✓ Congratulations! You passed!

TO PASS 80% or higher



GRADE 80%

Natural Language Processing & Word Embeddings

LATEST SUBMISSION GRADE

80%

1.	Suppose you learn a word embedding for a vocabulary of 10000 words. Then the embedding vectors should be 10000
	discount and an extra section the full section of contests and section in the contests.

0 / 1 point

True

○ False

No, the dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors ranges between 50 and 400.

2. What is t-SNE?

1/1 point

- A linear transformation that allows us to solve analogies on word vectors
- A non-linear dimensionality reduction technique
- A supervised learning algorithm for learning word embeddings
- An open-source sequence modeling library



Suppose you download a pre-trained word embedding which has been trained on a huge corpus of text. You then use this 1/1 point 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

○ False



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".

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

1/1 point

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



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

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

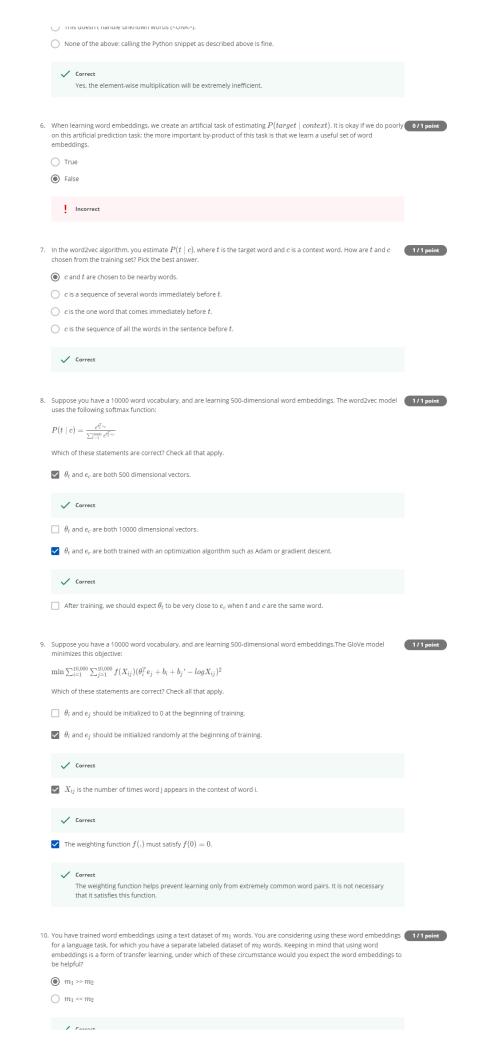
Yes!

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

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 1/1 point of word 1234, why don't we call $E \ast o_{1234}$ in Python?

It is computationally wasteful.

 $\bigcirc \ \ \, \text{ The correct formula is } E^T*o_{1234}.$



▼ correct