Exploring word2vec vector space

JULIA BAZIŃSKA

"a word is characterized by the company it keeps"

John Rupert Firth

Word similarity

P(a|c) – probability that word a appears in the neighbourhood of word c.

Words *a* and *b* have similar meanings



$$P(c|a) \approx P(c|b)$$

for every word c

But then we would need to know P(x|y) for every pair x, y...

Pointwise mutual information

$$PMI(x, y) = \log\left(\frac{P(x \land y)}{P(x)P(y)}\right) = \log\left(\frac{P(x \mid y)}{P(x)}\right)$$

How much more probable are words *x*, *y* to occur toghether than at random?

Vector approximation

For words x and y let's find vectors v_x and v_y satisfying:

$$PMI(x,y) = \overrightarrow{v_x} \cdot \overrightarrow{v_y}$$

Back to word similarity

$$P(c|a) \approx P(c|b)$$

for every word *c*.

$$PMI(a,c) \approx PMI(b,c)$$

$$\overrightarrow{v_a} \cdot \overrightarrow{v_c} \approx \overrightarrow{v_b} \cdot \overrightarrow{v_c}$$

$$\overrightarrow{v_c} \cdot (\overrightarrow{v_a} - \overrightarrow{v_b}) \approx 0$$

$$\overrightarrow{v_a} \approx \overrightarrow{v_b}$$

Cosine distance

- Similar words have similar vector values
- We can use cosine distance to measure similarity:

$$dist(a,b) = \frac{\overrightarrow{v_a} \cdot \overrightarrow{v_b}}{|\overrightarrow{v_a}||\overrightarrow{v_b}|}$$

```
In [24]: find most similar("europe", 10)
                           d most similar ("blue", 10)
                                                                           Out[24]: 0
4
                                                                                               1.000000
                                                                                   europe
                        blue
                                  1.000000
                                                                                               0.881251
                                                                                   european
                        red
                                  0.890182
                                                                                   asia
                                                                                               0.834660
                        black
                               0.864808
                                                                                   world
                                                                                               0.826442
                        pink
                                0.845264
                                                                                               0.819641
                                                                                   countries
                        green
                               0.834646
                                                                                               0.816004
                                                                                   britain
                        yellow 0.832033
                                                                                               0.798280
                                                                                   continent
                        purple 0.829353
                                                                                   america 0.794502
                        white
                                 0.822612
                                                                                   germany 0.791652
                        orange 0.811403
                                                                                   country 0.790833
                        bright
                                 0.799914
                                                                                   dtype: float64
                        dtype: float64
                                                                           In [14]: find most similar("vinci", 10)
                In [15]: find most similar("dance", 10)
                                                                           Out[14]: 0
                Out[15]: 0
                                                                                                 1.000000
                                                                                   vinci
                                  1.000000
                         dance
                                                                                   leonardo 0.739511
                        dancing
                                  0.906835
                                                                                   botticelli 0.672160
                                  0.871643
                         singing
                                                                                   michelangelo 0.661199
                                   0.853638
                        dances
                                                                                   caravaggio 0.658187
                                   0.853078
                        music
                                                                                   vaio
                                                                                                 0.641079
                                   0.839160
                        musical
                                                                                   andrea
                                                                                                 0.631471
                        dancers
                                   0.813865
                                                                                                 0.628268
                                                                                   giovanni
                                   0.797830
                        hop
                                                                                                 0.627567
                                                                                   vita
                                   0.788027
                        singers
                                                                                   francesca
                                                                                                  0.626247
                                   0.787464
                        pop
                                                                                   dtype: float64
                         dtype: float64
```

X is to Y as A is to...?

The nearest vector to $v_y - v_x + v_a$ is v_b with the highest value of $v_b \cdot (v_y - v_x + v_a)$

$$v_b \cdot (v_y - v_x + v_a) = v_b \cdot v_y + v_b \cdot v_x - v_b \cdot v_a$$

```
In [68]: riddle("warsaw", "poland", "moscow")
Out[68]: 0
                   0.955646
         russia
                  0.871337
        ukraine
                   0.849050
         russian
        poland
                   0.846253
         republic 0.842461
        dtype: float64
In [71]: riddle("good", "bad", "up")
Out[71]: 0
                    0.953188
        down
                   0.920470
        up
                   0.912033
         falling
                    0.893837
         out
        dropping 0.876727
        dtype: float64
```

```
In [100]: riddle("hope", "disappointment", "peace")
Out[100]: 0
         disagreement
                           0.764241
         disappointment
                          0.751991
          underlined
                           0.742288
          renewed
                           0.714227
         underscored
                          0.712070
         dtype: float64
In [102]: riddle("country", "language", "britain")
Out[102]: 0
         english
                          0.780426
                          0.777146
         language
                          0.762685
          translation
                         0.745039
          text
         pronunciation
                         0.729750
         dtype: float64
```

```
In [30]: riddle("science", "einstein", "painting")
Out[30]: 0
                   1.117469
        matisse
                   1.083420
        picasso
        duchamp 1.055209
        rembrandt 1.027414
                    1.001785
        titian
        dtype: float64
In [32]: riddle("astronomy", "copernicus", "philosophy")
Out[32]: 0
        nietzsche
                     0.897995
                    0.886114
        copernicus
                     0.872768
        freud
                    0.865735
        hegel
        kierkegaard 0.843774
        dtype: float64
```

