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 $\Leftrightarrow$ $P(c|a) \approx P(c|b)$ for every word *c*

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

Pointwise mutual information

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

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)
```

```
0
blue      1.000000
red       0.890182
black     0.864808
pink      0.845264
green     0.834646
yellow    0.832033
purple    0.829353
white     0.822612
orange    0.811403
bright    0.799914
dtype: float64
```

```
find_most_similar("dance", 10)
```

```
0
dance       1.000000
dancing     0.906835
singing     0.871643
dances      0.853638
music       0.853078
musical     0.839160
dancers     0.813865
hop         0.797830
singers     0.788027
pop         0.787464
dtype: float64
```

```
find_most_similar("europe", 10)
```

```
0
europe        1.000000
european      0.881251
asia          0.834660
world         0.826442
countries     0.819641
britain       0.816004
continent     0.798280
america       0.794502
germany       0.791652
country       0.790833
```

```
find_most_similar("vinci", 10)
```

```
0
vinci            1.000000
leonardo         0.739511
botticelli       0.672160
michelangelo     0.661199
caravaggio       0.658187
vaio             0.641079
andrea           0.631471
giovanni         0.628268
vita             0.627567
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	
russia	0.955646
ukraine	0.871337
russian	0.849050
poland	0.846253
republic	0.842461

```
riddle("good", "bad", "up")
```

0	
down	0.953188
up	0.920470
falling	0.912033
out	0.893837
dropping	0.876727

```
riddle("hope", "disappointment", "peace")
```

```
0
disagreement      0.764241
disappointment    0.751991
underlined        0.742288
renewed           0.714227
underscored       0.712070
```

```
riddle("country", "language", "britain")
```

```
0
english           0.780426
language          0.777146
translation       0.762685
text              0.745039
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")
```

```
0
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: lamyiowce.github.io/word2viz

Vector rejection

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

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
pop       0.674067
punk      0.661065
bands     0.645949
'n'       0.624487
rocks     0.616312
album     0.613320
albums    0.600090
music     0.599088
```

```
dfn.dot(reject('rock', ['music'], dfn)).sort_values(ascending = False).head(10)
```

```
0
rock           0.641093
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
```

```python
find_most_similar('python', dfn, 10)
```

```
0
python         1.000000
monty          0.689272
spamalot       0.561178
cleese         0.545438
php            0.525527
pythons        0.507684
perl           0.499981
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
c++           0.316876
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
```

```python
find_most_similar("polish", dfn)
```

```
0
polish        1.000000
lithuanian    0.725827
hungarian     0.701218
poland        0.694825
slovak        0.667826
dtype: float64
```

```python
dfn.dot(reject('polish', ['lithuanian'], dfn)).sort_values(ascending = False).head(10)
```

```
0
polish    0.473175
poland    0.383410
warsaw    0.344952
german    0.286086
jerzy     0.267660
walesa    0.263369
pope      0.262397
krakow    0.260597
lech      0.253146
iraqi     0.251096
```

Removing gender from vectors

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

```
find_most_similar('jane', dfn, 10)

0
jane         1.000000
elizabeth    0.608416
anne         0.601765
mary         0.572299
sarah        0.570372
eliza        0.558833
alice        0.555983
helen        0.553562
ellen        0.553515
fonda        0.537337
```

```
find_most_similar('jane', ungendered, 10)

0
jane         1.000000
elizabeth    0.556611
anne         0.545300
john         0.540406
bruce        0.537315
william      0.527928
henry        0.525371
eyre         0.515861
mary         0.515024
sarah        0.506965
```

```
find_most_similar('mother', dfn, 10)

0
mother         1.000000
daughter       0.864802
wife           0.856802
grandmother    0.837379
husband        0.805565
sister         0.802924
father         0.793677
her            0.783749
daughters      0.758976
woman          0.757987
```

```
find_most_similar('mother', ungendered, 10)

0
mother         1.000000
father         0.888926
daughter       0.842649
wife           0.837058
grandmother    0.811116
husband        0.804543
son            0.787408
sister         0.759808
brother        0.757006
her            0.738656
```

Vector quality

- Corpus size
- Vector dimensions
- Vector deriving algorithm

Thank you

Julia Bazińska
julia@bazinski.com
GH: lamyiowce