import os  
import sys  
import re  
from gensim.models import Word2Vec  
from gensim.models.phrases import Phraser, Phrases

TEXT\_DATA\_DIR = './20\_newsgroups/'

texts = [] # list of text samples  
labels\_index = {} # dictionary mapping label name to numeric id  
labels = [] # list of label ids  
label\_text = [] # list of label texts

# Go through each directory  
for name in sorted(os.listdir(TEXT\_DATA\_DIR)):  
 path = os.path.join(TEXT\_DATA\_DIR, name)  
 if os.path.isdir(path):  
 label\_id = len(labels\_index)  
 labels\_index[name] = label\_id  
 for fname in sorted(os.listdir(path)):  
 # News groups posts are named as numbers, with no extensions.  
 if fname.isdigit():  
 fpath = os.path.join(path, fname)  
 f = open(fpath, encoding='latin-1')  
 t = f.read()  
 i = t.find('\n\n') # skip header in file (starts with two newlines.)  
 if 0 < i:  
 t = t[i:]  
 texts.append(t)  
 f.close()  
 labels.append(label\_id)  
 label\_text.append(name)  
print('Found %s texts.' % len(texts))  
# >> Found 1997 texts.

Found 19997 texts.

len(texts)

19997

len(labels\_index)

20

labels\_index

{'alt.atheism': 0,  
 'comp.graphics': 1,  
 'comp.os.ms-windows.misc': 2,  
 'comp.sys.ibm.pc.hardware': 3,  
 'comp.sys.mac.hardware': 4,  
 'comp.windows.x': 5,  
 'misc.forsale': 6,  
 'rec.autos': 7,  
 'rec.motorcycles': 8,  
 'rec.sport.baseball': 9,  
 'rec.sport.hockey': 10,  
 'sci.crypt': 11,  
 'sci.electronics': 12,  
 'sci.med': 13,  
 'sci.space': 14,  
 'soc.religion.christian': 15,  
 'talk.politics.guns': 16,  
 'talk.politics.mideast': 17,  
 'talk.politics.misc': 18,  
 'talk.religion.misc': 19}

len(labels)

19997

len(label\_text)

19997

# Cleaning data - remove punctuation from every newsgroup text  
sentences = []  
# Go through each text in turn  
for ii in range(len(texts)):  
 sentences = [re.sub(pattern=r'[\!"#$%&\\*+,-./:;<=>?@^\_`()|~=]',   
 repl='',   
 string=x  
 ).strip().split(' ') for x in texts[ii].split('\n')   
 if not x.endswith('writes:')]  
 sentences = [x for x in sentences if x != ['']]  
 texts[ii] = sentences

print(texts[6])

[['The', 'motto', 'originated', 'in', 'the', 'StarSpangled', 'Banner', '', 'Tell', 'me', 'that', 'this', 'has'], ['something', 'to', 'do', 'with', 'atheists'], ['The', 'motto', 'oncoins', 'originated', 'as', 'a', 'McCarthyite', 'smear', 'which', 'equated', 'atheism'], ['with', 'Communism', 'and', 'called', 'both', 'unamerican'], ['No', 'it', "didn't", '', 'The', 'motto', 'has', 'been', 'on', 'various', 'coins', 'since', 'the', 'Civil', 'War'], ['It', 'was', 'just', 'required', 'to', 'be', 'on', 'all', 'currency', 'in', 'the', "50's"], ['keith']]

# concatenate all sentences from all texts into a single list of sentences  
all\_sentences = []  
for text in texts:  
 all\_sentences += text

len(all\_sentences)

564196

# Phrase Detection  
# Give some common terms that can be ignored in phrase detection  
# For example, 'state\_of\_affairs' will be detected because 'of' is provided here:   
common\_terms = ["of", "with", "without", "and", "or", "the", "a"]  
# Create the relevant phrases from the list of sentences:  
# phrases = Phrases(all\_sentences, connector\_words=common\_terms)  
phrases = Phrases(all\_sentences, connector\_words=phrases.connector\_words)  
# The Phraser object is used from now on to transform sentences  
bigram = Phraser(phrases)  
# Applying the Phraser to transform our sentences is simply  
all\_sentences = list(bigram[all\_sentences])

print(bigram[all\_sentences[5676]])

