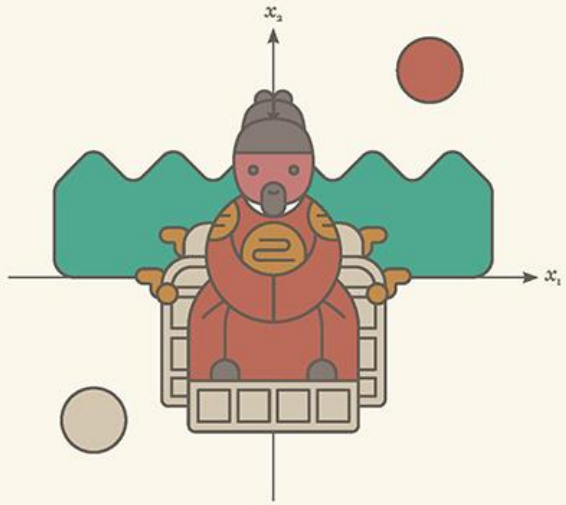
워드 임베딩 이론

한선아

목차

참고 자료

- 이기창, 2019, 한국어 임베딩
  https://ratsgo.github.io/embedding/



자연어 처리 모델의 성능을 높이는 핵심 비결
Word2Vec에서 ELMo, BERT까지

한국어 임베딩
Sentence Embeddings Using Korean Corpora

이기창 지음
NAVER Chatbot Model 감수

1. 임베딩(Embedding)이란

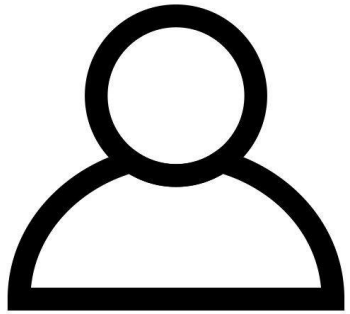
1. 임베딩(Embedding)이란

0010

1010

빠르고 효율적인 계산기

1. 임베딩(Embedding)이란



기차 → [0, 2, 10]

메밀꽃 → [12, 1, 0]

어머니 → [12, 15, 20]

1. 임베딩(Embedding)이란

사람이 쓰는 자연어를 기계가 이해할 수 있도록,

숫자의 나열인 벡터(vector)로 바꾼 결과 혹은 그 일련의 과정 전체

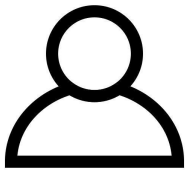
1. 임베딩(Embedding)이란

사람이 쓰는 자연어를 기계가 이해할 수 있도록,

숫자의 나열인 벡터(vector)로 바꾼 결과 혹은 그 일련의 과정 전체

컴퓨터과학적 관점에서,

벡터는 여러 개의 숫자를 하나로 묶어서 사용하는 것을 말합니다.



기차 → [0, 2, 10]

메밀꽃 → [12, 1, 0]

어머니 → [12, 15, 20]

TTA 한국정보통신기술협회 정보통신용어사전, 벡터
https://terms.tta.or.kr/dictionary/dictionaryView.do?word_seq=059390-1

2. 임베딩의 역할

임베딩으로 할 수 있는 일, 임베딩의 목적

2.1 단어/문장 간의 관련도 계산

코사인 유사도 등을 활용해 단어/문장 벡터들의 유사도를 알 수 있습니다.

2.2 의미/문법 정보 함축

단어/문장 벡터 간의 연산을 통해 의미적, 문법적 관계를 도출해낼 수 있습니다.

2.3 전이 학습

임베딩을 다른 모델의 입력값으로 사용해 성능을 높일 수 있습니다.

2.1 단어/문장 간의 관련도 계산

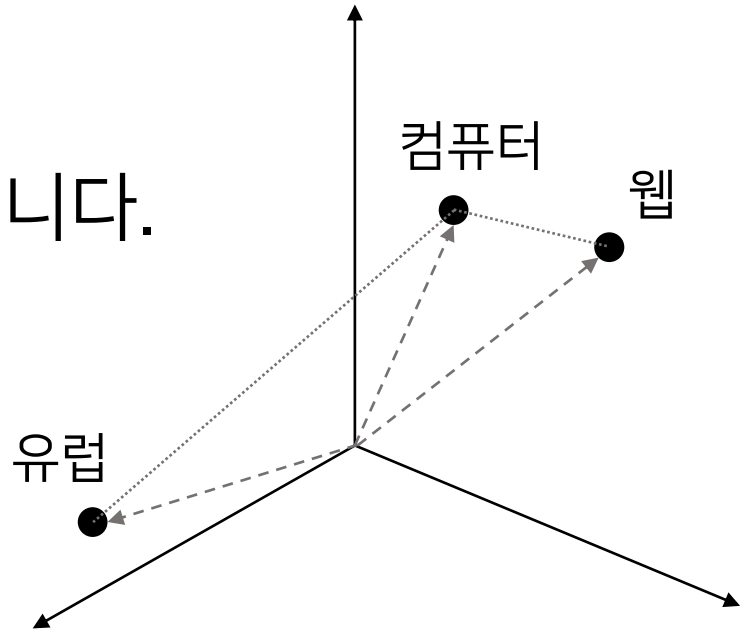
2.1 단어/문장 간의 관련도 계산

단어를 벡터로 임베딩하면,

1. 가상의 공간 안에 단어들의 위치를 찍을 수 있습니다.

2. 어떤 단어들이 가깝고 먼지 알 수 있습니다.

⇥ 즉, 단어들의 유사도를 알 수 있습니다.



2.1 단어/문장 간의 관련도 계산

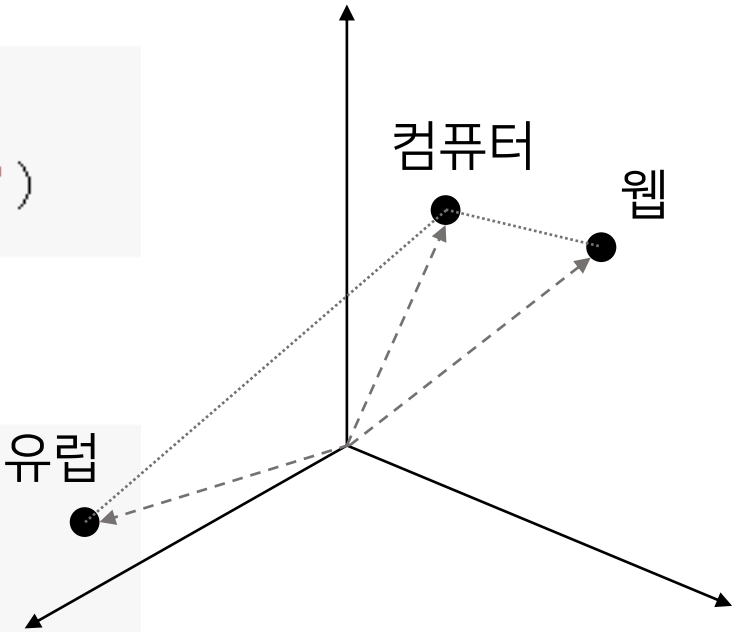
두 단어의 유사도 계산

```
[18]  1 # '컴퓨터'와 '유럽'의 유사도
      2 word2vec_vectors.similarity('컴퓨터', '유럽')
```

