mcdevitt_nlp_homework_06_document_similarity

November 5, 2018

0.1 Document Similarity

book title evaluation

MSDS 7337 - Natural Language Processing - Homework 06

Patrick McDevitt

05-Nov-2018

For this project we are requested to :

1. Evaluate text similarity of Amazon book search results by doing the following:
    a. Do a book search on Amazon. Manually copy the full book title (including subtitle) of each of the top 24 books listed in the first two pages of search results.

    b. In Python, run one of the text-similarity measures covered in this course, e.g., cosine similarity. Compare each of the book titles, pairwise, to every other one.

    c. Which two titles are the most similar to each other? Which are the most dissimilar? Where do they rank, among the first 24 results?
2. Now evaluate using a major search engine.
    a. Enter one of the book titles from question 1a into Google, Bing, or Yahoo!. Copy the capsule of the first organic result and the 20th organic result. Take web results only (i.e., not video results), and skip sponsored results.

    b. Run the same text similarity calculation that you used for question 1b on each of these capsules in comparison to the original query (book title).

    c. Which one has the highest similarity measure?

Submit all of your inputs and outputs and your code for this assignment, along with a brief written explanation of your findings.

0.1.1 Question 1.c responses :

- Which two titles are the most similar to each other ?
  - from the results shown in below section (after code execution), we can observe that there is one pair that has a perfect match (based on cosine similarity) due to only being separated by a digit in the title
    - Title 6 : Lords of the Underworld Collection Volume 2: An Anthology
    - Title 3 : Lords of the Underworld Collection Volume 1: An Anthology
    - Cosine similarity score : 1.0
  - Discounting those 2 titles, the next closest match occurs between these 2 titles :
    - Title 4 : The Darkest Warrior (Lords of the Underworld Book 14)
    - Title 13 : The Darkest Night (Lords of the Underworld Book 1)
    - Cosine similarity score : 0.525
- Which are the most dissimilar ?
  - most dissimilar, based on cosine similarity score are :
    - Title 9 : City of Devils: The Two Men Who Ruled the Underworld of Old Shanghai
    - Title 0 : Underworld: A Novel
    - Cosine similarity score : 0.031
- Where do they rank, among the first 24 results ?
  - the rank for each of the titles is listed above

0.1.2 Question 2.c response :

- Which one has the highest similarity measure ?
  - Of the 2 capsules that were compared to the selected title, we can observe from the results below that the 1st returned capsule has the highest cosine similarity score. From a visual comparison of the 2 capsules, the first one has most of the words of the book title embedded in the synopsis.
    - **Book title** : The Corporation: An Epic Story of the Cuban American Underworld

- **Capsule most similar of the 2 evaluated** : An epic story of gangsters, drugs, violence, sex, and murder rooted in the streets, The Corporation reveals how an entire generation of political exiles, refugees, racketeers, corrupt cops, hitmen, and their wives and girlfriends became caught up in an American saga of desperation and empire building.

- Cosine similarity score : 0.239

```
In [1]: # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... necessary packages for Ben Brock
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        import platform; print platform.platform()
        import sys; print "Python", sys.version
        import nltk; print "nltk", nltk.__version__
        #from bs4 import BeautifulSoup, SoupStrainer
        import requests; print "requests", requests.__version__

        try :
            from urllib2 import Request, urlopen
        except :
            from urllib.request import Request, urlopen

        import re; print "re", re.__version__

        from pattern.en import parsetree

        import os
        #print (os.environ['CONDA_DEFAULT_ENV'])
```

```
Linux-4.15.0-38-generic-x86_64-with-Ubuntu-16.04-xenial
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609]
nltk 3.3
requests 2.18.4
re 2.2.1
```

```
In [2]: import numpy as np
        import pattern

        import import_ipynb # supports importing other ipynb notebooks
```

```
In [3]: # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... import some packages
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        import re
        import nltk
        import string
        from nltk.stem import WordNetLemmatizer
        from HTMLParser import HTMLParser
        import unicodedata
```

```
In [4]: # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... import Sarkar functions : https://github.com/dipanjanS/text-analytics-with-python
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        #import normalization
        from normalization import normalize_corpus
```

```
importing Jupyter notebook from normalization.ipynb
importing Jupyter notebook from contractions.ipynb
```

```
In [5]: # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... ref : Text Analytics with Python, Sarker, p. 270
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

        def build_feature_matrix(documents, feature_type = 'frequency',
                                 ngram_range = (1, 1), min_df = 0.0, max_df = 1.0):

            feature_type = feature_type.lower().strip()

            if feature_type == 'binary':
                vectorizer = CountVectorizer(binary = True, min_df = min_df,
                                             max_df = max_df, ngram_range = ngram_range)
            elif feature_type == 'frequency':
                vectorizer = CountVectorizer(binary = False, min_df = min_df,
                                             max_df = max_df, ngram_range = ngram_range)
            elif feature_type == 'tfidf':
                vectorizer = TfidfVectorizer(min_df = min_df, max_df = max_df,
                                             ngram_range = ngram_range)
            else:
                raise Exception("Wrong feature type entered. Possible values: 'binary', 'frequen

            feature_matrix = vectorizer.fit_transform(documents).astype(float)

            return vectorizer, feature_matrix
```

```
In [6]: # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... ref : Text Analytics with Python, Sarker, p. 287
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        def compute_cosine_similarity(doc_features, corpus_features, top_n = 3):

            # get document vectors
            doc_features = doc_features.toarray()[0]
            corpus_features = corpus_features.toarray()

            # compute similarities
            similarity = np.dot(doc_features,
                                corpus_features.T)

            # get docs with highest similarity scores
            top_docs = similarity.argsort()[::-1][:top_n]

            top_docs_with_score = [(index, round(similarity[index], 3))
                                    for index in top_docs]

            return top_docs_with_score

In [7]: # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... list of book titles from Amazon
        # ... search key word entered : "Underworld"
        # ... list below represents the 1st 24 titles returned from Amazon search
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        corpus = ["Underworld: A Novel",
        "Underworld - Through the Belly of the Beast: A LitRPG Series",
        "Underworld: The Mysterious Origins of Civilization",
        "Lords of the Underworld Collection Volume 1: An Anthology",
        "The Darkest Warrior (Lords of the Underworld Book 14)",
        "The Underworld U.S.A. Trilogy, Volume I: American Tabloid, The Cold Six Thousand (Every
        "Lords of the Underworld Collection Volume 2: An Anthology",
        "The Social Order of the Underworld: How Prison Gangs Govern the American Penal System",
        "Lords of the Underworld: The Darkest Sampler",
        "City of Devils: The Two Men Who Ruled the Underworld of Old Shanghai",
        "The Corporation: An Epic Story of the Cuban American Underworld",
        "The Dark Net: Inside the Digital Underworld",
        "Kings of the Underworld: Alpha & Omega",
        "The Darkest Night (Lords of the Underworld Book 1)",
        "The Underworld U.S.A. Trilogy, Volume II: Blood's A Rover",
        "SAINT (Boston Underworld Book 4)",
        "REAPER (Boston Underworld Book 2)",
        "Servant of the Underworld (Obsidian and Blood) (Volume 1)",
        "Go to Hell: A Heated History of the Underworld",
        "GHOST (Boston Underworld Book 3)",
        "The Arraignment III: The Underworld",
        "THIEF (Boston Underworld Book 5)",
        "King Tut: The Journey through the Underworld",
"Underworld - Level Up or Die: A LitRPG Series"]
```

In [8]:

```
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... ref : Text Analytics with Python, Sarker, pp. 286, 287
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

query_docs = corpus

html_parser = HTMLParser()

# normalize and extract features from the toy corpus

norm_corpus = normalize_corpus(corpus, lemmatize = True)
#norm_corpus = normalize_corpus(query_docs, lemmatize = True)

tfidf_vectorizer, tfidf_features = build_feature_matrix(norm_corpus,
                                                        feature_type = 'tfidf',
                                                        ngram_range = (1, 1),
                                                        min_df = 0.0,
                                                        max_df = 1.0)

# normalize and extract features from the query corpus

norm_query_docs =  normalize_corpus(query_docs, lemmatize = True)
#norm_query_docs =  normalize_corpus(corpus, lemmatize = True)

query_docs_tfidf = tfidf_vectorizer.transform(norm_query_docs)

# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... cosine similarity
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

print ('='*60)
print ('Book Titles Similarity Analysis using Cosine Similarity')
print ('='*60)
print "\n"

# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
# ... return top 2 similar, then drop the 1st if same as primary document
# ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

for index, doc in enumerate(query_docs):

    doc_tfidf = query_docs_tfidf[index]

    top_similar_docs = compute_cosine_similarity(doc_tfidf,
                                             tfidf_features,
                                             top_n = 2)
```

```
        if(top_similar_docs[0][0] == index) :
            tsd = top_similar_docs[1]
        else :
            tsd = top_similar_docs[0]

        print ('-'*80)
        buffer = "Title %2d         : %s" % (index, doc)
        print buffer
        print 'Cosine similar   :', corpus[tsd[0]]
        print 'Similarity score :', tsd[1]
        print ('-'*80)
        print ("\n")
```

```
============================================================
Book Titles Similarity Analysis using Cosine Similarity
============================================================


--------------------------------------------------------------------------------
Title  0         : Underworld: A Novel
Cosine similar   : THIEF (Boston Underworld Book 5)
Similarity score : 0.054
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
Title  1         : Underworld - Through the Belly of the Beast: A LitRPG Series
Cosine similar   : Underworld - Level Up or Die: A LitRPG Series
Similarity score : 0.43
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
Title  2         : Underworld: The Mysterious Origins of Civilization
Cosine similar   : Underworld: A Novel
Similarity score : 0.044
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
Title  3         : Lords of the Underworld Collection Volume 1: An Anthology
Cosine similar   : Lords of the Underworld Collection Volume 2: An Anthology
Similarity score : 1.0
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
Title  4         : The Darkest Warrior (Lords of the Underworld Book 14)
Cosine similar   : The Darkest Night (Lords of the Underworld Book 1)
Similarity score : 0.525
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title  5         : The Underworld U.S.A. Trilogy, Volume I: American Tabloid, The Cold Six Thous
Cosine similar   : The Underworld U.S.A. Trilogy, Volume II: Blood's A Rover
Similarity score : 0.212
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title  6         : Lords of the Underworld Collection Volume 2: An Anthology
Cosine similar   : Lords of the Underworld Collection Volume 1: An Anthology
Similarity score : 1.0
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title  7         : The Social Order of the Underworld: How Prison Gangs Govern the American Pena
Cosine similar   : The Corporation: An Epic Story of the Cuban American Underworld
Similarity score : 0.12
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title  8         : Lords of the Underworld: The Darkest Sampler
Cosine similar   : The Darkest Night (Lords of the Underworld Book 1)
Similarity score : 0.5
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title  9         : City of Devils: The Two Men Who Ruled the Underworld of Old Shanghai
Cosine similar   : Underworld: A Novel
Similarity score : 0.031
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title 10         : The Corporation: An Epic Story of the Cuban American Underworld
Cosine similar   : The Social Order of the Underworld: How Prison Gangs Govern the American Pena
Similarity score : 0.12
--------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
Title 11         : The Dark Net: Inside the Digital Underworld
Cosine similar   : Underworld: A Novel
Similarity score : 0.038
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
Title 12         : Kings of the Underworld: Alpha & Omega
Cosine similar   : King Tut: The Journey through the Underworld
Similarity score : 0.302
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
Title 13         : The Darkest Night (Lords of the Underworld Book 1)
Cosine similar   : The Darkest Warrior (Lords of the Underworld Book 14)
Similarity score : 0.525
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
Title 14         : The Underworld U.S.A. Trilogy, Volume II: Blood's A Rover
Cosine similar   : Servant of the Underworld (Obsidian and Blood) (Volume 1)
Similarity score : 0.361
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
Title 15         : SAINT (Boston Underworld Book 4)
Cosine similar   : REAPER (Boston Underworld Book 2)
Similarity score : 0.511
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
Title 16         : REAPER (Boston Underworld Book 2)
Cosine similar   : SAINT (Boston Underworld Book 4)
Similarity score : 0.511
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
Title 17         : Servant of the Underworld (Obsidian and Blood) (Volume 1)
Cosine similar   : The Underworld U.S.A. Trilogy, Volume II: Blood's A Rover
Similarity score : 0.361
-------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
Title 18         : Go to Hell: A Heated History of the Underworld
Cosine similar   : Underworld: A Novel
Similarity score : 0.044
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title 19         : GHOST (Boston Underworld Book 3)
Cosine similar   : SAINT (Boston Underworld Book 4)
Similarity score : 0.511
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title 20         : The Arraignment III: The Underworld
Cosine similar   : Underworld: A Novel
Similarity score : 0.054
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title 21         : THIEF (Boston Underworld Book 5)
Cosine similar   : SAINT (Boston Underworld Book 4)
Similarity score : 0.511
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title 22         : King Tut: The Journey through the Underworld
Cosine similar   : Kings of the Underworld: Alpha & Omega
Similarity score : 0.302
--------------------------------------------------------------------------------


--------------------------------------------------------------------------------
Title 23         : Underworld - Level Up or Die: A LitRPG Series
Cosine similar   : Underworld - Through the Belly of the Beast: A LitRPG Series
Similarity score : 0.43
--------------------------------------------------------------------------------
```

```
/usr/local/lib/python2.7/dist-packages/sklearn/feature_extraction/text.py:1089: FutureWarning: C
  if hasattr(X, 'dtype') and np.issubdtype(X.dtype, np.float):
```

```
In [9]: # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... select a book from the above list
        # ... enter the book title into Yahoo search engine
        # ... return the 1st and 20th capsules returned from that search
        # ...
        # ... selected book title : "The Corporation: An Epic Story of the Cuban American Underw
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        capsule = ["The Corporation: An Epic Story of the Cuban American Underworld",
                   "An epic story of gangsters, drugs, violence, sex, and murder rooted in the s
                   "A fascinating, cinematic, multigenerational history of the Cuban mob in the

        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... repeat above procedure to compare selected book title to the 2 capsules returned
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        query_docs = capsule

        html_parser = HTMLParser()

        # normalize and extract features from the corpus

        norm_corpus = normalize_corpus(capsule, lemmatize = True)

        #norm_corpus = normalize_corpus(query_docs, lemmatize = True)

        tfidf_vectorizer, tfidf_features = build_feature_matrix(norm_corpus,
                                                                 feature_type = 'tfidf',
                                                                 ngram_range = (1, 1),
                                                                 min_df = 0.0,
                                                                 max_df = 1.0)

        # normalize and extract features from the query corpus

        norm_query_docs =  normalize_corpus(query_docs, lemmatize = True)
        #norm_query_docs =  normalize_corpus(corpus, lemmatize = True)

        query_docs_tfidf = tfidf_vectorizer.transform(norm_query_docs)

        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... cosine similarity
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        print ('='*80)
        print ('Book title comparison to search return capsule - cosine similarity')
        print ('='*80)
        print "\n"

        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
        # ... return top 2 similar, then drop the 1st if same as primary document
        # ... -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

        for index, doc in enumerate(query_docs):

            doc_tfidf = query_docs_tfidf[index]

            top_similar_docs = compute_cosine_similarity(doc_tfidf,
                                                     tfidf_features,
                                                     top_n = 3)

            if(top_similar_docs[0][0] == index) :
                tsd = top_similar_docs[1]
            else :
                tsd = top_similar_docs[0]

            print ('-'*80)
            buffer = "Title %2d         : %s" % (index, doc)
            print buffer
            print 'Cosine similar   :', capsule[tsd[0]]
            print 'Similarity score :', tsd[1]
            print ('-'*80)
            print ("\n")
```

```
================================================================================
Book title comparison to search return capsule - cosine similarity
================================================================================



--------------------------------------------------------------------------------
Title  0         : The Corporation: An Epic Story of the Cuban American Underworld
Cosine similar   : An epic story of gangsters, drugs, violence, sex, and murder rooted in the st
Similarity score : 0.239
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
Title  1         : An epic story of gangsters, drugs, violence, sex, and murder rooted in the st
Cosine similar   : The Corporation: An Epic Story of the Cuban American Underworld
Similarity score : 0.239
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
Title  2         : A fascinating, cinematic, multigenerational history of the Cuban mob in the U
```

```
Cosine similar   : The Corporation: An Epic Story of the Cuban American Underworld
Similarity score : 0.118
---------------------------------------------------------------------------------
```