TEXT MINING for PRACTICE

Python을 활용한 비정형 데이터 분석 - WEEK 10

전병진 FINGEREDMAN (fingeredman@gmail.com)

Part 9.

단어 임베딩: Word2Vec 임베딩 기반 연관어분석

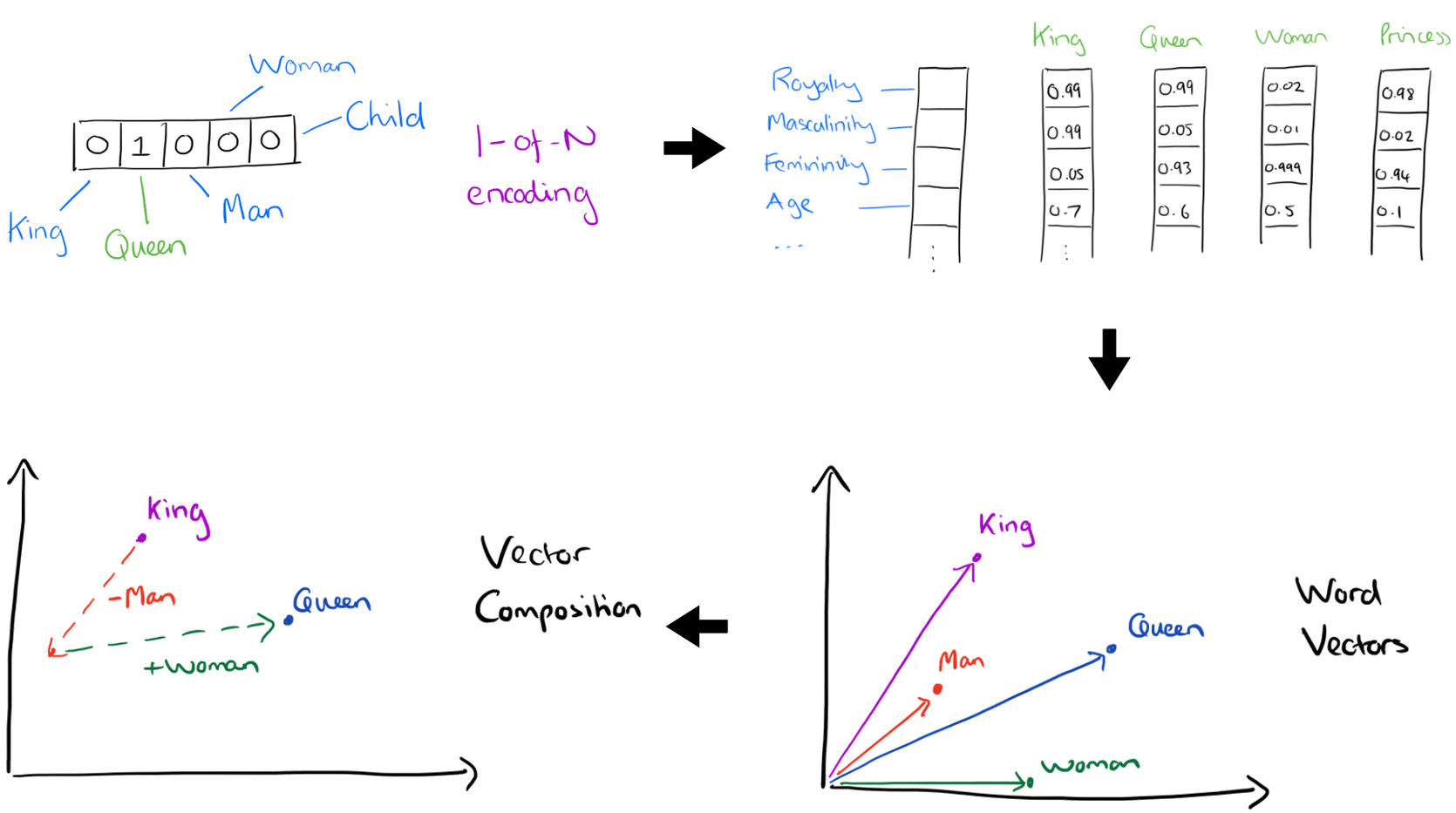
단어 임베딩 (Word Embedding)

단어의 분포를 가상의 벡터공간에 배치하는 방법

- 글 내부에서 가까이 위치해 있는 단어끼리는 유사한 의미를 지닌다는 가정 (distributional hypothesis)을 기반으로, 벡터 공간에서 각 단어들이 어떻게 분포해 있는지를 학습하는 방법

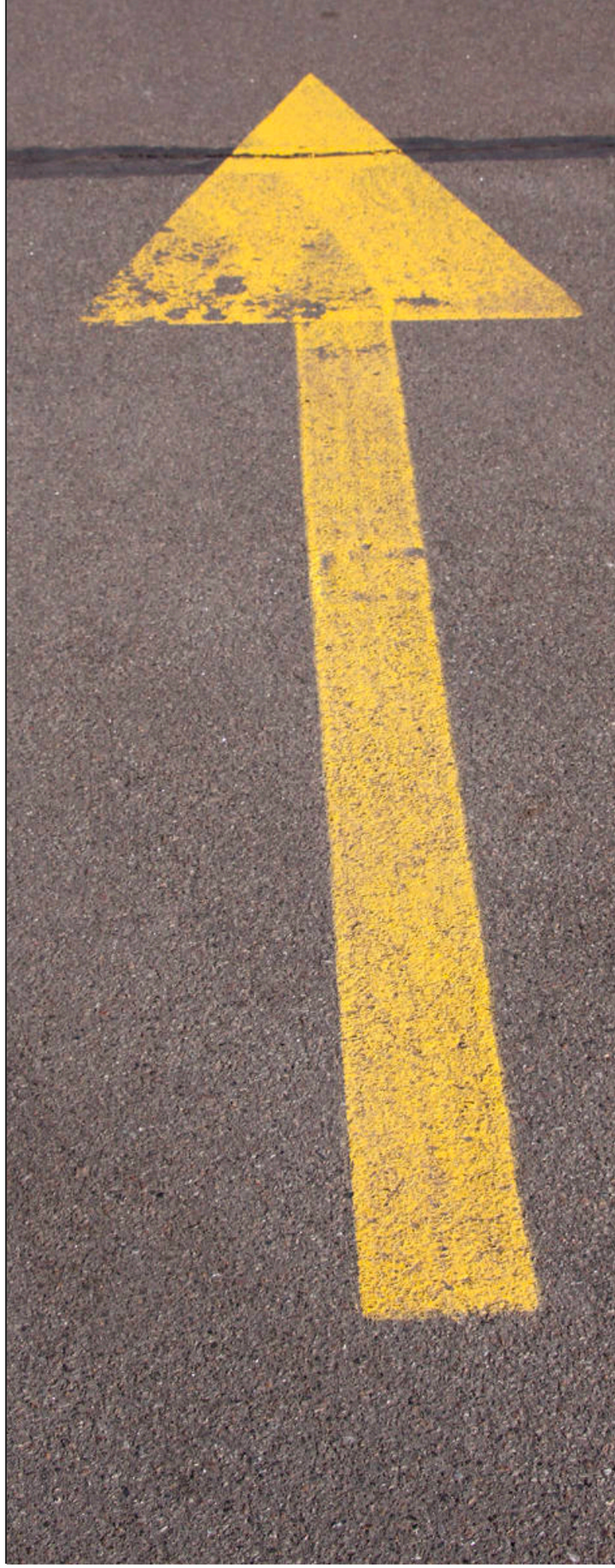
[One-hot encoding]

[Distributed Similarity-based Representation]



*Source : The Morning Paper, The amazing power of word vectors, 2016.4.21., https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/.

단어 임베딩 (Word Embedding)



One-hot Encoding

- 문서에 포함된 N개의 단어를 N차원 벡터로 표현하는 방법
- 영어의 토큰(단어) 수는 1,300만 이상이며, 모든 단어의 표현을 위해 1,300만 차원이 필요함

[Sentence]

A = 안녕하세요. 저는 워너원 멤버 강다니엘 입니다.
B = 안녕하세요. 얘는 트와이스 멤버 정연 입니다.
C = 안녕하세요. 저분은 워너원 멤버 황민현 입니다.

→

[Bag of Word]

A_{BoW} = [안녕, 저, 워너원, 멤버, 강다니엘]
B_{BoW} = [안녕, 얘, 트와이스, 멤버, 정연]
C_{BoW} = [안녕, 저, 워너원, 멤버, 황민현]

[One-hot Encoding]

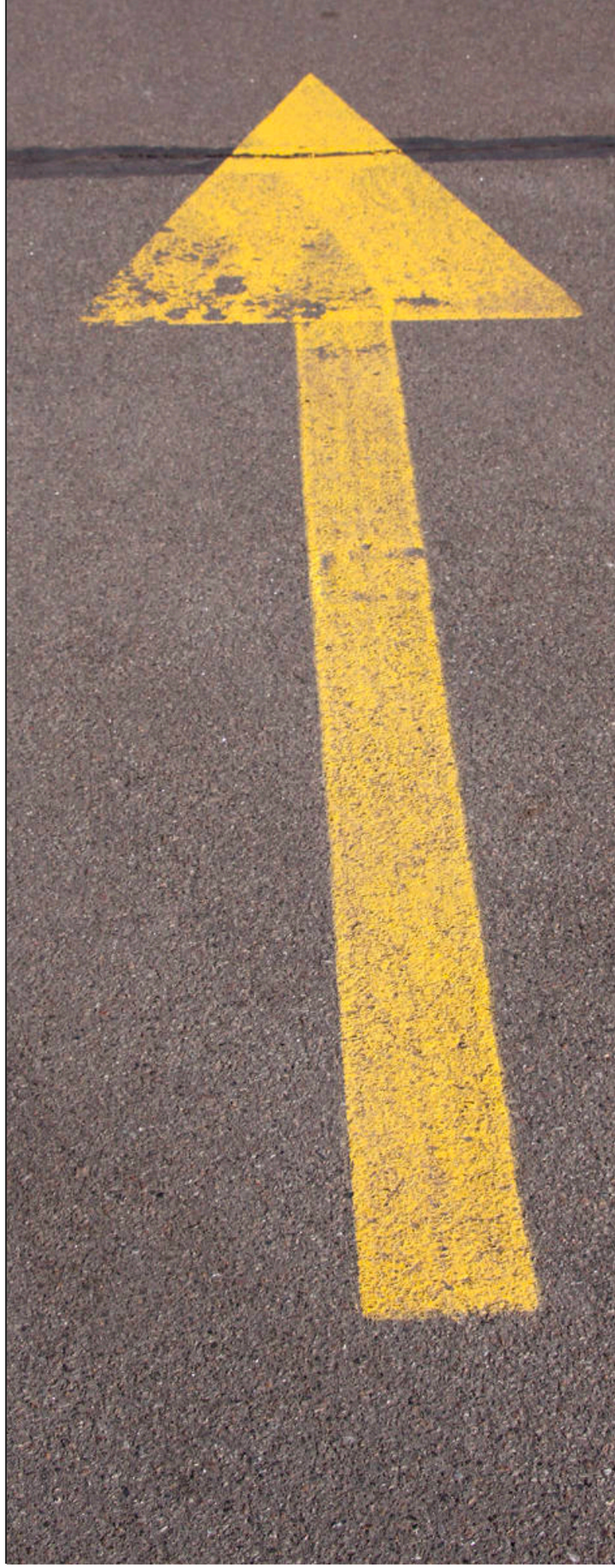
Vector Space(VS) = [안녕, 저, 얘, 워너원, 트와이스, 멤버, 정연, 강다니엘, 옹성우, 황민현]
안녕$_{VS}$ = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
저$_{VS}$ = [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
얘$_{VS}$ = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
워너원$_{VS}$ = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
트와이스$_{VS}$ = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
멤버$_{VS}$ = [0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
정연$_{VS}$ = [0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
강다니엘$_{VS}$ = [0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
옹성우$_{VS}$ = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
황민현$_{VS}$ = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

→ 단어를 각각 10차원 벡터로 표현

문제점

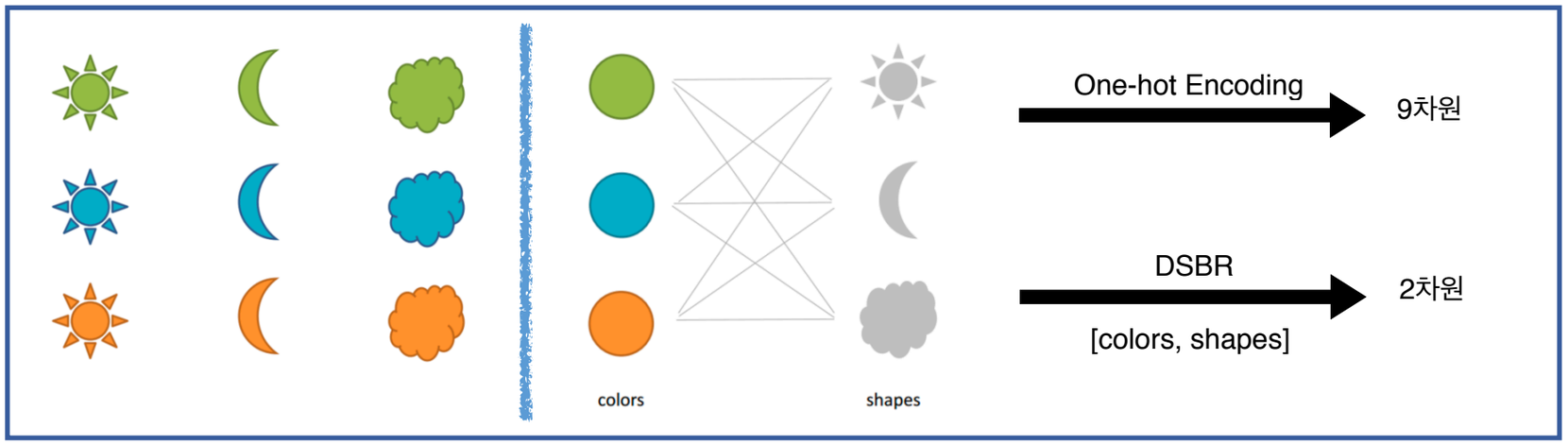
1.차원의 저주(curse of dimensionality) : 하나의 단어를 표현하기 위해 문서에 존재하는 단어의 수만큼 차원을 만들어야함
2.각 단어는 서로 관계를 가질 수 없기 때문에(VS_T = 0), 유의어, 반의어와 같은 관계를 표현할 수 없고 서로 독립으로만 존재함

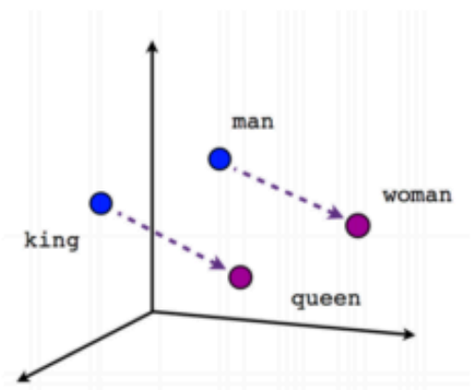
단어 임베딩 (Word Embedding)

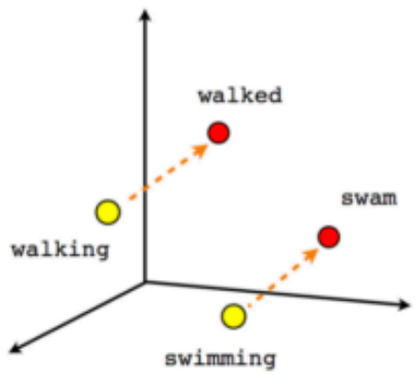


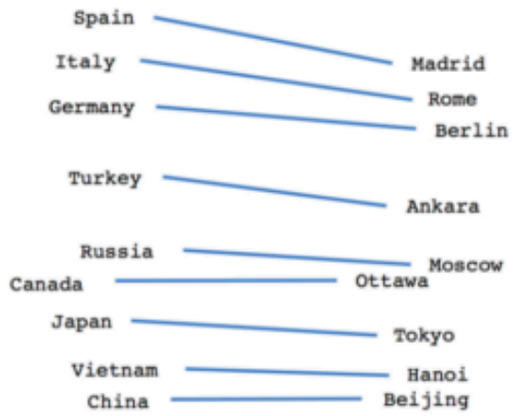
분산표상 (Distributed Similarity Based Representation)

- One-hot Encoding의 단어 사이에 관계를 설명할 수 없는 단점을 해결하기 위해, N차원 단어 벡터를 그보다 훨씬 적은 n차원 벡터로 표현하는 방법



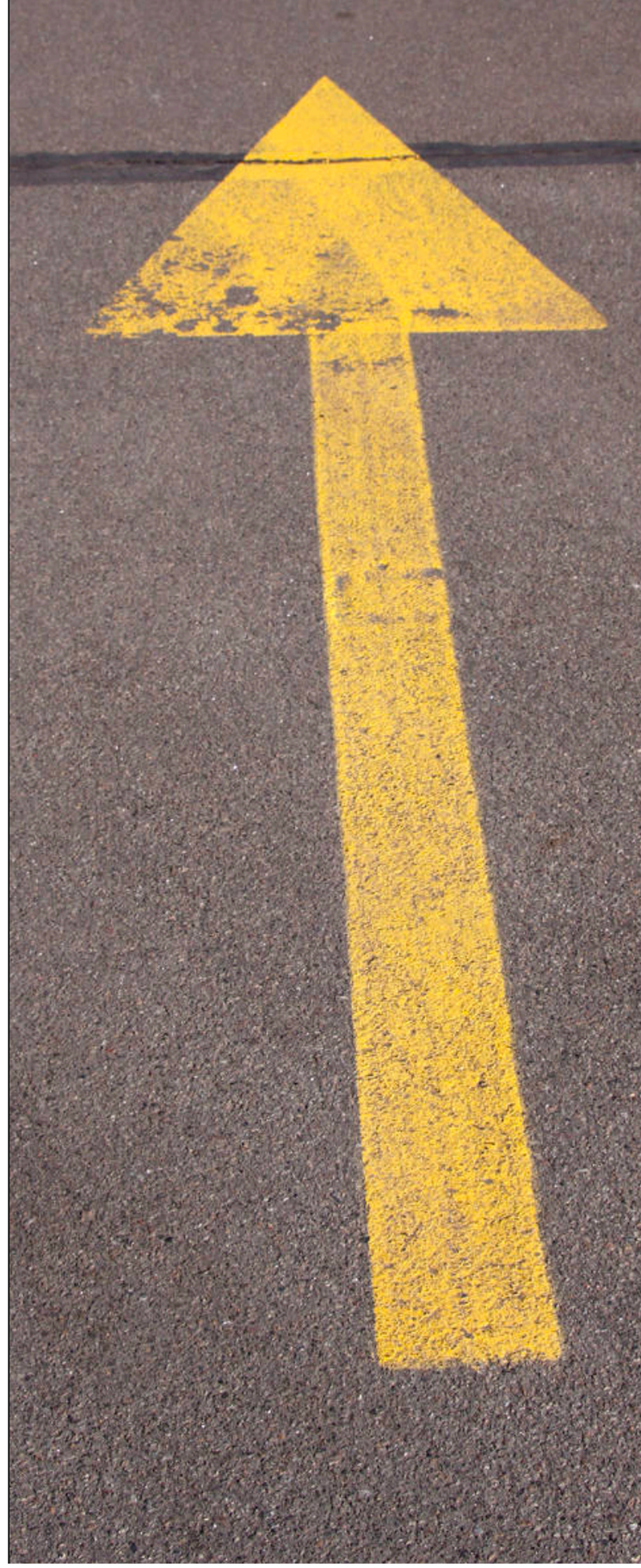






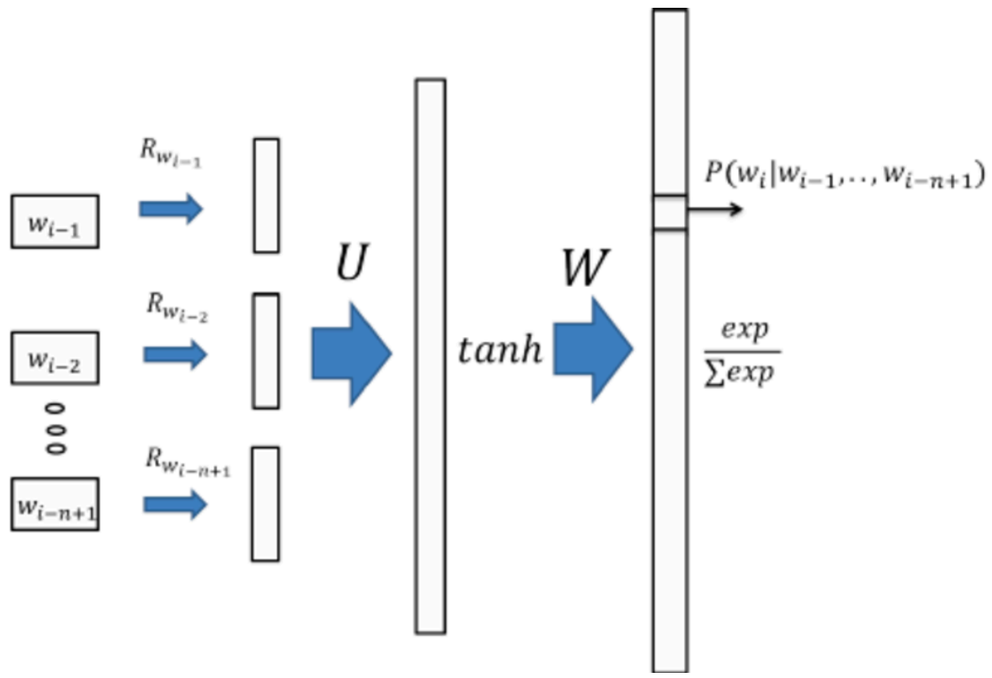
*Source : Tensor Flow, Vector Representations of Words, https://www.tensorflow.org/tutorials/representation/word2vec/.

