Natural Language Processing & Word Embeddings

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

- False
- True

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. True/False: t-SNE is a linear transformation that allows us to solve analogies on word vectors.

1 / 1 point

- False
- True

Expand

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

3.
Question 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 that my cat is ill.	0
Really enjoying this!	1

True/False: Then even if the word "upset" does not appear in your small training set, your RNN might reasonably be expected to recognize "I'm upset" as deserving a label y = 0.

- (•) True
- ( ) False

Expand

Correct
Yes, word vectors empower your model with an incredible ability to generalize. The vector for "upset" would contain a negative/unhappy connotation which will probably make your model classify the sentence as a "0".

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

1 / 1 point

- [x] $e_{man} - e_{king} \approx e_{woman} - e_{queen}$

  ✓ Correct
  The order of words is correct in this analogy.

- [x] $e_{man} - e_{woman} \approx e_{king} - e_{queen}$

  ✓ Correct
  The order of words is correct in this analogy.

- [ ] $e_{man} - e_{king} \approx e_{queen} - e_{woman}$
- [ ] $e_{man} - e_{woman} \approx e_{queen} - e_{king}$

Expand

Correct
Great, you got all the right answers.

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?

1 / 1 point

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

Expand

✓ 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

- ◉ False
- ○ True

Expand

✓ 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 | c)*, where *t* is the target word and *c* is a context word. *t* and *c* are chosen from the training set using *c* as the sequence of all the words in the sentence before *t*.

1 / 1 point

- ○ True
- ◉ False

Expand

✓ Correct
and are chosen from the training set to be nearby words.

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

0 / 1 point

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

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

- [ ] θ_t and e_c are both trained with an optimization algorithm such as Adam or gradient descent.
- [ ] 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 10000 dimensional vectors.
- [x] θ_t and e_c are both 500 dimensional vectors.

✓ Correct

Expand

Incorrect
You didn't select all the correct 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

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

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 t_1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of t_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?

- ○ When t_1 is smaller than t_2
- ○ When t_1 is equal to t_2
- ◉ When t_1 is larger than t_2

Expand

Correct
Transfer embeddings to new tasks with smaller training sets.

1 / 1 point

1. Suppose you learn a word embedding for a vocabulary of 10000 words. Then the embedding vectors could be 10000 dimensional, so as to capture the full range of variation and meaning in those words.

1 / 1 point

- ○ True
- ◉ False

Expand

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

1 / 1 point

- False
- True

Expand

Correct
tr-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=0
- y=1

Expand

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!

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

Correct

Yes!

Expand

Correct

Great, you got all the right answers.

5. True/False: The most computationally efficient formula for Python to get the embedding of word 1021, if C is an embedding matrix, and o_{1021} is a one-hot vector corresponding to word 1021, is $C^T * o_{1021}$.

0 / 1 point

- (•) True
- ( ) False

Expand

Incorrect

No, it is computationally wasteful because the element-wise multiplication will be extremely inefficient.

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.

1 / 1 point

- (•) True
- ( ) False

Expand

Correct

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.

1 / 1 point

- (•) c and t are chosen to be nearby words.
- ( ) 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

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

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

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

- [x] θ_t and e_c are both 500 dimensional vectors.

✓ **Correct**

- [ ] θ_t and e_c are both 10000 dimensional vectors.
- [x] θ_t and e_c are both trained with an optimization algorithm such as Adam or gradient descent.

✓ **Correct**

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

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

1 / 1 point

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

True/False: X_{ij} is the number of times word j appears in the context of word i.

- (•) True
- ( ) False

Expand

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

10. You have trained word embeddings using a text dataset of t_1 words. You are considering using these word embeddings for a language task, for which you have a separate labeled dataset of t_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

- ○ When t_1 is equal to t_2
- ◉ When t_1 is larger than t_2
- ○ When t_1 is smaller than t_2

Expand

Correct

Transfer embeddings to new tasks with smaller training sets.