- Embeddings
- ddings: Word2vec & GloVe
- ord Embeddings
- Lecture Notes (Optional)
- Quiz
  - ✓ 테스트: Natural Language Processing & Word Embeddings
    10개의 질문
- Programming Assignments

테스트테스트 • 30 min30 minutes

Natural Language Processing & Word Embeddings

✓
과제 제출
기한년 8월 30일 오후 3:59 KST년 8월 30일 오후 3:59 KST
시도하기8 hours당 3회

계속하기

✓
성적 받기
통과 점수:80% 이상
성적
100%

피드백 보기

최고 점수가 유지됩니다.

탐색 확인

이 페이지에서 나가시겠습니까?

이 페이지에 머물기 이 페이지에서 나가기

←

Natural Language Processing & Word Embeddings
성적 평가 퀴즈 • 30 min

만료 년 8월 30일 오후 3:59 KST

✓

축하합니다! 통과하셨습니다!
통과 점수: 80% 이상

학습 계속하기

성적
100%

Natural Language Processing & Word Embeddings

최신 제출물 성적
100%
1.
질문 1

Suppose you learn a word embedding for a vocabulary of 10000 words. Then the embedding vectors should be 10000 dimensional, so as to capture the full range of variation and meaning in those words.

1 / 1점

○○

True

◉●

False

✓
맞습니다

The dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors range between 50 and 400.

2.
질문 2

What is t-SNE?

1 / 1점

◉●

A non-linear dimensionality reduction technique

○○

An open-source sequence modeling library

○○

A linear transformation that allows us to solve analogies on word vectors

○○

A supervised learning algorithm for learning word embeddings

✓
맞습니다

Yes

3.
질문 3

Suppose you download a pre-trained word embedding which has been trained on a huge corpus of text. You then use this word embedding to train an RNN for a language task of recognizing if someone is happy from a short snippet of text, using a small training set.

x (input text)	y (happy?)
I'm feeling wonderful today!	1
I'm bummed my cat is ill.	0
Really enjoying this!	1

Then even if the word “ecstatic” does not appear in your small training set, your RNN might reasonably be expected to recognize “I’m ecstatic” as deserving a label $y = 1$y = 1.

1 / 1점

◉●

True

○○

False

✓

맞습니다

Yes, word vectors empower your model with an incredible ability to generalize. The vector for “ecstatic” would contain a positive/happy connotation which will probably make your model classify the sentence as a "1".

4.
질문 4

Which of these equations do you think should hold for a good word embedding? (Check all that apply)

1 / 1점

□□

$e_{boy} - e_{girl} \approx e_{sister} - e_{brother}$eboy − egirl ≈ esister − ebrother

☑☒

$e_{boy} - e_{brother} \approx e_{girl} - e_{sister}$eboy − ebrother ≈ egirl − esister

✓

맞습니다

Yes!

□□

$e_{boy} - e_{brother} \approx e_{sister} - e_{girl}$eboy − ebrother ≈ esister − egirl

☑☒

$e_{boy} - e_{girl} \approx e_{brother} - e_{sister}$eboy − egirl ≈ ebrother − esister

✓

맞습니다

Yes!

5.
질문 5

Let EE be an embedding matrix, and let o_{1234}o1234 be a one-hot vector corresponding to word 1234. Then to get the embedding of word 1234, why don’t we call $E * o_{1234}$E ∗ o1234 in Python?

1 / 1점

○○

The correct formula is $E^T * o_{1234}$ET ∗ o1234.

○○

This doesn’t handle unknown words (<UNK>).

○○

None of the above: calling the Python snippet as described above is fine.

◉●

It is computationally wasteful.

✓
맞습니다

Yes, the element-wise multiplication will be extremely inefficient.

6.
질문 6

When learning word embeddings, we create an artificial task of estimating $P(target \mid context)$P(target | context). It is okay if we do poorly on this artificial prediction task; the more important by-product of this task is that we learn a useful set of word embeddings.

1 / 1점
○○

False

◉●

True

✓
맞습니다
7.
질문 7

In the word2vec algorithm, you estimate $P(t \mid c)$P(t | c), where tt is the target word and cc is a context word. How are t t and cc chosen from the training set? Pick the best answer.

1 / 1점
○○

cc is the one word that comes immediately before tt.

○○

cc is the sequence of all the words in the sentence before tt.

○○

cc is a sequence of several words immediately before tt.

◉●

cc and tt are chosen to be nearby words.

✓
맞습니다
8.
질문 8

Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings. The word2vec model uses the following softmax function:

$P(t \mid c) = \frac{e^{\theta_t^T e_c}}{\sum_{t'=1}^{10000} e^{\theta_{t'}^T e_c}}$P (t | c) = ∑t’=110000 eθt’T ec eθtT ec

Which of these statements are correct? Check all that apply.

1 / 1점
□□

After training, we should expect θ_tθt to be very close to $e_c$ec when tt and cc are the same word.

☑☒

θ_tθt and $e_c$ec are both 500 dimensional vectors.

✓

맞습니다

□□

θ_tθt and $e_c$ec are both 10000 dimensional vectors.

☑☒

θ_tθt and $e_c$ec are both trained with an optimization algorithm such as Adam or gradient descent.

✓

맞습니다

9.

질문 9

Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings.The GloVe model minimizes this objective:

$\min \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij})(\theta_i^T e_j + b_i + b_j' - log X_{ij})^2$min ∑i=110,000 ∑j=110,000 f(Xij)(θiT ej + bi + bj’ − logXij)2

Which of these statements are correct? Check all that apply.

1 / 1점

□□

θ_iθi and $e_j$ej should be initialized to 0 at the beginning of training.

☑☒

The weighting function $f(.)$f(.) must satisfy $f(0) = 0$f(0) = 0.

✓

맞습니다

The weighting function helps prevent learning only from extremely common word pairs. It is not necessary that it satisfies this function.

☑☒

X_{ij}Xij is the number of times word j appears in the context of word i.

✓

맞습니다

☑☒

θ_iθi and $e_j$ej should be initialized randomly at the beginning of training.

✓

맞습니다

10.

질문 10

You have trained word embeddings using a text dataset of m_1m1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of m_2m2 words. Keeping in mind that using word embeddings is a form of transfer learning, under which of these circumstances would you expect the word embeddings to be helpful?

1 / 1점

○○

m_1m1 << m_2m2

◉●

m_1m1 >> m_2m2

✓

맞습니다