

✔ **Congratulations! You passed!**

Grade
received **80%**

Latest Submission
Grade 80%

To pass 80% or
higher

[Go to next item](#)

1. True/False: Suppose you learn a word embedding for a vocabulary of 20000 words. Then the embedding vectors could be 1000 dimensional, so as to capture the full range of variation and meaning in those words.

0 / 1 point

☐ True

☒ False

[Expand](#)

✘ **Incorrect**

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. What is t-SNE?

1 / 1 point

- ☐ An open-source sequence modeling library
- ☐ 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

[Expand](#)

✔ **Correct**

Yes

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?)
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

Expand

✓ 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 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

☒ $e_{man} - e_{king} \approx e_{woman} - e_{queen}$

✓ Correct

The order of words is correct in this analogy.

☐ $e_{man} - e_{king} \approx e_{queen} - e_{woman}$

☒ $e_{man} - e_{woman} \approx e_{king} - e_{queen}$

✓ Correct

The order of words is correct in this analogy.

Typesetting math: 100% $e_{queen} - e_{king}$

Expand

✓ Correct

Great, you got all the right answers.

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.

☒ It is computationally wasteful.

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

☐ The correct formula is $A^T * o_{4567}$

Expand

✓ Correct

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

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

1 / 1 point

☐ False

☒ True

Expand

✓ Correct

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.

1 / 1 point

- ☒ c and t are chosen to be nearby words.
- ☐ c is a sequence of several words immediately before t
- ☐ c is the sequence of all the words in the sentence before t
- ☐ c is the one word that comes immediately before t

Typesetting math: 100%

[Expand](#)

✓ Correct

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

1 / 1 point

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

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

- ☒ θ_t and e_c are both 500 dimensional vectors.

✓ Correct

- ☐ After training, we should expect θ_t to be very close to e_c when t and c are the same word.

- ☒ θ_t and e_c are both trained with an optimization algorithm such as Adam or gradient descent.

✓ Correct

- ☐ θ_t
- and

[Expand](#)

✓ Correct

Great, you got all the right answers.

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

0 / 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.

- ☒ True
- ☐ False

[Expand](#)

✗ Incorrect

No, 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 s_1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of s_2 words. Keeping in mind that using word embeddings is a form of transfer learning, under which of these circumstances would you expect the

1 / 1 point

word embeddings to be helpful?

☒ $s_1 \gg s_2$

☐ $s_1 \ll s_2$

Typesetting math: 100%

[↗ Expand](#)

✓ **Correct**

s_1 should transfer to s_2