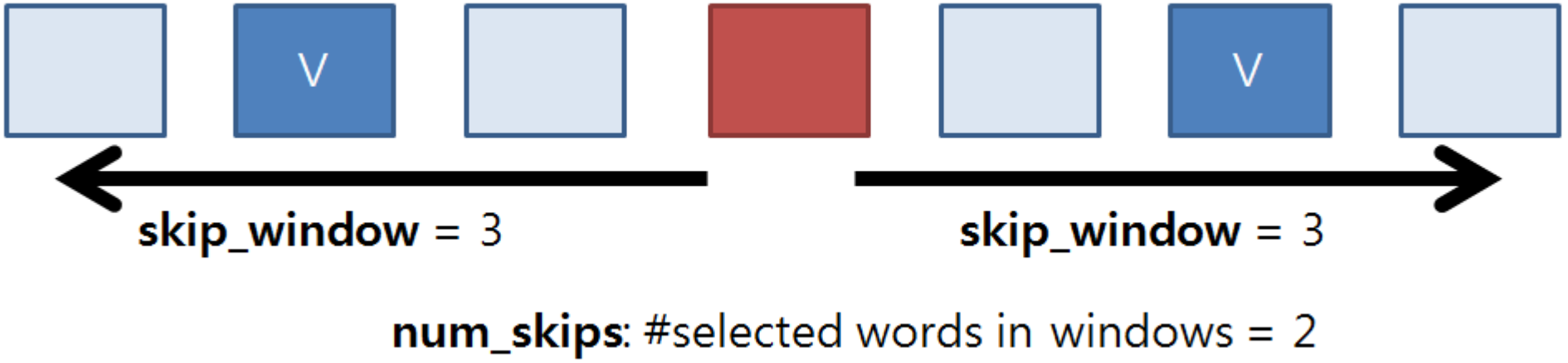


Skip-gram



How Word2Vec Algorithm Works

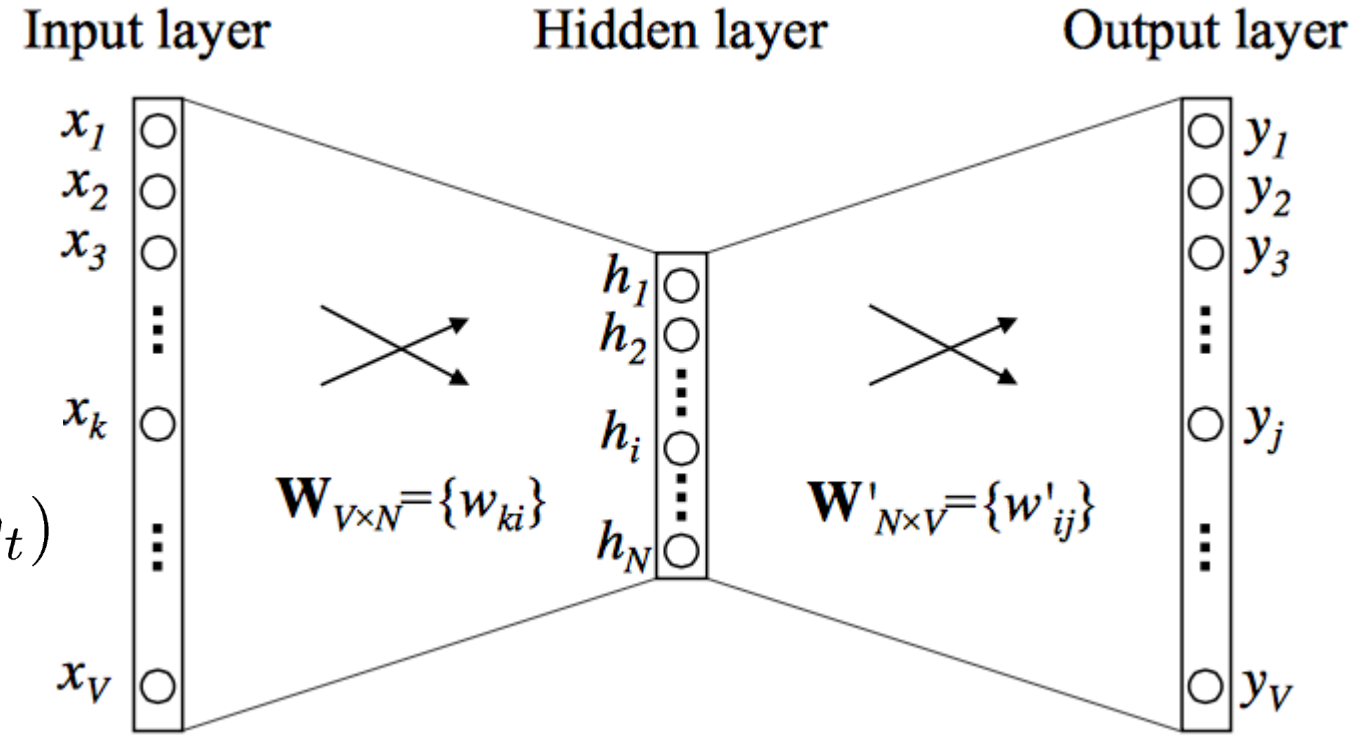
- 먼저 window 크기를 정하고



- Skipgram: 중심 단어로 window 내의 주변 단어 (context words)를 예측
  - Semantic task에 장점을 보이나, 학습이 느림.
- CBOW: 주변 단어들로 중심 단어를 예측
  - Syntactic task에 장점을 보이고, 학습이 상대적으로 빠름

How Word2Vec Algorithm Works

- V : vocabulary size
