Eksploracja tekstu i wyszukiwanie informacji w mediach społecznościowych

LABORATORIUM 9

- Model word2vec
- Model GloVe

Model word2vec

Jednym z częstych zagadnień związanych z text mining jest budowa tzw. *word embedding*, czyli sposobu reprezentacji wyrazów w postaci liczbowej. W pewien sposób mówiliśmy już o tym, kiedy rozpatrywaliśmy **model przestrzeni wektorowej** - jest to jednak bardzo ograniczona rezprezentacja.

Semantyka dystrybucyjna znalazła w ostatnich latach szerokie zastosowanie w rozwiązywaniu szeregu zadań związanych z przetwarzaniem języka naturalnego. U jej podstaw leży hipoteza, że słowa występujące w podobnych kontekstach w dużych zbiorach danych tekstowych mają **podobne znaczenie**. Znaczenia słów reprezentowane są przez wektory liczbowe.

Jednym z bardziej popularnych przykładów, należącym do tej grupy metod jest **word2vec**, stworzony przez Tomasa Mikolova z Google ok. 5 lat temu. Podejście to może wykorzystywać dwie różne techniki: **CBOW** - Continuous Bag of Words oraz **Skip-Gram**. Teoretycznie metody są algorytmicznie identyczne, oprócz tego, że CBOW przewiduje kluczowe słowo na podstawie kontekstu np. ``cat sits on the **mat**'', a Skip-Gram odwrotnie. Word2vec używa sieci neuronowej z pojedynczą warstwą jako podstawowej architektury, ale jako ogólny wskaźnik używane jest po prostu prawdopodobieństwo $p(w_t|h)$ słowa w_t pod warunkiem historii (otoczenia) h, które jest wyznaczane przy użyciu metody największej wiarygodności.

Model GloVe

GloVe, czyli **Global Vectors for Word Representation** jest datującą się na rok 2014 techniką zaproponowaną przez naukowców ze Stanford University i opierającą się na następującym algorytmie:

- wykonaj statystykę współwystepowania słów i zapisz ją w postaci macierzy $\mathbf{X}$; zwykle korpus jest skanowany w taki sposób, że szukmay słów kontekstowych w pewnym oknie zarówno przed jak i po interesującym nas słowie; zwykle też dalsze słowa wchodzą z mniejszą wagą np. $w = 1/\mathrm{przesunięcie}$
- zdefiniuj więzy dla każdej pary słów występującej w macierzy $\mathbf{X}$ jako $\mathbf{w}_i^T \mathbf{w}_j + b_i + b = \log X_{ij}$, gdzie $\mathbf{w}_i$ to wektor głównego słowa, a $\mathbf{w}_j$ to wektor słowa z kontekstu, natomiast b_i , b_j to skalary
- zdefinuj fukcję kosztu $J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(\mathbf{w}_i^T \mathbf{w}_j + b_i + b - \log X_{ij})^2$,

przy czym $f(X_{ij})$ jest pewną funkcją ważącą, zapobiegającą uczeniu się jedynie od najczęściej wystepujących słów. Funkcja zostało zaproponowana jako $f(X_{ij}) = (X_{ij}/x_{max})^\alpha$ dla $X_{ij} < x_{max}$ i $f(X_{ij}) = 1$ w przeciwnym razie.

Rozpoczynamy od zainstalowania pakietu **text2vec** (co może trochę potrwać), a następnie pobiramy plik potrzebny do analiz - uwaga, dane mają ok. 100MB!

```
# PRZYKŁAD 9.1

library(text2vec)
file.name <- "http://www.if.pw.edu.pl/~julas/TEXT/lab/text8"
file.in <- readLines(file.name, n = 1, warn = FALSE)
```

Wykorzystywany tekst jest wczytywany, a następnie usuwane są najrzadsze oraz naczęstsze słowa.

```
# PRZYKŁAD 9.2

tokens <- space_tokenizer(file.in)
tokens[[1]][1:10]
```

```
##  [1] ""           "anarchism"  "originated" "as"         "a"
##  [6] "term"       "of"         "abuse"      "first"      "used"
```

```
it <- itoken(tokens, progressbar = FALSE)
vocab <- create_vocabulary(it)
vocab
```

```
## Number of docs: 1
## 0 stopwords:  ...
## ngram_min = 1; ngram_max = 1
## Vocabulary:
##                term term_count doc_count
##      1:   piagarmi          1         1
##      2: manganiyar          1         1
##      3:      yavne          1         1
##      4:  aroostock          1         1
##      5: stollwerck          1         1
##     ---
## 253850:         in     372201         1
## 253851:        one     411764         1
## 253852:        and     416629         1
## 253853:         of     593677         1
## 253854:        the    1061396         1
```

```
vocab <- prune_vocabulary(vocab, term_count_min = 5L)
```

