Does Doc2Vec learn representations for the tags?

Asked 4 years, 8 months ago Active 4 years, 7 months ago Viewed 5k times

6

I'm using the Doc2Vec tags as an unique identifier for my documents, each document has a different tag and no semantic meaning. I'm using the tags to find specific documents so I can calculate the similarity between them.

Do the tags influence the results of my model?

In this tutorial they talk about a parameter `train_lbls=false`, with this set to false there are no representations learned for the labels (tags).

That tutorial is somewhat dated and I guess the parameter does no longer exist, how does Doc2Vec handle tags?

gensim doc2vec

Share Improve this question Follow

edited Apr 24 '17 at 7:32

asked Apr 21 '17 at 13:16

Stanko

3,497 ● 3 ● 17 ● 46

1 Answer

Active | Oldest | Votes

16

For gensim's Doc2Vec, your text examples must be objects similar to the example `TaggedDocument` class: with `words` and `tags` properties. The `tags` property should be a list of 'tags', which serve as keys to the doc-vectors that will be learned from the corresponding text.

In the classic/original case, each document has a single tag – essentially a unique ID for that one document. (Tags can be strings, but for very large corpuses, Doc2Vec will use somewhat less memory if you instead use tags that are plain Python ints, starting from 0, with no skipped values.)

The tags are used to look-up the learned vectors after training. If you had a document during training with the single tag `'mars'`, you'd look-up the learned vector with:

```
model.docvecs['mars']
```

If you were do a `model.docvecs.most_similar['mars']` call, the results will be reported by their tag keys, as well.

The tags are *just* keys into the doc-vectors collection – they have no semantic meaning, and even if a string is repeated from the word-tokens in the text, there's no necessary relation between this tag key and the word.

That is, if you have a document whose single ID tag is 'mars', there's no essential relationship between the learned doc-vector accessed via that key (`model.docvecs['mars']`), and any learned word-vector accessed with the same string key (`model.wv['mars']`) – they're coming from separate collections-of-vectors.

Share Improve this answer Follow

answered May 16 '17 at 5:30

gojomo

45.1k ● 12 ● 79 ● 102