- N : embedding dimension



- Objective criterion

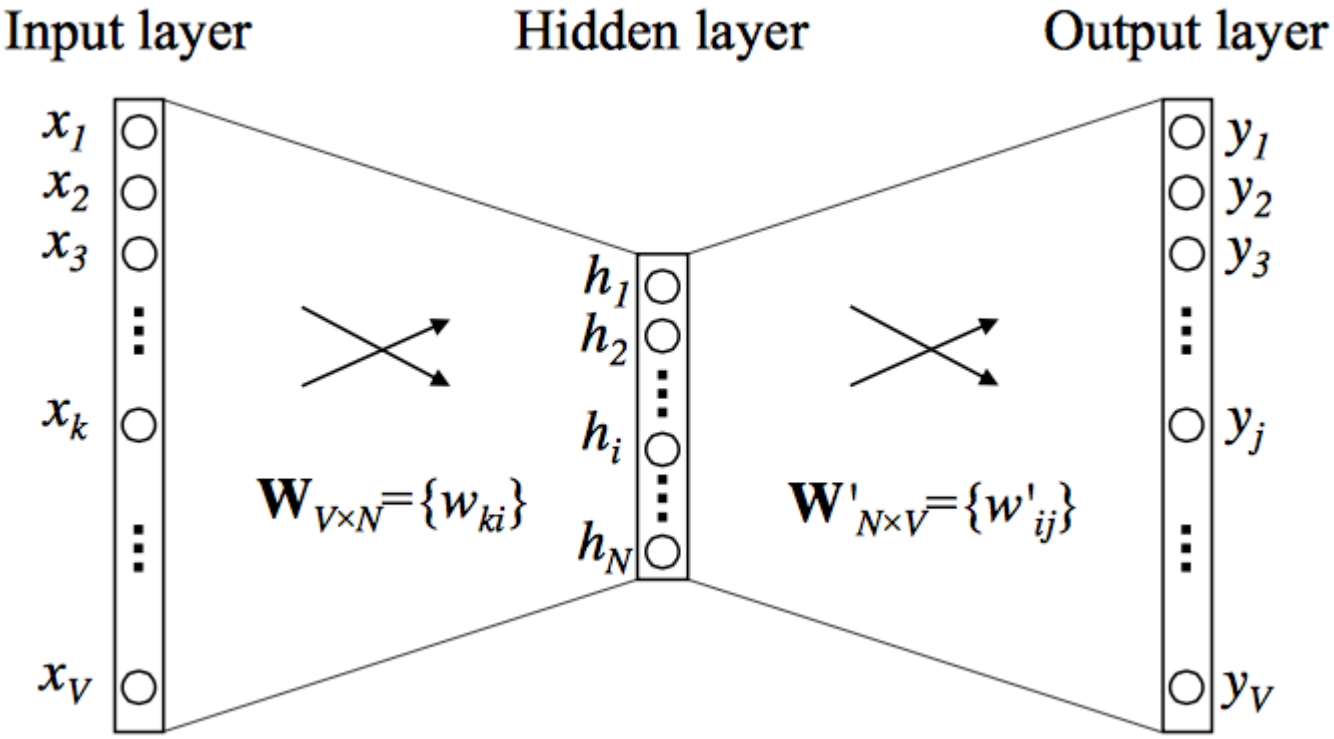
$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} \sum_{-m\leq j\leq m, j\neq 0} \log p(w_{t+j}|w_t)$$

where

$$\log p(o|c) = \log \frac{\exp\left(u_o^T v_c\right)}{\sum_{w=1}^{W} \exp\left(u_w^T v_c\right)}$$, which is a softmax function.

