In [17]:
```
from gensim.models.word2vec import Word2Vec
```

In [18]:
```
import re
from sklearn import feature_extraction
stop_words = feature_extraction.text.ENGLISH_STOP_WORDS
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer

def preprocess(text):
  text = text.lower() #lowercase
  text = re.sub(r'[^\w\s]', '', text) #remove punctuations
  text = re.sub(r'\d+', '', text) #remove numbers
  text = " ".join(text.split()) #stripWhitespace
  text = text.split()
  text = [x for x in text if x not in stop_words] #remove stopwords
  text = [x for x in text if x not in ["dr", "doctor"]] #remove task specific stopwords
  text = " ".join(text)
  # stemmer_ps = PorterStemmer()
  # text = [stemmer_ps.stem(word) for word in text.split()] #stemming
  # text = " ".join(text)
  # lemmatizer = WordNetLemmatizer()
  # text = [lemmatizer.lemmatize(word) for word in text.split()]  #lemmatization
  # text = " ".join(text)
  return(text)
```

In [19]:
```
import pandas as pd
data = pd.read_csv('universal_studio.csv')
```

In [20]:
```
data['review_processed']=data['review_text'].apply(lambda x:preprocess(x))
data['review_processed']=data['review_processed'].apply(lambda x:x.split())
```

In [21]:
```
data['review_processed']
```

Out[21]:
```
0        [went, universal, memorial, day, weekend, tota...
1        [food, service, horrible, im, reviewing, food,...
2        [booked, vacation, mainly, ride, hagrid, motor...
3        [person, tries, test, seat, rides, gets, green...
4        [ok, stress, universal, studios, orlando, make...
                              ...
50899    [visit, universal, studio, theme, park, went, ...
50900    [finally, visited, singapores, theme, park, un...
50901    [visited, week, soft, opening, unfortunately, ...
50902    [visited, rd, day, soft, opening, ticket, sale...
50903    [group, managed, tickets, february, sneak, pre...
Name: review_processed, Length: 50904, dtype: object
```

In [9]:
```
data
```

Out[9]:

	reviewer	rating	written_date	title	review_text	branch	review_processed
0	Kelly B	2	30-May-21	Universal is a complete Disaster - stick with ...	We went to Universal over Memorial Day weekend...	Universal Studios Florida	[went, universal, memorial, day, weekend, tota...
1	Jon	1	30-May-21	Food is hard to get.	The food service is horrible. I'm not reviewin...	Universal Studios Florida	[food, service, horrible, im, reviewing, food,...
2	Nerdy P	2	30-May-21	Disappointed	I booked this vacation mainly to ride Hagrid m...	Universal Studios Florida	[booked, vacation, mainly, ride, hagrid, motor...
3	ran101278	4	29-May-21	My opinion	When a person tries the test seat for the ride...	Universal Studios Florida	[person, tries, test, seat, rides, gets, green...
4	tammies20132015	5	28-May-21	The Bourne Stuntacular...MUST SEE	Ok, I can't stress enough to anyone and everyo...	Universal Studios Florida	[ok, stress, universal, studios, orlando, make...
...	...	...	...	...	...	...	...
50899	vinz20	4	29-Mar-10	I'll Be Back Only If ...	This is my first visit to a Universal Studio t...	Universal Studios Singapore	[visit, universal, studio, theme, park, went, ...
50900	betty l	4	29-Mar-10	Universal Studios Singapore Experience	We finally visited Singapore's very first them...	Universal Studios Singapore	[finally, visited, singapores, theme, park, un...
50901	spoonos65	4	28-Mar-10	Impressive but not quite finished!	We visited during the first week of its 'soft ...	Universal Studios Singapore	[visited, week, soft, opening, unfortunately, ...
50902	HeatSeekerWrexham_UK	4	22-Mar-10	Small but beautifully marked	We visited on the 3rd day of the 'soft' openin...	Universal Studios Singapore	[visited, rd, day, soft, opening, ticket, sale...
50903	sc_myinitial	5	24-Feb-10	Excellent Sneak Preview	My group managed to get the tickets for the 16...	Universal Studios Singapore	[group, managed, tickets, february, sneak, pre...