0.3540771

```
[19]  1 # '컴퓨터'와 '웹'의 유사도
      2 word2vec_vectors.similarity('컴퓨터', '웹')
```

0.7922803

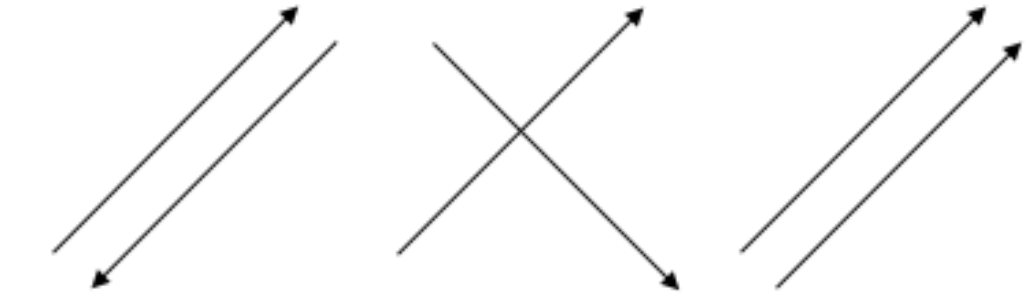


2.1 단어/문장 간의 관련도 계산

코사인 유사도(Cosine similarity)

두 벡터 간의 코사인 각도를 이용하여 구할 수 있는, 두 벡터의 유사도를 말합니다.

➡ 두 벡터가 가리키는 방향이 얼마나 유사한가를 의미합니다.



코사인유사도는 -1이상 1이하의 값을 가지며,

값이 1에 가까울수록 유사도가 높다고 판단할 수 있습니다.

2.2 의미/문법 정보 함축

단어를 벡터로 임베딩하면,

단어 벡터 간의 연산을 통해

단어들 사이의 의미적, 문법적 관계를 도출해낼 수 있습니다.

왕 + 여성 – 남성 = ???

2.2 의미/문법 정보 함축

단어벡터의 연산

[21] 0초

```
# 왕 + 여성 - 남성 = ???
word2vec_vectors.most_similar(positive=['왕', '여성'], negative=['남성'], topn=5)
```

```
[('여왕', 0.7429717779159546),
 ('왕비', 0.7376519441604614),
 ('국왕', 0.7237517833709717),
 ('왕국', 0.720409095287323),
 ('왕가', 0.7172703742980957)]
```

2.3 전이 학습(Transfer learning)

이미 만들어진 임베딩을

다른 딥러닝 모델의 입력값으로 쓰는 것을 말합니다.

2.3 전이 학습(Transfer learning)

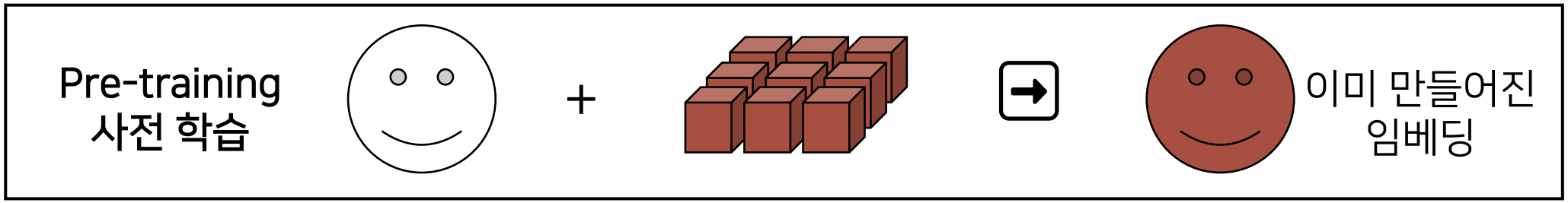
: 이미 만들어진 임베딩을 다른 딥러닝 모델의 입력값으로 쓰는 것을 말합니다.

🎼 목표 : 글의 주제 찾기

한글부터 가르치기  vs.  글을 읽고 이해할 수 있는 사람에게 가르치기

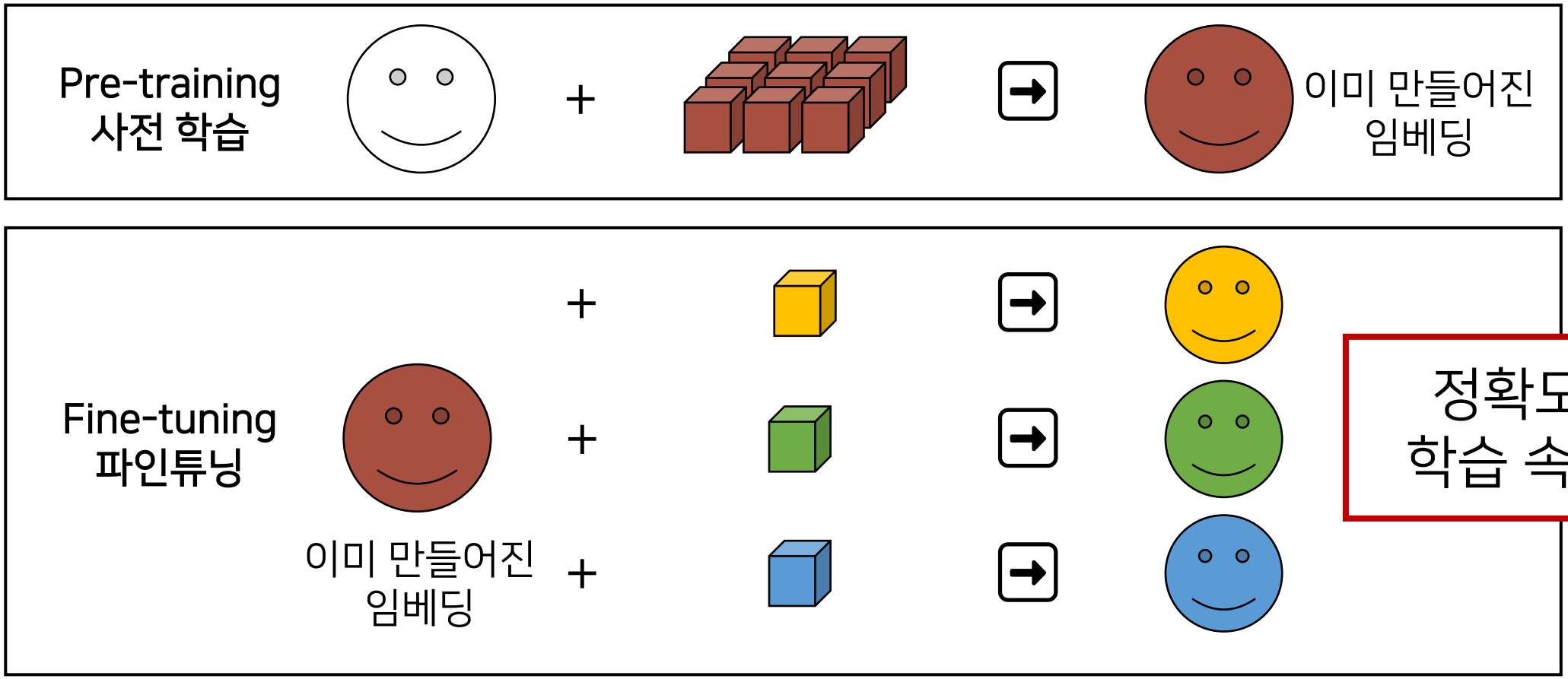
2.3 전이 학습(Transfer learning)

: 이미 만들어진 임베딩을 다른 딥러닝 모델의 입력값으로 쓰는 것을 말합니다.



