Semantic Similarity Using WordNet Ontology

Pragadesh Vasudevan Follow
Apr 20, 2018 · 4 min read

The main essence of this project is to make use of the Word Net® Ontology to work on the bag of words of the image data-set by reducing the number of elements with similar meanings that describes the image and replacing them with their respective synonym.

Research Problem:

This dimensionality reduction is very important as it would reduce the redundancy of the data as well as the size of the dataset. Finding semantic similarity of the word plays an important role in many applications of Artificial Intelligence, Knowledge Sharing, Web Mining.

Some of the applications are discussed below

· **Document Topic modeling**

Knowledge based and semantic information retrieval systems (identify optimal match for the query) by suitable concepts

· **Language Conversion**

NLP application — Conversion of one language to another using NLP technique.

· **Sense disambiguation**

The sense the word has been used (context) in the documents or query.

· **Bioinformatics**

Compare genes and proteins based on the similarity of their functions rather than on their sequence similarity.

Loading packages:

Importing a necessary package is the first step in developing this project. I highly recommend if you don't have installed in your python environment.

```
import numpy as np
import pandas as pd
import string
from string import digits
import re
import nltk
from nltk.corpus import stopwords
from sklearn import preprocessing
import itertools

import scipy
import csv
from nltk.tokenize import word_tokenize
import re, csv, yaml
from nltk.corpus import wordnet as wn
from nltk.metrics import edit_distance
nltk.download('punkt')
import itertools
nltk.download('wordnet')
```

Essential Python Libraries

Data Loading:

The data we would be working now is a *.csv* file which contains:

Cluster_size, Date, Interval, Distance, File, Category, Concept, Event

We are interested in the Concept column of the dataset. The first row of the concept column has a cluster of words which include: business, ceremony, festival, etc. We are interested in finding out the similarity between each pair of words in the clusters using the functions of WordNet Ontology package.

	cluster_size	Date	Interval	Distance	File	Category	Concept	Event
0	1	Sun Jun 14 11:25:19 PDT 2009	0 Min	0.00 Miles	[DSCN0657]	[Office=1]	[business=1, ceremony=1, festival=1, group=1,...	[]
1	6	Sun Oct 18 13:40:54 CST 2009	4 Min	0.04 Miles	[DSCN4196, DSCN4197, DSCN4198, DSCN4199, DSCN...	[Scenic Lookout=4, Hotel=1]	[mountain=6, outdoors=6, travel=6, tree=6, fo...	[URBAN_WALK]
2	5	Sun Sep 20 14:23:57 CST 2009	12 Min	0.20 Miles	[DSCN3925, DSCN3926, DSCN3927, DSCN3928, DSCN...	[Chinese Restaurant=2, Coffee Shop=2, Bus Sta...	[people=4, city=3, politics=3, architecture=2...	[URBAN_WALK]
3	44	Sun Sep 06 17:48:48 SGT 2009	63 Min	0.26 Miles	[DSCN3382, DSCN3383, DSCN3384, DSCN3385, DSCN...	[Beach=8, Cable Car=7, Park=6, Event Space=3,...	[people=28, politics=18, protest=13, travel=1...	[URBAN_WALK]
		Sun Jun 07		0.00			[automobile=1,	

Sample dataset

Text Pre-processing:

The Concept column of the dataset consists of rows of lists of words with frequencies which are describing the respective image taken. We observe that each of the words in a rows would have a similar meanings.

Removing special characters, numbers and tokenizing are the pre-processing technique we would be using for our project.

```
imfile['Concept'].head(5)

def convert_dict(s):
    return re.sub(r'\W',' ',s)

imfile['list'] = imfile['Concept'].apply(convert_dict)

def remove_num(s):
    s = re.sub("\d+"," ", s)
    return " ".join(s.split())

imfile['list'] = imfile['list'].apply(remove_num)

word_tokenize(imfile['list'][0])
```

The below image shows the final output after text pre-processing.

```
imfile['list'].tail(20)
```

```
372    adult cabinet facial expression furniture girl...
373    people politics travel coast group outdoors ac...
374    road people action street transportation trave...
375
376    city travel people architecture street politic...
377    people architecture travel building outdoors a...
378             adult clothing one people portrait women
379    nobody architecture garden house luxury modern...
380    beach coast leisure recreation relaxation reso...
381    architecture building business city cityscape ...
382    travel architecture business nobody building p...
383    adult architecture building city competition c...
384    adult bar beverage cafe commerce counter desk ...
385    carpet chair dining table easy chair fireplace...
386    people politics adult men police protest road ...
387    people recreation swimming water swimming pool...
388    adult people outdoors women portrait politics ...
389    dwelling furniture house indoors room display ...
390    chair contemporary desk dwelling floor furnitu...
391    outdoors nobody tree environment nature people...
Name: list, dtype: object
```

Sample Concept column after pre-processing.

