Natural Language Processing & Word Embeddings

| 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 ✓ 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 400. What is t-SNE? ♠ A non-linear dimensionality reduction technique ♠ A supervised learning algorithm for learning word embeddings ✓ Correct Yes 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?) True True False Y (happy?) Which of these equations do you think should hold for a good word embedding? (Check all that apply) Glosp - Gurster S Glord - Guister S |
|---|
| dimensional, so as to capture the full range of variation and meaning in those words. ↑ True ↑ False ✓ 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 400. What is t-SNE? ♠ A non-linear dimensionality reduction technique ♠ A linear transformation that allows us to solve analogies on word vectors ♠ An open-source sequence modeling library ♠ supervised learning algorithm for learning word embeddings ✓ Correct Yes 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) x (input text) y (happy?) I'm bummed my cat is iii. Really enjoying this! Then even if the word "ecstatic" does not appear in your small training set, your RNN might reasonably be expected to recognize "me estatic" as deserving a label y = 1. ↑ True ↑ True ↑ True ↑ False ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) Glog - Caracture ≈ Gaider - Ggirl Capture ≈ Gaider - Ggirl Correct Yes |
| The dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors range between 50 and 400. What is t-SNE? A non-linear dimensionality reduction technique A linear transformation that allows us to solve analogies on word vectors An open-source sequence modeling library A supervised learning algorithm for learning word embeddings ✓ Correct Yes 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/ I'm bearing dry act iill. 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 "Tm ecstatic" as deserving a label y = 1. True False ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) □ ⟨Elong → elondher ≈ ⟨Esister → egirl → Esister ✓ Correct Yes! |
| The dimension of word vectors is usually smaller than the size of the vocabulary. Most common sizes for word vectors range between 50 and 400. What is t-SNE? A non-linear dimensionality reduction technique A linear transformation that allows us to solve analogies on word vectors An open-source sequence modeling library A supervised learning algorithm for learning word embeddings Correct Yes 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) I'm feeling wonderful today! I'm bummed my cat is ill. 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 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) close = Glerother ≈ Guister = Ggirl = Guister Correct Yes! |
| A non-linear dimensionality reduction technique A linear transformation that allows us to solve analogies on word vectors An open-source sequence modeling library A supervised learning algorithm for learning word embeddings ✓ Correct Yes 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! 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 ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) □ choy - chrother ≈ csister - cpirl Choy - chrother ≈ csister - csister Yes! |
| A non-linear dimensionality reduction technique A linear transformation that allows us to solve analogies on word vectors An open-source sequence modeling library A supervised learning algorithm for learning word embeddings ✓ Correct Yes 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! 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 ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) □ choy - chrother ≈ csister - cpirl Choy - chrother ≈ csister - csister Yes! |
| A linear transformation that allows us to solve analogies on word vectors An open-source sequence modeling library A supervised learning algorithm for learning word embeddings ✓ Correct Yes 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 ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) □ clogy − clorother: ≈ csister − cgirl ✓ correct Yes! |
| A linear transformation that allows us to solve analogies on word vectors An open-source sequence modeling library A supervised learning algorithm for learning word embeddings ✓ correct Yes 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 ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) cloy - elrother ≈ esister - esister ✓ correct Yes, Correct Yes, ord elrother ≈ esister - esister |
| An open-source sequence modeling library A supervised learning algorithm for learning word embeddings ✓ Correct Yes 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! I'm bummed my cat is ill. 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 ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) chap ← chrother ≈ esister − egirl chap ← chrother ≈ esister − egirl chap ← chrother ≈ esister − esister ✓ Correct Yes! |
| A supervised learning algorithm for learning word embeddings ✓ Correct Yes 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! I'm bummed my cat is ill. 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 ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) cloy ← chrother ≈ esister − egirl cloy ← chrother ≈ esister − egirl cloy ← chrother ≈ esister − egirl cloy ← chrother ≈ esister − esister ✓ Correct Yes! |