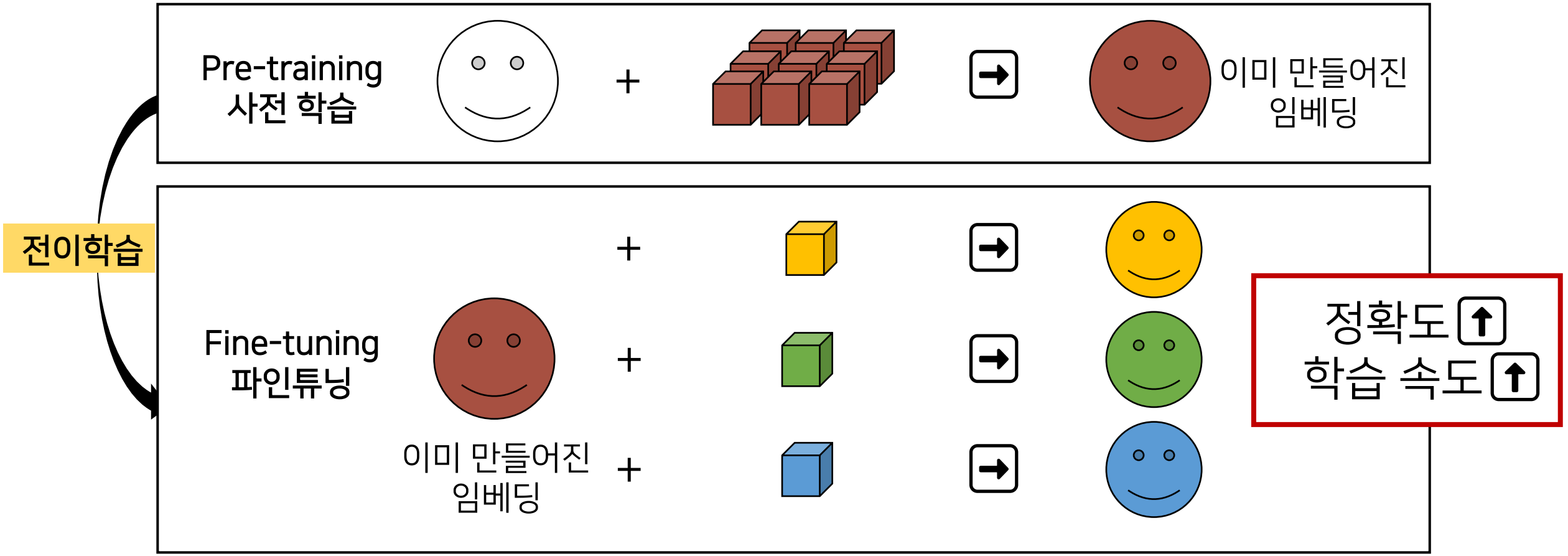
2.3 전이 학습(Transfer learning)

: 이미 만들어진 임베딩을 다른 딥러닝 모델의 입력값으로 쓰는 것을 말합니다.



2.3 전이 학습(Transfer learning)

: 이미 만들어진 임베딩을 다른 딥러닝 모델의 입력값으로 쓰는 것을 말합니다.



Review

2. 임베딩의 역할

임베딩으로 할 수 있는 일, 임베딩의 목적

2.1 단어/문장 간의 관련도 계산

코사인 유사도 등을 활용해 단어/문장 벡터들의 유사도를 알 수 있습니다.

2.2 의미/문법 정보 함축

단어/문장 벡터 간의 연산을 통해 의미적, 문법적 관계를 도출해낼 수 있습니다.

2.3 전이 학습

임베딩을 다른 모델의 입력값으로 사용해 성능을 높일 수 있습니다.

3. 임베딩에 의미를 어떻게 함축하는가

숫자가 어떻게 자연어의 의미를 담을 수 있을까?

단어의 사용 빈도, 순서, 주변 문맥 등

말뭉치(Corpus)의 통계적 패턴(statistical pattern)정보를 넣습니다.

3. 임베딩에 의미를 어떻게 함축하는가

숫자가 어떻게 자연어의 의미를 담을 수 있을까?

3.1 백오브워즈(bag of words) 가정

단어들의 순서를 고려하지 않고, 말뭉치에서 사용된 빈도를 세어 사용합니다.

3.2 언어 모델(language model)

순서를 가진, 단어 시퀀스가 자연스러울수록 더 높은 확률을 부여합니다.

3.3 분포 가정(distributional hypothesis)

앞 뒤 문맥에 어떤 단어가 같이 나왔는지 봅니다.

말뭉치의 통계적 패턴을 서로 다른 각도에서 분석하는 것이며, 상호보완적입니다.

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

단어들의 순서를 고려하지 않고,

단어의 사용 여부와 등장 빈도를 임베딩으로 쓰는 기법

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

중복 원소를 허용한 집합, multiset

별 하나 에 추억 과
별 하나 에 사랑 과
별 하나 에 쓸쓸함 과
별 하나 에 동경 과
별 하나 에 시 와
별 하나 에 어머니 , 어머니

순서 X

빈도 추출

별	6
하나	6
에	6
추억	1
과	4
쓸쓸	1
함	1
동경	1
시	1
와	1
어머니	2
,	1

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

문서 단어 행렬(Document-Term Matrix, DTM)

	별	하나	에	추억	과	사랑	쓸쓸	함	동경	시	와	어머니
별 헤는 밤	10	8	12	1	8	1	1	1	1	1	1	3
흰 바람 벽이 있어	0	0	7	0	5	5	4	0	0	0	0	2
님의 침묵	0	0	9	1	1	4	0	0	0	0	0	0

- 별 헤는 밤은 “별”이 주제임을 알 수 있습니다.

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

문서 단어 행렬(Document-Term Matrix, DTM)

	별	하나	에	추억	과	사랑	쓸쓸	함	동경	시	와	어머니
별 헤는 밤	10	8	12	1	8	1	1	1	1	1	1	3
흰 바람 벽이 있어	0	0	7	0	5	5	4	0	0	0	0	2
님의 침묵	0	0	9	1	1	4	0	0	0	0	0	0

- 그러나, 단어의 빈도수가 꼭 그 문서의 주제를 나타내지는 않는다는 단점이 있습니다.

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

TF-IDF(Term Frequency - Inverse Document Frequency)

다른 문서에 비해 해당 문서에서만 특별히 많이 등장하는 단어에
집중하기 위한 기법입니다.