Kolejnym krokiem jest stworzenie macierzy współwystępowania słów, opartej na wspomniane wcześniej ruchome okno - wartośc wynosi 5, co oznacza, że wejdzie do niego po 5 słów z lewej i prawej strony wyrazu.

```
# PRZYKŁAD 9.3
vectorizer <- vocab_vectorizer(vocab)
tcm <- create_tcm(it, vectorizer, skip_grams_window = 5L)
options(max.print = 1000)
tcm
```

```
## 71290 x 71290 sparse Matrix of class "dgTMatrix"
```

```
##    [[ suppressing 29 column names 'kentauros', 'tornatore', 'phantastica' ... ]]
##    [[ suppressing 29 column names 'kentauros', 'tornatore', 'phantastica' ... ]]
```

```
##
## kentauros      . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## tornatore      . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## phantastica    . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## steinhoff      . . . 0.25 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## minthe         . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## jizyah         . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## pommern        . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## iconodules     . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## palas          . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## splinters      . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## nears          . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## cashed         . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## dimasi         . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## yisra          . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## mistranslated  . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## adenocarcinoma . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## cheetham       . . . .    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
##
##  ..............................
##  ........suppressing columns and rows in show(); maybe adjust 'options(max.print= *
, width = *)'
##  ..............................
```

```
##    [[ suppressing 29 column names 'kentauros', 'tornatore', 'phantastica' ... ]]
```

```
##
## three . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## five  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## s     . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## for   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## eight . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## as    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## is    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## two   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## nine  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## zero  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## to    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## a     . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## in    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## one   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## and   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## of    . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
## the   . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .....
```

Wreszcie przechodzimy do samego uruchomienia algorytmu, ustawiając rozmiar wektora na 50 elementów

```
# PRZYKŁAD 9.4

glove_model <- GlobalVectors$new(word_vectors_size = 50, vocabulary = vocab, x_max = 1
0)
words_main <- glove_model$fit_transform(tcm, n_iter = 10, convergence_tol = 0.01)
```

```
## INFO [2019-01-07 06:23:09] 2019-01-07 06:23:09 - epoch 1, expected cost 0.0880
## INFO [2019-01-07 06:23:24] 2019-01-07 06:23:24 - epoch 2, expected cost 0.0614
## INFO [2019-01-07 06:23:39] 2019-01-07 06:23:39 - epoch 3, expected cost 0.0542
## INFO [2019-01-07 06:23:54] 2019-01-07 06:23:54 - epoch 4, expected cost 0.0502
## INFO [2019-01-07 06:24:11] 2019-01-07 06:24:11 - epoch 5, expected cost 0.0477
## INFO [2019-01-07 06:24:28] 2019-01-07 06:24:28 - epoch 6, expected cost 0.0459
## INFO [2019-01-07 06:24:44] 2019-01-07 06:24:44 - epoch 7, expected cost 0.0445
## INFO [2019-01-07 06:24:59] 2019-01-07 06:24:59 - epoch 8, expected cost 0.0434
## INFO [2019-01-07 06:25:15] 2019-01-07 06:25:15 - epoch 9, expected cost 0.0425
## INFO [2019-01-07 06:25:30] 2019-01-07 06:25:30 - epoch 10, expected cost 0.0418
```