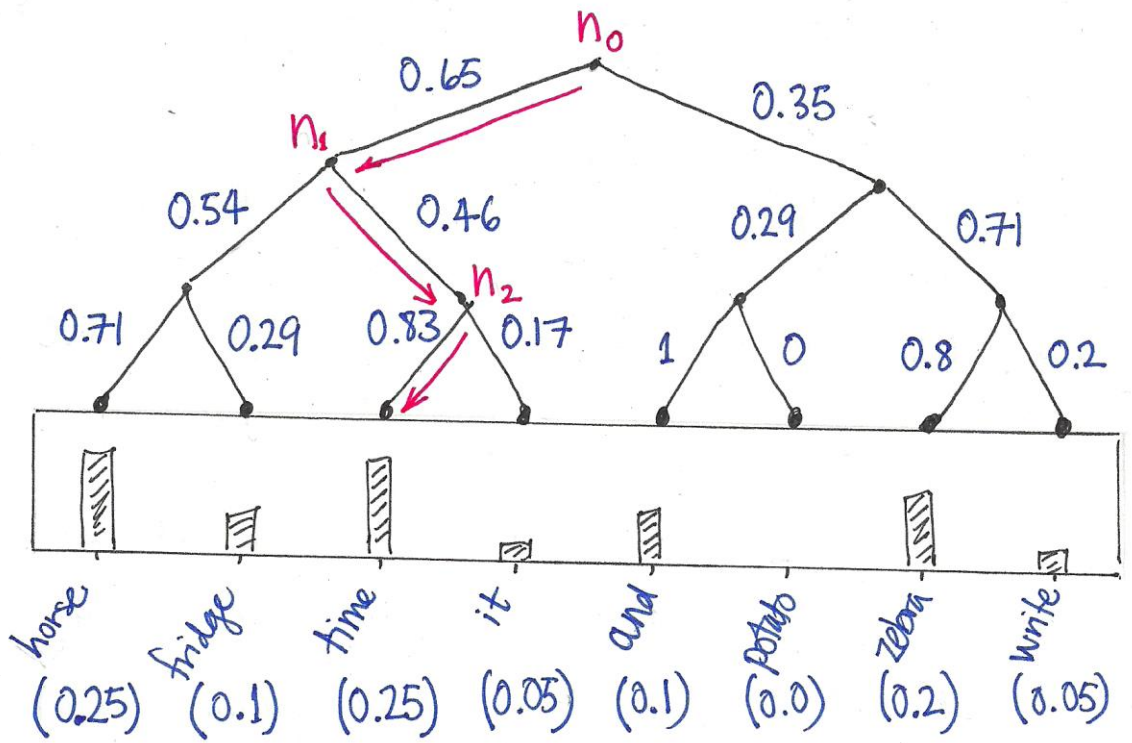
Softmax Implementations are Slow!

- 두번째 layer의 파라메터인 W'를 학습하는 것이 굉장히 느림.
- 즉, vocabulary size가 영어의 경우 총 개수는 백만개가 넘는데, 네트워크의 output layer에서 softmax 계산을 하기 위해서는 각 단어에 대해 전부 계산을 해서 normalization을 해주어야 하고, 이에 따라 추가적인 연산이 엄청나게 늘어나서 대부분의 연산을 이 부분에 쏟아야 함.



Efficient Solution 1: Hierarchical Softmax

- 각 단어들을 leaves로 가지는 binary tree를 생성함.
- 해당하는 단어의 확률을 계산할 때 root에서부터 해당 leaf로 가는 path에 따라서 확률을 곱해나가는 식으로 해당 단어가 나올 최종적인 확률을 계산함.
- 즉, 각각의 internal node를 왼쪽, 오른쪽 노드들로 분기하는 Bernoulli 확률 변수라 생각하여, 각 단어의 확률을 계산함