예) 을/를, 이/가 등의 조사는 대부분의 문서에 등장합니다.

문서 단어 행렬(Document-Term Matrix, DTM)

	별	하나	에	추억	과	사랑	쓸쓸	함	동경	시	와	어머니
별 헤는 밤	10	8	12	1	8	1	1	1	1	1	1	3
흰 바람 벽이 있어	0	0	7	0	5	5	4	0	0	0	0	2
님의 침묵	0	0	9	1	1	4	0	0	0	0	0	0

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

TF-IDF(Term Frequency - Inverse Document Frequency)

$$TF - IDF(w) = TF(w) \times \log\left(\frac{N}{DF(w)}\right)$$

TF가 클수록,

DF가 작을 수록,

결과값 TF-IDF가 커집니다.

$TF(w)$: Term Frequency, 특정 단어(w)가 특정 문서에서 나타난 빈도수

$DF(w)$: Document Frequency, 특정 단어(w)가 나타난 문서의 수

N : 전체 문서 수

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

TF-IDF(Term Frequency - Inverse Document Frequency)

별 헤는 밤에서 별이 나온 빈도수
TF = 10

	별	하나	에	추억
별 헤는 밤	10	8	12	1
흰 바람 벽이 있어	0	0	7	0
님의 침묵	0	0	9	1

...

전체 문서 수
N = 3

별이 나온 문서 수
DF = 1

$$TF - IDF(\text{별}) = TF(\text{별}) \times \log\left(\frac{N}{DF(\text{별})}\right)$$

$$= 10 \times \log\left(\frac{3}{1}\right) = 4.7712\ldots$$

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

TF-IDF(Term Frequency – Inverse Document Frequency)

별 헤는 밤에서 '에'가 나온 빈도수
TF = 12

	별	하나	에	추억
별 헤는 밤	10	8	12	1
흰 바람 벽이 있어	0	0	7	0
님의 침묵	0	0	9	1

...

전체 문서 수
N = 3

'에'가 나온 문서 수
DF = 3

$$TF - IDF(\text{에}) = TF(\text{에}) \times \log\left(\frac{N}{DF(\text{에})}\right)$$

$$= 12 \times \log\left(\frac{3}{3}\right) = 0$$

$\log 1 = 0$

단어의 사용 빈도

3.1 백오브워즈(Bag of words) 가정

TF-IDF(Term Frequency - Inverse Document Frequency)

TF-IDF 기법 적용시

	별	하나	에	추억	과
별 헤는 밤	10	8	12	1	8
흰 바람 벽이 있어	0	0	7	0	5
님의 침묵	0	0	9	1	1

	별	하나	에	추억	과
별 헤는 밤	4.7712	3.8170	0	0.1761	0
흰 바람 벽이 있어	0	0	0	0	0
님의 침묵	0	0	0	0.1761	0

Review

3. 임베딩에 의미를 어떻게 함축하는가

숫자가 어떻게 자연어의 의미를 담을 수 있을까?

3.1 백오브워즈(bag of words) 가정

단어들의 순서를 고려하지 않고, 말뭉치에서 사용된 빈도를 세어 사용합니다.

3.2 언어 모델(language model)

순서를 가진, 단어 시퀀스가 자연스러울수록 더 높은 확률을 부여합니다.

3.3 분포 가정(distributional hypothesis)

앞 뒤 문맥에 어떤 단어가 같이 나왔는지 봅니다.

말뭉치의 통계적 패턴을 서로 다른 각도에서 분석하는 것이며, 상호보완적입니다.

단어의 등장 순서

3.2 언어 모델(Language Model)

언어 모델은, 단어 시퀀스(순서)에 확률을 부여하는 모델입니다.

P(누명을 쓰다) > P(누명을 당하다)

0.41 0.02

단어의 등장 순서

3.2 언어 모델(Language Model)

언어 모델은, 단어 시퀀스(순서)에 확률을 부여하는 모델입니다.

언어 모델의 두 가지 분류

3.2.1 통계 기반 언어 모델

3.2.2 뉴럴 네트워크 기반 언어 모델

단어의 등장 순서

3.2.1 통계 기반 언어 모델

말뭉치에서 해당 단어 시퀀스가 얼마나 자주 등장하는지 빈도를 세어 확률을 구합니다.

$$P(\text{최고의 명작이다}) = ?$$

단어의 등장 순서

3.2.1 통계 기반 언어 모델

말뭉치에서 해당 단어 시퀀스가 얼마나 자주 등장하는지 빈도를 세어 확률을 구합니다.

$$P(\text{최고의 명작이다}) = P(\text{최고의}) \times P(\text{명작이다}|\text{최고의})$$

표현	빈도
최고의	3503
명작이다	298
최고의 명작이다	23

$$P(\text{명작이다}|\text{최고의}) = \frac{Freq(\text{최고의 명작이다})}{Freq(\text{최고의})} = \frac{23}{3503}$$

단어의 등장 순서

3.2.1 통계 기반 언어 모델

$$P(\text{내 마음 속에 영원히 기억될 최고의 명작이다}) = ?$$

단어의 등장 순서

3.2.1 통계 기반 언어 모델

희소 문제(sparsity Problem)

: 충분한 데이터를 관측하지 못하여 언어를 정확히 모델링하지 못하는 문제

표현	빈도
최고의	3503
명작이다	298
최고의 명작이다	23

내 마음 속에 영원히 기억될 최고의 명작이다	0

$$P(\text{명작이다}|\text{내, 마음, 속에, 영원히, 기억될, 최고의}) = 0$$

단어의 등장 순서

3.2.1 통계 기반 언어 모델

희소 문제(sparsity Problem)

$$P(\text{내 마음 속에 영원히 기억될 최고의 명작이다})$$

$$= P(\text{내}) \times P(\text{마음}|\text{내}) \times \cdots \underbrace{P(\text{명작이다}|\text{내 마음 속에 영원히 기억될 최고의})}_{=0}$$

$$= 0$$

"내 마음 속에 영원히 기억될 최고의 명작이다 " 는

문법적, 의미적으로 결함이 없는 문장임에도, 확률을 0으로 부여하게 됩니다.

단어의 등장 순서

3.2.1 통계 기반 언어 모델

N-gram 언어 모델

: 이전에 등장한 n-1개의 단어만 고려하여 통계적 접근 방식을 사용합니다.

