Untitled6

October 26, 2019

In [9]:
```
import nltk
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.stem import PorterStemmer
nltk.download('averaged_perceptron_tagger')
import os
import re
from gensim.models import Word2Vec
#get all subfolders with newsgroup data
d='20_newsgroups'
folders = list(filter(lambda x: os.path.isdir(os.path.join(d, x)), os.listdir(d)))
print(folders)

#Creating corpus
fulltext = []
for i in range(len(folders)):
    folder = '20_newsgroups/' + folders[i]
    files = list(filter(lambda x: os.path.isfile(os.path.join(folder, x)), os.listdir(fo
    for j in range(len(files)):
        file1 = open(folder + '/' + files[j], "r+", encoding="latin-1")
        text = file1.read()
        fulltext.append(text)
        file1.close()
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /Users/mollysrour/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

```
['talk.politics.mideast', 'rec.autos', 'comp.sys.mac.hardware', 'alt.atheism', 'rec.sport.baseba
```

In [4]:
```
def preprocess(corpus):
    tokenized_corpus = []
    for i in corpus:
        tokenized_document = word_tokenize(i)
        tokenized_corpus.append(tokenized_document)
    normalized_corpus = []
```

```
            for i in tokenized_corpus:
                normalized_document = []
                for j in i:
                    j = re.sub(r'\d+', '', j) # remove numbers
                    j = re.sub(r'[^\w\s]','', j.lower().strip()) #remove everything except words
                    j = re.sub(r'\_','', j)
                    normalized_document.append(j)
                normalized_document = [i for i in normalized_document if i] # remove empty strin
                normalized_document = [i for i in normalized_document if len(i) > 0]
                normalized_corpus.append(normalized_document)
            with open('output.txt', 'w') as f:
                for i in normalized_corpus:
                    for j in i:
                        f.write('%s ' % j)
                    f.write('\n\n')
            return normalized_corpus
```

```
In [5]: normalized_corpus = preprocess(fulltext)
```

```
In [13]: def test_word2vec(normalized_corpus):
             for i in [0,1]:
                 for j in [10, 100, 200]:
                     word2vec_model = Word2Vec(normalized_corpus, size=j, window=5, min_count=5,
                     print('Model: ' + str(i) + 'Size: ' + str(j))
                     print(word2vec_model.wv.most_similar('happy'))
                     print(('\n\n'))
                     print(word2vec_model.wv.most_similar('truth'))
                     print(('\n\n'))
                     print(word2vec_model.wv.most_similar('schedule'))
                     print(('\n\n'))
                     print(word2vec_model.wv.most_similar('time'))
                     print(('\n\n'))
                     print(word2vec_model.wv.most_similar('friend'))
```

```
In [14]: test_word2vec(normalized_corpus)
```

```
Model: 0Size: 10
[('afraid', 0.9386643171310425), ('quick', 0.9271412491798401), ('nice', 0.9190751910209656), ('
```

```
[('what', 0.9564932584762573), ('sense', 0.9395981431007385), ('any', 0.9383403658866882), ('not
```

```
[('patches', 0.9580537676811218), ('item', 0.9476672410964966), ('generation', 0.937206864356994
```

[('place', 0.9739801287651062), ('score', 0.9599364399909973), ('face', 0.9517949223518372), ('p

[('thread', 0.9718018770217896), ('talks', 0.9078044295310974), ('received', 0.9029957056045532)
Model: 0Size: 100
[('pleased', 0.7705234289169312), ('willing', 0.6964993476867676), ('afraid', 0.6890759468078613

[('bible', 0.637065052986145), ('meaning', 0.6107625961303711), ('god', 0.6050542593002319), ('e

[('mission', 0.7189797163009644), ('conference', 0.7145601511001587), ('launch', 0.6921563744544

[('moment', 0.5961534380912781), ('year', 0.5918624401092529), ('step', 0.5836614966392517), ('d

[('wife', 0.7439003586769104), ('doctor', 0.7331457138061523), ('colleague', 0.7275895476341248)
Model: 0Size: 200
[('pleased', 0.7534517049789429), ('lucky', 0.6772536635398865), ('satisfied', 0.676540970802307

[('meaning', 0.6257511973381042), ('bible', 0.6051745414733887), ('scripture', 0.602840006351471

[('conference', 0.7305501699447632), ('mission', 0.7190903425216675), ('broadcast', 0.6987998485

[('moment', 0.5984171032905579), ('step', 0.5412589907646179), ('stage', 0.5305915474891663), ('

[('colleague', 0.730625569820404), ('doctor', 0.723242998123169), ('wife', 0.7207539677619934),
Model: 1Size: 10
[('jeez', 0.9817153215408325), ('cheat', 0.9696027040481567), ('scoop', 0.9690260887145996), ('s

[('absolutes', 0.980271577835083), ('truths', 0.9801130294799805), ('strongly', 0.97387909889221

[('chevrolet', 0.9512313604354858), ('quest', 0.949804425239563), ('stats', 0.948480486869812),

[('place', 0.9824861884117126), ('photograph', 0.9813970327377319), ('cracks', 0.980284035205841

[('girlfriend', 0.9378345608711243), ('lately', 0.9375348091125488), ('thread', 0.93465662002563
Model: 1Size: 100
[('willing', 0.682263195514679), ('pleased', 0.6817788481712341), ('reluctant', 0.67631703615188

[('falsehood', 0.6905444860458374), ('truths', 0.6674897074699402), ('morals', 0.662748694419860

[('hosting', 0.711611270904541), ('devilsislanders', 0.7109827399253845), ('camden', 0.701209664

[('consuming', 0.6898764967918396), ('blasted', 0.6826618909835815), ('hike', 0.6821691989898682

[('girlfriend', 0.7085487842559814), ('coworker', 0.7020944356918335), ('colleague', 0.685582756
Model: 1Size: 200
[('pleased', 0.6112673282623291), ('unhappy', 0.5872986316680908), ('disappointed', 0.5590735673

[('ture', 0.6050390601158142), ('falsehood', 0.5950509309768677), ('objectively', 0.594140648841

[('devilsislanders', 0.668843150138855), ('boxscores', 0.6663817763328552), ('televise', 0.66590

[('fiddling', 0.5520125031471252), ('consuming', 0.5506227016448975), ('barbeque', 0.54897457361

[('girlfriend', 0.6274957656860352), ('coworker', 0.622255802154541), ('colleague', 0.6070100665

In [ ]: