Deep Learning for NLP 2020
Exercise 04 Solution

May 19, 2020

1 Pingo

- In neural networks, the choice between different pretrained embeddings can be considered like a hyperparameter.
  - ☑ True
  - □ False
- Which of the following approaches for word representations are extrinsic methods?
  - ☑ Testing as part of named entity recognition
  - □ Calculating vector distances for similar words
  - □ Evaluation by humans, for example by word analogy
  - ☑ Performance comparison of two word representations trained within the same model
- Which of the following statements on the skip-gram model are correct?
  - ☑ It is used to learn a lower-dimensional representation for words
  - □ It attempts to infer the missing word from its context
  - ☑ It attempts to infer the missing context from a word
  - ☑ It is implemented in word2vec
  - □ It learns exactly one matrix
- Which of the following statements apply on negative sampling?
  - □ Frequent words are omitted in the calculation
  - ☑ Random combinations for contexts are used as negative training examples
  - □ It is applied to reduce the number of trainable parameters of the skip-gram model
  - ☑ It is applied because a full computation of all probabilities in skip-gram is infeasible

2 Word Embeddings

1. What is unsupervised pre-training and why is it desirable?

   Answer: What is it?

   unsupervised: learning a (word) representation from unlabeled data

   pre-training: representation is trained in a separate step, can afterwards be used for target tasks

   Why is it desirable?

   - for having reusable, meaningfully initialized representations for multiple target tasks → might lower amount of training data necessary for target tasks, improve convergence, etc.

2. Vectorized representations can be trained for many applications and on all kinds of different data. State at least three different kinds of embeddings that one could train.

   Answer:

   - what is being represented: words, characters, sentences, documents, ...
   - languages: monolingual (, bilingual, multilingual) → more on this in next week's lecture
   - data source / domain: newswire, social media, recipes, ...
   - time span: using texts from last year, 15th century, ancient text, ...
   - focused on: syntax, morphology, semantics

3 Tokenization

Tokenization is the NLP task of splitting a sentence (or an arbitrary character sequence) into useful parts, called tokens.

1. A fellow student of yours uses `str.split(" ")` to tokenize his/her input sentences, which come from children's novels. How would you convince him/her that this is a bad idea?

   Answer: Consider the sentence `That's $3.25, but you're broke.`:

   - contractions are mishandled (`That's` should be tokenized as `That 's`, etc.)
   - punctuation is mishandled (`broke.` should be tokenized as `broke .`, etc.)
   - currencies are mishandled
   - ...

   The choice of a tokenizer depends on the input data. In most cases, one should use a "proper" tokenizer (such as the one included in NLTK or the Stanford tokenizer) to correctly handle the above cases. `str.split(" ")` can be appropriate in some cases, but one needs to be aware of its consequences.

2. A neural network of yours should classify sentences. You decide to use pretrained embeddings downloaded from the internet (say, word2vec). How would you preprocess your input sentences in order to reach the best possible performance? Explain in up to three sentences.

   Answer: One should use the same tokenization strategy that was used to create the vocabulary of the pretrained embeddings. If the tokenization strategy is different, one runs into the risk of having a lot more out-of-vocabulary tokens which hurts performance. Disregarding the casing of the embeddings is also a major source for trouble.

   More comments: Unfortunately, tokenization strategies are rarely mentioned in publications about word embeddings. If in doubt, `grep` the embedding vocabulary for words like `don't` or `it's`.