# 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

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

$$\Rightarrow \text{ for every word } c$$

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:

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

```
find most similar("blue", 10)
blue
        1.000000
         0.890182
red
black
        0.864808
pink
       0.845264
      0.834646
green
      0.832033
yellow
purple
      0.829353
white 0.822612
orange 0.811403
bright 0.799914
dtype: float64
```

```
find most similar("dance", 10)
0
          1.000000
dance
dancing
          0.906835
singing
         0.871643
dances
         0.853638
          0.853078
music
musical
          0.839160
         0.813865
dancers
          0.797830
hop
singers
         0.788027
          0.787464
pop
dtype: float64
```

```
find most similar("europe", 10)
0
            1.000000
europe
            0.881251
european
            0.834660
asia
world
            0.826442
countries
            0.819641
britain
            0.816004
            0.798280
continent
            0.794502
america
            0.791652
germany
            0.790833
country
```

```
find most similar("vinci", 10)
               1.000000
vinci
leonardo
               0.739511
botticelli
               0.672160
michelangelo
               0.661199
caravaggio
               0.658187
               0.641079
vaio
andrea
               0.631471
               0.628268
giovanni
               0.627567
vita
francesca
               0.626247
```

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

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

```
riddle("warsaw", "poland", "moscow")
            0.955646
russia
ukraine
            0.871337
                          riddle("good", "bad", "up")
russian
            0.849050
poland
            0.846253
                          0
republic
            0.842461
                          down
                                       0.953188
                                       0.920470
                          up
                          falling
                                       0.912033
                                       0.893837
                          out
                          dropping
                                       0.876727
```

```
riddle("hope", "disappointment", "peace")
0
                  0.764241
disagreement
disappointment
                  0.751991
underlined
                  0.742288
                  0.714227
renewed
underscored
                  0.712070
riddle("country", "language", "britain")
english
                  0.780426
                  0.777146
language
translation
                  0.762685
                  0.745039
text
pronunciation
                  0.729750
```

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

### 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: <a href="mailto:lamyiowce.github.io/word2viz">lamyiowce.github.io/word2viz</a>

### 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!

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

```
find most similar('rock', dfn, 10)
0
rock
          1.000000
band
          0.694770
          0.674067
pop
punk
          0.661065
bands
          0.645949
'n'
          0.624487
          0.616312
rocks
album
          0.613320
albums
          0.600090
music
          0.599088
```

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

```
find most similar('python', dfn, 10)
0
               1.000000
python
               0.689272
monty
spamalot
               0.561178
cleese
               0.545438
               0.525527
php
pythons
               0.507684
               0.499981
perl
scripting
               0.485102
skit
               0.475383
reticulatus
             0.470973
```

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

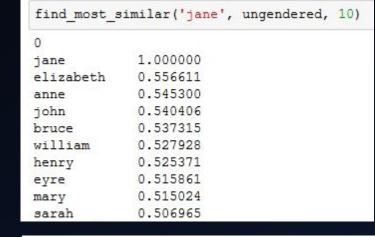
```
dfn.dot(reject('python', ['monty', 'php'], dfn)).sort_values(ascending = False).head(10)
0
python
               0.307225
pythons
               0.263030
reticulated
               0.236272
crocodile
               0.232522
snake
               0.230709
monkey
               0.227266
burmese
   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

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

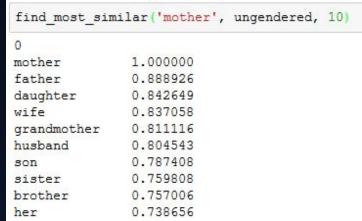
#### Removing gender from vectors

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

```
find most similar ('jane', dfn, 10)
0
             1.000000
jane
elizabeth
             0.608416
             0.601765
anne
             0.572299
mary
sarah
             0.570372
eliza
             0.558833
alice
             0.555983
             0.553562
helen
             0.553515
ellen
             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
woman
```



## Vector quality

- Corpus size
- Vector dimensions
- Vector deriving algorithm

# Thank you

Julia Bazińska julia@bazinski.com GH: lamyiowce