50904 rows × 7 columns

In [13]:
```
model = Word2Vec(sentences=data['review_processed'].tolist(), vector_size=100, sg=1,min_count=5,window=1
```

In [15]:
```
vocab = model.wv.index2word
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/var/folders/cf/s1wshv2j2bz4cbfxfg5qgrgm0000gn/T/ipykernel_67518/3125669496.py in <module>
----> 1 vocab = model.wv.index2word

~/opt/anaconda3/lib/python3.9/site-packages/gensim/models/keyedvectors.py in index2word(self)
    648     @property
    649     def index2word(self):
--> 650         raise AttributeError(
    651             "The index2word attribute has been replaced by index_to_key since Gensim 4.0.0.\n"
    652             "See https://github.com/RaRe-Technologies/gensim/wiki/Migrating-from-Gensim-3.x-to
-4" (https://github.com/RaRe-Technologies/gensim/wiki/Migrating-from-Gensim-3.x-to-4")

AttributeError: The index2word attribute has been replaced by index_to_key since Gensim 4.0.0.
See https://github.com/RaRe-Technologies/gensim/wiki/Migrating-from-Gensim-3.x-to-4 (https://github.co
m/RaRe-Technologies/gensim/wiki/Migrating-from-Gensim-3.x-to-4)
```

In [16]:
```
len(vocab)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/var/folders/cf/s1wshv2j2bz4cbfxfg5qgrgm0000gn/T/ipykernel_67518/2434904630.py in <module>
----> 1 len(vocab)

NameError: name 'vocab' is not defined
```

In [108]:
```
model.wv.most_similar('pregnancy', topn=10)
```

Out[108]:
```
[('acknowledged', 0.371005654335022),
 ('colostomy', 0.3642195463180542),
 ('airhead', 0.34320884943008423),
 ('vannuyen', 0.33772072196006775),
 ('rudei', 0.3354911208152771),
 ('dummy', 0.33476555347442627),
 ('urogyn', 0.32517358660697937),
 ('grayson', 0.3176262080669403),
 ('visitors', 0.3155759871006012),
 ('advancements', 0.31440863013267517)]
```

In [122]:
```
model.wv.most_similar('surgery', topn=10)
```

Out[122]:
```
[('dncs', 0.3832515478134155),
 ('positives', 0.3712007999420166),
 ('relying', 0.36110588908195496),
 ('corporate', 0.35009485483169556),
 ('reversed', 0.3414647579193115),
 ('willis', 0.32585233449935913),
 ('folic', 0.31733620166778564),
 ('rattled', 0.3168574571609497),
 ('sivkin', 0.3136730194091797),
 ('sum', 0.31158018112182617)]
```

In [110]:
```
v_time = model.wv['time']
```

In [111]:
```
v_time
```

Out[111]:
```
array([ 4.7564772e-03, -3.8165010e-03, -4.7649667e-03, -4.8181256e-03,
       -2.8230886e-03, -1.8766781e-03, -1.3006260e-03,  4.7683897e-03,
        8.0386625e-04, -9.9461712e-04, -3.0890570e-03, -1.2697545e-03,
        7.9923199e-04,  3.2912386e-03,  4.5572720e-03,  1.6915080e-03,
       -1.3097618e-03, -2.7664637e-03,  2.7784656e-03, -1.7129695e-03,
        3.8893814e-03,  2.3633067e-04, -2.5447879e-03, -2.8300688e-03,
       -3.4407559e-03, -3.2346160e-03, -1.1493486e-03, -1.6840914e-03,
        1.9813152e-03,  2.8048202e-03, -3.2215840e-03, -3.6928281e-03,
        1.3219353e-03, -3.5431802e-03, -1.1252342e-03,  4.2588734e-03,
        3.9281724e-03,  2.3391158e-03, -2.4282334e-03,  3.6887042e-03,
       -4.1654343e-03,  3.9304616e-03, -1.6826278e-04,  1.2218139e-05,
       -4.4432604e-03, -3.9079869e-03, -3.1208389e-03,  3.6213261e-03,
       -3.5648337e-03, -1.9867413e-03, -2.8872963e-03, -8.0989813e-04,
       -1.3387596e-03,  3.5741113e-04,  1.0048251e-03,  3.3559240e-03,
       -3.5359573e-03, -4.6250760e-03, -3.7983588e-03, -3.9949752e-03,
        2.3829269e-03,  4.6502822e-03, -1.3237691e-03, -4.9677426e-03,
       -4.7566628e-04, -2.9111947e-03,  3.7544481e-03, -3.0353647e-03,
        4.8340796e-04, -1.4626822e-03,  3.5994123e-03,  2.2377125e-03,
        4.5515732e-03, -4.4592475e-03,  4.1288417e-03, -4.9708760e-04,
       -7.0245209e-04,  3.7344843e-03, -2.3325463e-03,  3.4572284e-03,
       -2.4633193e-03,  3.1881065e-03,  1.8683851e-03,  2.5859675e-03,
        1.3144470e-03, -4.6450356e-03, -3.7226293e-03, -1.4327405e-03,
        2.9552169e-03, -3.3057157e-03, -4.6403399e-03, -4.5972117e-03,
       -1.2277535e-03, -4.8469095e-03, -4.4800672e-03, -4.7206804e-03,
       -9.2028431e-04, -1.6980399e-03, -1.8314410e-03,  4.4337558e-03],
      dtype=float32)
```

