## Natural Language Processing & Word Embeddings Quiz, 10 questions

3.

1 point
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
1 point
2. What is t-SNE?
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
1 point

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

False

1 point

4

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

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

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

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

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

1 point

5.

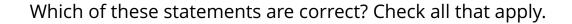
Let E be an embedding matrix, and let  $o_{1234}$  be a one-hot vector corresponding to word 1234. Then to get the embedding of word 1234, why don't we call  $E*o_{1234}$  in Python?

	It is computationally wasteful.
	The correct formula is $E^Tst o_{1234}.$
	This doesn't handle unknown words ( <unk>).</unk>
	None of the above: calling the Python snippet as described above is fine.
1 point	
we do p	earning word embeddings, we create an artificial task of estimating $P(target \mid context)$ . It is okay if boorly on this artificial prediction task; the more important by-product of this task is that we learn a set of word embeddings.
	True
	False
1 point	
	word2vec algorithm, you estimate $P(t\mid c)$ , where $t$ is the target word and $c$ is a context word. How and $c$ chosen from the training set? Pick the best answer.
	c and $t$ are chosen to be nearby words.
	c is the sequence of all the words in the sentence before $t.$
	c is a sequence of several words immediately before $t.$
	c is the one word that comes immediately before $t.$
1	
point	

8.

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

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



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

lacksquare  $heta_t$  and  $e_c$  are both 10000 dimensional vectors.

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

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

1 point

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.

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

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

The weighting function f(.) must satisfy f(0)=0.

1 point

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?
$\bigcirc m_1 >> m_2$
$\bigcirc m_1 \lessdot m_2$
Upgrade to submit

You have trained word embeddings using a text dataset of  $m_1$  words. You are considering using these word