Projection on a difference axis

To measure whether v_{x} is associated more with v_{a} or v_{b} we can calculate

$$v_x \cdot (v_a - v_b)$$

- Name gender
- Good bad vs up down
- Interactive viz: <u>lamyiowce.github.io/word2viz</u>

Vector rejection

$$v'_a = v_a - \frac{v_a \cdot v_b}{v_b \cdot v_b} \cdot v_b \Rightarrow v'_a \perp v_b$$
projection of a on b

Vector rejection

Can help with polysemy problems!

$$v'_{rock} = v_{rock} - \frac{v_{rock} \cdot v_{music}}{v_{music} \cdot v_{music}} \cdot v_{music}$$

```
In [159]: find most similar('rock', dfn, 10)
Out[159]: 0
          rock
                    1.000000
                    0.694770
          band
                   0.674067
          pop
                  0.661065
          punk
                   0.645949
          bands
          'n'
                   0.624487
                  0.616312
          rocks
                   0.613320
          album
                  0.600090
          albums
                    0.599088
          music
          dtype: float64
```

```
dfn.dot(reject('rock', ['music'], dfn)).sort_values(ascending = False).head(10)
In [160]:
Out[160]: 0
                         0.641093
          rock
                         0.498883
          rocks
                         0.444147
          outcropping
          limestone
                         0.431764
                         0.419863
          outcrops
                         0.409561
          outcrop
          cliffs
                         0.404085
          fraggle
                         0.400296
          rockers
                         0.399462
                         0.399351
          granite
          dtype: float64
```

```
In [167]: find most similar('python', dfn, 10)
Out[167]: 0
         python
                        1.000000
                        0.689272
         monty
          spamalot
                        0.561178
          cleese
                        0.545438
                        0.525527
         php
                        0.507684
         pythons
         perl
                        0.499981
                        0.485102
          scripting
          skit
                        0.475383
         reticulatus
                        0.470973
         dtype: float64
```

```
In [168]: dfn.dot(reject('python', ['monty'], dfn)).sort_values(ascending = False).head(10)
Out[168]: 0
         python
                       0.524904
                       0.414212
         php
         scripting
                      0.398441
                       0.336685
         java
         server-side 0.334762
         pythons
                      0.323510
                      0.322435
         perl
         javascript 0.317562
         c++
                       0.316876
         bindings
                       0.316854
         dtype: float64
```

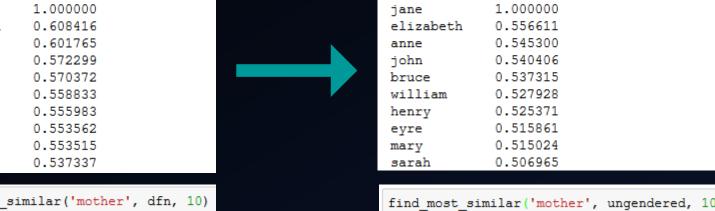
```
In [169]: dfn.dot(reject('python', ['monty', 'php'], dfn)).sort_values(ascending = False).head(10)
Out[169]: 0
                     0.307225
         python
         pythons
                       0.263030
         reticulated 0.236272
         crocodile
                       0.232522
         snake
                     0.230709
                     0.227266
         monkey
         burmese
                    0.225066
                   0.219475
         lizard
         turtles
                    0.216014
         tortoise
                     0.214143
         dtype: float64
```

```
In [247]: find_most_similar("polish", dfn)
Out[247]: 0
                       1.000000
         polish
         lithuanian
                      0.725827
                      0.701218
          hungarian
                      0.694825
         poland
          slovak
                       0.667826
          dtype: float64
In [252]: dfn.dot(reject('polish', ['lithuanian'], dfn)).sort_values(ascending = False).head(10)
Out[252]: 0
                   0.473175
         polish
         poland
                   0.383410
                   0.344952
          warsaw
                   0.286086
          german
                   0.267660
          jerzy
          walesa
                  0.263369
                   0.262397
          pope
                  0.260597
          krakow
          lech
                   0.253146
          iraqi
                   0.251096
          dtype: float64
```

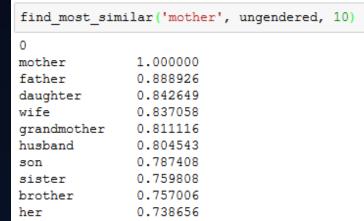
Removing gender from vectors

We can reject $v_{she} - v_{he}$ from all vectors in data

```
find most similar ('jane', dfn, 10)
jane
             1.000000
elizabeth
             0.608416
             0.601765
anne
             0.572299
mary
             0.570372
sarah
             0.558833
eliza
alice
             0.555983
             0.553562
helen
ellen
             0.553515
             0.537337
fonda
```



```
find most similar ('mother', dfn, 10)
0
mother
               1.000000
daughter
               0.864802
wife
               0.856802
grandmother
               0.837379
husband
               0.805565
               0.802924
sister
father
               0.793677
               0.783749
her
daughters
               0.758976
               0.757987
```



find most similar ('jane', ungendered, 10)

Vector quality

- Corpus size
- Vector dimensions
- Vector deriving algorithm

Thank you

Julia Bazińska julia@bazinski.com GH: lamyiowce