In [5]:
```
import numpy as np
import keras.backend as K
from keras.models import Sequential
from keras.layers import Dense, Embedding, Lambda
from keras.utils import np_utils
from keras.preprocessing import sequence
from keras.preprocessing.text import Tokenizer
import gensim
```

In [6]:
```
data=open('covid.txt','r')
corona_data = [text for text in data if text.count(' ') >= 2]
vectorize = Tokenizer()
vectorize.fit_on_texts(corona_data)
corona_data = vectorize.texts_to_sequences(corona_data)
total_vocab = sum(len(s) for s in corona_data)
word_count = len(vectorize.word_index) + 1
window_size = 2
```

In [7]:
```
def cbow_model(data, window_size, total_vocab):
    total_length = window_size*2
    for text in data:
        text_len = len(text)
        for idx, word in enumerate(text):
            context_word = []
            target   = []
            begin = idx - window_size
            end = idx + window_size + 1
            context_word.append([text[i] for i in range(begin, end) if 0 <= i < text_l
            target.append(word)
            contextual = sequence.pad_sequences(context_word, total_length=total_lengt
            final_target = np_utils.to_categorical(target, total_vocab)
            yield(contextual, final_target)
```

In [8]:
```
model = Sequential()
model.add(Embedding(input_dim=total_vocab, output_dim=100, input_length=window_size*2)
model.add(Lambda(lambda x: K.mean(x, axis=1), output_shape=(100,)))
model.add(Dense(total_vocab, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam')
for i in range(10):
    cost = 0
    for x, y in cbow_model(data, window_size, total_vocab):
        cost += model.train_on_batch(contextual, final_target)
    print(i, cost)
```

```
0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
```

In [13]:
```
dimensions=100
vect_file = open('vectors.txt' ,'w')
```

```
vect_file.write('{} {}\n'.format(total_vocab,dimensions))
```

Out[13]:
```
8
```

In [14]:
```
weights = model.get_weights()[0]
for text, i in vectorize.word_index.items():
    final_vec = ' '.join(map(str, list(weights[i, :])))
    vect_file.write('{} {}\n'.format(text, final_vec))
vect_file.close()
```

In [19]:
```
cbow_output = gensim.models.KeyedVectors.load_word2vec_format('vectors.txt', binary=Fa
cbow_output.most_similar(positive=['virus'])
```

Out[19]:
```
[('covid', 0.223189115524292),
 ('understood', 0.20812445878982544),
 ('-', 0.20038005709648132),
 ('or', 0.18580959737300873),
 ('interval', 0.1851491630077362),
 ('5', 0.17845691740512848),
 ('successive', 0.17402121424674988),
 ('we', 0.16869743168354034),
 ('median', 0.1599927544593811),
 ('19', 0.15923653542995453)]
```

In [ ]: