Discussion Forums

Week 4

SUBFORUMS

All

Assignment: Assignment 4 Submission

Week 4

Week 4 Notebook Provided Here

Uwe F Mayer Mentor Week 4 · 2 months ago

There isn't a provided one. I assembled a workbook when I took the class from the material presented. Here's the code, each block is a cell.

```
import re
import pandas as pd
import numpy as np
import nltk
from nltk.corpus import wordnet as wn

# Use path length in wordnet to find word similarity
# find sense of words via synonym set
# n=noun, 01=synonym set for first meaning of the word
deer = wn.synset('deer.n.01')
deer

elk = wn.synset('elk.n.01')
deer.path_similarity(elk)

horse = wn.synset('horse.n.01')
deer.path_similarity(horse)

# Use an information criteria to find word similarity
from nltk.corpus import wordnet_ic
brown_ic = wordnet_ic.ic('ic-brown.dat')
deer.lin_similarity(elk, brown_ic)

deer.lin_similarity(horse, brown_ic)

# Use NLTK Collocation and Association Measures
from nltk.collocations import *
# load some text for examples
from nltk.book import *
# text1 is the book "Moby Dick"
# extract just the words without numbers and sentence marks and make them lower case
text = [w.lower() for w in list(text1) if w.isalpha()]

bigram_measures = nltk.collocations.BigramAssocMeasures()
finder = BigramCollocationFinder.from_words(text)
finder.nbest(bigram_measures.pmi,10)

# find all the bigrams with occurrence of at least 10, this modifies our "finder" object
finder.apply_freq_filter(10)
finder.nbest(bigram_measures.pmi,10)

# Working with Latent Dirichlet Allocation (LDA) in Python
# Several packages available, such as gensim and lda. Text needs to be
# preprocessed: tokenizing, normalizing such as lower-casing, stopword
# removal, stemming, and then transforming into a (sparse) matrix for
# word (bigram, etc) occurences.
# generate a set of preprocessed documents
from nltk.stem.porter import PorterStemmer
from nltk.corpus import stopwords
from nltk.book import *

len(stopwords.words('english'))

stopwords.words('english')

# extract just the stemmed words without numbers and sentence marks and make them lower case
p_stemmer = PorterStemmer()
sw = stopwords.words('english')
doc1 = [p_stemmer.stem(w.lower()) for w in list(text1) if w.isalpha() and not w.lower() in sw]
doc2 = [p_stemmer.stem(w.lower()) for w in list(text2) if w.isalpha() and not w.lower() in sw]
doc3 = [p_stemmer.stem(w.lower()) for w in list(text3) if w.isalpha() and not w.lower() in sw]
doc4 = [p_stemmer.stem(w.lower()) for w in list(text4) if w.isalpha() and not w.lower() in sw]
doc5 = [p_stemmer.stem(w.lower()) for w in list(text5) if w.isalpha() and not w.lower() in sw]
doc_set = [doc1, doc2, doc3, doc4, doc5]

# under Windows this generates a warning
import gensim
from gensim import corpora, models

```

```
dictionary = corpora.Dictionary(doc_set)
dictionary

# transform each document into a bag of words
corpus = [dictionary.doc2bow((doc)) for doc in doc_set]

# The corpus contains the 5 documents
# each document is a list of indexed features and occurrence count (freq)
print(type(corpus))
print(type(corpus[0]))
print(type(corpus[0][0]))
print(corpus[0][::2000])

# let's try 4 topics for our 5 documents
# 50 passes takes quite a while, let's try less
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics=4, id2word=dictionary, passes=10)

print(ldamodel.print_topics(num_topics=4, num_words=10))
```

11 Upvotes Reply Follow this discussion

Earliest **Top** **Most Recent**

Kedar Joshi · 5 days ago

This certainly was a great help! Thanks Uwe and thank for your insights overall on various discussion threads throughout the course.

0 Upvotes Reply

OP Oscar Rene Chamberlain Pravia · a month ago

Excellent contribution!!

0 Upvotes Reply

Kakoli · a month ago

This is a great help.

0 Upvotes Reply

1

HW Reply

Reply