Na wyjściu otrzymujemy tak naprawdę dwa wektory $\mathbf{w}_i$ oraz $\mathbf{w}_j$ - teoretycznie powinny byc symetryczne. W praktyce wykorzystuje się średnią lub ich sumę.

```
# PRZYKŁAD 9.5

words_components <- glove_model$components
W <- words_main + t(words_components)
options(max.print = 1000)
round(W, 3)
```

```
##                          [,1]   [,2]   [,3]   [,4]   [,5]   [,6]
## kentauros               0.325 -0.013  0.071  0.146  0.683  0.187
## tornatore              -0.384 -0.477 -0.169  0.817  0.279  0.116
## phantastica             0.024 -0.420  0.169  0.771  0.277 -0.440
## steinhoff               0.074 -1.078  0.093 -0.416  0.680 -0.837
## minthe                  0.346 -0.616  0.126  1.045 -0.006  0.485
## jizyah                 -0.167 -0.437  0.550  0.481  0.307 -0.120
## pommern                -0.473  0.073  0.512  0.302 -0.497  0.238
## iconodules              0.267  0.001  0.095 -0.178 -0.190 -0.622
## palas                   0.274  0.716 -0.064 -0.314  0.462 -0.048
## splinters              -0.042 -0.426  1.159  0.041  0.002 -0.161
## nears                  -0.151  0.060  0.305  0.060  0.259  0.028
## cashed                  0.072 -0.526  0.286  0.425  0.637  0.253
## dimasi                  0.640 -0.681 -0.124  0.010  0.433  0.291
## yisra                  -0.996  0.364 -0.341  0.675 -0.031 -0.677
## mistranslated          -0.733 -0.014  0.046  0.053 -0.262 -0.800
## adenocarcinoma          0.319 -0.535 -0.358  0.356  0.113  0.122
## cheetham                0.104  0.292  0.311  1.217 -0.490 -0.187
## cowbell                 0.009 -0.710 -0.276  0.680  0.060  0.010
## wallerstein             0.701 -0.137  0.315 -0.019  0.832 -0.808
## bluie                   0.621  0.050 -0.389  0.047  0.027  0.661
##                          [,7]   [,8]   [,9]  [,10]  [,11]  [,12]
## kentauros               0.225 -0.095 -0.136 -0.180 -0.213 -0.010
## tornatore              -0.327 -0.127  0.125 -0.034  0.401 -0.319
## phantastica             0.003  0.245 -0.112 -1.028  0.316 -0.130
## steinhoff              -0.466  0.117 -0.054  0.000  0.635 -0.682
## minthe                 -0.033 -0.091  0.650 -0.818  0.272  0.086
## jizyah                  0.622  0.080 -0.149  0.245  0.291 -0.274
## pommern                 0.620 -0.155  0.499 -0.176  0.044 -0.160
## iconodules              0.366  0.312 -0.094 -0.595  0.189 -0.177
## palas                  -0.031  0.587  0.420 -0.456 -0.487 -0.161
## splinters              -0.402  0.417 -0.551  0.114  0.515 -0.431
## nears                  -0.115  0.232  1.087 -0.401  0.427  0.005
## cashed                  0.101  0.337  0.086 -0.302 -0.103 -0.900
## dimasi                  0.371  0.637 -0.407 -0.353  1.087 -0.267
## yisra                   0.701  0.289  0.930  0.275  0.695 -0.306
## mistranslated           0.301  0.293  0.658  0.166  0.289  0.175
## adenocarcinoma          0.539 -0.373  0.104  0.813  0.526 -0.742
## cheetham                0.881  0.438  0.903 -0.420  1.140 -0.748
## cowbell                -0.594 -0.223 -0.657 -0.093  0.517  0.122
## wallerstein             0.521  0.209 -0.308  0.049  0.847 -0.029
## bluie                  -0.241 -0.188  0.140  0.424  0.597 -0.294
##                         [,13]  [,14]  [,15]  [,16]  [,17]  [,18]
## kentauros              -0.121  0.683  0.575 -0.291  0.546  0.221
## tornatore              -0.313  0.870 -0.367 -0.559  1.004 -0.637
## phantastica            -0.955  0.462 -0.561 -0.166  0.110 -0.750
## steinhoff               0.142  0.055 -0.263 -0.051  0.005 -0.370
## minthe                  0.108  0.107 -0.358  0.936  0.402 -0.621
## jizyah                 -0.165 -0.230 -0.270 -0.428  0.295 -0.296
## pommern                 0.004  0.072 -0.070 -0.383 -0.622 -0.572
```

