

# Natural Language Processing & Word Embeddings

10/10 分 (100%)

测验, 10 个问题

✓ **恭喜！您通过了！**

下一项



1 / 1 分

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.



True



False

正确



1 / 1 分

2。

What is t-SNE?



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



A non-linear dimensionality reduction technique

正确



A supervised learning algorithm for learning word embeddings



An open-source sequence modeling library

## Natural Language Processing & Word Embeddings

10/10 分 (100%)

测验, 10 个问题



1 / 1 分

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



正确



False



1 / 1 分

4。

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



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



正确

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

## Natural Language Processing & Word Embeddings

10/10 分 (100%)

测验, 10 个问题

未选择的是正确的

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

正确

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

未选择的是正确的



1 / 1 分

5。

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



It is computationally wasteful.

正确



The correct formula is  $E^T * e_{1234}$ .



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



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



1 / 1 分

## Natural Language Processing & Word Embeddings

测验, 10 个问题

10/10 分 (100%)

6。

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



正确



False



1 / 1 分

7。

In the word2vec algorithm, you estimate  $P(t | c)$ , 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.



$c$  and  $t$  are chosen to be nearby words.



正确



$c$  is the sequence of all the words in the sentence before  $t$ .



$c$  is the one word that comes immediately before  $t$ .



$c$  is a sequence of several words immediately before  $t$ .



1 / 1 分

## Natural Language Processing & Word Embeddings

测验, 10 个问题

10/10 分 (100%)

8。

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

$$P(t | c) = \frac{e^{\theta_t^T e_c}}{\sum_{t'=1}^{10000} e^{\theta_{t'}^T e_c}}$$

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



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



正确



$\theta_t$  and  $e_c$  are both 10000 dimensional vectors.



未选择的是正确的



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



正确



After training, we should expect  $\theta_t$  to be very close to  $e_c$  when  $t$  and  $c$  are the same word.



未选择的是正确的



1 / 1 分

9。

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

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

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



☐  $\theta_i$  and  $e_j$  should be initialized to 0 at the beginning of training.

## Natural Language Processing & Word Embeddings

10/10 分 (100%)

测验, 10 个问题

未选择的是正确的

☒  $\theta_i$  and  $e_j$  should be initialized randomly at the beginning of training.



正确

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



正确

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



正确

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



1 / 1 分

10。

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



$m_1 \gg m_2$



正确



$m_1 \ll m_2$

---

# Natural Language Processing & Word Embeddings

10/10 分 (100%)

测验, 10 个问题

---