Semantic Similarity Methods:

In order to know if two words are similar, we will calculate the Semantic Similarity between those two words. The similarities that we are going to use are:

a) Wu-Palmar Similarity

The principle of similarity computation is based on the edge counting method which is defined as follows: Given an ontology Ω formed by a set of nodes and a root node (R). C1 and C2 represent two ontology elements of which we will calculate the similarity. The principle of similarity computation is based on the distance (N1 and N2) which separates nodes C1 and C2 from the root node and the distance (N) which separates the closest common ancestor (CS) of C1 and C2 from the node R.[1] The similarity measure of Wu and Palmer is defined by the following expression:

$$SimWP = 2*N/(N1+N2)$$

Wu-Palmer similarity calculation gives a similarity score from 0 to 1. If the similarity score is greater than 0.5

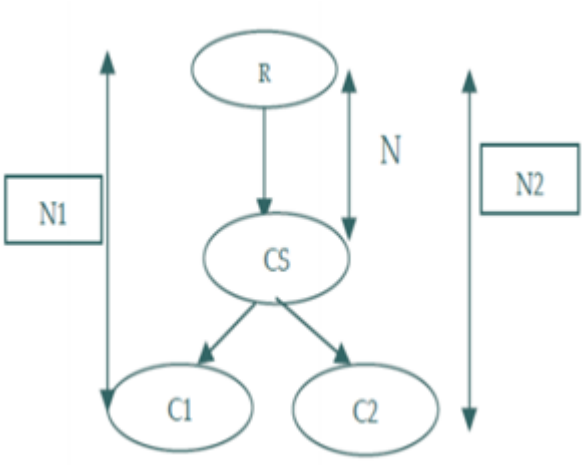

Fig. Example of a concept hierarchy

b) Leacock Chodorow Similarity:

Leacock and Chodorow rely on the length len(c1,c2) of the shortest path between two Synsets for their measure of similarity. However, they limit their attention to is–a links and scale the path length by the overall depth D of the taxonomy.

$$SimLch(c1, c2) = -\log(length/2*D)$$

Where length is the length of the shortest path between the two concepts (using node-counting) and D is the maximum depth of the taxonomy. Based on this measure, the shortest path between two concepts of the ontology restricted to taxonomic links is normalized by introducing a division by the double of the maximum hierarchy depth.

The LCH similarity scores are between 0 and 3.689. Hence, we can consider the threshold value as 2.

```
n=392
count=0
newlist = []
while(count<n):
    tokens = word_tokenize(imfile['list'][count])
    for a in range(0, len(tokens)-2):
        for b in range(a+1, len(tokens)-1):
            #print(a)
            #print(b)
            x = wn.synsets(tokens[a])
            y = wn.synsets(tokens[b])
            print(x)
            print(y)
            if (len(x) != 0) and (len(y) != 0):
                ai = x[0]
                bi = y[0]
                xyz = ai.wup_similarity(bi)
                if xyz is None:
                    xyz = 0
                if xyz > 0.5:
                    tokens[b] = tokens[a]
                    #tokens = list(set(tokens))
                    #len(tokens)
                print(ai)
                print(bi)
                print(xyz)
    #print(tokens)
    newlist.append(tokens)
    print(newlist)
    #print(len(tokens))
    count+=1
```

The code compares the first synset of the word in wordNet Ontology as shown below

```
[Synset('business.n.01'), Synset('commercial_enterprise.n.02'), Synset('occupation.n.01'), Synset('business.n.04'), Synset('b
usiness.n.05'), Synset('business.n.06'), Synset('business.n.07'), Synset('clientele.n.01'), Synset('business.n.09')]
[Synset('group.n.01'), Synset('group.n.02'), Synset('group.n.03'), Synset('group.v.01'), Synset('group.v.02')]
Synset('business.n.01')
Synset('group.n.01')
0.6
[Synset('business.n.01'), Synset('commercial_enterprise.n.02'), Synset('occupation.n.01'), Synset('business.n.04'), Synset('b
usiness.n.05'), Synset('business.n.06'), Synset('business.n.07'), Synset('clientele.n.01'), Synset('business.n.09')]
[Synset('inside.r.01')]
Synset('business.n.01')
Synset('inside.r.01')
0
[Synset('business.n.01'), Synset('commercial_enterprise.n.02'), Synset('occupation.n.01'), Synset('business.n.04'), Synset('b
usiness.n.05'), Synset('business.n.06'), Synset('business.n.07'), Synset('clientele.n.01'), Synset('business.n.09')]
[Synset('people.n.01'), Synset('citizenry.n.01'), Synset('people.n.03'), Synset('multitude.n.03'), Synset('people.v.01'), Syn
set('people.v.02')]
Synset('business.n.01')
Synset('people.n.01')
0.5454545454545454
[Synset('business.n.01'), Synset('commercial_enterprise.n.02'), Synset('occupation.n.01'), Synset('business.n.04'), Synset('b
```

Result:

After the output of each similarity measures, we found the length of each list to check for the number of items. The WuP similarity shows the best results compared to LcH.

	Concept_wup	Concept_lch	concept	wup_len	lch_len	Concepts
0	business ceremony festival business indoo...	business ceremony festival business indoo...	9	6	7	[business=1, ceremony=1, festival=1, group=1,...
1	mountain outdoors travel tree forest lan...	mountain outdoors travel tree forest lan...	22	14	18	[mountain=6, outdoors=6, travel=6, tree=6, fo...
2	people city politics architecture archite...	people city politics architecture archite...	37	15	27	[people=4, city=3, politics=3, architecture=2...
3	people politics protest protest adult ad...	people politics protest travel adult adu...	110	26	51	[people=28, politics=18, protest=13, travel=1...
4	automobile business business automobile v...	automobile business business transportatio...	5	3	4	[automobile=1, business=1, police=1, transpor...
5	aircraft aircraft flight sky	aircraft aircraft flight sky	4	3	3	[aircraft=1, airplane=1, flight=1, sky=1]

Further from here:

•Deciding the threshold for the LCH and WUP similarity scores is a major problem in the project. (For the convenience, we have used 0.5 for WUP and 2 for LCH)

- Another problem is that the frequency of the words.

. . .

You can reach me with details below if you have any questions/suggestions:

LinkedIn: https://bit.ly/2q8ykMi

Github: *https://bit.ly/2GwvDzm*

Email: pragadheeswaransv@gmail.com

Data Science Wordnet Ontology Python Semantics