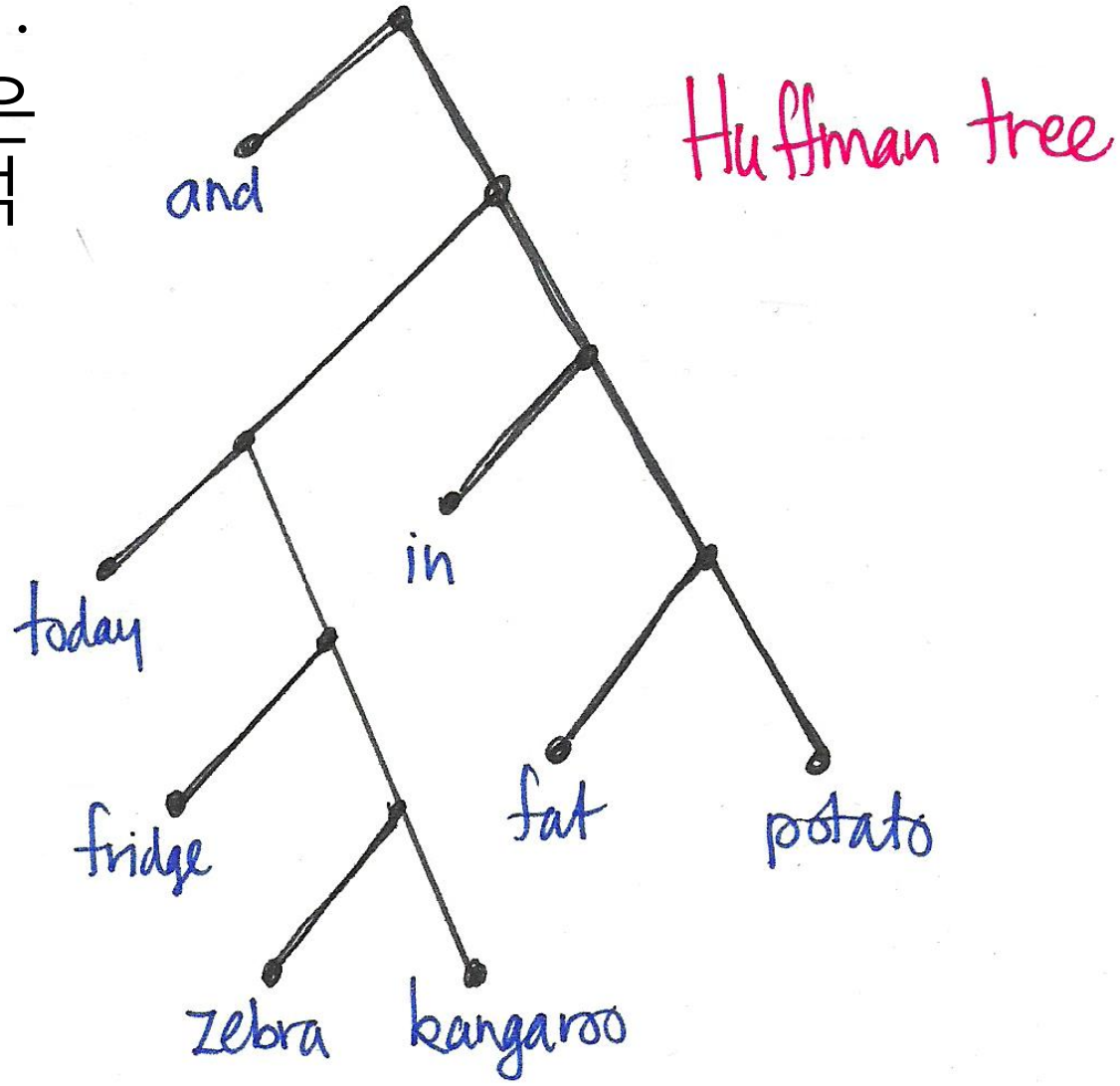
$$P(\texttt{time}|\mathcal{C}) = P_{n_0}(\text{left}|\mathcal{C})P_{n_1}(\text{right}|\mathcal{C})P_{n_2}(\text{left}|\mathcal{C})$$

- 이를 통해 V번의 계산을 log V번의 계산으로 줄일 수 있음.

Efficient Solution 1: Hierarchical Softmax

- Binary tree는 Huffman tree를 사용함.
- 즉, 자주 등장하는 단어들은 보다 짧은 path로 도달할 수 있기 때문에 전체적인 계산량이 더 낮아지는 효과를 냄.

word	count
fat	3
fridge	2
zebra	1
potato	3
and	14
in	7
today	4
kangaroo	2



Efficient Solution 2: Negative Sampling

- Hierarchial softmax의 대체 방법.
- 일부 단어만 뽑아서 softmax 계산을 하고 normalization을 해줌. 즉, 계산량은 N x V 에서 N x K (K : 뽑는 샘플의 갯수)로 줄어들게 됨.
- 이 때, 실제 target으로 사용하는 단어의 경우 반드시 계산을 해야 하므로 이를 'positive sample' 이라고 부르고, 나머지 'negative sample' 들을 어떻게 뽑느냐가 관건임.

$$\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^\top v_{w_I}) \right]$$

- Negative Sampling에서 샘플들을 뽑는 것은, 'Noise Distribution' 을 정의하고, 그 분포 $P_n(w)$를 이용하여 단어들을 일정 갯수 뽑아서 사용하는데, 실험적으로 'Unigram Distribution의 3/4 승'으로 설정.

Word2Vec의 적용 분야

자연어 처리 거의 모든 분야에서 성능 향상을 가져왔음.