```
## iconodules                -0.268  0.323 -0.019  0.532  0.208 -1.128
## palas                     -0.730  0.115 -0.334 -0.281  0.843 -0.260
## splinters                  0.409  0.687 -0.175  0.151  0.054 -0.494
## nears                      0.089  0.506 -0.435 -0.247 -0.978 -0.299
## cashed                    -0.512 -0.002 -0.067 -0.242  0.214 -0.618
## dimasi                    -0.577 -0.251 -0.886 -0.113 -0.019 -0.208
## yisra                      0.088  0.920 -0.549  0.488  0.039 -0.587
## mistranslated             -0.814  0.195  0.119  0.270 -0.235 -0.715
## adenocarcinoma            -0.259 -0.158 -0.682 -0.028  0.424 -0.457
## cheetham                   0.091  0.495 -0.036 -0.136  0.000  0.653
## cowbell                   -0.577  0.208 -0.576  0.566 -0.025 -0.142
## wallerstein               -0.595 -0.176  0.608  0.189  0.001 -0.758
## bluie                      0.233 -0.148 -0.526 -0.285  0.054 -0.961
##                             [,19]  [,20]  [,21]  [,22]  [,23]  [,24]
## kentauros                 -0.519 -0.167 -0.946 -0.163 -0.200 -0.555
## tornatore                  0.840 -0.089 -0.137  0.378  0.228 -0.793
## phantastica               -0.419  0.322 -0.121  0.040  0.509 -0.433
## steinhoff                  0.272  0.014 -0.602 -0.582 -0.074  0.369
## minthe                     0.156 -0.230 -0.907 -0.142 -0.134  0.258
## jizyah                    -0.234 -1.060 -0.029 -0.099 -0.332 -0.458
## pommern                    0.199 -0.302 -0.657 -0.841 -0.553 -0.510
## iconodules                -0.351 -0.037  0.290 -0.090  0.308 -0.039
## palas                      0.216  0.073 -0.286 -0.064  0.379 -0.329
## splinters                  0.707  0.031 -1.178 -0.064  0.536  0.356
## nears                     -0.012  0.276 -0.701  0.812 -0.072 -0.083
## cashed                    -0.862  0.637 -0.387  0.158  0.105 -0.252
## dimasi                    -0.675  0.201 -0.085 -0.051 -0.509 -0.045
## yisra                     -0.671  0.379 -0.271 -0.477  0.014 -0.900
## mistranslated             -0.567  0.633 -0.951 -0.095  0.324  0.367
## adenocarcinoma            -0.050  0.135 -0.692  0.068 -0.083  0.019
## cheetham                   0.276 -0.343  0.277 -0.587 -0.483 -0.535
## cowbell                    0.445  0.400 -0.574 -0.308 -0.503 -0.274
## wallerstein                0.376 -0.282 -0.511 -0.412 -0.811  0.186
## bluie                      0.126 -0.587  0.048 -0.454  1.028  0.319
##                             [,25]  [,26]  [,27]  [,28]  [,29]  [,30]
## kentauros                 -0.689  0.066  0.542 -0.820  0.148  0.192
## tornatore                  0.464 -0.225  0.529  0.386 -0.432  1.310
## phantastica               -0.281  0.222  0.007 -0.822 -0.203  0.874
## steinhoff                 -0.071 -0.331 -0.328 -0.863  0.104  1.111
## minthe                     0.034  0.156  0.309 -0.173 -0.390  0.386
## jizyah                     0.265  0.243  0.840 -0.190  0.029  0.037
## pommern                    0.188  0.021  0.719 -0.195  0.494  0.876
## iconodules                -0.648  0.160  0.005 -0.503  0.200  0.398
## palas                     -0.450 -0.009  0.186 -0.204 -0.482  0.096
## splinters                  0.151  0.016  1.072  0.337 -0.307  0.593
## nears                     -0.600  0.438  0.115 -0.463  0.176 -0.068
## cashed                    -0.044 -0.062  0.839 -0.259 -0.082 -0.166
## dimasi                    -0.563 -0.406 -0.175 -0.878 -0.045 -0.151
## yisra                      0.110 -0.163  0.566 -0.258 -0.133  1.204
## mistranslated             -0.300  0.792 -0.236 -0.219  0.088 -0.387
## adenocarcinoma            -0.129 -0.029 -0.329 -0.052  0.183  0.970
```

```
