Natural Language Processing & Word Embeddings

10/10 points (100%)

Quiz, 10 questions

Congratulations! You passed!	Next Item
1/1	
points	
1. Suppose you learn a word embedding for a vocabulary of 100 the embedding vectors should be 10000 dimensional, so as to full range of variation and meaning in those words.	
True	
False	
Correct	
1/1 points	
2.	
What is t-SNE?	
A linear transformation that allows us to solve analogi vectors	es on word
A non-linear dimensionality reduction technique	
Correct	
A supervised learning algorithm for learning word eml	oeddings
An open-source sequence modeling library	

Natural Language Processing & Word Embeddings

10/10 points (100%)

Quiz, 10 questions



points

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.

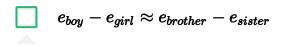
0	True	
Corr	ect	

False



4.

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



Correct

 $oxed{egin{array}{c} e_{boy} - e_{girl} pprox e_{sister} - e_{brother} \end{array}}$

Natural Language Processing & Word Embeddings

10/10 points (100%)

Quiz, 10 questions

$e_{boy}-e_{brother}$	≈	e_{girl}	_	e_{sister}

Correct



Un-selected is correct

Un-selected is correct



1/1 points

5.

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



It is computationally wasteful.

Correct

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

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

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



1/1 points

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 Natural Language Processings by Word Fundedings learn a useful seto/10 points (100%) of word embeddings.

	True
Corr	rect
0	False
~	1 / 1 points
	word2vec algorithm, you estimate $P\left(t\mid c ight)$, where t is the target word
	is a context word. How are $m{t}$ and $m{c}$ chosen from the training set? Pick est answer.
	$oldsymbol{c}$ and $oldsymbol{t}$ are chosen to be nearby words.
the be	$oldsymbol{c}$ and $oldsymbol{t}$ are chosen to be nearby words.
the be	$oldsymbol{c}$ and $oldsymbol{t}$ are chosen to be nearby words.
the be	$m{c}$ and $m{t}$ are chosen to be nearby words. rect $m{c}$ is the one word that comes immediately before $m{t}$.
the be	$m{c}$ and $m{t}$ are chosen to be nearby words. ect $m{c}$ is the one word that comes immediately before $m{t}$. $m{c}$ is a sequence of several words immediately before $m{t}$.

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

Natural Language Processing & Word Embeddings

10/10 points (100%)

Quiz, 10 questions

$$P\left(t\mid c
ight) = rac{e^{ heta_{t}^{T}e_{c}}}{\sum_{t'=1}^{10000}e^{ heta_{t'}^{T}e_{c}}}$$

Which of these statements are correct? Check all that apply. $heta_t$ and e_c are both 500 dimensional vectors. Correct $heta_t$ and $extbf{\emph{e}}_c$ are both 10000 dimensional vectors. **Un-selected is correct** $heta_t$ and e_c are both trained with an optimization algorithm such as Adam or gradient descent. Correct After training, we should expect $heta_t$ to be very close to e_c when tand c are the same word. **Un-selected is correct**



1/1 points

9.

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

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

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

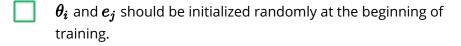
 θ_i and e_i should be initialized to 0 at the beginning of training.

Un-selected is correct

Natural Language Processing & Word Embeddings

10/10 points (100%)

Quiz, 10 questions



Correct

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

Correct

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

Correct

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



1/1 points

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?



 $m_1 >> m_2$

Correct

 $m_1 \ll m_2$