- Word similarity
- Machine translation
- Part-of-speech tagging and named entity recognition
- Sentiment analysis
- Clustering
- Semantic lexicon building

Word2Vec의 적용 분야 – Machine Translation

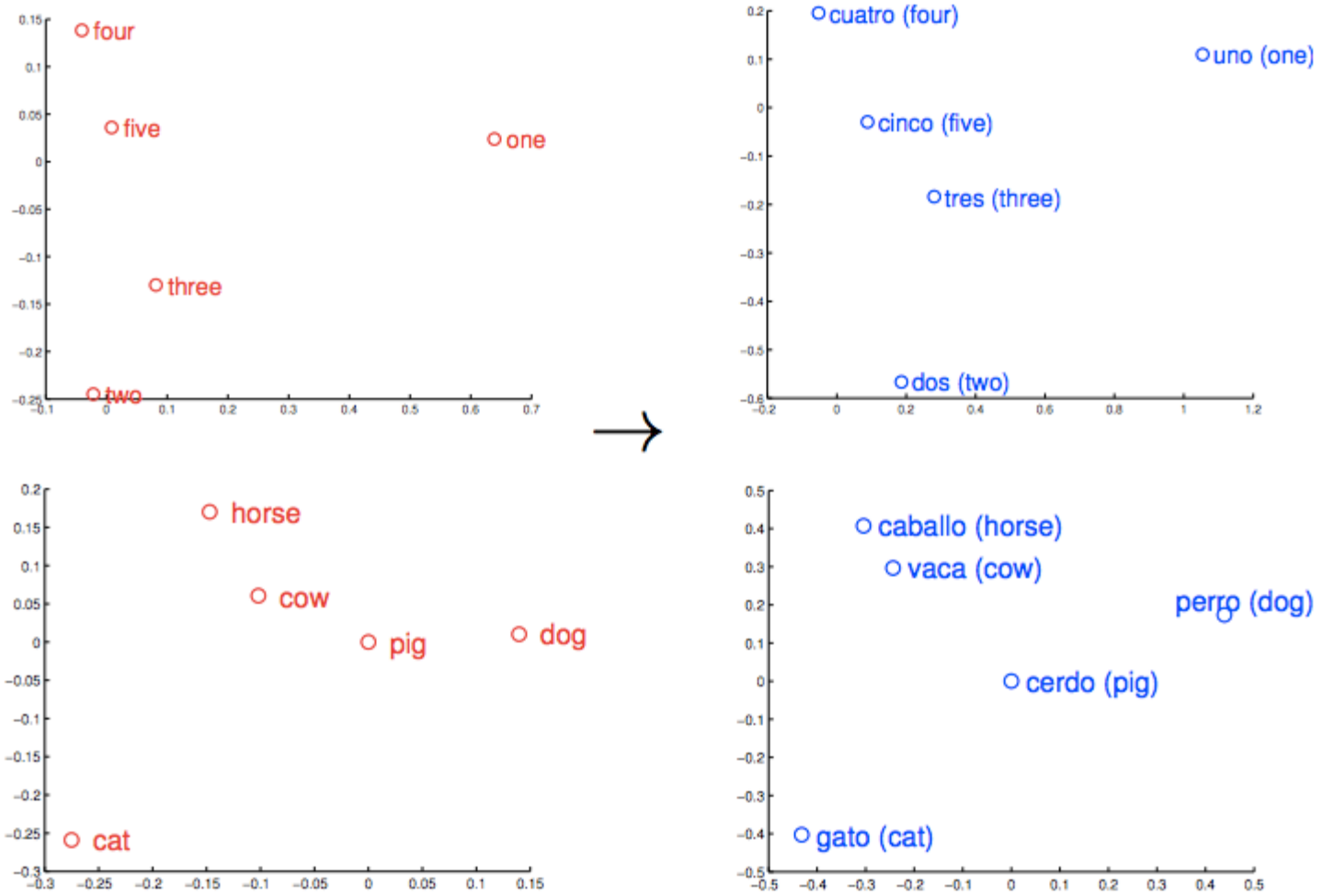
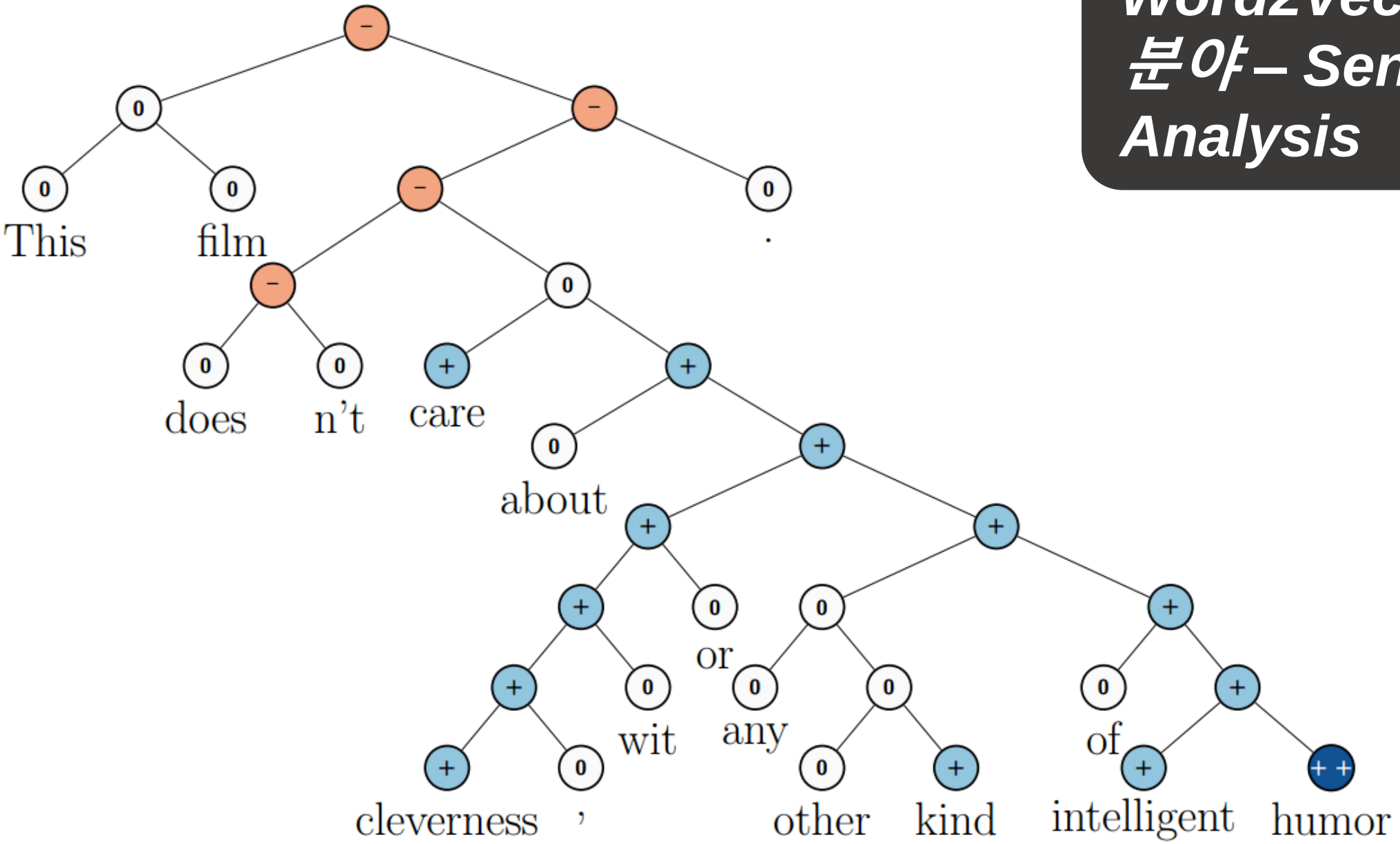


Figure 1: Distributed word vector representations of numbers and animals in English (left) and Spanish (right). The five vectors in each language were projected down to two dimensions using PCA, and then manually rotated to accentuate their similarity. It can be seen that these concepts have similar geometric arrangements in both spaces, suggesting that it is possible to learn an accurate linear mapping from one space to another. This is the key idea behind our method of translation.

Word2Vec의 적용 분야 – Sentiment Analysis



Word2Vec의 적용 분야 – Image Captioning

Figure 5. A selection of evaluation results, grouped by human rating.

02

2. Other Models

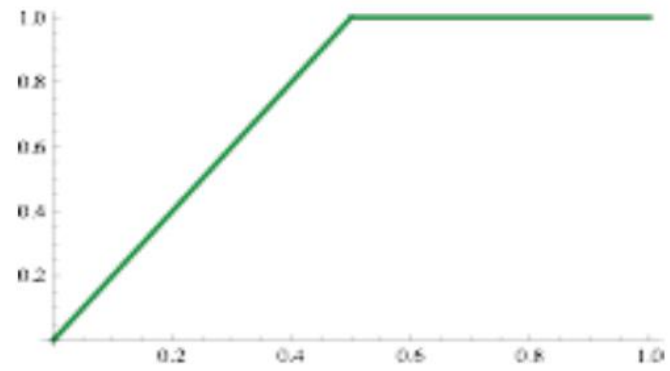
2-1. GloVe
2-2. Word2Vec for Polysemy
2-3. Doc2Vec