In [126]:
```
model.wv.similarity('std', 'pregnancy')
```

Out[126]:
```
-0.04429632
```

In [128]:
```
model.wv.similarity('pregnancy', 'exam')
```

Out[128]:
```
0.13613759
```

In [132]:
```
model.wv.similarity('std', 'prescribe')
```

Out[132]:
```
0.16172192
```

In [133]:
```
model.wv.similarity('exam', 'prescribe')
```

Out[133]:
```
0.06621391
```

In [134]:
```
v_std = model.wv['std']
v_pregnancy = model.wv['pregnancy']
import numpy
numpy.dot(v_std, v_pregnancy)/(numpy.linalg.norm(v_std)* numpy.linalg.norm(v_pregnancy))
```

Out[134]:
```
-0.044296324
```

In [135]:
```
v_std = model.wv['std']
v_pregnancy = model.wv['pregnancy']
v_exam = model.wv['exam']
v_prescribe = model.wv['prescribe']
created_condition = v_std - v_prescribe + v_exam
numpy.dot(created_condition, v_pregnancy)/(numpy.linalg.norm(created_condition)* numpy.linalg.norm(v_pre
```

Out[135]:
```
0.18944451
```

In [151]:
```
v_husband = model.wv['husband']
v_wife = model.wv['wife']

numpy.dot(v_husband, v_wife)/(numpy.linalg.norm(v_husband)* numpy.linalg.norm(v_wife))
```

Out[151]:
```
0.20052034
```

In [152]:
```
v_woman = model.wv['female']
v_man = model.wv['male']
v_husband = model.wv['husband']
v_wife = model.wv['wife']
created_husband = v_wife - v_woman + v_man
numpy.dot(created_husband, v_husband)/(numpy.linalg.norm(created_husband)* numpy.linalg.norm(v_husband))
```

Out[152]:
```
0.023384754
```

In [154]:
```
v_husband = model.wv['son']
v_wife = model.wv['wife']

numpy.dot(v_husband, v_wife)/(numpy.linalg.norm(v_husband)* numpy.linalg.norm(v_wife))
```

Out[154]:
```
0.15959325
```

In [155]:
```
v_woman = model.wv['female']
v_man = model.wv['male']
v_husband = model.wv['son']
v_wife = model.wv['wife']
created_husband = v_wife - v_woman + v_man
numpy.dot(created_husband, v_husband)/(numpy.linalg.norm(created_husband)* numpy.linalg.norm(v_husband))
```

Out[155]:
```
0.1376494
```

In [156]:
```
model.wv.vectors.shape
```

Out[156]:
```
(15487, 100)
```

In [157]:
```
outdata=pd.DataFrame(model.wv.vectors)
```

In [158]:
```
outdata
```

Out[158]:

	0	1	2	3	4	5	6	7	8	9	...	90	9
0	0.004756	-0.003817	-0.004765	-0.004818	-0.002823	-0.001877	-0.001301	0.004768	0.000804	-0.000995	...	-0.004640	-0.00459
1	-0.004130	-0.002402	0.001780	0.000797	0.003601	-0.002496	-0.000197	0.003979	0.001421	0.002025	...	-0.004242	0.00320
2	-0.000419	-0.000079	-0.003681	-0.004363	-0.002970	-0.004543	0.004725	-0.004172	0.001544	0.004264	...	-0.003051	0.00386
3	-0.004816	0.003951	-0.002790	0.000608	-0.001526	-0.004824	-0.001972	-0.002428	-0.000499	-0.001269	...	-0.003716	0.00302
4	0.000152	-0.000779	0.001041	-0.002821	-0.001447	-0.000327	0.002964	-0.004830	0.002010	0.004742	...	-0.001018	0.00233
...	...	...	...	...	...	...	...	...	...	...	...	...	..
15482	-0.003204	0.000121	0.002318	-0.004018	0.001655	0.004634	0.003111	0.001279	-0.002178	0.002107	...	-0.002881	0.00133
15483	-0.004814	-0.001408	0.002516	-0.003346	-0.004372	-0.004012	0.001252	0.003076	-0.000845	0.000875	...	0.003621	-0.00475
15484	-0.002439	0.002359	0.002921	-0.001349	-0.001690	0.003099	-0.000334	0.002380	0.004341	-0.004256	...	-0.003179	0.00058
15485	0.001950	-0.002748	0.004256	0.001879	0.003050	0.003382	0.004705	0.002413	0.001877	0.000163	...	-0.002020	0.00431
15486	-0.001261	0.002125	0.004221	-0.000501	-0.003958	-0.002050	-0.004938	0.002158	0.003188	-0.000095	...	0.004573	0.00226

15487 rows × 100 columns

In [159]:
```
outdata.to_csv('word2vec_ratemds.tsv',sep='\t',index=False,header=False)
```

In [160]:
```
pd.DataFrame(model.wv.index2word).to_csv('word2vec_ratemds_words.tsv',sep='\t',index=False,header=False)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```
https://projector.tensorflow.org/
```