단어 임베딩 (Word Embedding)



단어 임베딩 기법: NNLM (Neural Network Language Model)

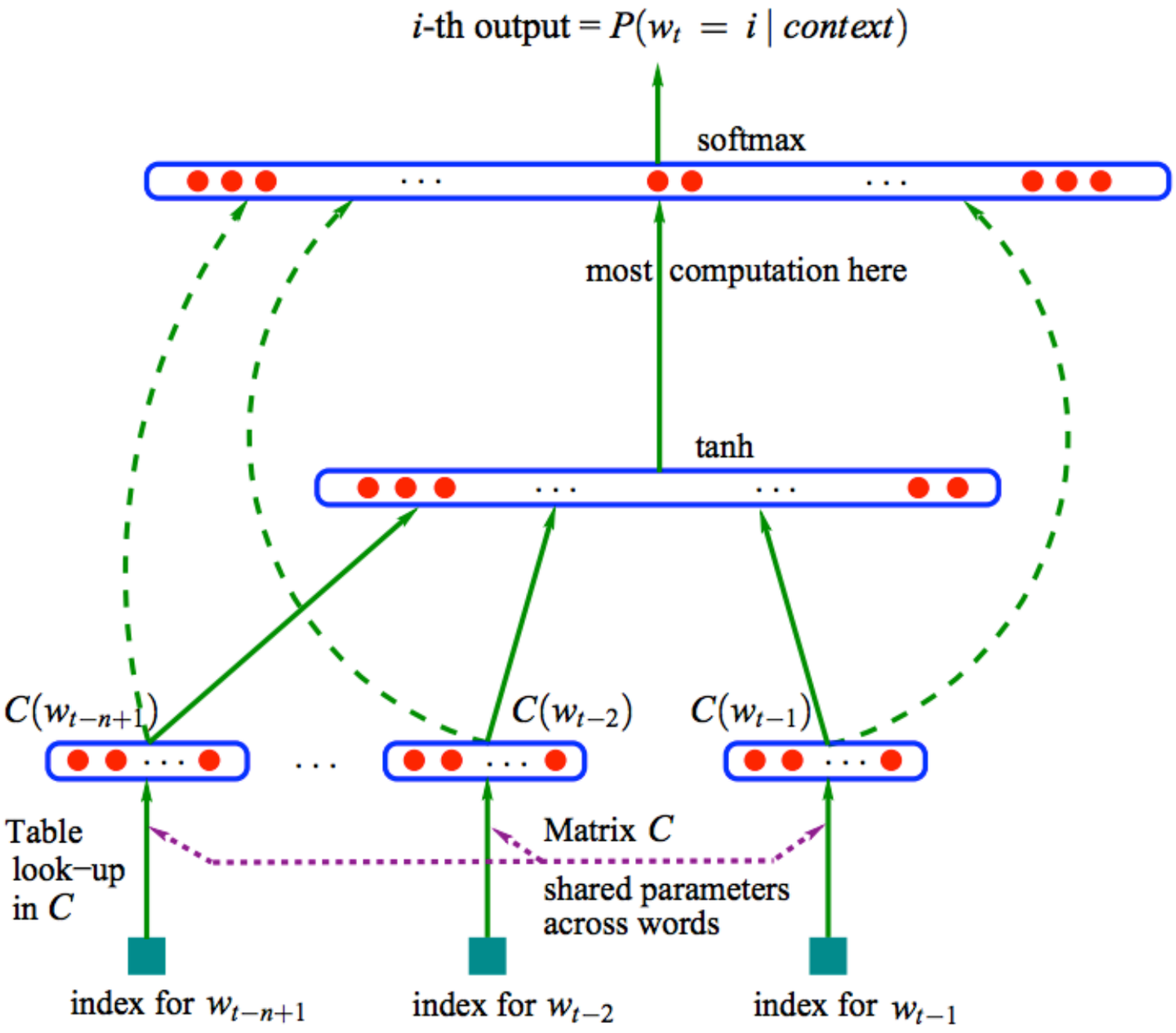
- 가장 초기에 제안된 단어 임베딩 기법으로, 단어의 벡터화 모델이라는 개념이 출현함
- 연속된 단어들이 주어질때 그 다음 단어가 무엇인지 맞추는 과정에서 분상표상으로 표현된 단어의 벡터를 만드는 임베딩 기법
- 대부분의 신경망 (neural network) 기반의 단어 학습모델은 NNLM의 아이디어를 발전해 탄생함
- 단점
  - 몇 개의 단어를 취급할것인지를 미리 정해줘야함 (N 선정)
  - 이전에 등장한 단어로부터만 영향을 받으며, 현재의 단어 앞에있는 단어들을 고려하지 못함
  - 임베딩 과정이 복잡하고 오래걸려 대규모 문서에 적용하는 데 매우 불리함



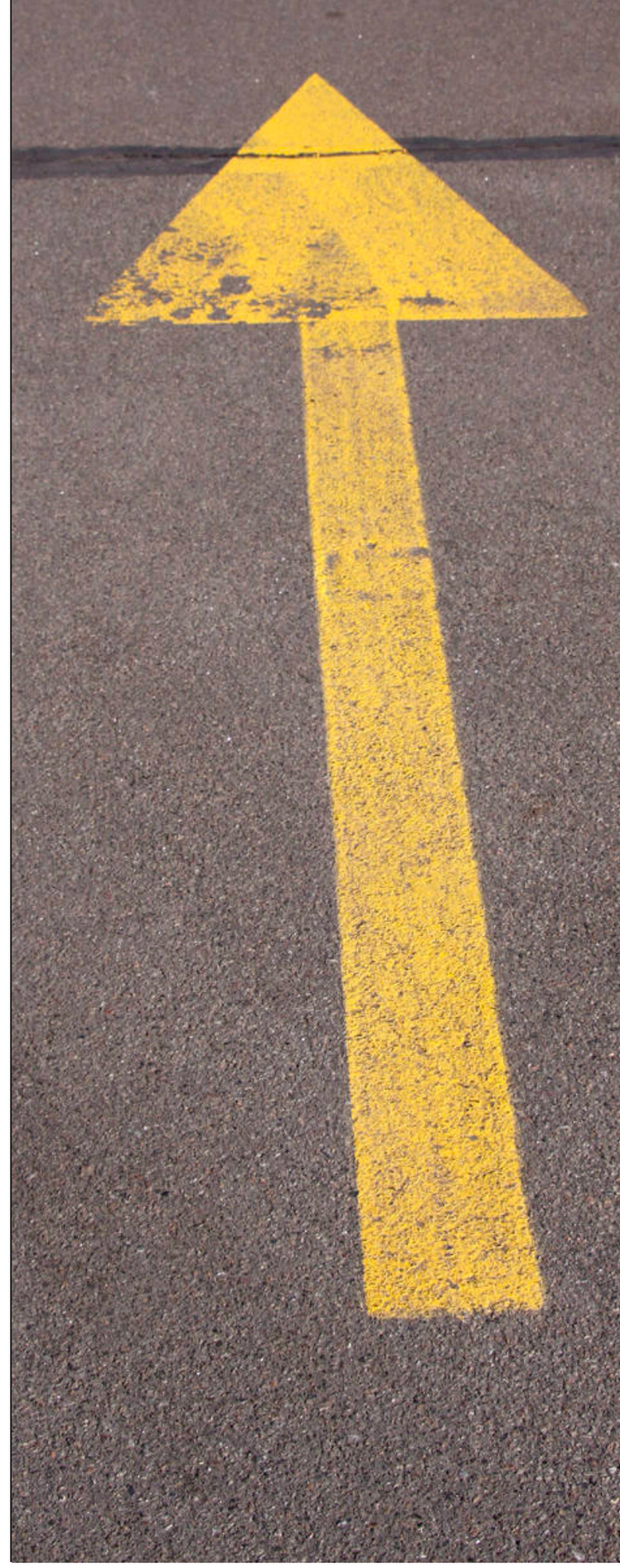
*Source : The Morning Paper, The amazing power of word vectors, 2016.4.21., https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/.
**Source : Puyang Xu et al., Efficient Subsampling for Training Complex Language Models., 2011.01., https://nlp.stanford.edu/projects/glove/.

단어 임베딩 (Word Embedding)

단어 임베딩 기법: NNLM (Neural Network Language Model)



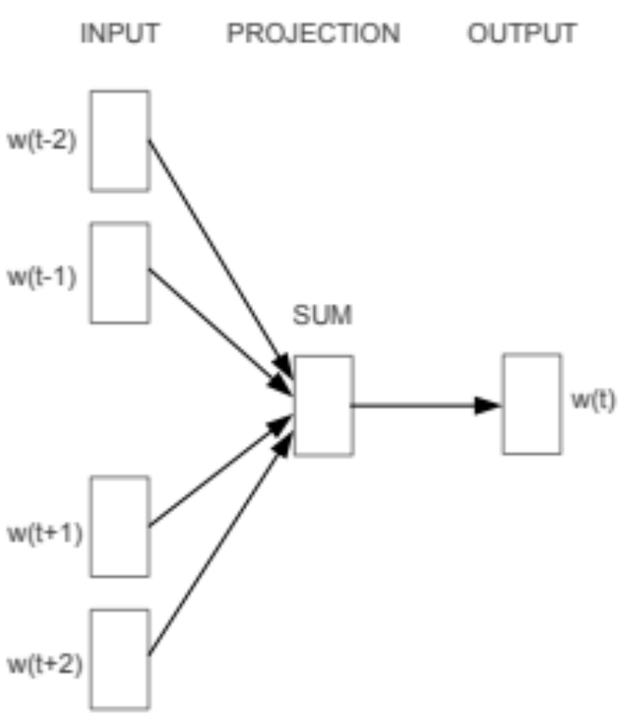
*Source : Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. Journal of machine learning research, 3(Feb), 1137-1155.



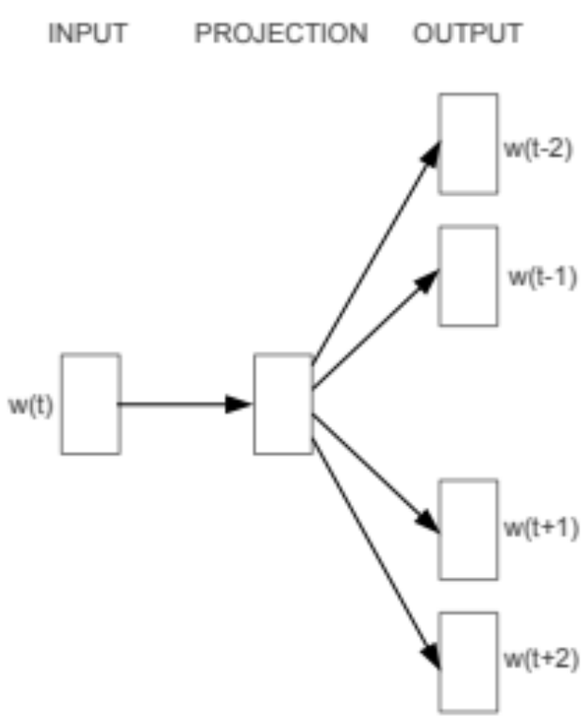
단어 임베딩 (Word Embedding)

단어 임베딩 기법: Word2Vec

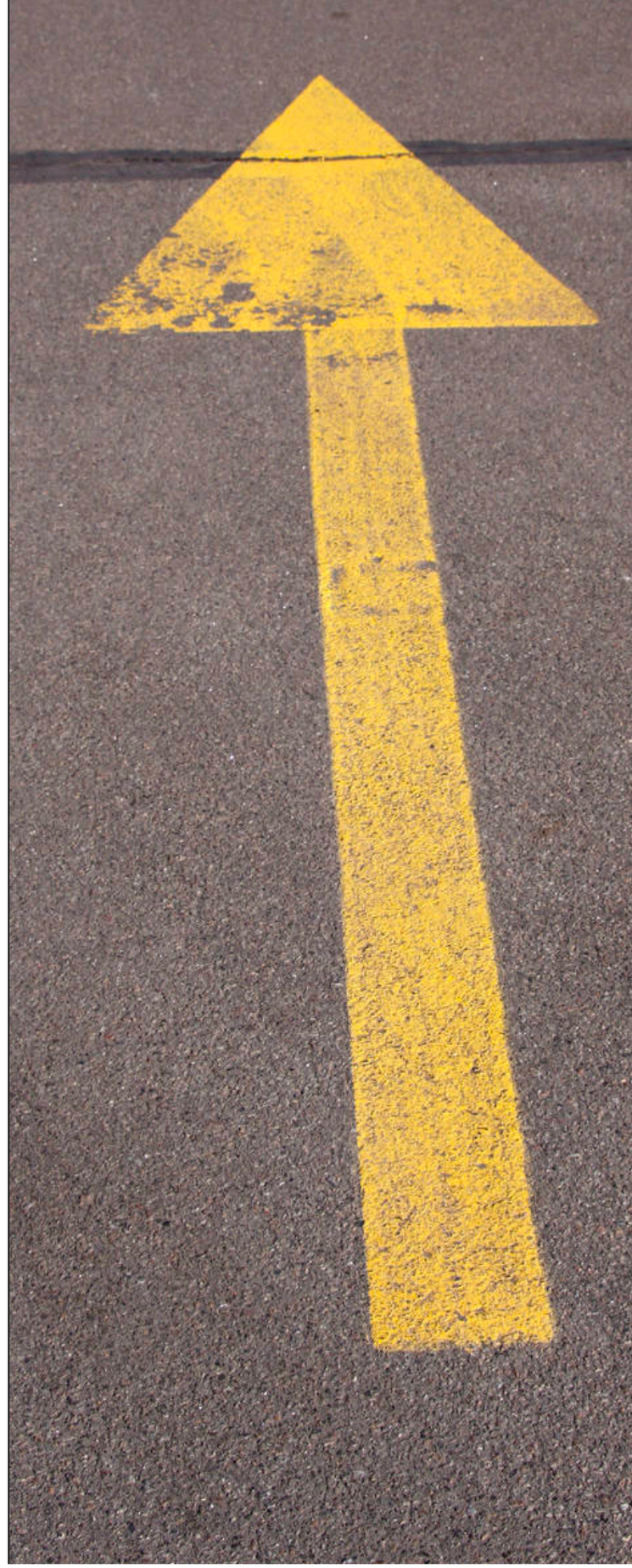
- 2013년 구글 (Google)에서 발표된 연구주제로, Continuous Word Embedding 학습 모형
- 기존 신경망 기반의 방법론에 비해 연산량을 대폭 감소시켜 몇 배 이상 빠른 학습이 가능하며, 현재 가장 많이 활용되는 단어 임베딩 모델
- Word2Vec 학습을 위한 두 가지 네트워크 모델 제시
  - CBOW (Continuous Bag-of-Words)
  - Skip-gram



CBOW



Skip-gram

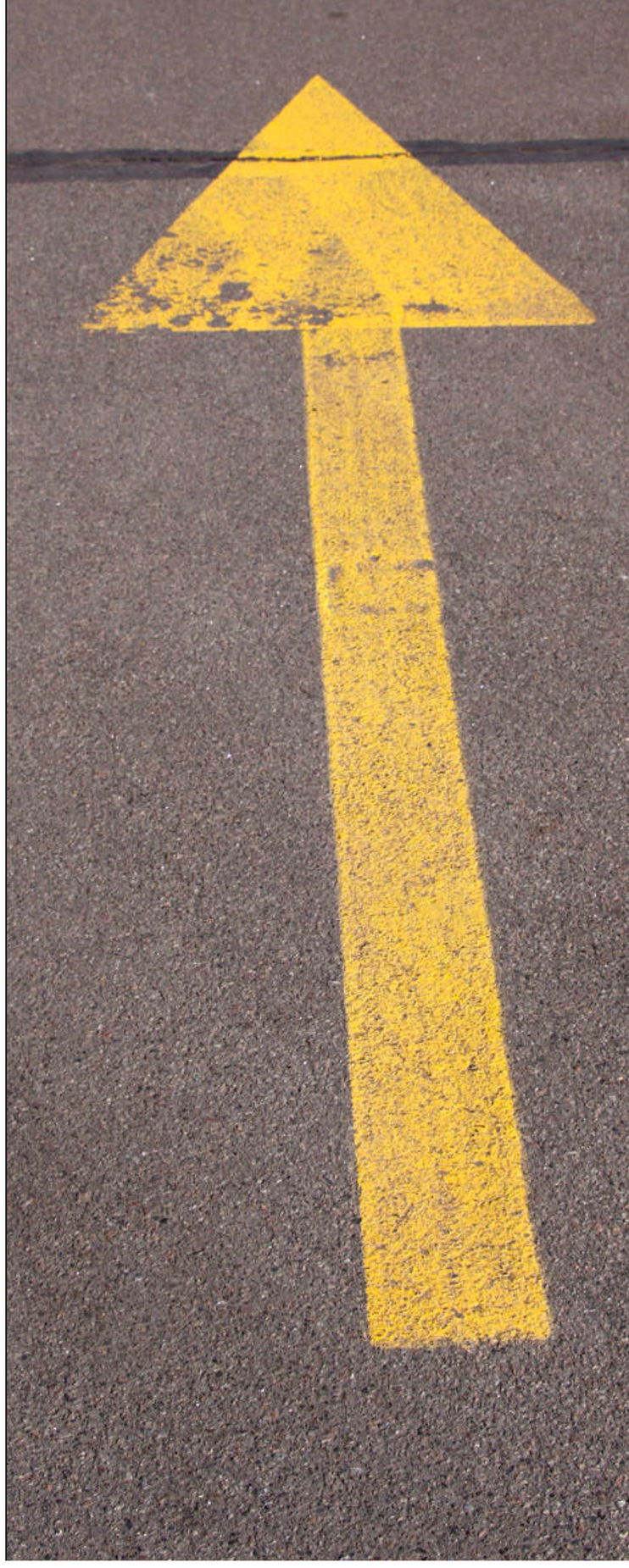


*Source : Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

단어 임베딩 (Word Embedding)

단어 임베딩 기법: Word2Vec의 한계점

- 단어의 형태학적 특성을 반영하지 못함
  - 의미적으로 유사한 단어들이라도 개별적으로 Embedding 하기 때문에 이 단어들의 벡터가 유사하게 구성되지 않음
  - ex) teach, teacher, teachers
- 희소한 단어를 Embedding하기 어려움
  - Distribution hypothesis를 기반으로 학습하기 때문에 출현횟수가 적은 단어에 대해서 제대로 Embedding이 되지 않음
- Out-of-Vocabulary (OOV)를 처리할 수 없는 단점
  - Word2Vec은 단어 단위로 어휘집(Vocabulary)를 구성하기 때문에, 어휘집에 없는 새로운 단어가 등장하면 데이터 전체를 다시 학습시켜야 함



*Source : Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

단어 임베딩 (Word Embedding)

단어 임베딩 기법: Word2Vec

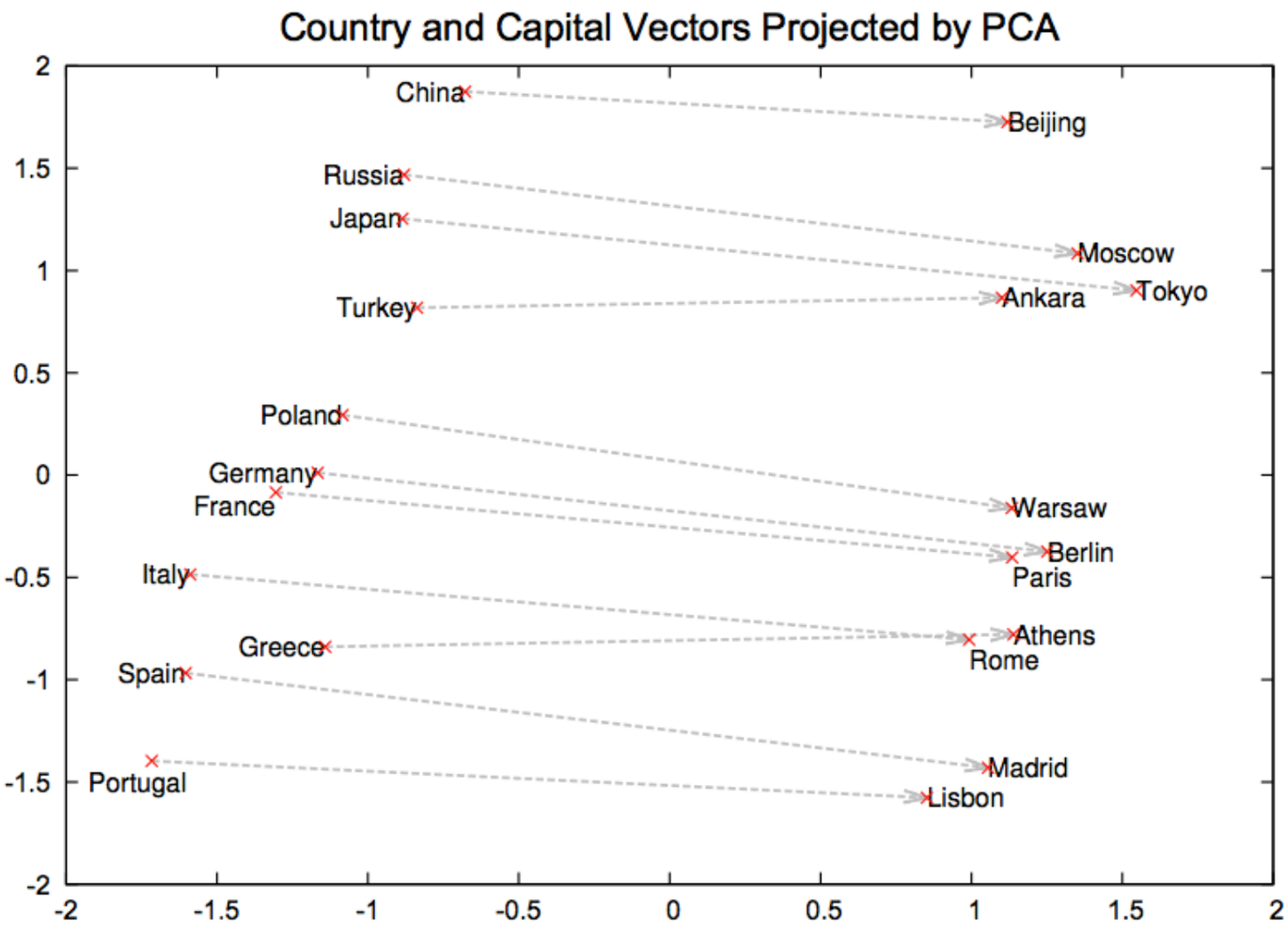
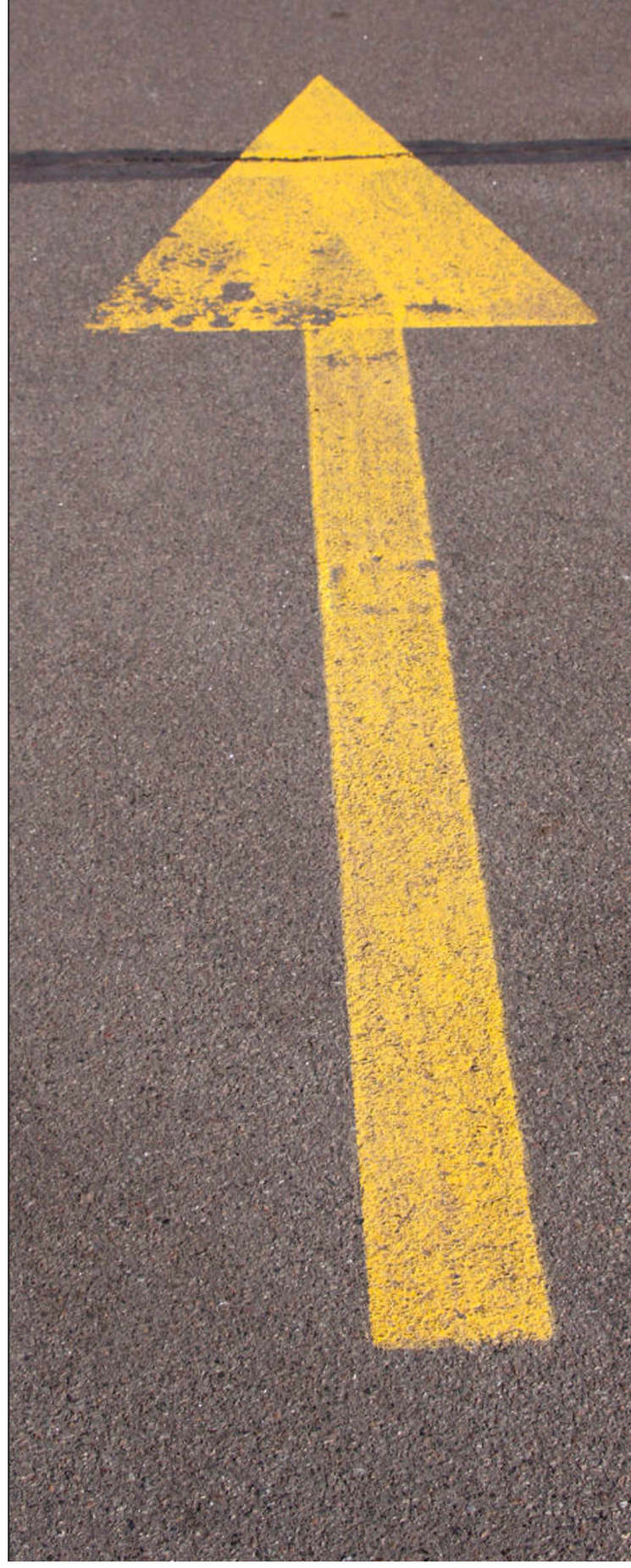
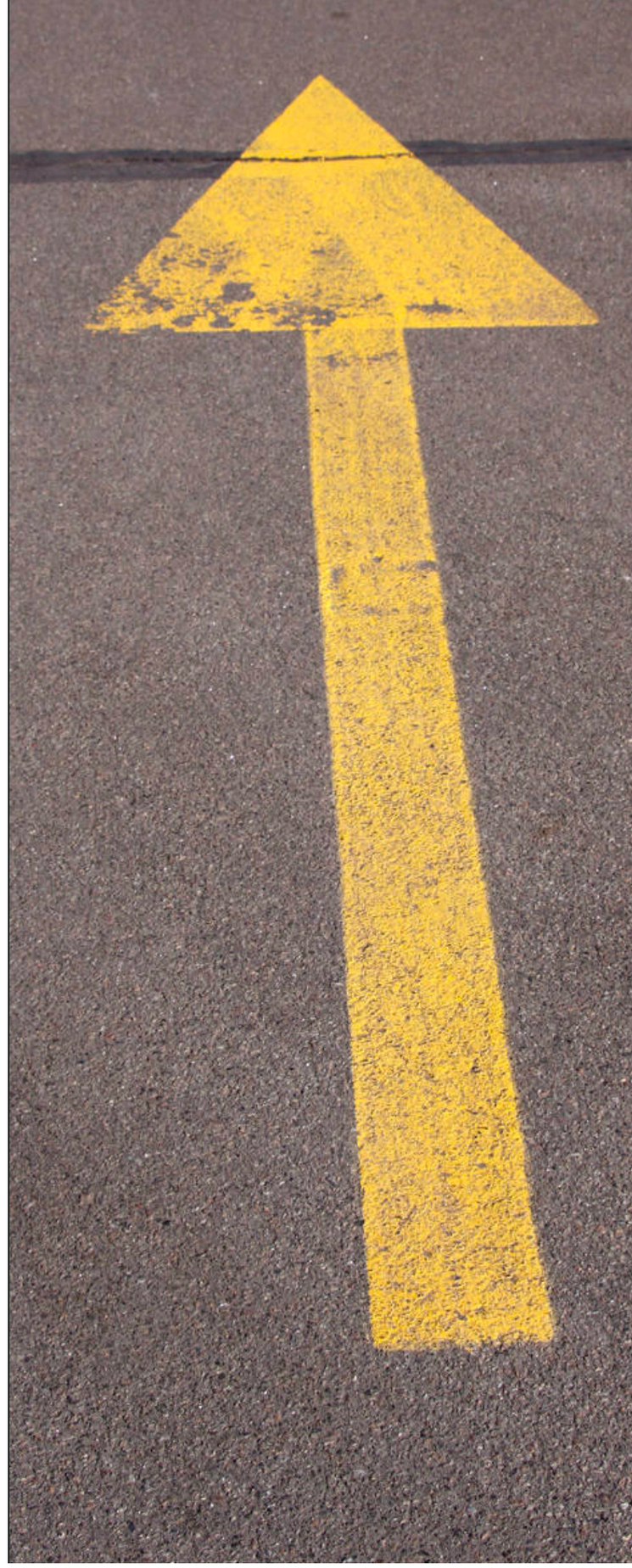


Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.



*Source : Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

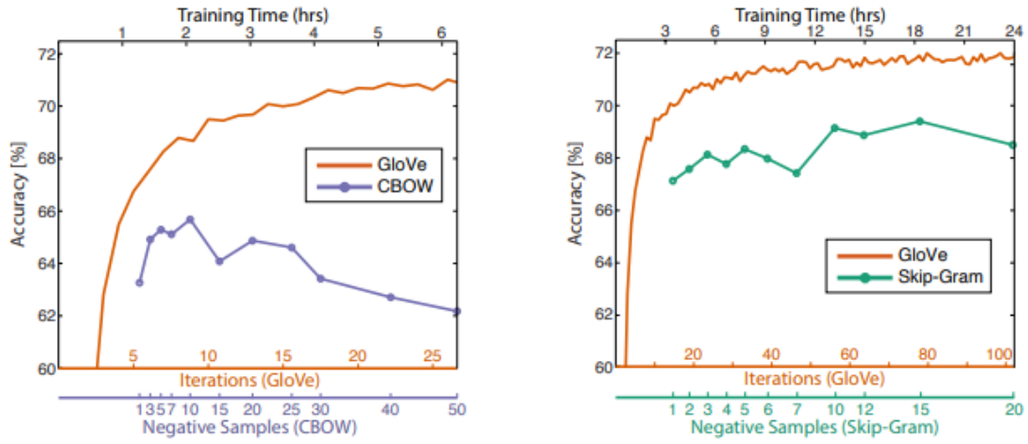
단어 임베딩 (Word Embedding)



단어 임베딩 기법: GloVe

- 기존의 Word2Vec에서와는 달리, 전체 corpus의 통계정보를 활용해 효율성 증대 및 성능 향상
- 주어진 corpus와 window size 가지고 co-occurrence matrix 생성함
- Word2Vec과 유사한 방법으로 학습 대상이 되는 단어들을 window size 안에서 선택함
- 선택한 단어와 matrix를 기반으로 objective function을 활용해 학습함
- 장점

  - count기반의 방법과 direct prediction 기반 방법을 결합해 성능이 크게 향상됨
  - 학습 시간이 빠르며, 큰 corpus에 대해 적용 가능함
  - 작은 corpus에서도 좋은 성능을 보임 (vector size가 작아도 성능이 좋음)
  - 출현빈도가 적은 단어에 대해서도 비교적 좋을 결과가 나옴



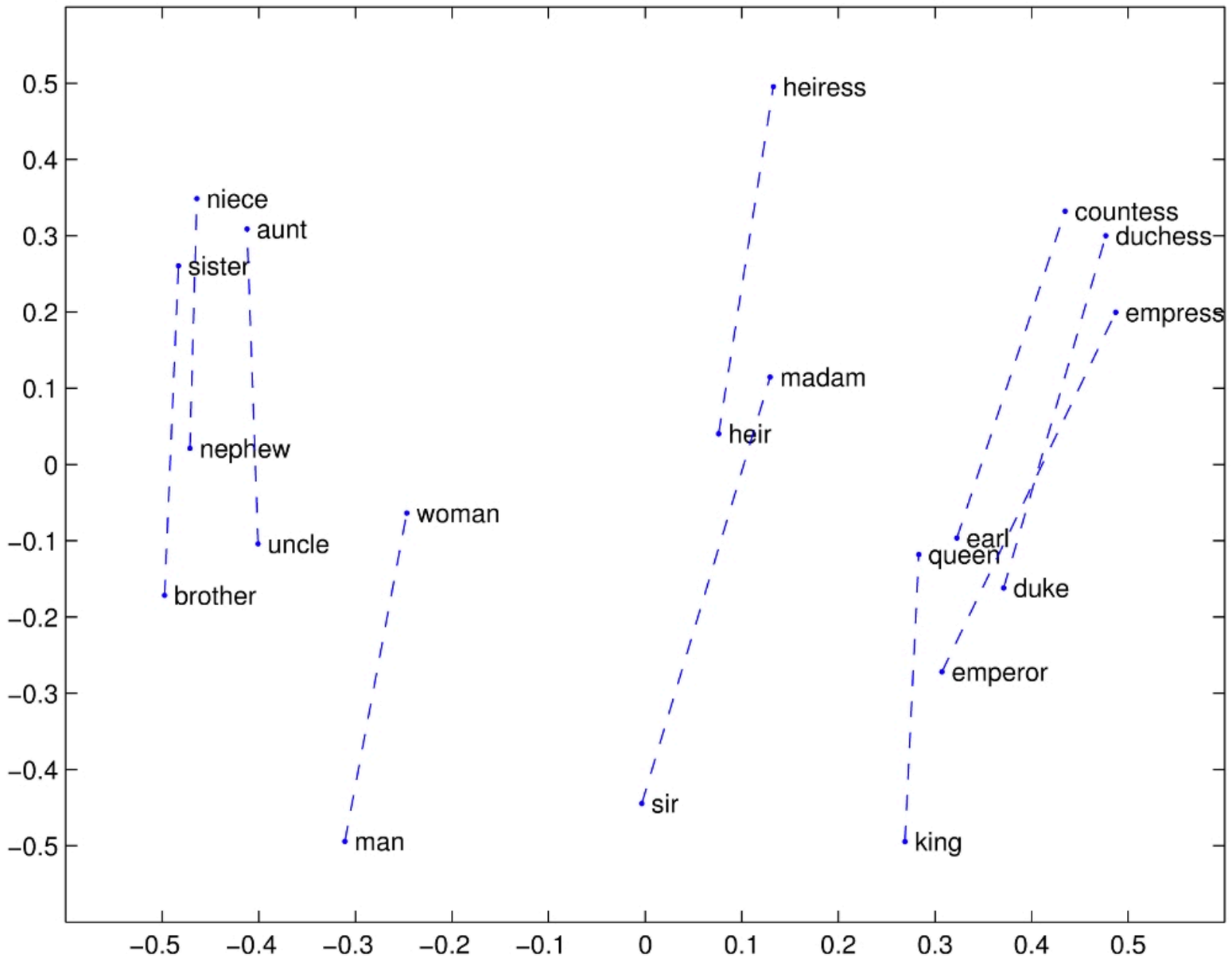
(a) GloVe vs CBOW (b) GloVe vs Skip-Gram

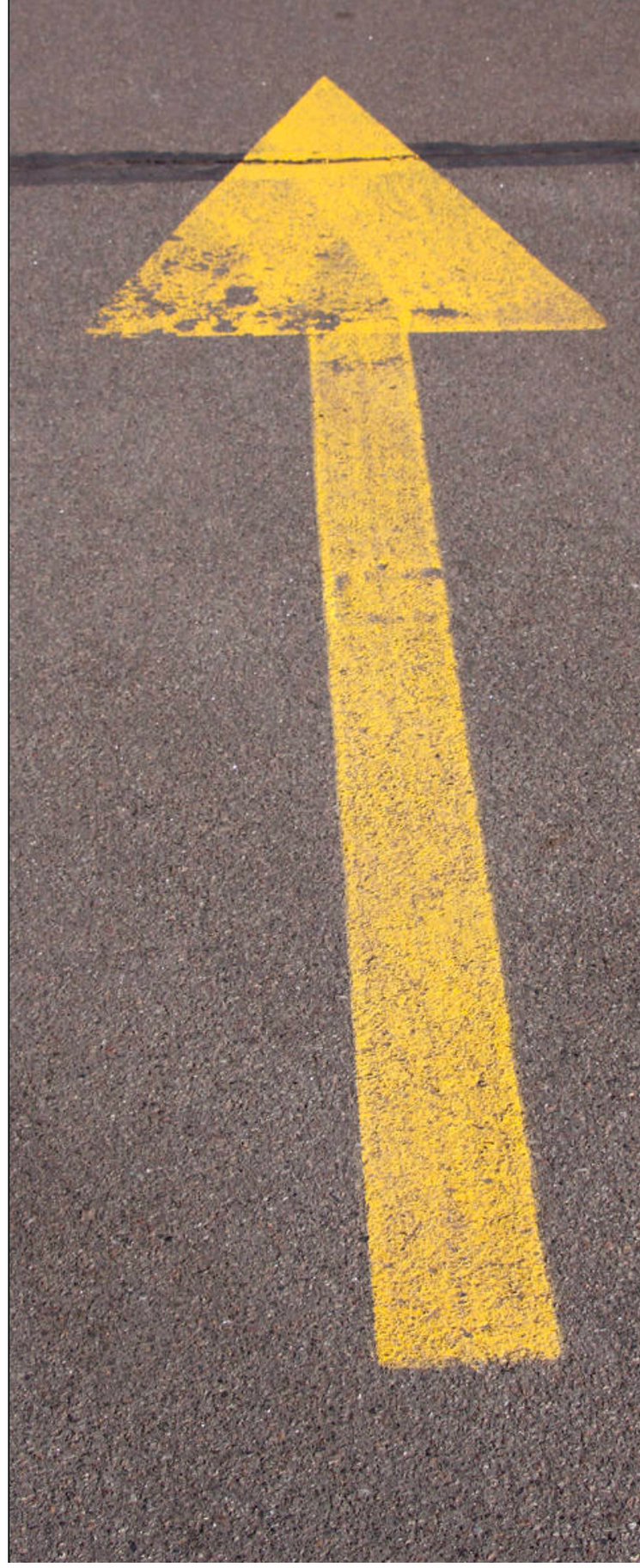
Figure 4: Overall accuracy on the word analogy task as a function of training time, which is governed by the number of iterations for GloVe and by the number of negative samples for CBOW (a) and skip-gram (b). In all cases, we train 300-dimensional vectors on the same 6B token corpus (Wikipedia 2014 + Gigaword 5) with the same 400,000 word vocabulary, and use a symmetric context window of size 10.

*Source : Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).

단어 임베딩 (Word Embedding)

단어 임베딩 기법: GloVe





*Source : Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
**Source : Jeffrey Pennington, Richard Socher, and Christopher D. Manning, GloVe: Global Vectors for Word Representation, https://nlp.stanford.edu/projects/glove/.

단어 임베딩 (Word Embedding)

단어 임베딩 기법: Word2Vec vs GloVe

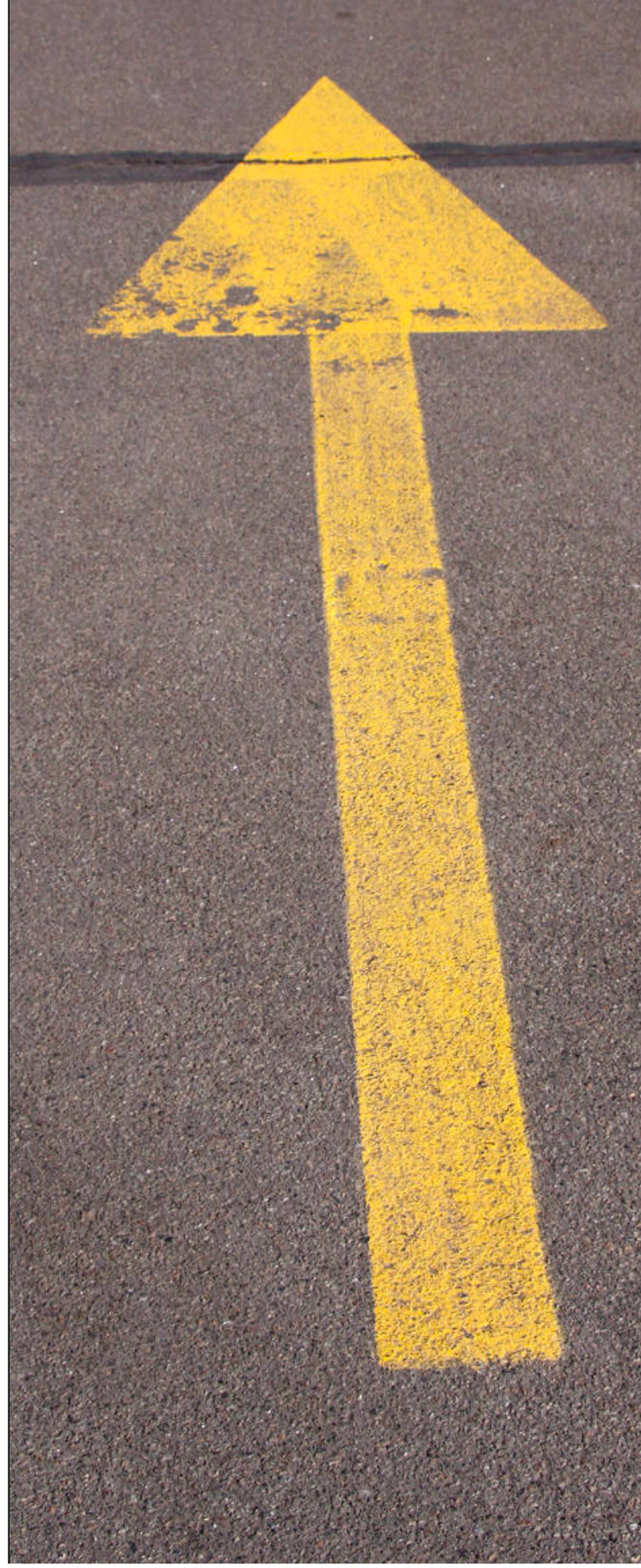
Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	67.5	54.3	60.3
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	61.5	70.3
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW†	300	6B	63.6	67.4	65.7
SG†	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7
CBOW	1000	6B	57.3	68.9	63.7
SG	1000	6B	66.1	65.1	65.6
SVD-L	300	42B	38.4	58.2	49.2
GloVe	300	42B	**81.9**	**69.3**	**75.0**

Table 3: Spearman rank correlation on word similarity tasks. All vectors are 300-dimensional. The CBOW* vectors are from the `word2vec` website and differ in that they contain phrase vectors.

Model	Size	WS353	MC	RG	SCWS	RW
SVD	6B	35.3	35.1	42.5	38.3	25.6
SVD-S	6B	56.5	71.5	71.0	53.6	34.7
SVD-L	6B	65.7	72.7	75.1	56.5	37.0
CBOW†	6B	57.2	65.6	68.2	57.0	32.5
SG†	6B	62.8	65.2	69.7	58.1	37.2
GloVe	6B	65.8	72.7	77.8	53.9	38.1
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	**75.9**	**83.6**	**82.9**	**59.6**	**47.8**
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

Table 4: F1 score on NER task with 50d vectors. *Discrete* is the baseline without word vectors. We use publicly-available vectors for HPCA, HSMN, and CW. See text for details.

Model	Dev	Test	ACE	MUC7
Discrete	91.0	85.4	77.4	73.4
SVD	90.8	85.7	77.3	73.7
SVD-S	91.0	85.5	77.6	74.3
SVD-L	90.5	84.8	73.6	71.5
HPCA	92.6	**88.7**	81.7	80.7
HSMN	90.5	85.7	78.7	74.7
CW	92.2	87.4	81.7	80.2
CBOW	93.1	88.2	82.2	81.1
GloVe	**93.2**	88.3	**82.9**	**82.2**



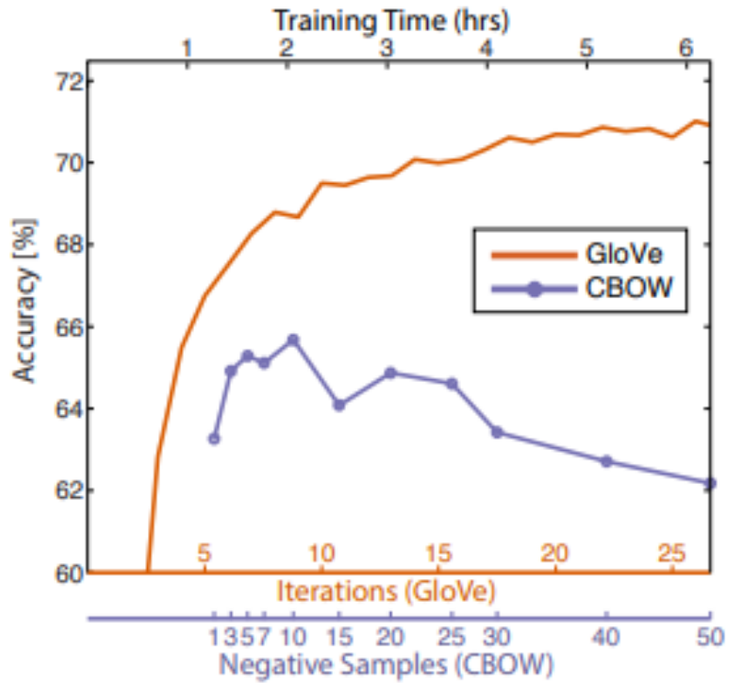
*Source : Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
**Source : Jeffrey Pennington, Richard Socher, and Christopher D. Manning, GloVe: Global Vectors for Word Representation, https://nlp.stanford.edu/projects/glove/.

단어 임베딩 (Word Embedding)

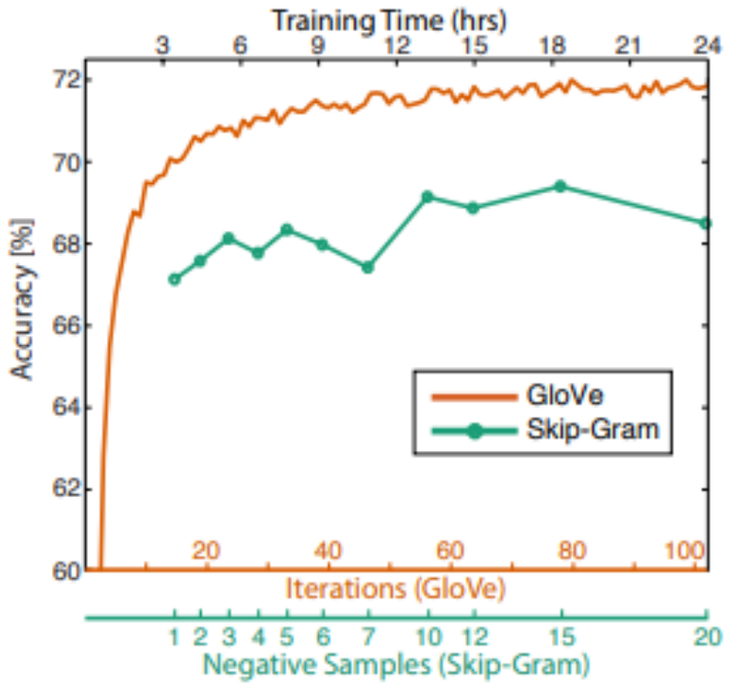
단어 임베딩 기법: Word2Vec vs GloVe

Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	67.5	54.3	60.3
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	61.5	70.3

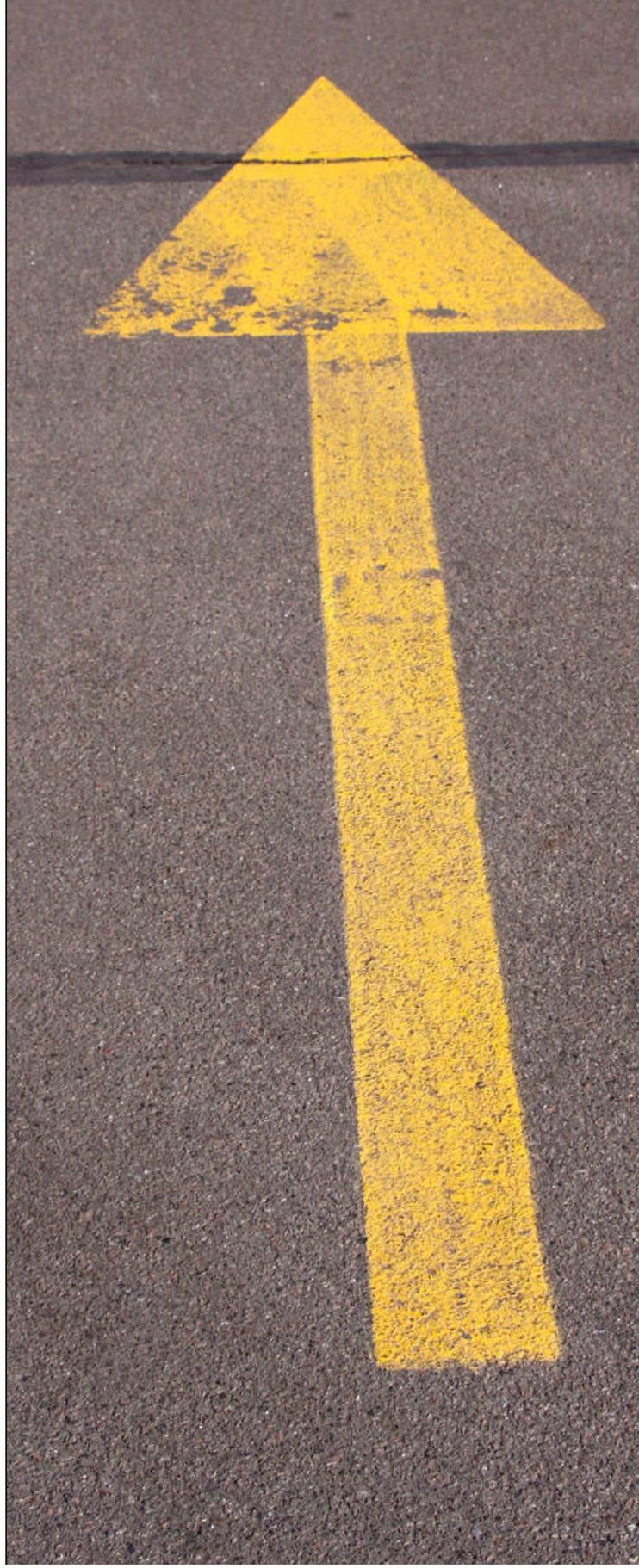
Model	Dim.	Size	Sem.	Syn.	Tot.
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW†	300	6B	63.6	67.4	65.7
SG†	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7
CBOW	1000	6B	57.3	68.9	63.7
SG	1000	6B	66.1	65.1	65.6
SVD-L	300	42B	38.4	58.2	49.2
GloVe	300	42B	**81.9**	**69.3**	**75.0**



(a) GloVe vs CBOW

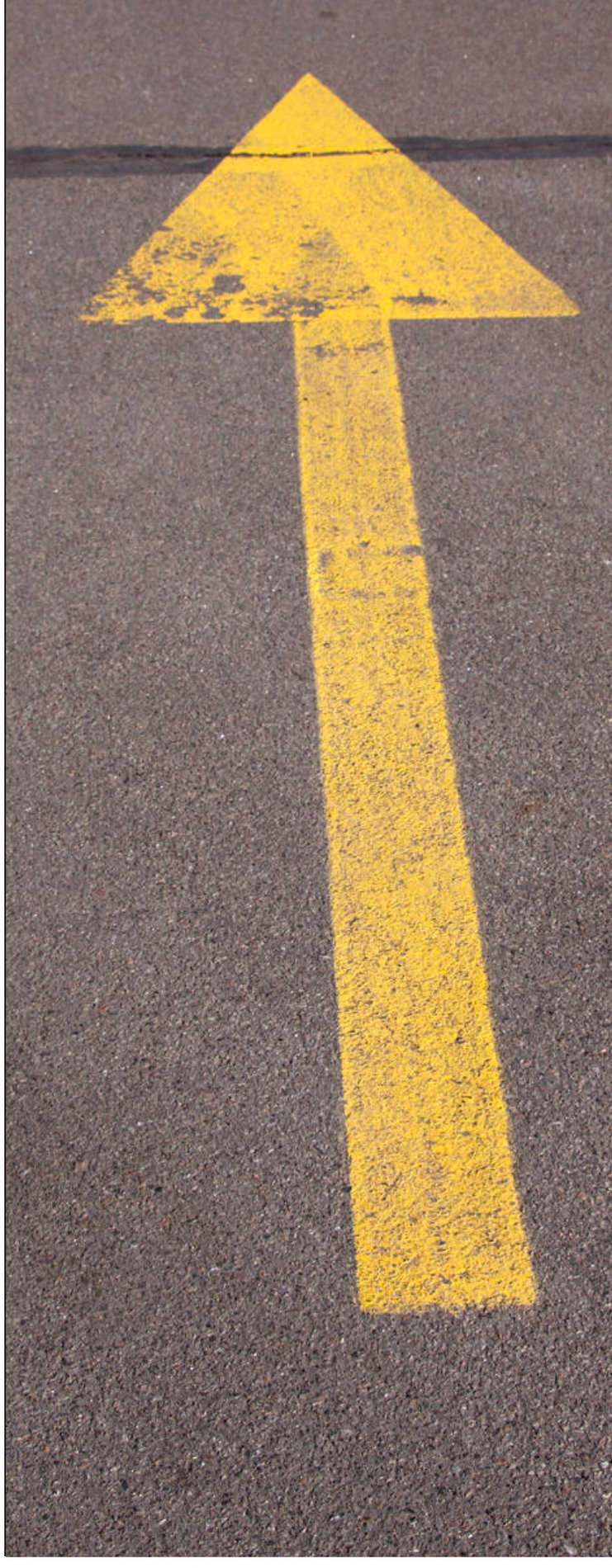


(b) GloVe vs Skip-Gram



*Source : Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
**Source : Jeffrey Pennington, Richard Socher, and Christopher D. Manning, GloVe: Global Vectors for Word Representation, https://nlp.stanford.edu/projects/glove/.