['guilty', 'in', 'a', 'court\_of\_law', 'As', 'his', 'guilt', 'has', 'not', 'been\_established', 'it', 'is']

all\_sentences = list(bigram[all\_sentences])

model = Word2Vec(all\_sentences,   
 min\_count=3, # Ignore words that appear less than this  
 vector\_size=200, # Dimensionality of word embeddings  
 workers=2, # Number of processors (parallelisation)  
 window=5, # Context window for words during training  
 epochs=30) # Number of epochs training over corpus

model

<gensim.models.word2vec.Word2Vec at 0x7f2f914426a0>

model.vector\_size

200

len(model.wv)

83896

model.wv.most\_similar(positive="New\_York")

[('England', 0.5623193383216858),  
 ('County', 0.5407129526138306),  
 ('Munich', 0.4977436363697052),  
 ('in\_New\_York', 0.4767480492591858),  
 ('Allegheny', 0.4763221740722656),  
 ('Atlanta', 0.4754171371459961),  
 ('Michigan', 0.4743909537792206),  
 ('Pool\_A', 0.4742652475833893),  
 ('Kentucky', 0.47199633717536926),  
 ('London', 0.4704380929470062)]

model.wv.most\_similar(positive="Los\_Angeles")

[('County', 0.6109175682067871),  
 ('Baltimore', 0.5546572804450989),  
 ('Central', 0.5483584403991699),  
 ('Maine', 0.5367791652679443),  
 ('York', 0.5323377251625061),  
 ('Hampton', 0.5303636193275452),  
 ('P\_A1', 0.5292092561721802),  
 ('San', 0.5285807251930237),  
 ('Memorial', 0.5282953977584839),  
 ('San\_Diego', 0.5272114872932434)]

model.wv.most\_similar(positive="engine")

[('motor', 0.5326041579246521),  
 ('car', 0.501063883304596),  
 ('suspension', 0.482737272977829),  
 ('bike', 0.475795179605484),  
 ('tires', 0.4540516138076782),  
 ('battery', 0.4513896405696869),  
 ('engines', 0.4501799941062927),  
 ('fuel', 0.44321951270103455),  
 ('fluid', 0.44040802121162415),  
 ('turbo', 0.4354288578033447)]

model.wv.most\_similar(positive="oil")

[('fuel', 0.5508603453636169),  
 ('water', 0.5092196464538574),  
 ('wind', 0.49241453409194946),  
 ('air', 0.4824705123901367),  
 ('heat', 0.46240293979644775),  
 ('gas', 0.4604598879814148),  
 ('pressure', 0.45390766859054565),  
 ('fluid', 0.4530622363090515),  
 ('lamp', 0.4386850893497467),  
 ('intake', 0.4361421465873718)]

model.wv.most\_similar(positive="man")

[('woman', 0.5961380004882812),  
 ('person', 0.5530979037284851),  
 ('himself', 0.5357284545898438),  
 ('child', 0.5261008143424988),  
 ('mother', 0.51578289270401),  
 ('flesh', 0.49853241443634033),  
 ('god', 0.4695480763912201),  
 ('God', 0.46365979313850403),  
 ('Satan', 0.460511177778244),  
 ('he', 0.45530590415000916)]

model.wv.most\_similar(positive="human")

[('physical', 0.4816420078277588),  
 ('conscious', 0.48163077235221863),  
 ('sin', 0.46208181977272034),  
 ('spiritual', 0.46029600501060486),  
 ('child', 0.43867719173431396),  
 ('humanity', 0.4284389913082123),  
 ('divine', 0.42350345849990845),  
 ('mankind', 0.41785159707069397),  
 ('humans', 0.41349247097969055),  
 ('life', 0.41205161809921265)]