Knapp, Stephen

Deep Learning Assignment 9

In [0]:
```
import numpy as np
import gensim
# Get the interactive Tools for Matplotlib
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from gensim.test.utils import datapath, get_tmpfile
from gensim.models import KeyedVectors
from gensim.scripts.glove2word2vec import glove2word2vec
import spacy
from spacy.lang.en import English
```

In [2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client
_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&
redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=code&scope=ema
il%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.
googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdri
ve.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.read
only

Enter your authorization code:
··········
Mounted at /content/drive
```

Problem 1

Cygwin code:

$make$ time ./word2vec -train text8 -output vectors.bin -cbow 1 -size 300 -window 8 -negative 25 -hs 0 -sample 1e-4 -threads 20 -binary 0 -iter 15 Starting training using file text8 Vocab size: 71291 Words in train file: 16718843 Alpha: 0.000005 Progress: 100.10% Words/thread/sec: 40.90k real 15m55.780s user 104m50.421s sys 0m6.765s $ ./word-analogy.exe GoogleNews-vectors-negative300.bin Enter three words (EXIT to break): france paris russia

Word: france Position in vocabulary: 225534

Word: paris Position in vocabulary: 198365

Word: russia Position in vocabulary: 294451

```
                                            Word              Distance
```

```
       north_korea       0.471760
        tom_cruise       0.449580
             lohan       0.448753
              joel       0.445634
     lindsay_lohan       0.445479
             heidi       0.440514
         megan_fox       0.438607
           britney       0.429737
          russians       0.429316
            moscow       0.428812
   natalie_portman       0.426762
           lil_kim       0.425724
           rihanna       0.425628
           lindsay       0.420228
          thailand       0.419693
               wiv       0.417865
            jessie       0.417515
     nicole_richie       0.417459
           lindsey       0.417458
         alexandra       0.416858
         christina       0.415890
           whitney       0.415334
      paris_hilton       0.415204
         brad_pitt       0.413550
            monica       0.412652
               tmz       0.411524
             paula       0.410811
              gwen       0.409483
     charlie_sheen       0.409303
            hilton       0.409228
     washington_dc       0.408328
            ronnie       0.407938
             nikki       0.407641
           ukraine       0.407487
           natalie       0.406954
       barbie_doll       0.406901
           michele       0.406771
    angelina_jolie       0.406008
              ktla       0.405925
  jennifer_aniston       0.404747
```

Enter three words (EXIT to break): USA washington_dc UK

Word: USA Position in vocabulary: 2276

Word: washington_dc Position in vocabulary: 840882

Word: UK Position in vocabulary: 928

Word	Distance
tesco	0.492387
scotland	0.487275
gordon_brown	0.486360
british	0.486181
britain	0.473281
westminster	0.470460
london	0.468258
barclays	0.458864
washington	0.456587
uk	0.451748
tasmania	0.448211
somerset	0.447870
malta	0.444047
northern_ireland	0.443387
Heddlu	0.440386
UKs	0.439778
russell	0.439609
Uk	0.438327
Fylde_coast	0.436810
£_6billion	0.431206
lehman_brothers	0.430657
albuquerque	0.430636
australian	0.430463
essex	0.428367
taj_mahal	0.427947
wot	0.424913
adams	0.424050
europe	0.424024
Francis_Maude	0.423974
blackpool	0.423024
el_paso	0.422961
devon	0.422773
dodd	0.422234
ron_paul	0.422220
sri_lanka	0.422184
XXXXXXing	0.421631
£_###k	0.420904
iraqi	0.420791
belfast	0.420452
hev	0.420234

Enter three words (EXIT to break): china beijing italy

Word: china Position in vocabulary: 32952

Word: beijing Position in vocabulary: 537874

Word: italy Position in vocabulary: 283535

Word	Distance
barcelona	0.566662
diego	0.534301
spain	0.525933
montreal	0.510384
sweden	0.509806
real_madrid	0.504179
orlando	0.502051
inter_milan	0.501926
croatia	0.500982
juve	0.499925
ronaldo	0.493477
luis	0.493169
lebron	0.491013
ac_milan	0.490685
europe	0.489157
france	0.488852
madrid	0.488293
epl	0.488080
forza	0.486562
lyon	0.485370
bayern	0.482659
milano	0.482116
santa_cruz	0.481644
malta	0.480842
carlos	0.480308
perth	0.480103
argentina	0.479513
italian	0.478078
zidane	0.477602
italians	0.475284
athens	0.474737
usa	0.474526
portuguese	0.473479
eto'o	0.472582
minutos	0.472089
portugal	0.471547
ireland	0.471434
liverpool	0.471409
holland	0.471090
greece	0.469304

Enter three words (EXIT to break): Canada ottawa spain

Word: Canada Position in vocabulary: 732

Word: ottawa Position in vocabulary: 572391

Word: spain Position in vocabulary: 261628

```
                                              Word       Distance
------------------------------------------------------------------------
                                            madrid		0.646454
                                            carlos		0.641303
                                           sanchez		0.632822
                                          valencia		0.616917
                                              alex		0.609036
                                          florence		0.605642
                                             diego		0.603246
                                          martinez		0.600520
                                          thompson		0.598823
                                           holland		0.597541
                                              luis		0.595870
                                         barcelona		0.593049
                                          williams		0.590833
                                             ramos		0.590415
                                            thomas		0.587581
                                           gilbert		0.587537
                                              raul		0.586900
                                              lyon		0.586287
                                              juan		0.585569
                                         hernandez		0.583346
                                        birmingham		0.583141
                                              dunn		0.582263
                                          columbia		0.582214
                                              jose		0.581086
                                            samuel		0.581084
                                           orlando		0.580889
                                                os		0.580558
                                            athens		0.580102
                                            miguel		0.580005
                                           rosario		0.579041
                                            joseph		0.578241
                                            bolton		0.575849
                                             marco		0.575431
                                        portsmouth		0.574388
                                            jacobs		0.572217
                                            walton		0.572086
                                             perez		0.571837
                                            arthur		0.570803
                                             eddie		0.570745
                                            torres		0.570596
```

Modified demo-word.sh file:

make if [! -e text8]; then wget http://mattmahoney.net/dc/text8.zip (http://mattmahoney.net/dc/text8.zip) -O text8.gz gzip -d text8.gz -f fi time ./word2vec -train text8 -output vectors.txt -cbow 1 -size 300 -window 8 -negative 25 -hs 0 -sample 1e-4 -threads 20 -binary 0 -iter 15 ./distance vectors.bin

Cygwin code:

$./demo-word.sh

Last line of output vectors.txt file:

-0.181675 0.285970 -0.012680 -0.157904 0.222176 0.106366 0.131103 -0.100605 -0.118704 0.400066 -0.170049 0.627699 0.006325 -0.150162 -0.406182 -0.049522 -0.192137 0.328455 -0.140763 0.288435 0.150895 0.046974 -0.003062 0.000353 0.288762 -0.218790 -0.183712 0.672979 -0.021745 0.292123 -0.223208 0.080553 0.309973 -0.111925 0.222427 0.281203 -0.405554 -0.262017 -0.035659 0.022486 -0.189647 0.106794 0.467505 -0.274588 -0.093628 0.260080 -0.231895 -0.209200 0.073046 -0.348545 0.413612 0.082884 0.067275 -0.001451 0.175938 -0.154404 -0.123044 0.191053 -0.240070 0.124044 -0.150376 -0.225028 -0.107731 0.126623 -0.100367 -0.267967 -0.127621 -0.006478 0.094554 0.015414 -0.434812 -0.095637 -0.215647 -0.057668 0.312453 0.088009 -0.401597 -0.128320 -0.240175 -0.520783 -0.161839 -0.391290 -0.272384 -0.044675 0.089029 0

Problem 2

In [3]:
```
glove_file = 'drive/My Drive/Colab Notebooks/glove.6B.100d.txt'
word2vec_glove_file = get_tmpfile("glove.6B.100d.word2vec.txt")
glove2word2vec(glove_file, word2vec_glove_file)
```

```
/usr/local/lib/python3.6/dist-packages/smart_open/smart_open_lib.py:253: User
Warning: This function is deprecated, use smart_open.open instead. See the mi
gration notes for details: https://github.com/RaRe-Technologies/smart_open/bl
ob/master/README.rst#migrating-to-the-new-open-function
  'See the migration notes for details: %s' % _MIGRATION_NOTES_URL
```

Out[3]:
```
(400000, 100)
```

In [4]:
```
model = KeyedVectors.load_word2vec_format(word2vec_glove_file)
```

```
/usr/local/lib/python3.6/dist-packages/smart_open/smart_open_lib.py:253: User
Warning: This function is deprecated, use smart_open.open instead. See the mi
gration notes for details: https://github.com/RaRe-Technologies/smart_open/bl
ob/master/README.rst#migrating-to-the-new-open-function
  'See the migration notes for details: %s' % _MIGRATION_NOTES_URL
```

In [0]:
```
capitals = ['canada', 'ottawa', 'spain', 'madrid', 'france', 'paris', 'russia'
, 'moscow',
         'usa', 'washington', 'uk', 'london', 'china', 'beijing', 'italy', 'ro
me']
```

In [0]:
```
words = capitals
```

In [0]:
```
word_vectors = np.array([model[w] for w in words])
twodim = PCA().fit_transform(word_vectors)[:,:2]
```

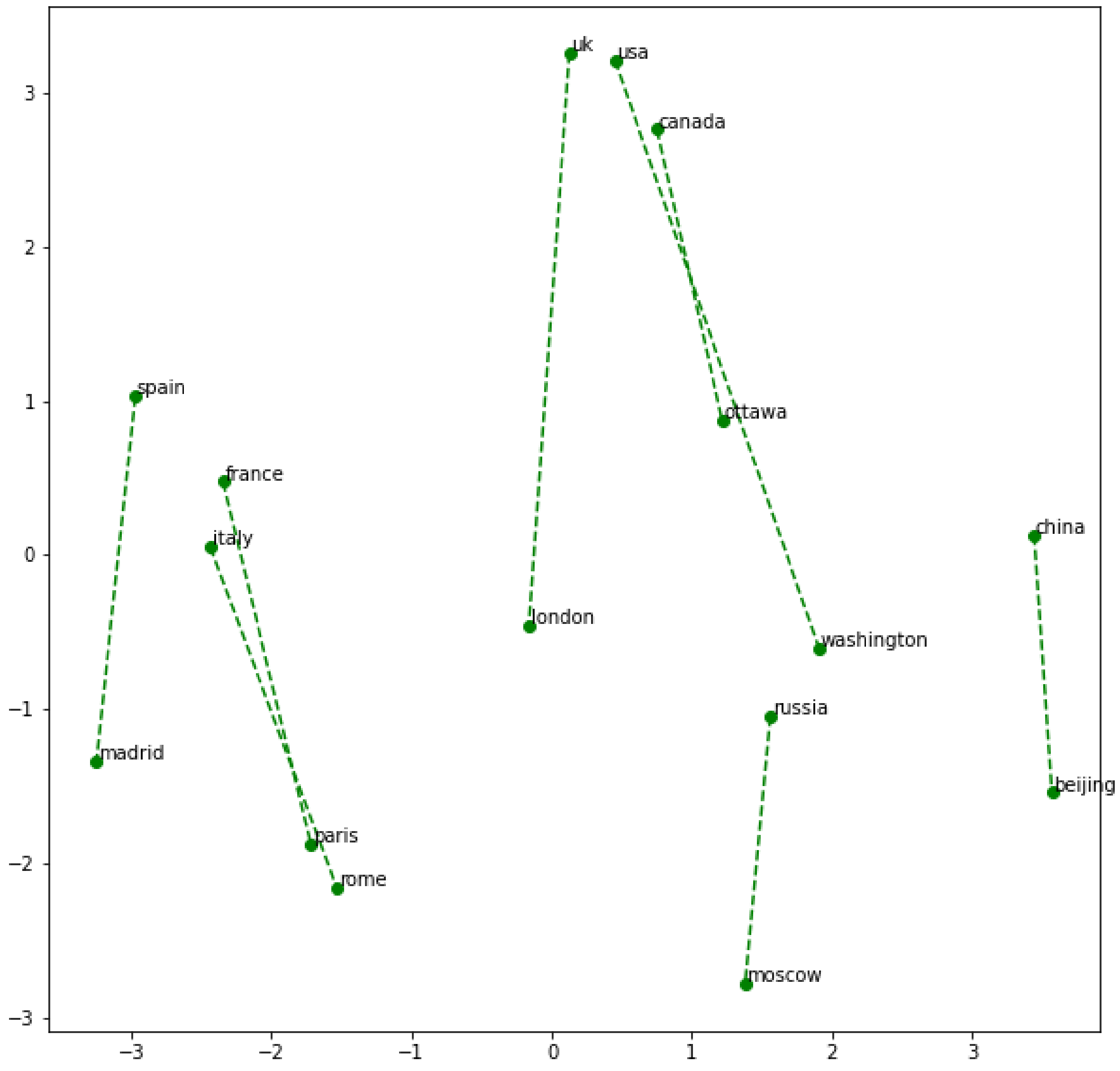
In [0]:
```
def divide_chunks(l, n):

    # looping till length l
    for i in range(0, len(l), n):
        yield l[i:i + n]

# How many elements each
# list should have
n = 2
```

In [0]:
```
twodimpoints = list(divide_chunks(twodim, n))
```

In [10]:
```
pairs=len(words)/2
i=0
j=0
plt.figure(figsize=(10,10))
while i < pairs:
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  j = j + 2
  i = i + 1;
plt.show();
```



The lines are relatively parallel to each other. For each the country is the higher points and the capital is the lower points. The lines range from approximately -75 to -105 degrees (relative to the country point).

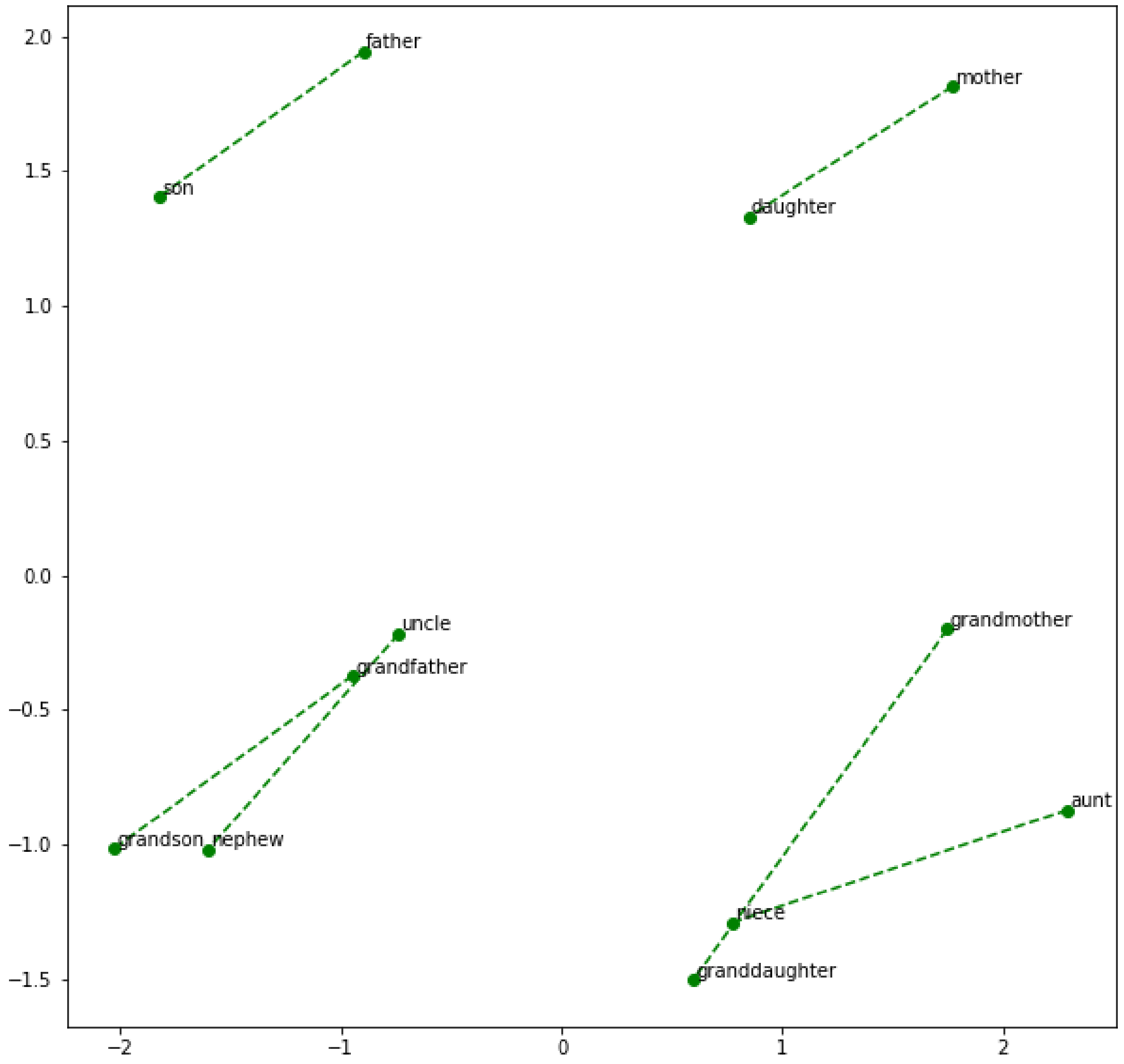
In [0]:
```
relatives = ['father', 'son', 'mother', 'daughter', 'grandmother', 'granddaugh
ter', 'grandfather', 'grandson',
         'uncle', 'nephew', 'aunt', 'niece']
```

In [0]:
```
words = relatives
```

In [0]:
```
word_vectors = np.array([model[w] for w in words])
twodim = PCA().fit_transform(word_vectors)[:,:2]
```

In [0]:
```
twodimpoints = list(divide_chunks(twodim, n))
```

In [15]:
```
pairs=len(words)/2
i=0
j=0
plt.figure(figsize=(10,10))
while i < pairs:
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  j = j + 2
  i = i + 1;
plt.show();
```



Here we see again almost parallel lines. The older person's title is higher and to the right of the younger person's title. The lines vary from approximately 30 to 45 degrees (from the younger title to the older).

Problem 3

In [16]:
```
glove_file = 'drive/My Drive/Colab Notebooks/glove.6B.300d.txt'
word2vec_glove_file = get_tmpfile("glove.6B.300d.word2vec.txt")
glove2word2vec(glove_file, word2vec_glove_file)
```

```
/usr/local/lib/python3.6/dist-packages/smart_open/smart_open_lib.py:253: User
Warning: This function is deprecated, use smart_open.open instead. See the mi
gration notes for details: https://github.com/RaRe-Technologies/smart_open/bl
ob/master/README.rst#migrating-to-the-new-open-function
  'See the migration notes for details: %s' % _MIGRATION_NOTES_URL
```

Out[16]:
```
(400000, 300)
```

In [17]:
```
model = KeyedVectors.load_word2vec_format(word2vec_glove_file, limit=143488, u
nicode_errors='ignore')
```

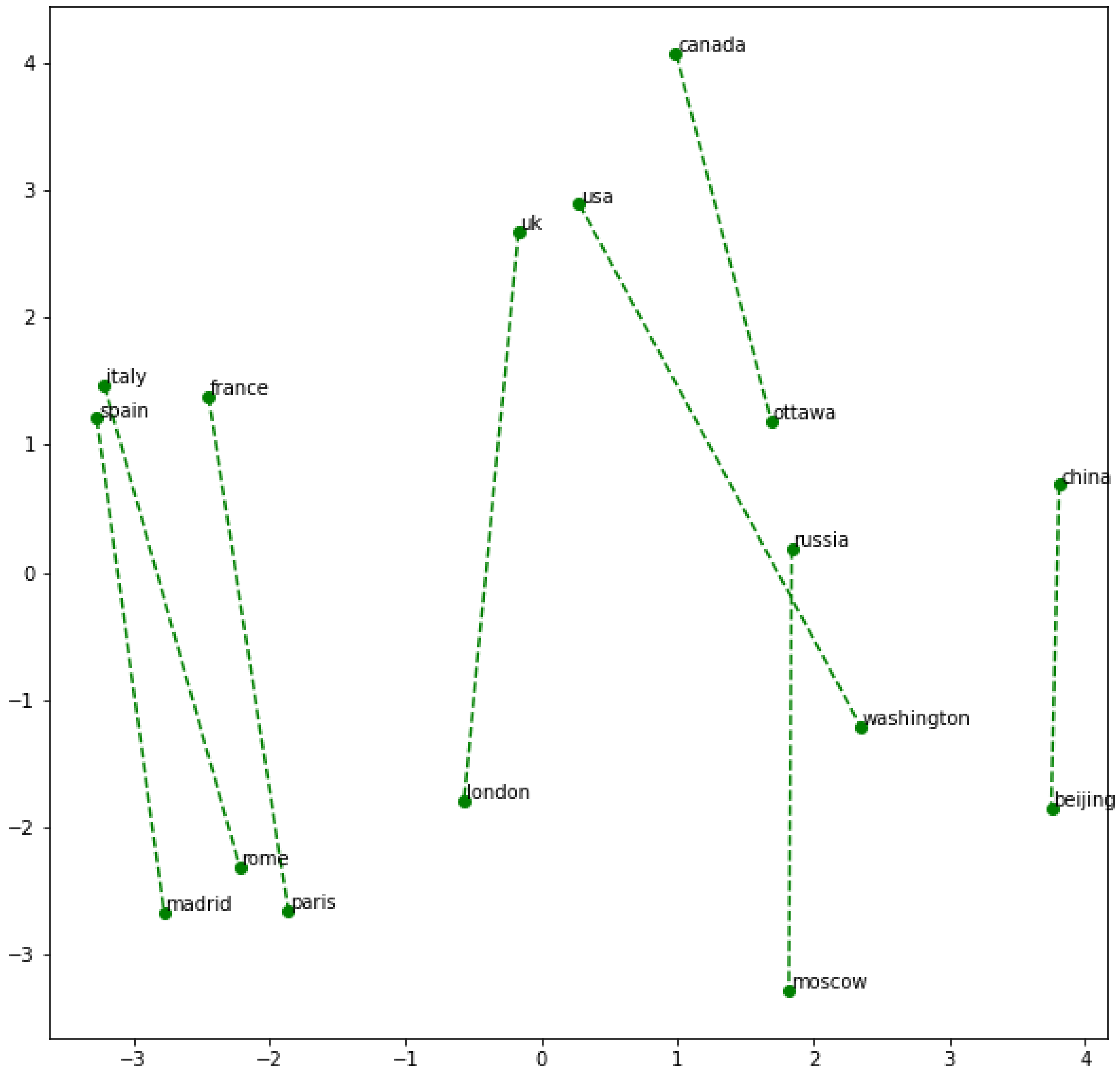
```
/usr/local/lib/python3.6/dist-packages/smart_open/smart_open_lib.py:253: User
Warning: This function is deprecated, use smart_open.open instead. See the mi
gration notes for details: https://github.com/RaRe-Technologies/smart_open/bl
ob/master/README.rst#migrating-to-the-new-open-function
  'See the migration notes for details: %s' % _MIGRATION_NOTES_URL
```

In [0]:
```
words = capitals
```

In [0]:
```
word_vectors = np.array([model[w] for w in words])
twodim = PCA().fit_transform(word_vectors)[:,:2]
```

In [0]:
```
twodimpoints = list(divide_chunks(twodim, n))
```

In [21]:
```
pairs=len(words)/2
i=0
j=0
plt.figure(figsize=(10,10))
while i < pairs:
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  j = j + 2
  i = i + 1;
plt.show();
```



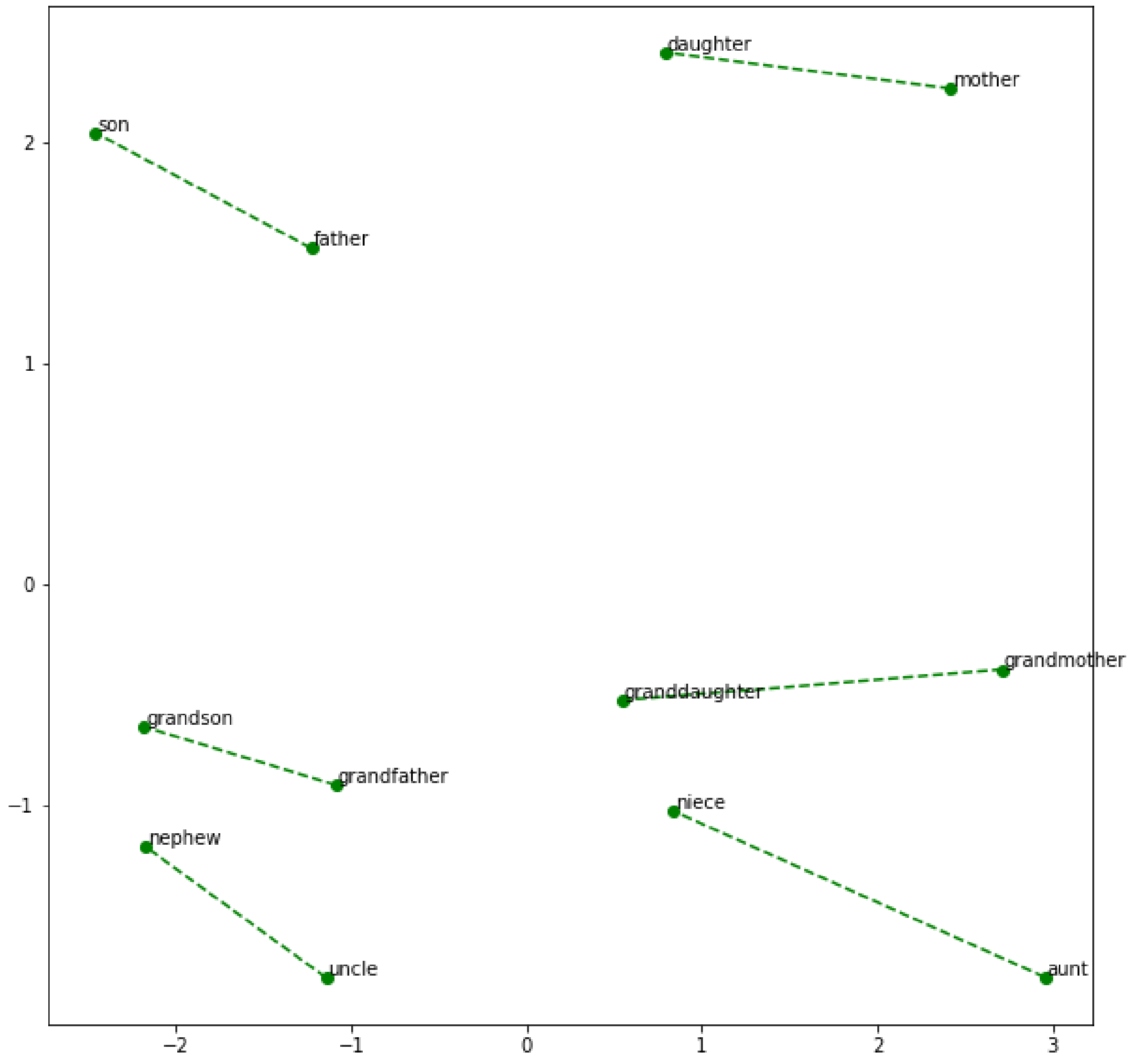
Using the 300 dimensional vector we see much more parallelism between the lines.

In [0]:
```
words = relatives
```

In [0]:
```
word_vectors = np.array([model[w] for w in words])
twodim = PCA().fit_transform(word_vectors)[:,:2]
```

In [0]:
```
twodimpoints = list(divide_chunks(twodim, n))
```

In [25]:
```
pairs=len(words)/2
i=0
j=0
plt.figure(figsize=(10,10))
while i < pairs:
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  j = j + 2
  i = i + 1;
plt.show();
```



In this case the results look very similar to using the 100 dimensional vector with respect to parallelism between the lines.

Problem 4

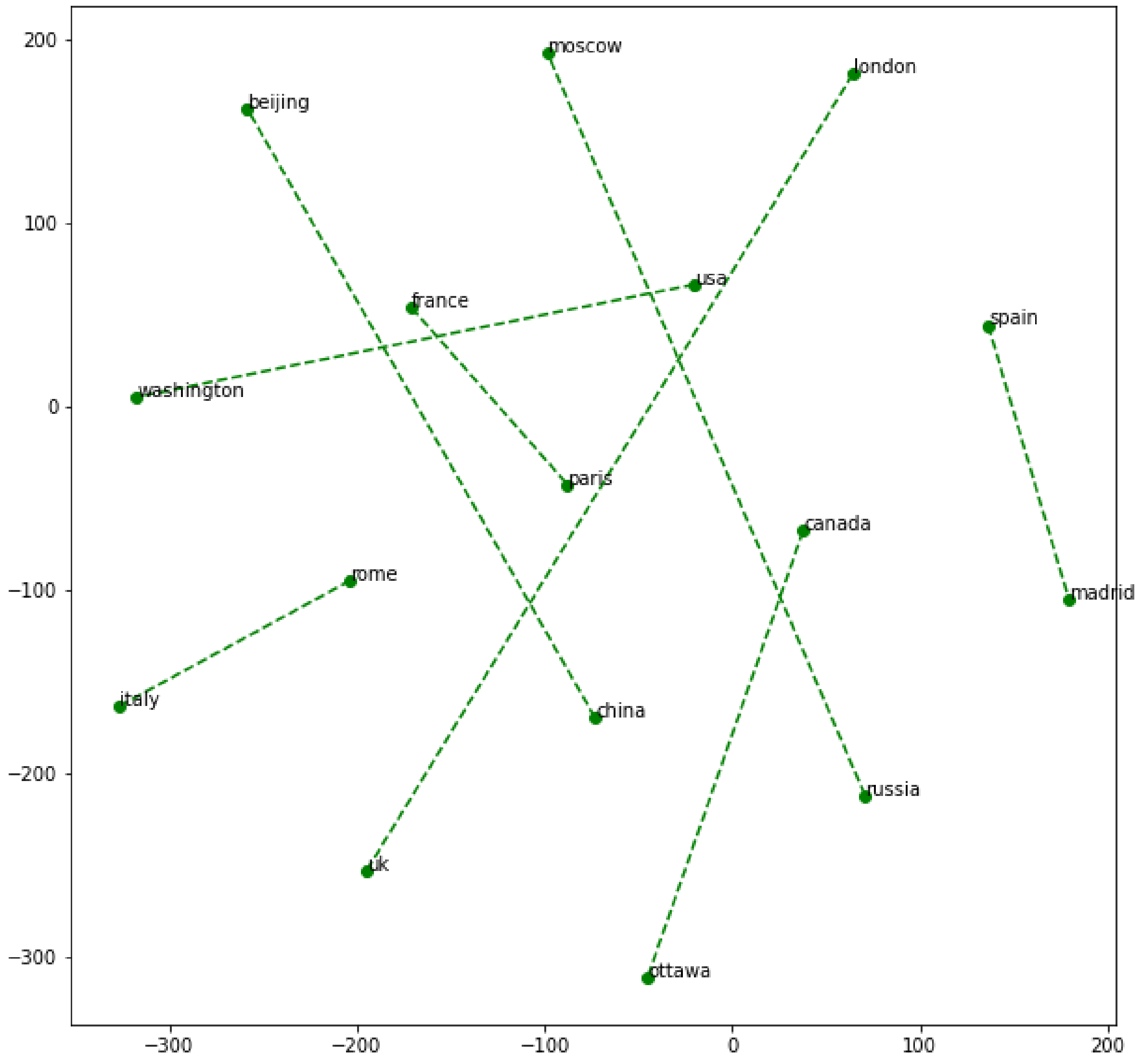
In [0]:
```
words = capitals
```

In [0]:
```
word_vectors = np.array([model[w] for w in words])
```

In [0]:
```
X_embedded = TSNE(n_components=2, perplexity=50.0).fit_transform(word_vectors)
```

In [0]:
```
twodimpoints = list(divide_chunks(X_embedded, n))
```

In [30]:
```
pairs=len(words)/2
i=0
j=0
plt.figure(figsize=(10,10))
while i < pairs:
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  j = j + 2
  i = i + 1;
plt.show();
```



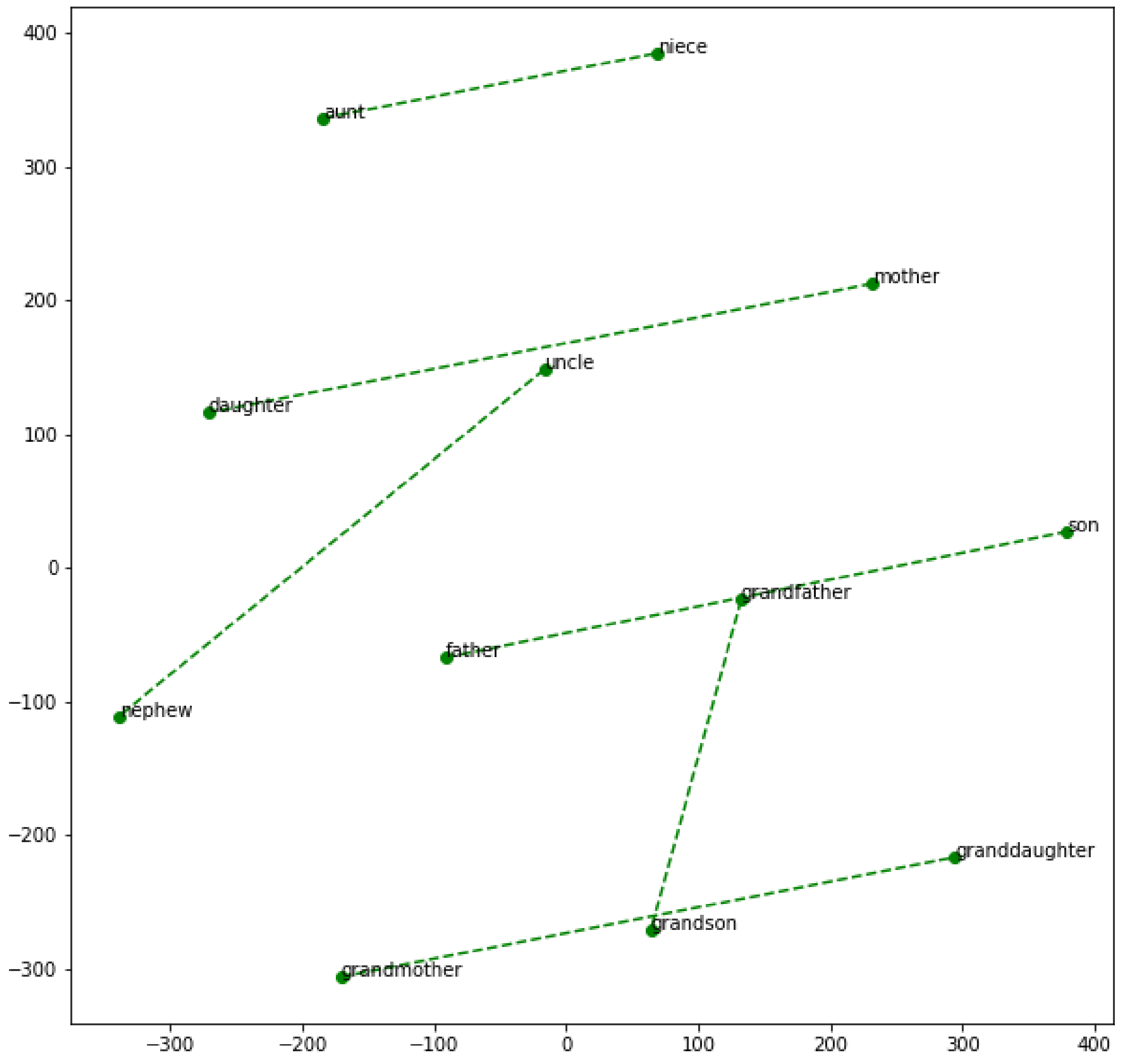
The parallelization pattern is almost non-existent despite trying a range of perplexities from 5 to 300.

In [0]:
```
words = relatives
```

In [0]:
```
word_vectors = np.array([model[w] for w in words])
X_embedded = TSNE(n_components=2).fit_transform(word_vectors)
```

In [0]:
```
twodimpoints = list(divide_chunks(X_embedded, n))
```

In [34]:
```
pairs=len(words)/2
i=0
j=0
plt.figure(figsize=(10,10))
while i < pairs:
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  j = j + 2
  i = i + 1;
plt.show();
```



Here we see strong parallelism between 4 of the relationships but then the other two are almost perpendicular to them. There is much less of an obvious relationship using the T-SNE method.

Problem 5

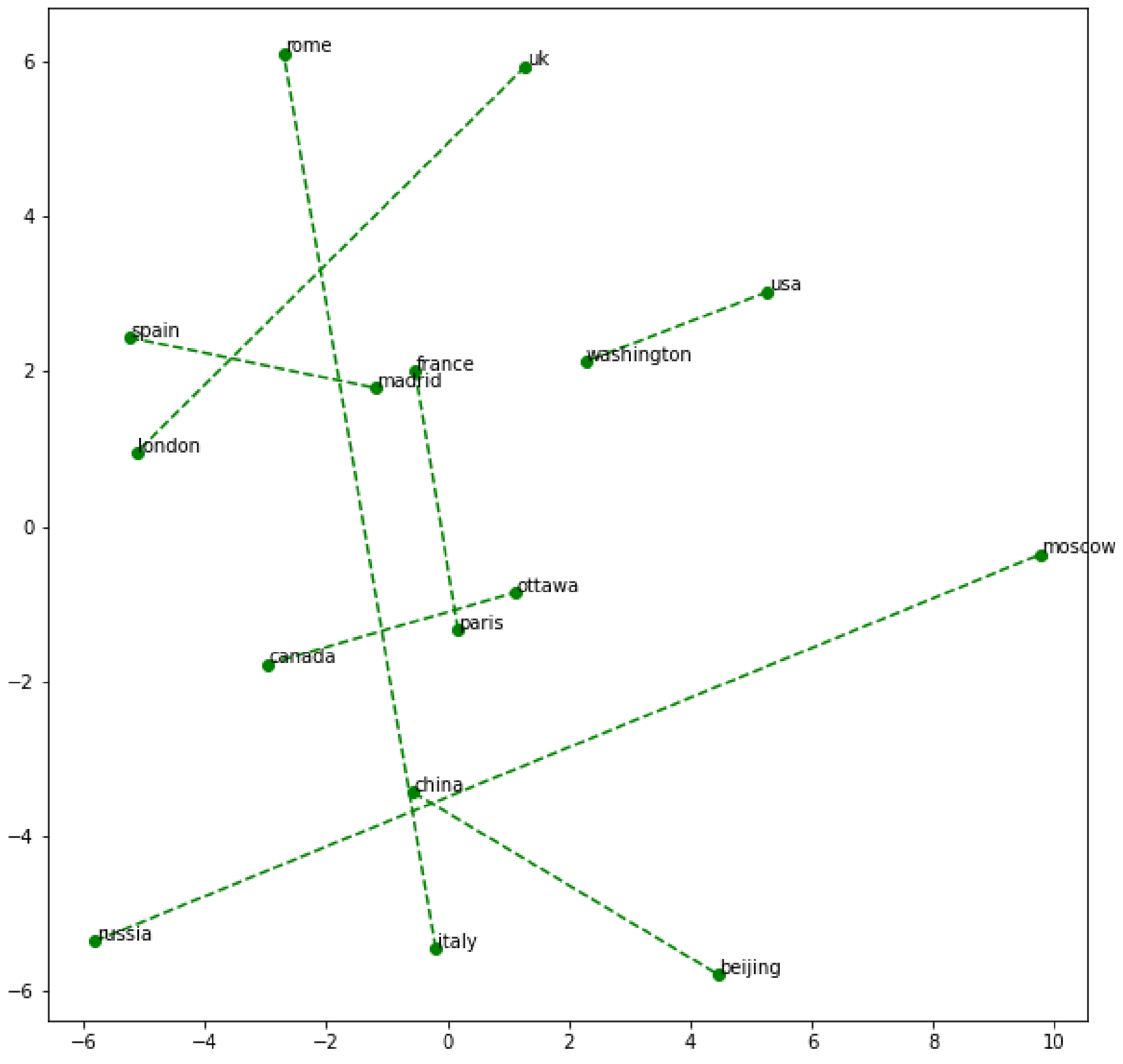
In [0]:
```
#set parser to english
parser = spacy.load('en')
#load the model
nlp = spacy.load("en_core_web_sm")
```

In [0]:
```
#define word bank
words = capitals
```

In [0]:
```
#turn word strings into arrays
word_vectors = [nlp(w).vector for w in words]
#analyze relative distances between words using PCA
twodim = PCA().fit_transform(word_vectors)[:,:2]
```

In [0]:
```
#chunk locations into word pairs
twodimpoints = list(divide_chunks(twodim, n))
```

In [88]:
```
#set number of word pairs
pairs=len(words)/2
#initialize counter for words
i=0
#initialize counter for word pairs
j=0
#provide plotting canvas
plt.figure(figsize=(10,10))
#while loop to plot all pairs and draw lines
while i < pairs:
  #plot a pair with line connecting
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  #add point labels
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  #advance both counters
  j = j + 2
  i = i + 1;
#now show the plot
plt.show();
```



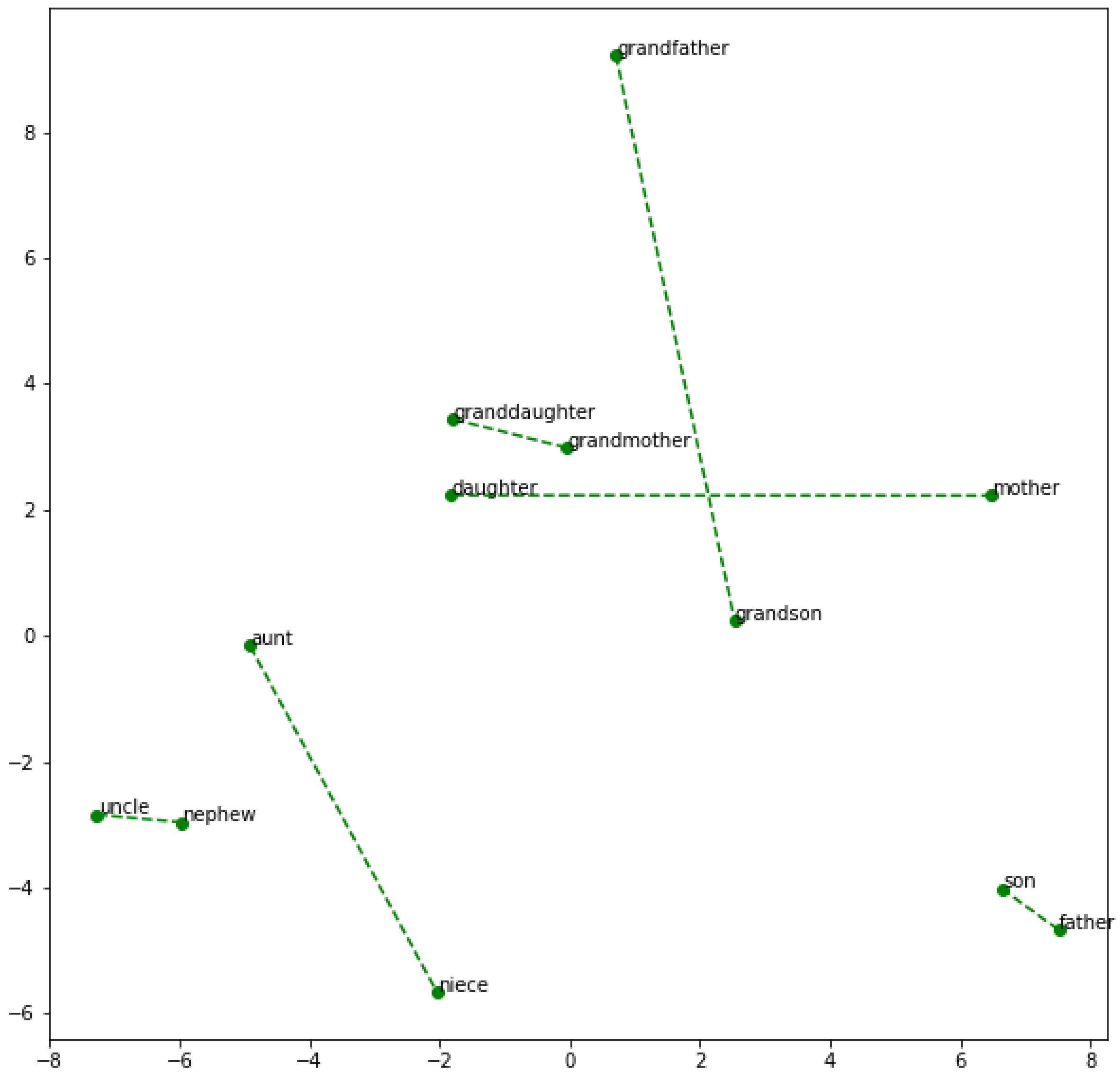
There is essentially no descernable parallel pattern.

In [0]:
```
#define word bank
words = relatives
```

In [0]:
```
#turn word strings into arrays
word_vectors = [nlp(w).vector for w in words]
#analyze relative distances between words using PCA
twodim = PCA().fit_transform(word_vectors)[:,:2]
```

In [0]:
```
#chunk locations into word pairs
twodimpoints = list(divide_chunks(twodim, n))
```

In [92]:
```
#set number of word pairs
pairs=len(words)/2
#initialize counter for words
i=0
#initialize counter for word pairs
j=0
#provide plotting canvas
plt.figure(figsize=(10,10))
#while loop to plot all pairs and draw lines
while i < pairs:
  #plot a pair with line connecting
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  #add point labels
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  #advance both counters
  j = j + 2
  i = i + 1;
#now show the plot
plt.show();
```



There is no discernable parallel pattern using the small model.

In [71]:
```
#download model
!python -m spacy download en_vectors_web_lg
```

```
Collecting en_vectors_web_lg==2.1.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_
vectors_web_lg-2.1.0/en_vectors_web_lg-2.1.0.tar.gz (661.8MB)
     |████████████████████████████████| 661.8MB 1.1MB/s
Requirement already satisfied: spacy<3.0.0,>=2.1.0 in /usr/local/lib/python3.
6/dist-packages (from en_vectors_web_lg==2.1.0) (2.2.4)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/py
thon3.6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0)
(1.0.2)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.
6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (2.0.3)
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /usr/local/lib/python3.
6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (0.4.1)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /usr/local/lib/python3.
6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (1.1.3)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.
6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (4.38.0)
Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.6/dist-
packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (7.4.0)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /usr/local/lib/pyth
on3.6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (1.
0.0)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.6/dist
-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (1.18.2)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /usr/local/lib/python3.
6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (1.0.2)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python
3.6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (3.0.
2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/pyth
on3.6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (2.2
1.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-pa
ckages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (46.1.3)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /usr/local/lib/python
3.6/dist-packages (from spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (0.6.
0)
Requirement already satisfied: importlib-metadata>=0.20; python_version < "3.
8" in /usr/local/lib/python3.6/dist-packages (from catalogue<1.1.0,>=0.0.7->s
pacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (1.6.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.
6/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.0.0,>=2.1.0->en_vector
s_web_lg==2.1.0) (2020.4.5.1)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python
3.6/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.0.0,>=2.1.0->en_vect
ors_web_lg==2.1.0) (3.0.4)
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dis
t-packages (from requests<3.0.0,>=2.13.0->spacy<3.0.0,>=2.1.0->en_vectors_web
_lg==2.1.0) (2.8)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/local/lib/python
3.6/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.0.0,>=2.1.0->en_vect
ors_web_lg==2.1.0) (1.24.3)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.6/dist-pac
kages (from importlib-metadata>=0.20; python_version < "3.8"->catalogue<1.1.
0,>=0.0.7->spacy<3.0.0,>=2.1.0->en_vectors_web_lg==2.1.0) (3.1.0)
Building wheels for collected packages: en-vectors-web-lg
  Building wheel for en-vectors-web-lg (setup.py) ... done
  Created wheel for en-vectors-web-lg: filename=en_vectors_web_lg-2.1.0-cp36-
none-any.whl size=663461747 sha256=03799805c57ba085ddf4784d2f4401102d4db5b259
5e44a6ec15f5fa20f94d72
  Stored in directory: /tmp/pip-ephem-wheel-cache-g8iavzqu/wheels/ce/3e/83/59
647d0b4584003cce18fb68ecda2866e7c7b2722c3ecaddaf
Successfully built en-vectors-web-lg
Installing collected packages: en-vectors-web-lg
Successfully installed en-vectors-web-lg-2.1.0
✔ Download and installation successful
You can now load the model via spacy.load('en_vectors_web_lg')
```

In [104]:
```
!python -m spacy download en
```