GloVe: Another Word Embedding Model

GloVe: Global Vectors for Word Representation

- Matrix decomposition on co-occurrence matrix

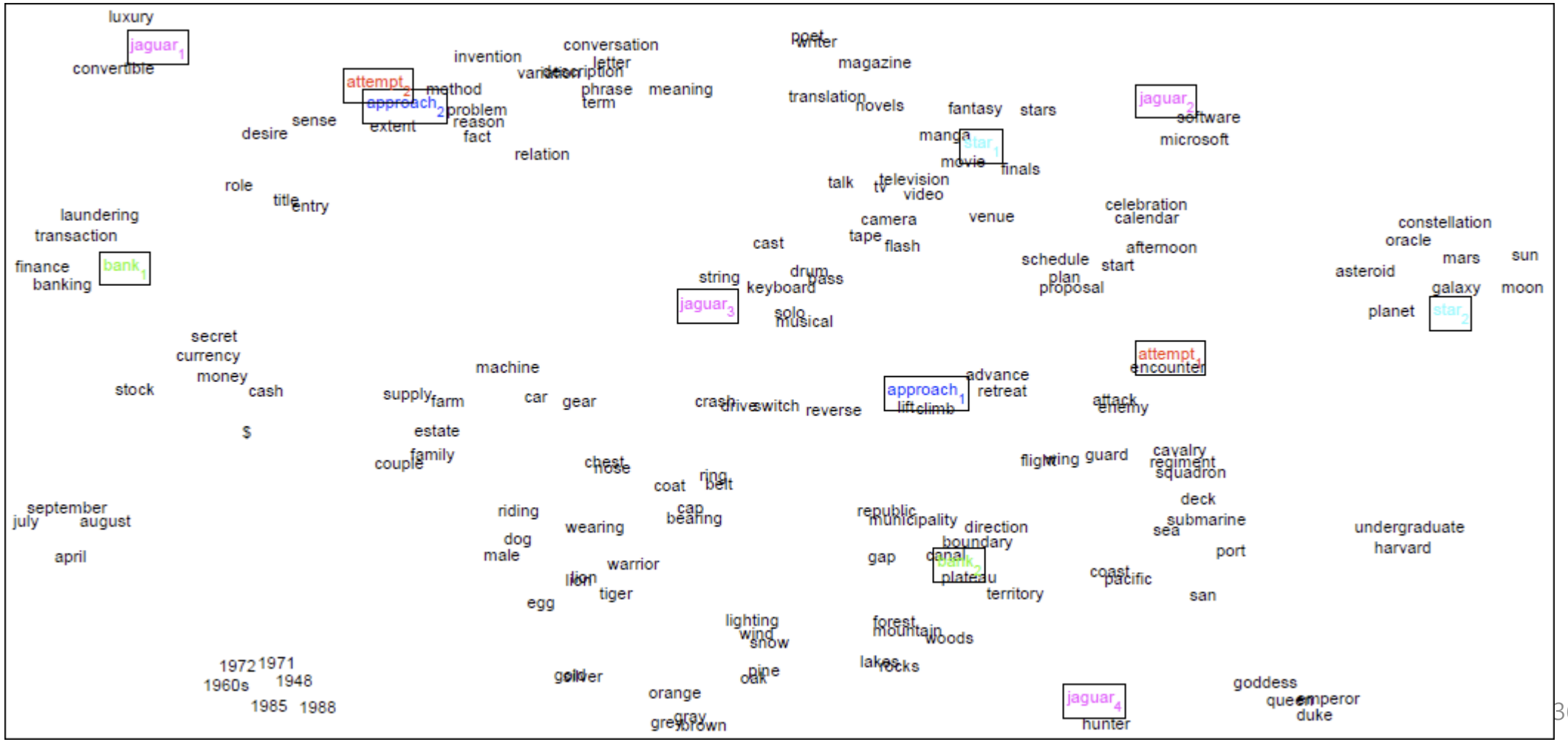
$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{W} f(P_{ij})(u_i^T v_j - \log P_{ij})^2$$

$f \sim$ 

- 학습이 빠름.
- 작은 corpus에서도 잘 동작함.

