SI630 Homework 2: Word2vec Vector Analysis

Important Note: Start this notebook only after you've gotten your word2vec model up and running!

Many NLP packages support working with word embeddings. In this notebook you can work through the various problems assigned in Task 3. We've provided the basic functionality for loading word vectors using Gensim (https://radimrehurek.com/gensim/models/keyedvectors.html), a good library for learning and using word vectors, and for working with the vectors.

One of the fun parts of word vectors is getting a sense of what they learned. Feel free to explore the vectors here!

In [2]:

```
from gensim.models import KeyedVectors
from gensim.test.utils import datapath
```

In [3]:

```
word_vectors = KeyedVectors.load_word2vec_format('trained_vector.txt', binary=False)
```

In [4]:

```
word_vectors['the']
```

Out[4]:

```
array([-0.04774104,  0.01113729, -0.0854168 , -0.02429663, -0.0077821
9,
        0.05570809,  0.04408854, -0.05375911, -0.09509112, -0.0485270
1,
       -0.01797679, -0.0983202 , -0.00569021,  0.07048442,  0.0846359
4,
       -0.05371277,  0.09819099, -0.09880042, -0.03294989,  0.0578516
2,
       -0.08029769, -0.01153355, -0.02386614,  0.06651969,  0.0423924
4,
       -0.02899984, -0.09507273,  0.0387025 , -0.01575761,  0.0716698
9,
        0.01501471,  0.08801247,  0.0760511 , -0.00837099, -0.0965379
8,
       -0.08626907, -0.0358784 ,  0.01846038,  0.02201049,  0.0024855
,
       -0.07104871,  0.07536174,  0.04764704, -0.02533592,  0.0657372
8,
       -0.0819521 , -0.08719935,  0.00117939,  0.08668534, -0.0702076
2],
      dtype=float32)
```

In [5]:

```
word_vectors
```

Out[5]:

```
<gensim.models.keyedvectors.Word2VecKeyedVectors at 0x7feb170436d0>
```

In [6]:

```
word_vectors.similar_by_word("books")
```

Out[6]:

```
[('essays', 0.8325610160827637),
 ('articles', 0.8254350423812866),
 ('novellas', 0.7723618745803833),
 ('magazines', 0.7717729806900024),
 ('monographs', 0.76274573802948),
 ('illustrating', 0.7334399819374084),
 ('biographies', 0.7303065061569214),
 ('journals', 0.7293117046356201),
 ('pages', 0.7265263199806213),
 ('columns', 0.7254587411880493)]
```

For the word ‘book’, the qualitatively most similar words are mostly the nouns with the semantic meaning ‘things can be read’ like essays, articles.

In [7]:

```
word_vectors.similar_by_word("April")
```

Out[7]:

```
[('October', 0.8920010328292847),
 ('December', 0.8778380155563354),
 ('February', 0.8765127062797546),
 ('August', 0.8728427886962891),
 ('November', 0.8700879812240601),
 ('March', 0.8650733232498169),
 ('June', 0.8603971004486084),
 ('May', 0.8477078676223755),
 ('September', 0.8467978239059448),
 ('July', 0.839765191078186)]
```

For the word ‘April’, the qualitatively most similar words are mostly the nouns indicating month like October.

In [8]:

```
word_vectors.similar_by_word("basketball")
```

Out[8]:

```
[('softball', 0.8399666547775269),
 ('football', 0.8360031843185425),
 ('soccer', 0.8227754831314087),
 ('volleyball', 0.8048578500747681),
 ('baseball', 0.8044962882995605),
 ('lacrosse', 0.7955363988876343),
 ('hockey', 0.7667567133903503),
 ('Goodenbour', 0.7493417263031006),
 ('tennis', 0.741902768611908),
 ('sledge', 0.7320674061775208)]
```

For the word ‘basketball’, the qualitatively most similar words are mostly the nouns indicating sports items like softball.

In [9]:

```
word_vectors.similar_by_word("He")
```

Out[9]:

```
[('She', 0.7889261245727539),
 ('Daccord', 0.6885871291160583),
 ('Lambertsen', 0.6511487364768982),
 ('Hailston', 0.6508619785308838),
 ('Galbally', 0.6429628133773804),
 ('Pepín', 0.6400373578071594),
 ('Romell', 0.6359394788742065),
 ('Moncewicz', 0.6355791091918945),
 ('Mauborgne', 0.6326525211334229),
 ('Dário', 0.6298860907554626)]
```

For the word ‘He’, the qualitatively most similar words are she and other person names.

