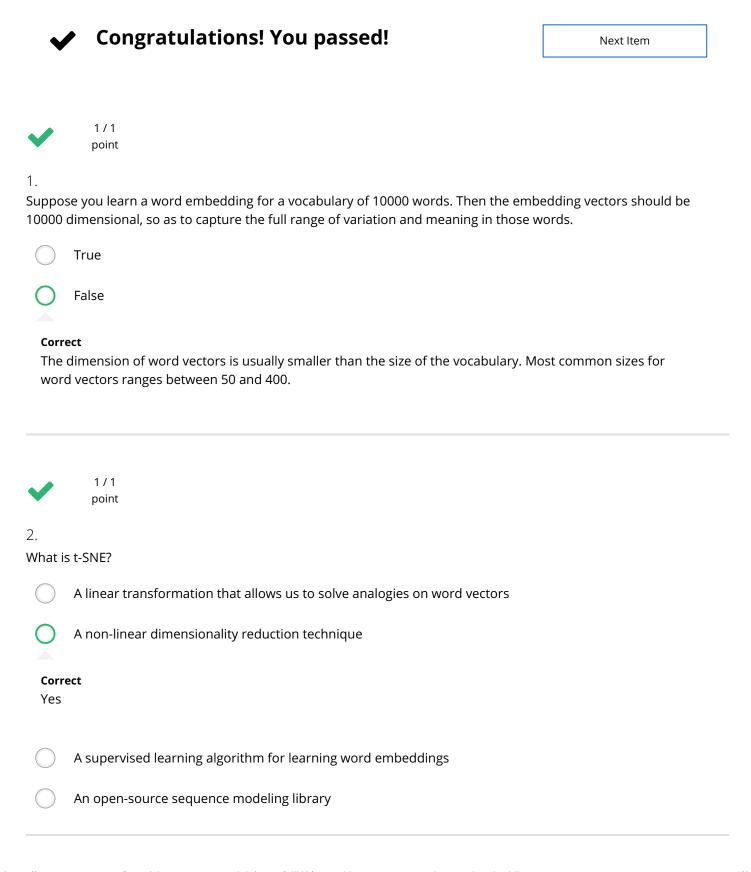
Quiz, 10 questions



Quiz, 10 questions

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.

_	

True

### Correct

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 classified the sentence as a "1".





1/1 point

4

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

$$e_{boy} - e_{girl} pprox e_{brother} - e_{sister}$$

### Correct

Yes!

$$oxed{egin{array}{c} e_{boy} - e_{girl} pprox e_{sister} - e_{brother} \end{array}}$$

**Un-selected is correct** 

Quiz, 10 questions **Correct** Yes!



$$e_{boy} - e_{brother} \approx e_{sister} - e_{girl}$$

**Un-selected is correct** 



1/1 point

5.

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



It is computationally wasteful.



#### Correct

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

- The correct formula is  $E^T * o_{1234}$ .
- This doesn't handle unknown words (<UNK>).
- None of the above: calling the Python snippet as described above is fine.



1/1 point

6.

When learning word embeddings, we create an artificial task of estimating  $P(target \mid context)$ . 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.



True

Correct

False

Quiz, 10 questions



1/1 point

7.

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

- igcap c is a sequence of several words immediately before t.
- igcup c and t are chosen to be nearby words.

Correct

- igcap c is the one word that comes immediately before t.
- c is the sequence of all the words in the sentence before t.



1/1 point

8

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

$$P(t \mid c) = rac{e^{ heta_t^T e_c}}{\sum_{t'=1}^{10000} e^{ heta_t^T e_c}}$$

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

 $\theta_t$  and  $e_c$  are both 500 dimensional vectors.

Correct

 $igcup_t$  and  $e_c$  are both 10000 dimensional vectors.

**Un-selected is correct** 

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

Correct

Natural Language Processing & Word Embeddings are the same word.

Quiz, 10 questions

Un-selected is correct



1/1 point

9.

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

$$\min \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij}) ( heta_i^T e_j + b_i + b_j' - log X_{ij})^2$$

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

hinspace hin

### **Un-selected is correct**

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

### Correct

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

## Correct

 $oxed{igcap}$  The weighting function f(.) must satisfy f(0)=0.

#### Correct

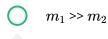
The weighting function helps prevent learning only from extremely common word pairs. It is not necessary that it satisfies this function.



1/1 point

10.

You have trained word embeddings using a text dataset of  $m_1$  words. You are considering using these word Native and Branch and Branch as  $m_2$  words. Keeping in mind that  $m_2$  words is a form of transfer learning, under which of these circumstance would you expect the word embeddings to be helpful?



Correct

