



courserd

Sequence Models

Natural Language Processing & Word Embeddings

〈 이전 다음 〉



Introduction to Word Embeddings

。 🔮

동영상: 강의Word Representation

. Duration: 10 minutes10 min

igstyle

동영상: 강의Using Word Embeddings

. Duration: 9 minutes 9 min

. .

동영상: 강의Properties of Word Embeddings

. Duration: 11 minutes11 min

동영상: 강의Embedding Matrix

. Duration: 3 minutes3 min

Learning Word Embeddings: Word2vec & GloVe

。 **②**

동영상: 강의Learning Word Embeddings

. Duration: 10 minutes 10 min

. 🔽

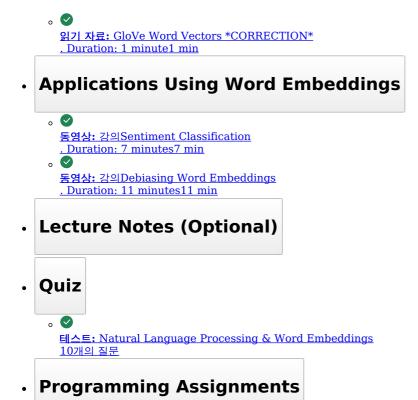
동영상: 강의Word2Vec

. Duration: 12 minutes12 min

동영상: 강의Negative Sampling . Duration: 11 minutes11 min

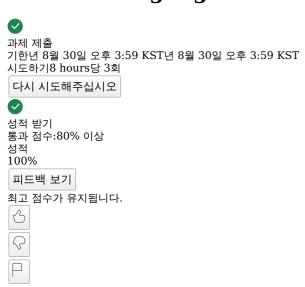
。 🔮

동영상: 강의GloVe Word Vectors . Duration: 11 minutes11 min



테스트테스트 • 30 min30 minutes

Natural Language Processing & Word Embeddings



탐색 확인

이 페이지에서 나가시겠습니까?

이 페이지에 머물기

이 페이지에서 나가기

 \leftarrow

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

1. 질문 1	
Suppose you learn a word en capture the full range of var	mbedding for a vocabulary of 10000 words. Then the embedding vectors should be 10000 dimensional, so as to riation and meaning in those words.
1/1 <u>2</u>	
True	
⊙ ○	
False	
✓ 맞습니다	
The dimension of word vector 400.	ors is usually smaller than the size of the vocabulary. Most common sizes for word vectors range between 50 and
2. 질문 2	
What is t-SNE?	
(1/1점) 〇 _〇	
A non-linear dimensionality	reduction technique
00	
An open-source sequence me	odeling library
00	
A linear transformation that	allows us to solve analogies on word vectors
00	
A supervised learning algori	thm for learning word embeddings
✓ 맞습니다	
Yes	
3.	
질문 3	e-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 langua	age 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 "ecstate ecstatic" as deserving a labe	atic" does not appear in your small training set, your RNN might reasonably be expected to recognize "I'm $y = 1y = 1$.
(1/1절) 〇 _〇	
True	
00	
False	
✓	
맞습니다	
Yes, word vectors empower connotation which will proba	your model with an incredible ability to generalize. The vector for "ecstatic" would contain a positive/happy ably make your model classify the sentence as a "1".

질문 4 Which of these equations do you think should hold for a good word embedding? (Check all that apply) 1 / 1점 $e_{boy} - e_{girl} \approx e_{sister} - e_{brother} \text{eboy} - \text{egirl} \approx \text{esister} - \text{ebrother}$ $e_{boy} - e_{brother} \approx e_{girl} - e_{sister} \text{eboy} - \text{ebrother} \approx \text{egirl} - \text{esister}$ 맞습니다 Yes! $e_{boy} - e_{brother} \approx e_{sister} - e_{girl} \text{eboy} - \text{ebrother} \approx \text{esister} - \text{egirl}$ $e_{boy} - e_{girl} \approx e_{brother} - e_{sister} \text{eboy} - \text{egirl} \approx \text{ebrother} - \text{esister}$ 맞습니다 Yes! 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 00 The correct formula is $E^T * o_{1234}$ ET * o1234. 00 This doesn't handle unknown words (<UNK>). 00 None of the above: calling the Python snippet as described above is fine. **O**O 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 00

False

O

True

맞습니다

질문 7

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

\circ_{\circ}
cc is the one word that comes immediately before t t.
\circ_{\circ}
cc is the sequence of all the words in the sentence before t t.
\circ_{\circ}
$c\mathrm{c}$ is a sequence of several words immediately before $t\mathrm{t}$.
$lackbox{0}_{\bigcirc}$
cc and t t 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(\mathbf{t} \mid \mathbf{c}) = \sum_{t'=1}^{10000} e^{\theta_{t'}^T e_c} P(\mathbf{t} \mid \mathbf{c}) = \sum_{t'=1}^{100000} e^{\theta_{t'}^T e_c} P(\mathbf{t} \mid \mathbf{c}) = \sum_{t'=1}^{1000000} e^{\theta_{t'}^T e_c} P(\mathbf{t} \mid \mathbf{c}) = \sum_{t'=1}^{100000} e^{\theta_{t'}^T e_c} P(\mathbf{t} \mid \mathbf{c}) = \sum_{t'=1}^{100000} e^{\theta_{t'}^T e_c} P(\mathbf{c} \mid \mathbf{c}) = \sum_{t'=1}^{1000000} e^{\theta_{t'}^T e_c} P(\mathbf{c} \mid \mathbf{c}) = \sum_{t'=1}^{1000000} e^{\theta_{t'}^T e_c} P(\mathbf{c} \mid $
Which of these statements are correct? Check all that apply.
After training, we should expect $\theta_t \theta t$ to be very close to $e_c ec$ when tt and cc are the same word.
$ heta_t heta t$ and $e_c ext{ec}$ are both 500 dimensional vectors.
✓ PG-IF
$ heta_t heta t$ and $e_c ext{ec}$ are both 10000 dimensional vectors.
$\theta_t \theta t$ 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 \sum_{j=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij}) (\theta_i^T e_j + b_i + b_j' - log X_{ij})^2 $
Which of these statements are correct? Check all that apply.
θ_i θ i and e_j ej should be initialized to 0 at the beginning of training.
The weighting function $f(.)f(.)$ must satisfy $f(0) = 0f(0) = 0$.
✓
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 ☑

 $\theta_i \theta i$ 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_1 m1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of m_2 m2 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?



 m_1 m1 << m_2 m2



 m_1 m1 >> m_2 m2



맞습니다