바이그램(bigram), n=2

"내 마음 속에 영원히 기억될 최고의 명작이다 "

$$P(\text{명작이다}|\text{내}, \text{마음}, \text{속에}, \text{영원히}, \text{기억될}, \text{최고의}) \approx P(\text{명작이다}|\text{최고의}) = \frac{23}{3503}$$

단어의 등장 순서

3.2.1 통계 기반 언어 모델

N-gram 언어 모델

: 이전에 등장한 n-1개의 단어만 고려하여 통계적 접근 방식을 사용합니다.

$$P(\text{내 마음 속에 영원히 기억될 최고의 명작이다}) \approx P(\text{내}) \times P(\text{마음}|\text{내}) \times P(\text{속에}|\text{마음}) \times P(\text{영원히}|\text{속에}) \times P(\text{기억될}|\text{영원히}) \times P(\text{최고의}|\text{기억될}) \times P(\text{명작이다}|\text{최고의})$$

바이그램(bigram), n=2

내 마음 속에 영원히 기억될 최고의 명작이다

내 마음 속에 영원히 기억될 최고의 명작이다

내 마음 속에 영원히 기억될 최고의 명작이다

내 마음 속에 영원히 기억될 최고의 명작이다

내 마음 속에 영원히 기억될 최고의 명작이다

내 마음 속에 영원히 기억될 최고의 명작이다

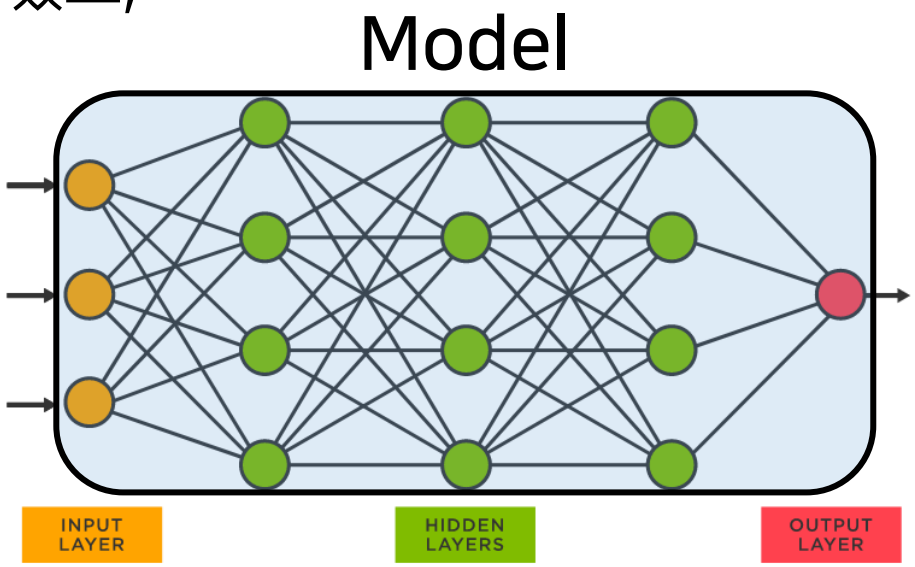
단어의 등장 순서

3.2.2 뉴럴 네트워크 기반 언어 모델

뉴럴 네트워크(neural network)

: 입력과 출력 사이의 관계를 유연하게 포착할 수 있고,

그 자체로 확률 모델로 기능할 수 있습니다.



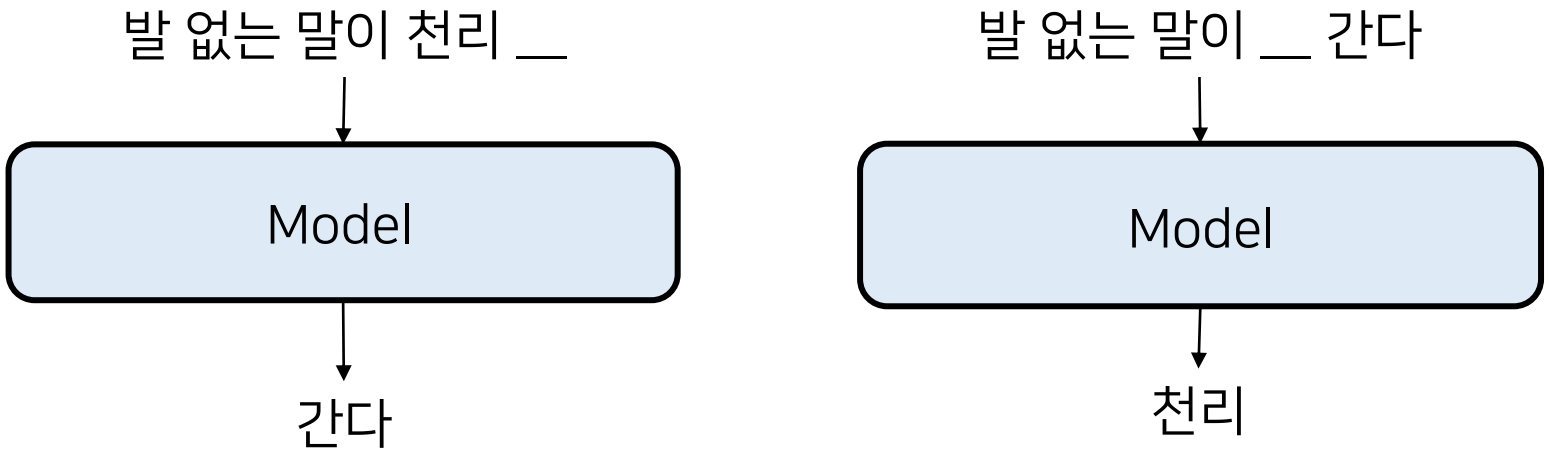
단어의 등장 순서

3.2.2 뉴럴 네트워크 기반 언어 모델

- 주어진 단어 시퀀스를 가지고 다음 단어를 맞추거나, (GPT, ELMo),

  문장의 중간에 가려진 단어를 양방향으로 예측하는 과정에서 학습됩니다.(BERT)

- 학습 후, 모델들의 중간 혹은 맨 마지막 층의 계산 결과물을 임베딩으로 활용합니다.



Review

3. 임베딩에 의미를 어떻게 함축하는가

숫자가 어떻게 자연어의 의미를 담을 수 있을까?

3.1 백오브워즈(bag of words) 가정

단어들의 순서를 고려하지 않고, 말뭉치에서 사용된 빈도를 세어 사용합니다.

3.2 언어 모델(language model)

순서를 가진, 단어 시퀀스가 자연스러울수록 더 높은 확률을 부여합니다.

3.3 분포 가정(distributional hypothesis)

앞 뒤 문맥에 어떤 단어가 같이 나왔는지 봅니다.

말뭉치의 통계적 패턴을 서로 다른 각도에서 분석하는 것이며, 상호보완적입니다.

단어의 주변 문맥

3.3 분포 가정(Distributional Hypothesis)

자연어처리에서 분포(distribution)

: 특정 범위(window, 윈도우) 내에 동시에 등장하는 이웃 단어 또는 문맥(context)의 집합

Window = 3

정부에서 적극 권장하였기 때문에, 더욱 아름답고 특징적인 다리들이 가설되기 시작하였다.

단어의 주변 문맥

3.3 분포 가정(Distributional Hypothesis)

분포 가정의 전제

“ 자연어의 의미는 그 주변 문맥을 통해 유추해볼 수 있다.”

“어떤 단어 쌍이 비슷한 문맥 환경에서 자주 등장한다면, 그 의미 또한 유사할 것이다.”