In [10]:

```
word_vectors.similar_by_word("25th")
```

Out[10]:

```
[('30th', 0.8240333199501038),
 ('64th', 0.8238258361816406),
 ('23rd', 0.8140838146209717),
 ('33rd', 0.8038157820701599),
 ('9th', 0.8005726337432861),
 ('10th', 0.7996435761451721),
 ('116th', 0.7987253069877625),
 ('59th', 0.7891724705696106),
 ('60th', 0.7865711450576782),
 ('36th', 0.7852972149848938)]
```

For the word ‘25th’, the qualitatively most similar words are all ordinal numerals.

In [11]:

```
word_vectors.similar_by_word("1967")
```

Out[11]:

```
[('1962', 0.8537935018539429),
 ('1951', 0.8409759998321533),
 ('1950', 0.8306080102920532),
 ('1955', 0.8281768560409546),
 ('1949', 0.8229819536209106),
 ('1934', 0.8175702691078186),
 ('1933', 0.8130149245262146),
 ('1946', 0.8127145767211914),
 ('1961', 0.8122072219848633),
 ('1959', 0.8111141324043274)]
```

For the word ‘1967’, the qualitatively most similar words are all years.

In [12]:

```
word_vectors.similar_by_word("mathematics")
```

Out[12]:

```
[('physics', 0.8251581192016602),
 ('sociology', 0.8141902685165405),
 ('medicine', 0.8090916872024536),
 ('zoology', 0.7975834608078003),
 ('chemistry', 0.7947500348091125),
 ('economics', 0.7866847515106201),
 ('geography', 0.7846213579177856),
 ('botany', 0.7817147970199585),
 ('engineering', 0.7719976902008057),
 ('divinity', 0.7719128131866455)]
```

For the word ‘mathematics’, the qualitatively most similar words are mostly subjects.

In [13]:

```
word_vectors.similar_by_word("Between")
```

Out[13]:

```
[('From', 0.7463487386703491),
 ('Until', 0.6833612322807312),
 ('By', 0.6592501401901245),
 ('Excavations', 0.634077787399292),
 ('Obergruppenführer', 0.6191043853759766),
 ('Appointed', 0.617178201675415),
 ('Starting', 0.6139140129089355),
 ('Saydam', 0.6130807399749756),
 ('Participated', 0.6110838055610657),
 ('Düren', 0.6091176867485046)]
```

For the word ‘Between’, the first three qualitatively most similar words are prepositions.

In [14]:

```
word_vectors.similar_by_word("Finland")
```

Out[14]:

```
[('Czechoslovakia', 0.8479709029197693),
 ('Croatia', 0.7815450429916382),
 ('Latvia', 0.7672635912895203),
 ('Switzerland', 0.7577182054519653),
 ('Yugoslavia', 0.7551692128181458),
 ('Ukraine', 0.7487151026725769),
 ('Slovenia', 0.7282400131225586),
 ('Morocco', 0.7266643047332764),
 ('Moldova', 0.7254005074501038),
 ('Serbia', 0.7202527523040771)]
```

For the word ‘Finland’, the qualitatively most similar words are other countries.

In [15]:

```
word_vectors.similar_by_word("pewter")
```

Out[15]:

```
[('Hockenheimring', 0.7922357320785522),
 ('criterium', 0.7663637399673462),
 ('Nurburgring', 0.7603985071182251),
 ('eights', 0.7548940181732178),
 ('10m', 0.750497579574585),
 ('T37', 0.7494640946388245),
 ('placings', 0.7486727237701416),
 ('Alm', 0.7444251775741577),
 ('hoop', 0.743412971496582),
 ('relays', 0.7425753474235535)]
```

For the rare word ‘Pewter’, the qualitatively most similar words are not similar to it since the frequency of it is too low.

In [16]:

```
def get_analogy(a, b, c):
    return word_vectors.most_similar(positive=[b, c], negative=[a])[0][0]
```

In [17]:

```
get_analogy('man','woman','king')
```

Out[17]:

```
'queen'
```

In [18]:

```
get_analogy('boy','girl','Prince')
```

Out[18]:

```
'Princess'
```

Prince-boy+girl=princess

In [19]:

```
get_analogy('boy','girl','actor')
```

Out[19]:

```
'actress'
```

actor-boy+girl=actress

In [20]:

```
get_analogy('snail','rabbit','slow')
```

Out[20]:

```
'fast'
```

slow-snail+rabbit=fast. Rabbit is fast, snail is slow

In [34]:

```
get_analogy('father','mother','man')
```

Out[34]:

```
'woman'
```

man-father+mother=woman

In [42]:

```
get_analogy('sing','teach','singer')
```

Out[42]:

```
'educator'
```

singer-sing+teach=educator. People who sing well is a singer. People who teach well is educator

In [22]:

```
get_analogy('winter','summer','December')
```

Out[22]:

```
'June'
```

December-winter+summer=June. December is the beginning of winter and June is the beginning of summer.

In [ ]:

```

```