Lenning_JM6

October 7, 2021

1 Homework 6

1.1 1

[ ]:
```
## Import fasttext and train on cooking data

## cooking data comes from fasttext tutorial

import fasttext

model = fasttext.train_supervised(input="../data/cooking.train")
```

```
Read 0M words
Number of words:  8952
Number of labels: 735
Progress: 100.0% words/sec/thread:   79687 lr:  0.000000 avg.loss: 10.032363
ETA:   0h 0m 0s
```

[ ]:
```
## Predict some sentences
print(model.predict("Which baking dish is best to bake a banana bread ?"))
print(model.predict("Why not put knives in the dishwasher?"))
```

```
(('__label__bread',), array([0.18973885]))
(('__label__food-safety',), array([0.09966777]))
```

[ ]:
```
## Test model

model.test("../data/cooking.valid", k=5)
```

[ ]:
```
(3000, 0.07186666666666666, 0.15539858728557013)
```

1.2 2

[ ]:
```
## sklearn has a cosine_similarity that is easier to get working than gensim in␣
 ↪my opinion

from sklearn.metrics.pairwise import cosine_similarity

## save word vectors as variables
asparagus = model.get_word_vector('asparagus')
artichoke = model.get_word_vector('artichoke')
spatula = model.get_word_vector('spatula')
baking = model.get_word_vector('baking')
cooking = model.get_word_vector('cooking')


# print comparisons of words
print("asparagus/artichoke distance:", cosine_similarity([asparagus],␣
 ↪[artichoke]))
print("asparagus/spatula distance:", cosine_similarity([asparagus], [spatula]))
print("artichoke/baking distance:", cosine_similarity([artichoke], [baking]))
print("baking/cooking distance:", cosine_similarity([baking], [cooking]))

# get neareset neighbors for asparagus
print("asparagus nearest neighbors:", model.get_nearest_neighbors('asparagus'))
```

```
asparagus/artichoke distance: [[0.83391684]]
asparagus/spatula distance: [[-0.88749397]]
artichoke/baking distance: [[-0.7999045]]
baking/cooking distance: [[0.3023894]]
asparagus nearest neighbors: [(0.9712576270103455, 'lunchboxes'),
(0.9701305031776428, 'distance'), (0.9701305031776428, 'mints'),
(0.9700727462768555, 'marker'), (0.9700727462768555, 'writing'),
(0.9700527191162109, 'content"'), (0.9700527191162109, '"not'),
(0.9696813225746155, 'mcdonald'), (0.9696146845817566, 'unwise'),
(0.9695428609848022, 'fertilizer')]
```

1.3 3 Fasttext trained on the complete works of Jane Austen

data found at: https://www.gutenberg.org/ebooks/31100

[ ]:
```
## trains fasttext on complete works of jane austen
model = fasttext.train_unsupervised(input="../data/31100/31100.txt")

## print out first 10 words of model
print(model.words[:10])
```

```
Read 0M words
Number of words:  9722
Number of labels: 0

['</s>', 'the', 'to', 'of', 'and', 'a', 'I', 'in', 'her', 'was']

Progress: 100.0% words/sec/thread:  118629 lr:  0.000000 avg.loss:  2.486263
ETA:   0h 0m 0s
```

[ ]:
```
## create new word vectors for words
home = model.get_word_vector('home')
estate = model.get_word_vector('estate')

marriage = model.get_word_vector('marriage')
happy = model.get_word_vector('happy')

widow = model.get_word_vector('widow')
wife = model.get_word_vector('wife')

sister = model.get_word_vector('sister')
family = model.get_word_vector('family')

soldier = model.get_word_vector('soldier')
gentleman = model.get_word_vector('gentleman')


## calculate distance between two words
print("home/estate distance:", cosine_similarity([home], [estate]))
print("marriage/happy distance:", cosine_similarity([marriage], [happy]))
print("widow/wife distance:", cosine_similarity([widow], [wife]))
print("sister/family distance:", cosine_similarity([sister], [family]))
print("soldier/gentleman distance:", cosine_similarity([soldier], [gentleman]))

## find nearest neighbors to marriage
print("marriage nearest neighbors:", model.get_nearest_neighbors('marriage'))

## finish analogy
print("life is to death, as wife is to {}".format(model.get_analogies("life",␣
 ↪"death", "wife")[0]))
```

```
home/estate distance: [[0.41620523]]
marriage/happy distance: [[0.6293156]]
widow/wife distance: [[0.7647355]]
sister/family distance: [[0.47695524]]
soldier/gentleman distance: [[0.6093536]]
marriage nearest neighbors: [(0.9746325612068176, 'marriage;'),
(0.9561651349067688, 'marriage.'), (0.9505626559257507, 'marriage,'),
(0.8580401539802551, 'Marriage'), (0.8474940657615662, 'Carriage'),
(0.8399662971496582, 'accident'), (0.8366764783859253, 'fame'),
```

```
(0.8355134129524231, 'hit'), (0.8335829973220825, 'proposals'),
(0.8305620551109314, 'management')]
life is to death, as wife is to (0.7963750958442688, 'marry,')
```

In retrospect it seems from marriage's nearest neighbors that more preprocessing work needs to be done to separate punctuation and capitalization, but it is interesting to note the following words and to see how Jane Austen purveys marriage in her writing. It is also interesting to see that for Jane Austen, the antonym of life, death, is mirrored closely by the opposite of wife, marry, apparently. That is a bit ironic, and quite funny.