Window = 3

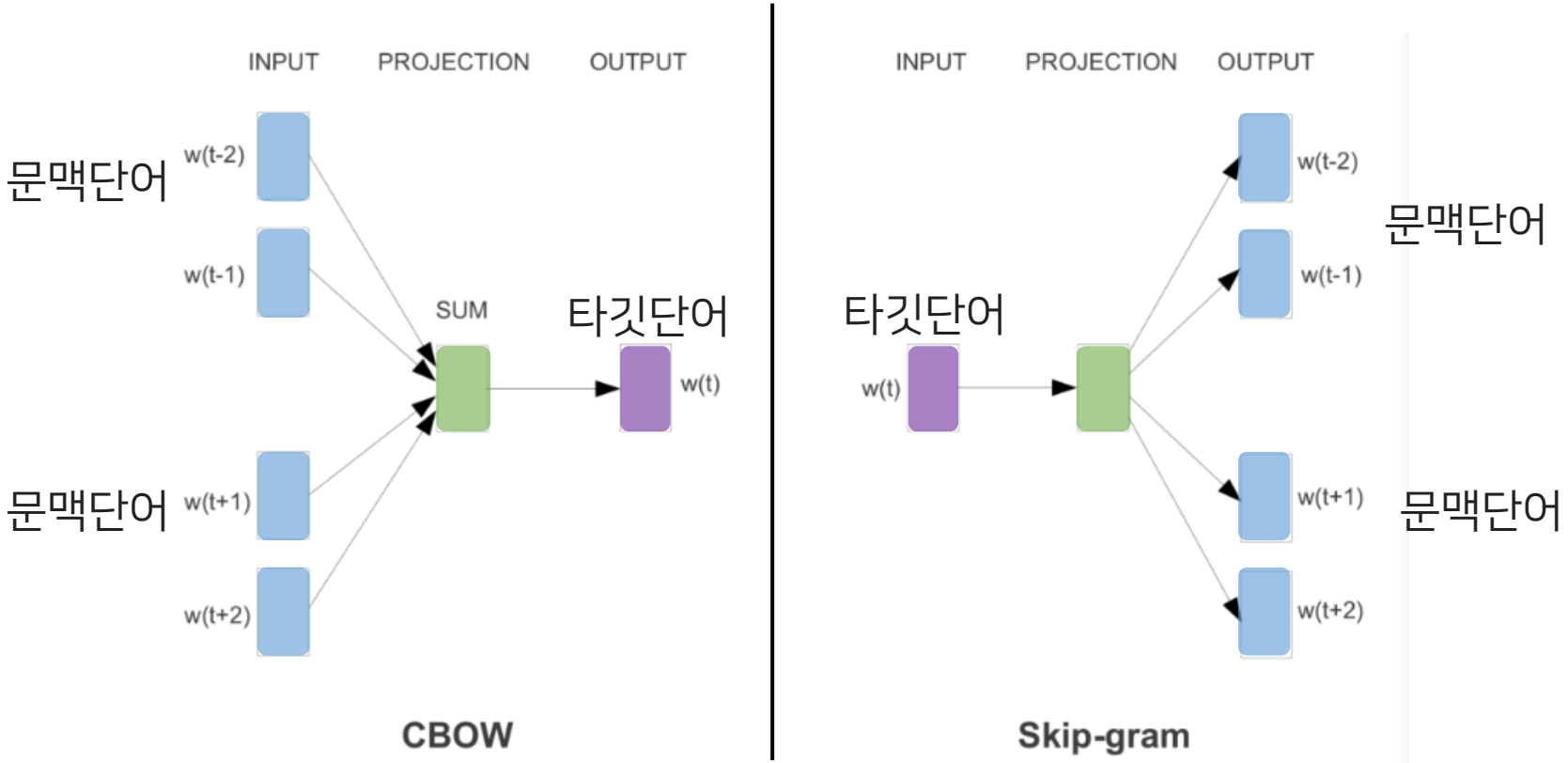
정부에서 적극 권장하였기 때문에, 더욱 아름답고 특징적인 다리들이 가설되기 시작하였다.

평발 때문에 아이가 다리를 아파한다며 병원에 오는데, 진단해보면 성장통이 원인인 게 대부분이다.

단어의 주변 문맥

3.3 분포 가정(Distributional Hypothesis)

Word2Vec : 문맥단어나, 타깃단어를 맞추는 과정에서 학습됩니다.



Review

3. 임베딩에 의미를 어떻게 함축하는가

숫자가 어떻게 자연어의 의미를 담을 수 있을까?

3.1 백오브워즈(bag of words) 가정

단어들의 순서를 고려하지 않고, 말뭉치에서 사용된 빈도를 세어 사용합니다.

3.2 언어 모델(language model)

순서를 가진, 단어 시퀀스가 자연스러울수록 더 높은 확률을 부여합니다.

3.3 분포 가정(distributional hypothesis)

앞 뒤 문맥에 어떤 단어가 같이 나왔는지 봅니다.

말뭉치의 통계적 패턴을 서로 다른 각도에서 분석하는 것이며, 상호보완적입니다.

4. 워드 임베딩 모델

워드 임베딩

희소한(sparse) 단어벡터를 밀집 벡터(dense vector)의 형태로

표현하는 방법을 말합니다.

4. 워드 임베딩 모델

희소한(sparse) 단어벡터를 밀집 벡터(dense vector)의 형태로 표현하는 방법을 말합니다.

희소 표현 (sparse representation)	밀집 표현 (dense representation)
벡터 또는 행렬이 대부분 0으로 표현된 것	사용자가 임의로 설정한 차원 크기에 맞추어 실수값으로 표현된 것
예) 원-핫 벡터 [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 …]	[-0.296575, 0.157551 , -0.239403, 0.371547 …]

4. 워드 임베딩 모델

4.1 Word2Vec

문맥단어나, 타깃단어를 맞추는 과정에서 학습됩니다.

4.2 FastText

Word2Vec과 유사하지만, 단어를 문자 단위 n-gram으로 표현합니다.

4.3 Glove

유사도 계산 성능이 좋으면서도,
말뭉치 전체 통계정보를 반영하고자 고안되었습니다.