단어 임베딩 (Word Embedding)

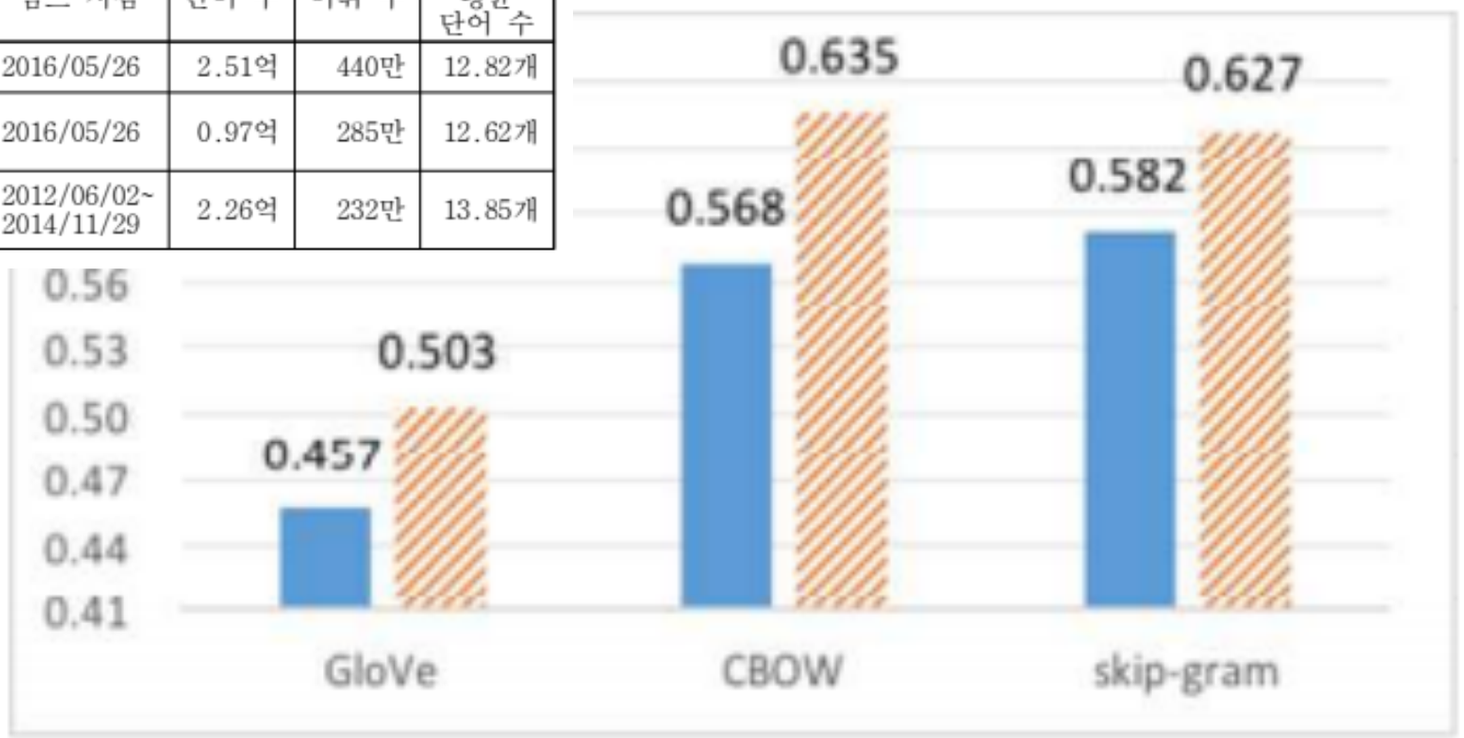


단어 임베딩 기법: Word2Vec vs GloVe

- 한국어는 GloVe보다 Word2Vec에서 더 높은 성능을 보임
  (영어의 경우 skip-gram 기반의 Word2Vec이 가장 높은 성능 보임)
- 한국어에 적합한 단어 임베딩 모델 및 파라미터 튜닝에 관한 연구 참고(2016, 최상혁, 설진석, 이상구)

<표 3: 학습에 사용된 말뭉치 분석>

말뭉치	덤프 시점	단어 수	어휘 수	문장당 평균 단어 수
나무위키	2016/05/26	2.51억	440만	12.82개
한국어 위키백과	2016/05/26	0.97억	285만	12.62개
뉴스 기사	2012/06/02~ 2014/11/29	2.26억	232만	13.85개



<그림 4: 모델별 성능비교>

*Source : 최상혁, 설진석, & 이상구. (2016). 한국어에 적합한 단어 임베딩 모델 및파라미터 튜닝에 관한 연구. 한글 및 한국어 정보처리 학술대회논문집], 제 28 회.

단어 임베딩 (Word Embedding)

단어 임베딩 기법: Word2Vec vs GloVe

- GloVe의 성능이 Word2Vec 보다 높게 나타남
- 어떤 Corpus로 어떻게 전처리 하느냐에 따라 성능 크게 달라짐
- A Study on Word Vector Models for Representing Korean Semantic Information 참고 (2015, Yang, Hejung et al.)

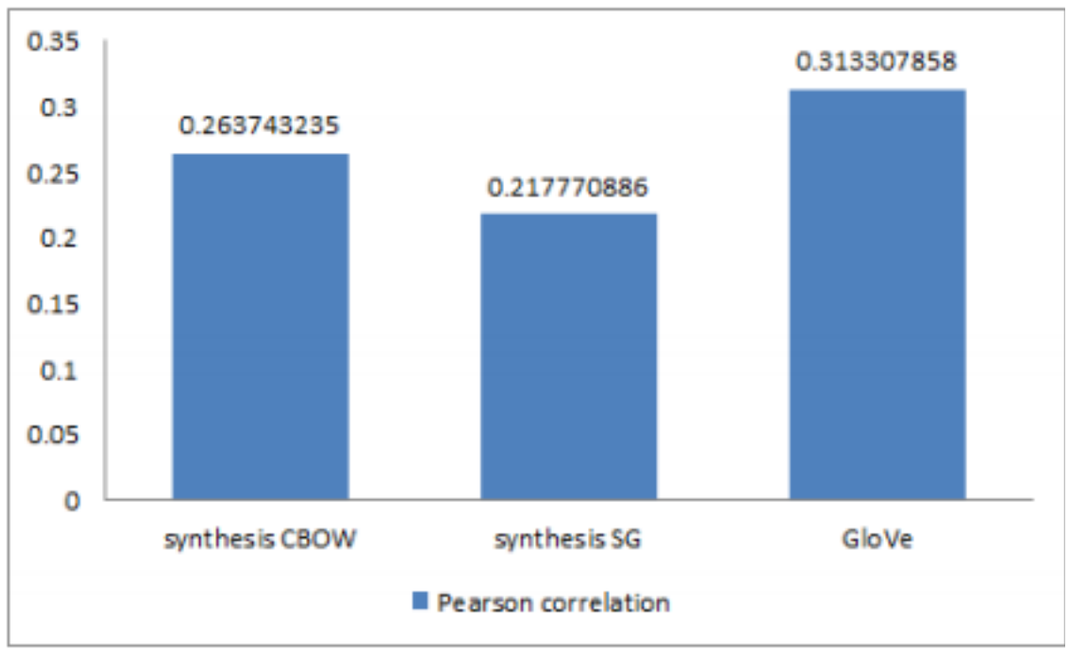


Figure 3. Pearson correlation coefficient of word2vec models

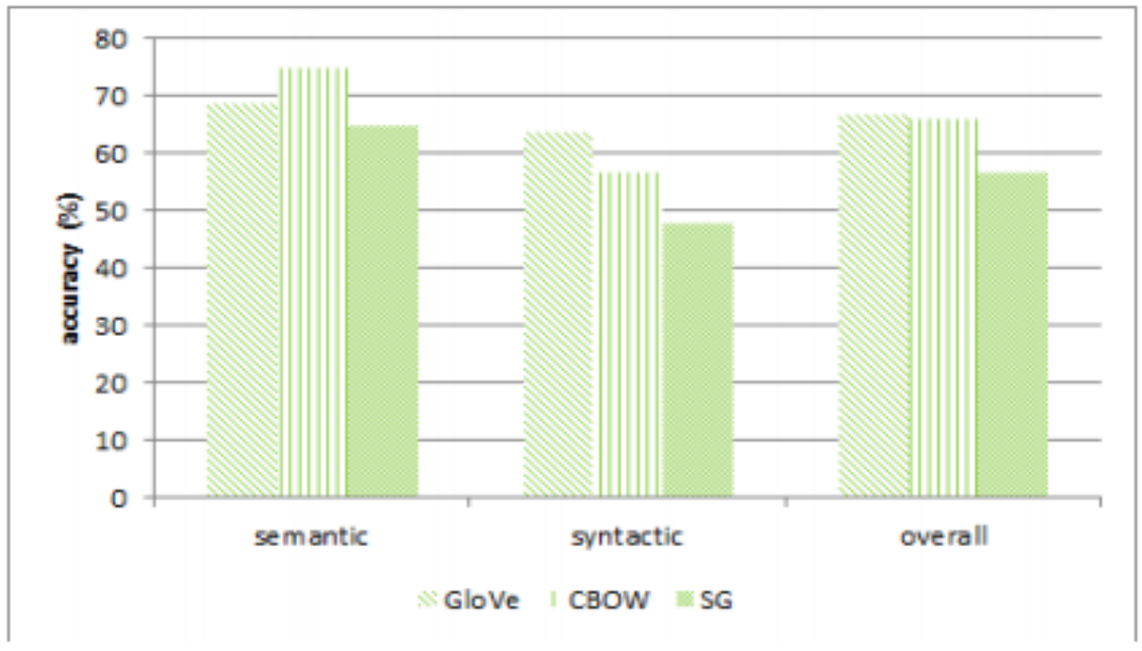


Figure 6. Accuracy on the word analogy task of word2vec and GloVe model.

*Source : Yang, H., Lee, Y. I., Lee, H. J., Cho, S. W., & Koo, M. W. (2015). A study on word vector models for representing korean semantic information. 말소리와 음성과학, 7(4), 41-47.

E.O.D