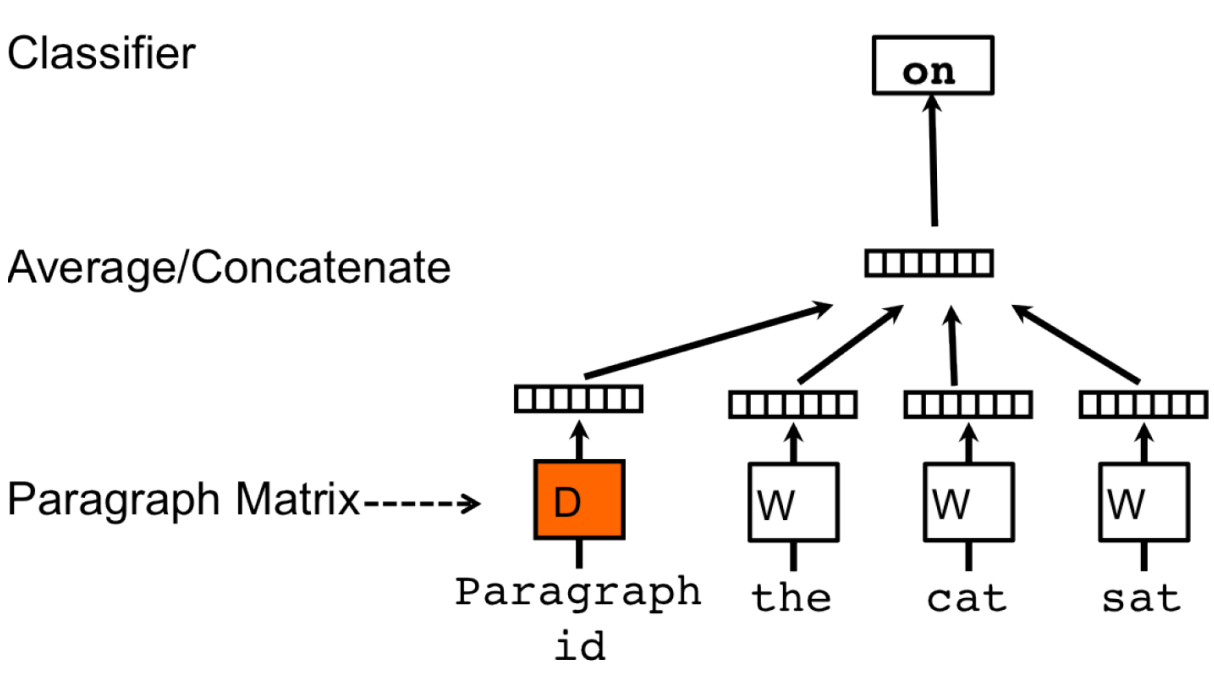
Word2Vec for Polysemy (동음이의어)

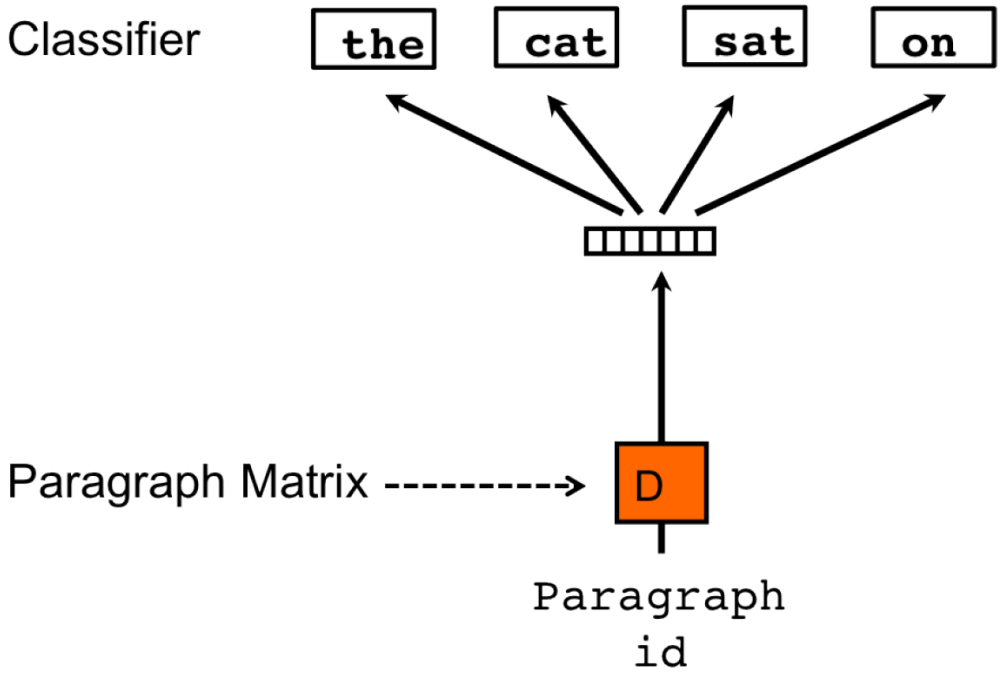
Cluster word windows around words, retrain with each word assigned to different clusters bank1, bank2, etc. (Huang et al., 2012)



Doc2Vec (Paragraph2Vec)

- Idea: 문서 (또는 문단) 벡터를 마치 단어인 양 학습시키자.
- 특성 및 장점
  - 같은 문단 및 문서에 나타난 단어는 유사도가 높게 학습된다.
  - 단어 벡터와 같은 공간에 문서를 전사할 수 있다.





References

코드

- Python: https://radimrehurek.com/gensim/models/word2vec.html
- C++: https://code.google.com/archive/p/word2vec/
- FastText: https://github.com/facebookresearch/fastText

Useful resources

- https://shuuki4.wordpress.com/2016/01/27/word2vec-%EA%B4%80%EB%A0%A8-%EC%9D%B4%EB%A1%A0-%EC%A0%95%EB%A6%AC/
- https://code.facebook.com/posts/1438652669495149/fair-open-sources-fasttext/
- http://cs224d.stanford.edu/
- https://ronxin.github.io/wevi/
- https://www.lucypark.kr/slides/2015-pyconkr/