4.1 Word2Vec

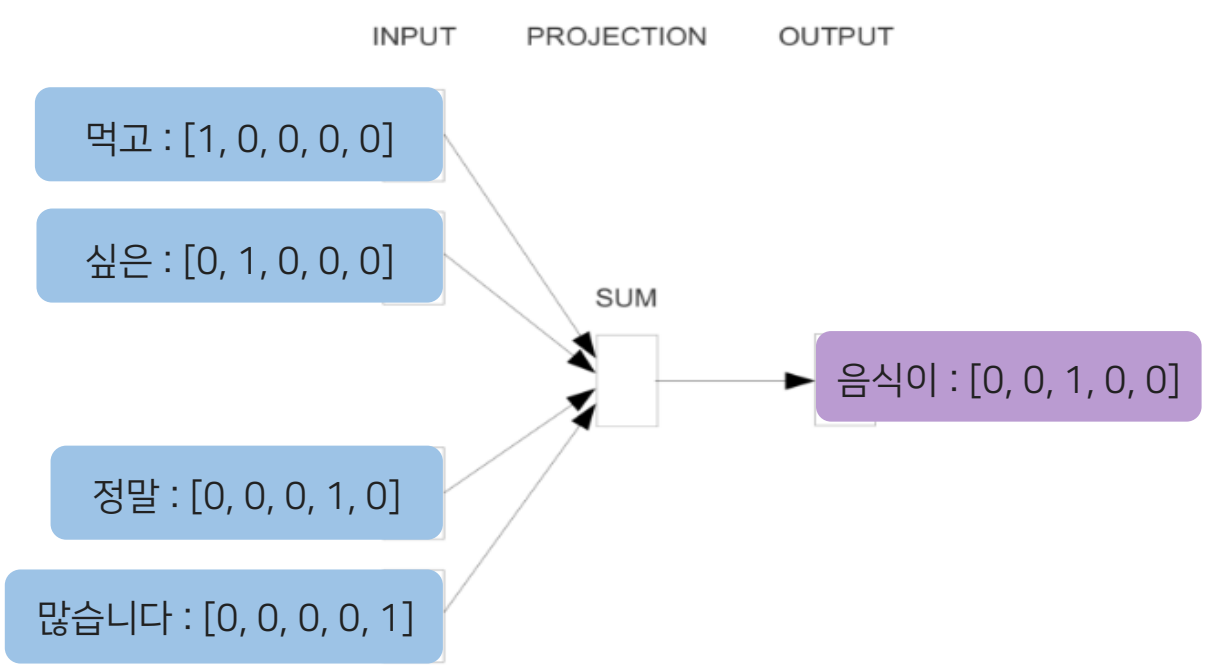
먼저, 단어를 희소벡터(sparse vector)인 원-핫 벡터(one-hot vector)로 만듭니다.

어휘집합 : [“먹고”, “싶은”, “음식이”, “정말”, “많습니다”]

- 먹고 : [1, 0, 0, 0, 0]
- 싶은 : [0, 1, 0, 0, 0]
- 음식이 : [0, 0, 1, 0, 0]
- 정말 : [0, 0, 0, 1, 0]
- 많습니다 : [0, 0, 0, 0, 1]

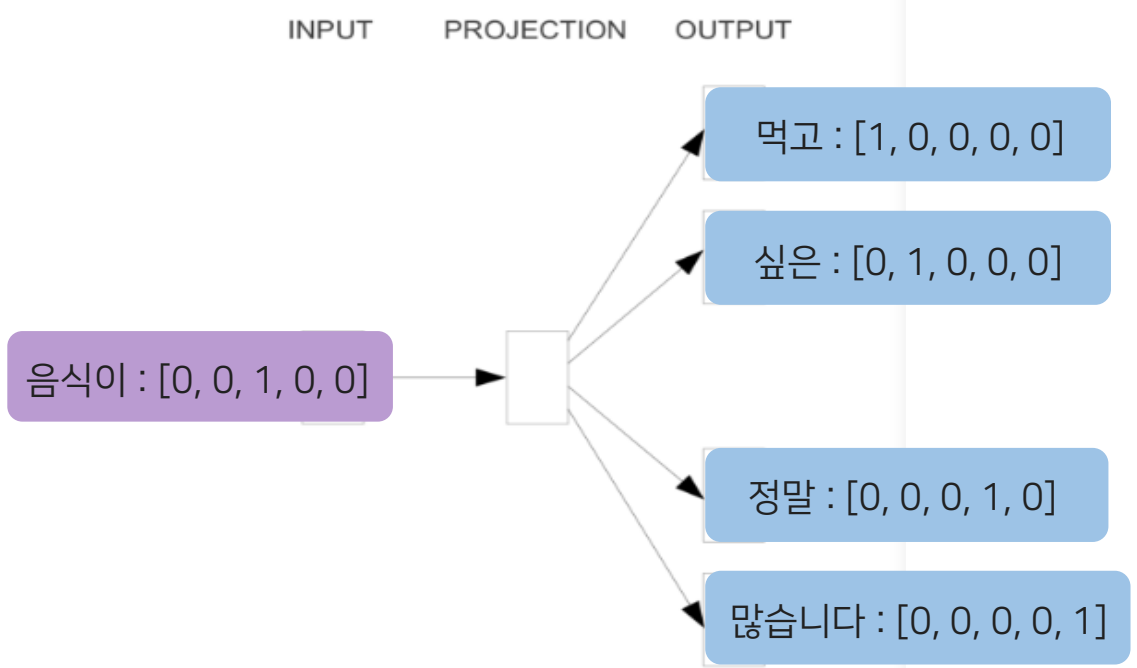
어휘집합의 크기가 커질 수록,
벡터의 차원도 커지는 문제점이 있습니다.

4.1 Word2Vec



CBOW

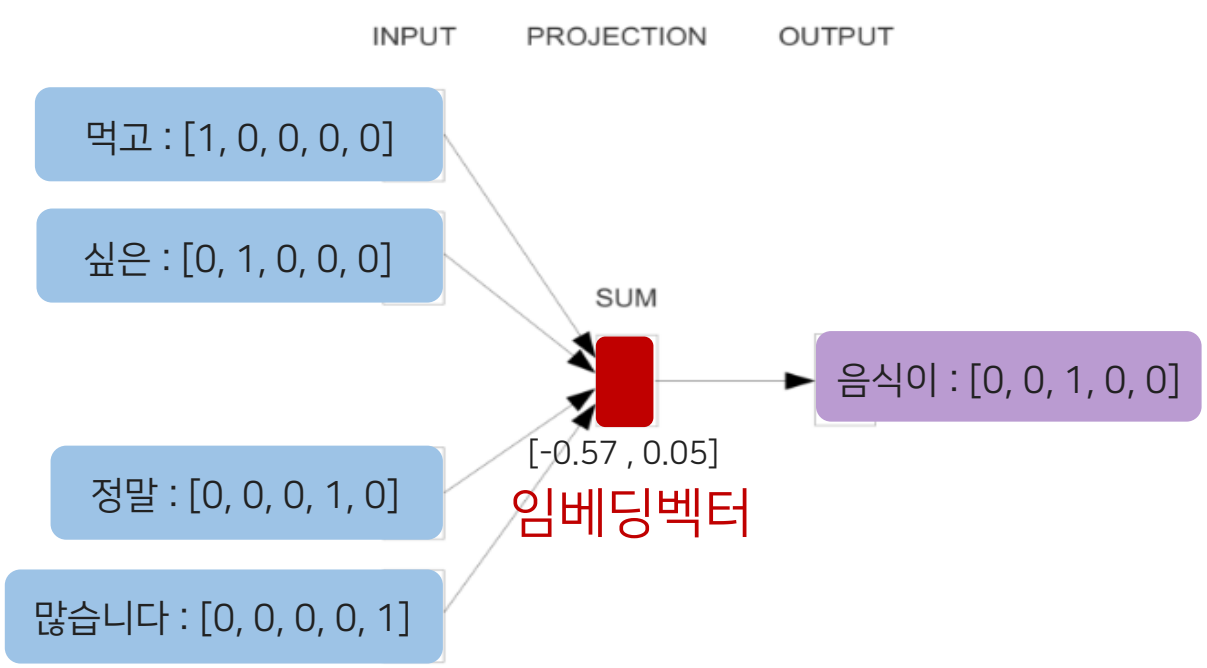
문맥 단어로 타깃 단어를 맞추며 학습합니다.



