Quiz, 10 questions

Congratulations! You passed! Next Item 1/1 points 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** The dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors ranges between 50 and 400. 1/1 points 2. What is t-SNE? A linear transformation that allows us to solve analogies on word vectors A non-linear dimensionality reduction technique Correct Yes A supervised learning algorithm for learning word embeddings

Natural Language Processing & Word Embeddings

10/10 points (100.00%)

Quiz, 10 questions



1/1 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.



True

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 classified the sentence as a "1".





1/1 points

4

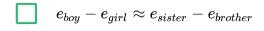
Which of these equations do you think should hold for a good word embedding? (Check all that apply)

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

Natural Language Processing & Word Embeddings

10/10 points (100.00%)

Quiz, 10 questions



Un-selected is correct



Correct

Yes!



Un-selected is correct



1/1 points

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.

Correct

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

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

Natural Language Processing & Word Embeddings When learning word embeddings, we create an artificial task of estimating

10/10 points (100.00%)

Quiz, 10 questions

 $P(target \mid context)$. 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.

| 0 | True |
|----------|--|
| Corre | ect |
| | False |
| ~ | 1 / 1 points |
| | word2vec algorithm, you estimate $P(t\mid c)$, where t is the target word and ontext word. How are t and c chosen from the training set? Pick the best r. |
| | c is a sequence of several words immediately before $t.$ |
| | c is the sequence of all the words in the sentence before $t.$ |
| 0 | \emph{c} and \emph{t} are chosen to be nearby words. |
| Corr | ect |
| | \emph{c} is the one word that comes immediately before \emph{t} . |
| | |

1/1 points

8.

Natural Language Processing & Word Embeddings

10/10 points (100.00%)

Quiz, 10 questions

$$P(t \mid c) = rac{e^{ heta_t^T e_c}}{\sum_{t'=1}^{10000} e^{ heta_t^T e_c}}$$

Which of these statements are correct? Check all that apply. $heta_t$ and e_c are both 500 dimensional vectors. Correct $heta_t$ and 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 t and care the same word. Un-selected is correct 1/1 points 9.

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

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

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 Lan Quiz, 10 questions | guag corre | $	heta_i$ and e_j should be initialized randomly at the beginning of training. ${f e}$ Processing ${f \&}$ Word Embeddings | 10/10 |
|--------------------------------|-----------------------------|--|--------|
| | Corre | X_{ij} is the number of times word i appears in the context of word j. | |
| | | The weighting function $f(.)$ must satisfy $f(0)=0.$ Lect weighting function helps prevent learning only from extremely mon word pairs. It is not necessary that it satisfies this function. | |
| | conside a separ embed | 1 / 1 points ve trained word embeddings using a text dataset of m_1 words. You are ring using these word embeddings for a language task, for which you rate labeled dataset of m_2 words. Keeping in mind that using word dings is a form of transfer learning, under which of these circumstance you expect the word embeddings to be helpful? $m_1 >> m_2$ | ı have |

points (100.00%)





 $m_1 \ll m_2$