Congratulations! You passed!

Grade received 100% **To pass** 80% or higher

Natural Language Processing & Word Embeddings

Quiz • 30 min

Natural Language Processing & Word Embeddings

Latest Submission Grade 100%

Due Apr 18, 2:59 PM CST **Attempts** 3 every 8 hours

1. True/False: Suppose you learn a word embedding for a vocabulary of 60000 words. Then the embedding vectors could be 1000 dimensional, so as to capture the full range of variation and meaning in those words. **1 / 1 point**

- False
- True

Correct
The dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors range between 50 and 1000.

2. True/False: t-SNE is a non-linear dimensionality reduction technique. **1 / 1 point**

- True
- False

Correct
t-SNE is a non-linear dimensionality reduction technique.

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. **1 / 1 point**

x (input text)	y (happy?)
Having a great time!	1
I'm sad it's raining.	0
I'm feeling awesome!	1

Even if the word "wonderful" does not appear in your small training set, what label might be reasonably expected for the input text "I feel wonderful!"?

- y=1
- y=0

Correct
Yes, word vectors empower your model with an incredible ability to generalize. The vector for "wonderful" would contain a negative/unhappy connotation which will probably make your model classify 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**

- [x] $e_{boy} - e_{girl} \approx e_{brother} - e_{sister}$

Correct
Yes!

- [x] $e_{boy} - e_{brother} \approx e_{girl} - e_{sister}$

Correct
Yes!

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

5. Let A be an embedding matrix, and let o_{4567} be a one-hot vector corresponding to word 4567. Then to get the embedding of word 4567, why don't we call $A * o_{4567}$ in Python? **1 / 1 point**

- None of the answers are correct: calling the Python snippet as described above is fine.
- This doesn't handle unknown words (<UNK>).
- The correct formula is $A^T * o_{4567}$.
- It is computationally wasteful.

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

6. When learning word embeddings, words are automatically generated along with the surrounding words. **1 / 1 point**

- True
- False

Correct
We pick a given word and try to predict its surrounding words or vice versa.

7. True/False: In the word2vec algorithm, you estimate $P(t \mid c)$, where t is the target word and c is a context word. t and c are chosen from the training set to be nearby words. **1 / 1 point**

- True
- False

Correct
Yes, t and c are chosen from the training set to be nearby words.

8. Suppose you have a 10000 word vocabulary, and are learning 100-dimensional word embeddings. The word2vec model uses the following softmax function: **1 / 1 point**

$$P(t \mid c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

True/False: After training, we should expect to be very close to when t and c are the same word.

- True
- False

Correct
To review this concept watch the Word2Vec lecture.

9. Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings.The GloVe model minimizes this objective: **1 / 1 point**

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

True/False: θ_i and e_j should be initialized to 0 at the beginning of training.

- False
- True

Correct
θ_i and e_j should be initialized randomly at the beginning of training.

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 circumstances would you expect the word embeddings to be helpful? **1 / 1 point**

- $m_1 << m_2$
- $m_1 >> m_2$

Correct