| 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. |
| 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. |
| 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. |
| 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. |
| 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. |
| I'm feeling wonderful today! I'm bummed my cat is ill. 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 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) $e_{looy} - e_{brother} \approx e_{sister} - e_{girl}$ $e_{looy} - e_{brother} \approx e_{sister} - e_{sister}$ Correct Yes! |
| I'm beeling wonderful today! I'm bummed my cat is ill. 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 "Tm ecstatic" as deserving a label $y=1$. True False 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) $e_{logy} - e_{brother} \approx e_{sister} - e_{girl}$ $e_{logy} - e_{brother} \approx e_{sister} - e_{sister}$ Correct Yes! |
| The neven if the word "ecstatic" does not appear in your small training set, your RNN might reasonably be expected to recognize "Tm ecstatic" as deserving a label $y=1$. True False 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) $e_{logy} - e_{brother} \approx e_{sister} - e_{girl}$ $e_{logy} - e_{brother} \approx e_{sister} - e_{sister}$ Correct Yes! |
| 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 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) $c_{boy} - c_{brother} \approx c_{sister} - c_{girl}$ $c_{boy} - c_{brother} \approx c_{sister} - c_{sister}$ Correct Yes! |
| recognize "I'm ecstatic" as deserving a label $y=1$. True False 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) $e_{looy} - e_{brother} \approx e_{sister} - e_{girl}$ $e_{looy} - e_{brother} \approx e_{girl} - e_{sister}$ Correct Yes! |
| ✓ 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) $c_{boy} - c_{brother} \approx c_{sister} - c_{girl}$ ✓ $c_{boy} - c_{brother} \approx c_{girl} - c_{sister}$ ✓ $Correct$ Yes! |
| 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". Which of these equations do you think should hold for a good word embedding? (Check all that apply) $e_{bog} - e_{brother} \approx e_{sister} - e_{girl}$ \checkmark Correct Yes! |
| |
| $igsqcup e_{boy} - e_{brother} pprox e_{sister} - e_{girl}$ $igsqcup e_{boy} - e_{brother} pprox e_{girl} - e_{sister}$ $igsqcup Correct$ Yes! |
| $ \mathbf{v} \in \epsilon_{boy} - \epsilon_{brother} \approx \epsilon_{girl} - \epsilon_{sister} $ $ \mathbf{v} \in \mathbf{v} \in \mathbf{v} $ Ves! |
| ✓ Correct Yes! |
| Yes! |
| $ ightharpoons e_{boy} - e_{girl} pprox e_{brother} - e_{sister}$ |
| |
| ✓ Correct Yes! |
| $igcup_{eboy} - e_{girl} pprox e_{sister} - e_{brother}$ |

| 5. Let E be an embedding matrix, and let o_{1234} be a one-hot vector corresponding to word 1234. Then to get the emb of word 1234, why don't we call $E*o_{1234}$ in Python? | edding 1/1 point |
|---|------------------|
| It is computationally wasteful. | |
| $igcap$ The correct formula is $E^T*o_{1234}.$ | |
| None of the above: calling the Python snippet as described above is fine. | |
| This doesn't handle unknown words (<unk>).</unk> | |
| Correct Yes, the element-wise multiplication will be extremely inefficient. | |
| 6. When learning word embeddings, we create an artificial task of estimating P(target context). It is okay if we do on this artificial prediction task; the more important by-product of this task is that we learn a useful set of word embeddings. True | poorly 1/1 point |
| O False | |
| ✓ 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 chosen from the training set? Pick the best answer. | C 1/1 point |
| c is the one word that comes immediately before t. | |
| c and t are chosen to be nearby words. | |
| c is a sequence of several words immediately before t. | |
| \bigcirc c is the sequence of all the words in the sentence before t . | |
| ✓ Correct | |
| | |
| Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings. The word2vec mouses the following softmax function: | odel 1/1 point |
| $P(t\mid c)=rac{e^{q^2_{T_c}}}{\sum_{i=0}^{M_c}e^{q^2_{T_c}c_i}}$ Which of these statements are correct? Check all that apply. | |
| | |
| $m{arphi}$ $	heta_t$ and e_c are both trained with an optimization algorithm such as Adam or gradient descent. | |
| ✓ Correct | |
| $ ightharpoonsdef{ } 	heta_{ m t}$ and $e_{ m c}$ are both 500 dimensional vectors. | |
| ✓ Correct | |
| $\hfill \Box$ After training, we should expect θ_t to be very close to e_c when t and c are the same word. | |
| $\hfill \theta_t$ and e_c are both 10000 dimensional vectors. | |
| Suppose you have a 10000 word vocabulary, and are learning 500-dimensional word embeddings. The GloVe model minimizes this objective: | 1/1 point |
| $\min \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij}) (heta_i^T e_j + b_i + b_j' - log X_{ij})^2$ | |
| Which of these statements are correct? Check all that apply. | |
| $ ightharpoons X_{ij}$ is the number of times word j appears in the context of word i. | |
| ✓ Correct | |
| $\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $ | |
| $\ensuremath{ \swarrow} \ensuremath{ \theta_i}$ and e_j should be initialized randomly at the beginning of training. | |