Skip-gram

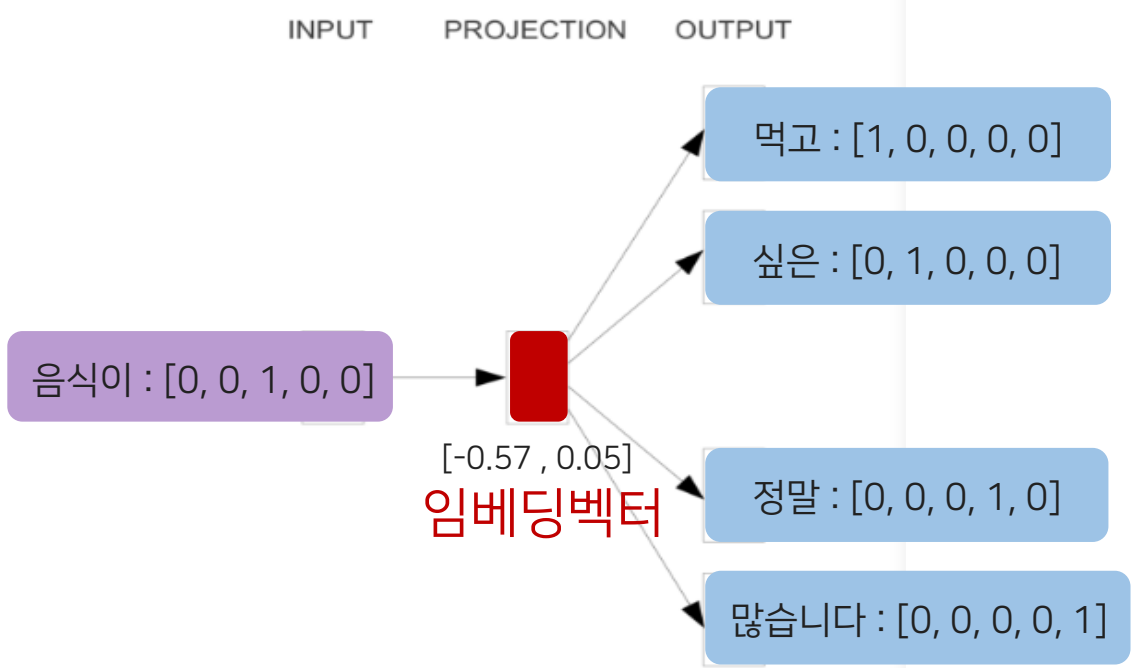
타깃 단어로 문맥 단어를 맞추며 학습합니다.

4.1 Word2Vec



CBOW

문맥 단어로 타깃 단어를 맞추며 학습합니다.



Skip-gram

타깃 단어로 문맥 단어를 맞추며 학습합니다.

4.2 FastText

- 페이스북에서 개발해 공개한 단어 임베딩 기법입니다.

- word2vec과 기본적으로 동일하나,

참고) <자연 vs <자연>

<자연, 자연어, 연어처, 어처리, 처리>

각 단어를 문자(Character) 단위 n-gram으로 표현합니다.

- FastText는 하나의 단어 안에도 여러 단어들이 존재하는 것으로 간주합니다.

내부 단어, 즉 서브워드(subword)를 고려하여 학습합니다.

- 코퍼스에 없는 모르는 단어(Out Of Vocabulary)에도 대처할 수 있다는 장점이 있습니다.

4.3 Glove

- 미국 스탠포드대학교연구팀에서 개발한 단어 임베딩 기법입니다.
- 유사도 계산의 성능이 좋으면서도, 윈도우 내의 로컬문맥(local context)만 학습하지 않고 전체의 통계정보를 반영하고자 고안된 기법입니다.
- 단어-문맥 행렬(동시 등장 행렬, co-occurrence matrix)을 사용합니다.
- 단정적으로 Word2Vec와 GloVe 중에서 어떤 것이 더 뛰어나다고 말할 수는 없고, 이 두 가지 전부를 사용해보고 성능이 더 좋은 것을 사용하는 것이 바람직합니다

4.3 Glove

-단어-문맥 행렬(동시 등장 행렬, co-occurrence matrix)을 사용합니다.

-오늘 뭐 먹고 싶어

-나는 오늘 연어 먹고 싶어

-나는 어제 연어 먹었어

	오늘	뭐	먹고	싶어	나는	연어	어제	먹었어
오늘	0	1	0	0	1	1	0	0
뭐	1	0	1	0	0	0	0	0
먹고	0	1	0	2	0	1	0	0
싶어	0	0	2	0	0	0	0	0
나는	1	0	0	0	0	0	1	0
연어	1	0	1	0	0	0	1	1
어제	0	0	0	0	1	1	0	0
먹었어	0	0	0	0	0	1	0	0

워드임베딩 이론 정리

1. 임베딩(Embedding)이란

사람이 쓰는 자연어를 기계가 이해할 수 있도록,

숫자의 나열인 벡터(vector)로 바꾼 결과 혹은 그 일련의 과정 전체

2. 임베딩의 역할

임베딩으로 할 수 있는 일, 임베딩의 목적

2.1 단어/문장 간의 관련도 계산

코사인 유사도 등을 활용해 단어/문장 벡터들의 유사도를 알 수 있습니다.

2.2 의미/문법 정보 함축

단어/문장 벡터 간의 연산을 통해 의미적, 문법적 관계를 도출해낼 수 있습니다.

2.3 전이 학습

임베딩을 다른 모델의 입력값으로 사용해 성능을 높일 수 있습니다.

3. 임베딩에 의미를 어떻게 함축하는가

숫자가 어떻게 자연어의 의미를 담을 수 있을까?

3.1 백오브워즈(bag of words) 가정

단어들의 순서를 고려하지 않고, 말뭉치에서 사용된 빈도를 세어 사용합니다.

3.2 언어 모델(language model)

순서를 가진, 단어 시퀀스가 자연스러울수록 더 높은 확률을 부여합니다.

3.3 분포 가정(distributional hypothesis)

앞 뒤 문맥에 어떤 단어가 같이 나왔는지 봅니다.

말뭉치의 통계적 패턴을 서로 다른 각도에서 분석하는 것이며, 상호보완적입니다.

4. 워드 임베딩 모델

워드 임베딩

희소한(sparse) 단어벡터를 밀집 벡터(dense vector)의 형태로

표현하는 방법을 말합니다.

4. 워드 임베딩 모델

4.1 Word2Vec

문맥단어나, 타깃단어를 맞추는 과정에서 학습됩니다.

4.2 FastText

Word2Vec과 유사하지만, 단어를 문자 단위 n-gram으로 표현합니다.

4.3 Glove

유사도 계산 성능이 좋으면서도,
말뭉치 전체 통계정보를 반영하고자 고안되었습니다.

워드 임베딩 실습

한선아