```
Requirement already satisfied: en_core_web_sm==2.2.5 from https://github.com/
explosion/spacy-models/releases/download/en_core_web_sm-2.2.5/en_core_web_sm-
2.2.5.tar.gz#egg=en_core_web_sm==2.2.5 in /usr/local/lib/python3.6/dist-packa
ges (2.2.5)
Requirement already satisfied: spacy>=2.2.2 in /usr/local/lib/python3.6/dist-
packages (from en_core_web_sm==2.2.5) (2.2.4)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python
3.6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (3.0.2)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /usr/local/lib/python
3.6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (0.6.0)
Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.6/dist-
packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (7.4.0)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/py
thon3.6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.2)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /usr/local/lib/python3.
6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.1.3)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /usr/local/lib/python3.
6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/pyth
on3.6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (2.21.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-pa
ckages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (46.1.3)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.
6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (2.0.3)
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /usr/local/lib/python3.
6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (0.4.1)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.6/dist
-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.18.2)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /usr/local/lib/pyth
on3.6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (1.0.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.
6/dist-packages (from spacy>=2.2.2->en_core_web_sm==2.2.5) (4.38.0)
Requirement already satisfied: idna<2.9,>=2.5 in /usr/local/lib/python3.6/dis
t-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==2.2.
5) (2.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.
6/dist-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm==
2.2.5) (2020.4.5.1)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /usr/local/lib/python
3.6/dist-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm
==2.2.5) (1.24.3)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /usr/local/lib/python
3.6/dist-packages (from requests<3.0.0,>=2.13.0->spacy>=2.2.2->en_core_web_sm
==2.2.5) (3.0.4)
Requirement already satisfied: importlib-metadata>=0.20; python_version < "3.
8" in /usr/local/lib/python3.6/dist-packages (from catalogue<1.1.0,>=0.0.7->s
pacy>=2.2.2->en_core_web_sm==2.2.5) (1.6.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.6/dist-pac
kages (from importlib-metadata>=0.20; python_version < "3.8"->catalogue<1.1.
0,>=0.0.7->spacy>=2.2.2->en_core_web_sm==2.2.5) (3.1.0)
✓ Download and installation successful
You can now load the model via spacy.load('en_core_web_sm')
✓ Linking successful
/usr/local/lib/python3.6/dist-packages/en_core_web_sm -->
/usr/local/lib/python3.6/dist-packages/spacy/data/en
You can now load the model via spacy.load('en')
```

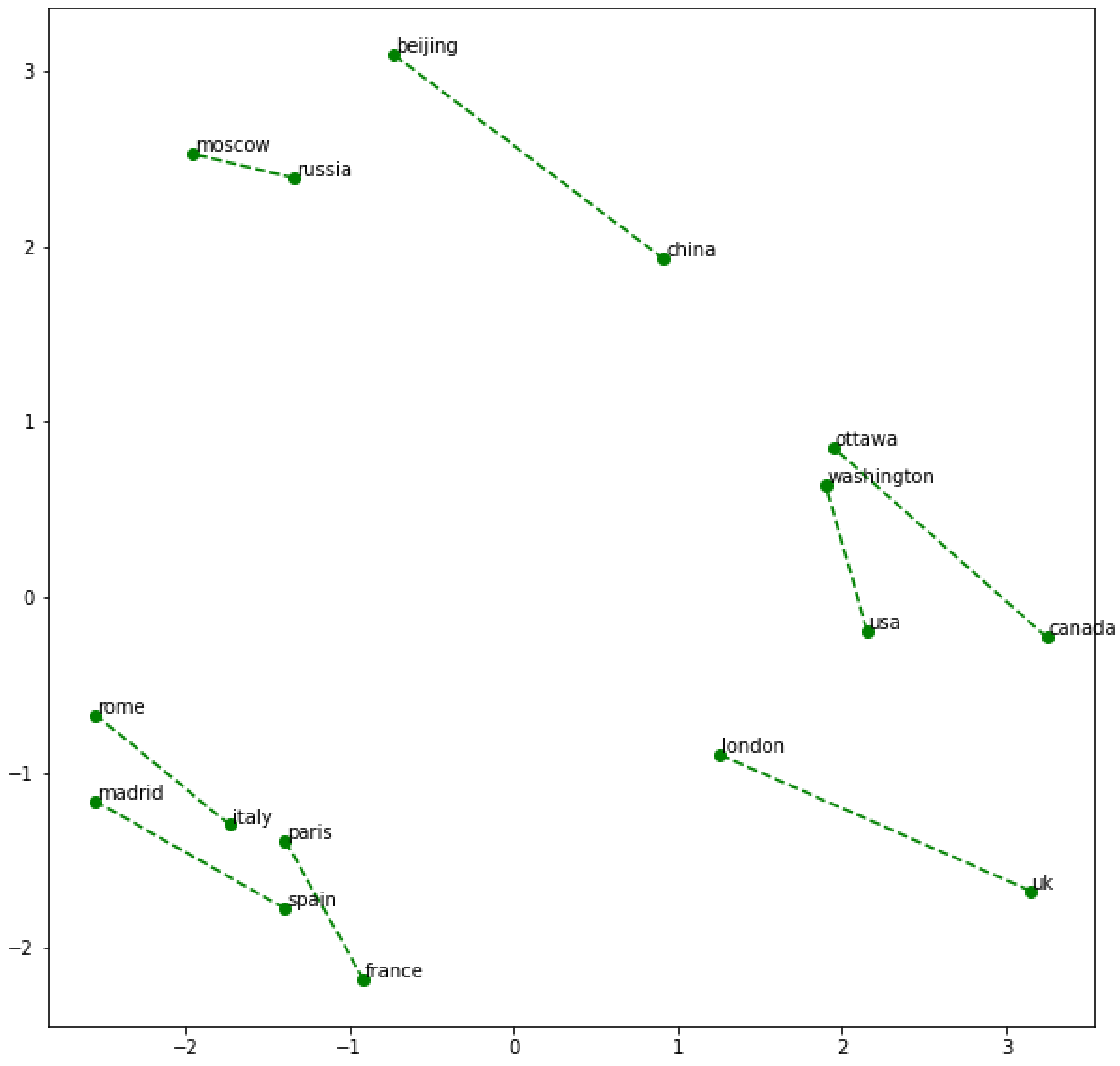
In [0]:
```
#define the model
nlp = en_vectors_web_lg.load()
```

In [0]:
```
#define word bank
words = capitals
```

In [0]:
```
#turn word strings into arrays
word_vectors = [nlp(w).vector for w in words]
#analyze relative distances between words using PCA
twodim = PCA().fit_transform(word_vectors)[:,:2]
```

In [0]:
```
#chunk locations into word pairs
twodimpoints = list(divide_chunks(twodim, n))
```

In [114]:
```
#set number of word pairs
pairs=len(words)/2
#initialize counter for words
i=0
#initialize counter for word pairs
j=0
#provide plotting canvas
plt.figure(figsize=(10,10))
#while loop to plot all pairs and draw lines
while i < pairs:
  #plot a pair with line connecting
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  #add point labels
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  #advance both counters
  j = j + 2
  i = i + 1;
#now show the plot
plt.show();
```



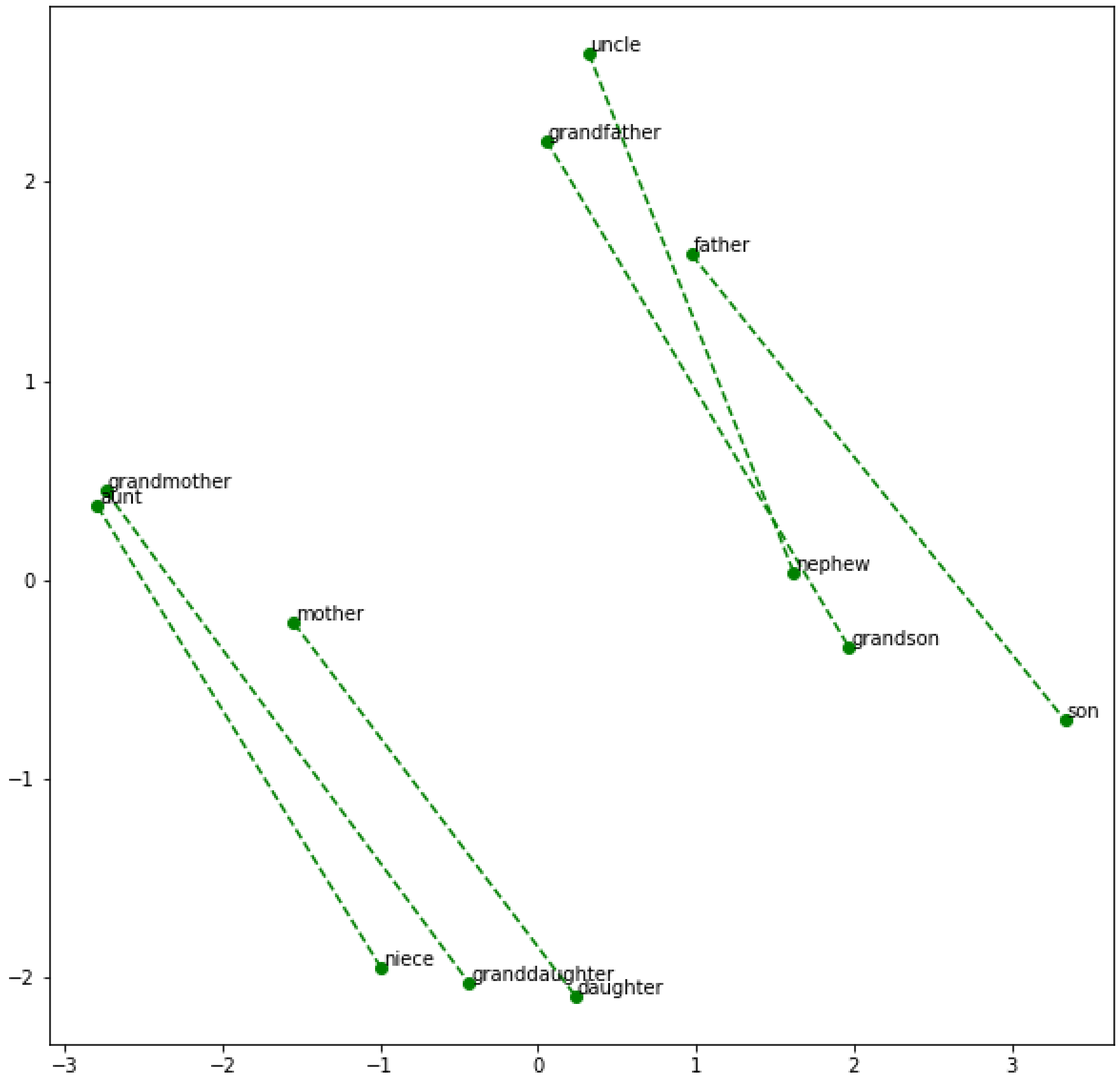
Using the larger model we see much more parallelism. There also appears to be possible grouping based on geography and possibly language.

In [0]:
```
#define word bank
words = relatives
```

In [0]:
```
#turn word strings into arrays
word_vectors = [nlp(w).vector for w in words]
#analyze relative distances between words using PCA
twodim = PCA().fit_transform(word_vectors)[:,:2]
```

In [0]:
```
#chunk locations into word pairs
twodimpoints = list(divide_chunks(twodim, n))
```

In [118]:
```
#set number of word pairs
pairs=len(words)/2
#initialize counter for words
i=0
#initialize counter for word pairs
j=0
#provide plotting canvas
plt.figure(figsize=(10,10))
#while loop to plot all pairs and draw lines
while i < pairs:
  #plot a pair with line connecting
  plt.plot([twodimpoints[i][0,0], twodimpoints[i][1,0]], [twodimpoints[i][0,1
], twodimpoints[i][1,1]], 'go--')
  #add point labels
  plt.text(twodimpoints[i][0,0]+0.01, twodimpoints[i][0,1]+0.01, words[j])
  plt.text(twodimpoints[i][1,0]+0.01, twodimpoints[i][1,1]+0.01, words[j+1])
  #advance both counters
  j = j + 2
  i = i + 1;
#now show the plot
plt.show();
```



We see significant parallelism using the larger model. It also grouped by gender too.

This model appears to be superior to all previous models used.

In [0]:

```
```