## cheetham                -0.330  0.101  0.803 -0.155 -0.058  0.693
## cowbell                 -0.103  0.681 -0.188 -0.616  0.017  0.388
## wallerstein              0.204 -0.007  0.665 -0.115 -0.488  0.297
## bluie                   -0.323 -0.175  0.340  0.028 -0.249  0.923
##                          [,31]  [,32]  [,33]  [,34]  [,35]  [,36]
## kentauros               -0.184 -0.184  0.956 -0.201  0.160  0.502
## tornatore               -0.403  0.728  0.284 -0.061 -0.510  0.414
## phantastica             -0.210  0.110 -0.554 -0.746 -0.451  0.428
## steinhoff                0.180  0.781  0.467 -0.625  0.598  0.230
## minthe                   0.149 -0.650  0.349  0.350  0.139  0.516
## jizyah                  -0.603  0.786  0.227 -0.522  0.286 -0.058
## pommern                 -1.080  0.464  0.419  0.359  0.725  0.137
## iconodules               0.528  0.789 -0.284 -0.755 -0.145  0.085
## palas                    0.449  0.673  0.331 -0.349 -0.135  0.093
## splinters               -0.836 -0.532  0.513 -0.380  0.197  0.195
## nears                   -0.043  1.238  0.340  0.129 -0.097  0.141
## cashed                  -0.439  0.327 -0.206 -0.612  0.927 -0.841
## dimasi                  -0.113  0.394 -0.069  0.098 -0.841 -0.059
## yisra                    0.662  0.523 -0.583  0.122 -0.696  0.554
## mistranslated           -0.490  1.081 -0.392  0.604  0.160 -0.185
## adenocarcinoma          -0.555 -0.594  0.432 -0.033  0.613 -0.562
## cheetham                -0.847  0.742  0.104 -0.386  0.031  0.385
## cowbell                 -0.117 -0.081  0.199  0.572 -0.020  0.733
## wallerstein             -0.333  0.552  0.542 -0.127 -0.159  0.577
## bluie                    0.091  0.901  0.435 -0.014 -0.035  0.116
##                          [,37]  [,38]  [,39]  [,40]  [,41]  [,42]
## kentauros                0.665  0.397 -0.368 -0.009  0.367  0.273
## tornatore               -0.317  0.598 -0.125 -0.484  0.518  0.018
## phantastica              0.872  0.141 -0.733 -0.456  0.306 -0.259
## steinhoff                0.549  0.621 -0.350 -0.274  0.274  0.245
## minthe                   0.339  0.165  0.420 -0.332  0.850  0.302
## jizyah                   0.980 -0.188 -0.244 -0.197  0.587  0.129
## pommern                  0.038  0.443 -0.073 -0.511  0.029  0.459
## iconodules              -0.136 -0.493 -0.503  0.075  0.717  0.127
## palas                   -0.164 -0.319  0.160 -1.176  0.583 -0.385
## splinters               -0.025  0.402  0.347  0.088  0.401  0.091
## nears                    0.708  0.576 -0.185 -0.479  0.308  0.973
## cashed                  -0.220  0.315 -0.108  0.399  0.334  0.251
## dimasi                   0.302 -0.019  0.195 -0.680  1.085 -0.151
## yisra                    0.279 -0.219 -0.621 -0.227 -0.056  0.526
## mistranslated            0.955  0.361 -0.841 -0.045 -0.108  0.131
## adenocarcinoma           0.724  0.519 -0.090 -0.162  0.074 -0.035
## cheetham                 0.506  0.462 -0.312  0.411  0.801  0.715
## cowbell                  0.954 -0.171 -0.613 -0.772  0.171  0.329
## wallerstein              0.119  0.495 -0.268  0.574  0.400  0.101
## bluie                    0.942  1.147 -0.721 -0.129  0.572 -0.206
##                          [,43]  [,44]  [,45]  [,46]  [,47]  [,48]
## kentauros               -0.567  0.605  0.354 -0.505 -0.083 -0.311
## tornatore               -0.325 -0.840 -0.302 -0.217  0.053  0.243
## phantastica             -0.484 -0.025 -0.276  0.270 -0.054 -0.127
## steinhoff               -0.394 -0.427 -0.780 -0.011  0.295  0.261
```

```
## minthe                          0.396  0.524 -0.255 -0.131 -0.432  0.440
## jizyah                         -0.550  0.165 -0.310 -0.481  0.830  0.404
## pommern                        -0.397  0.438 -0.188  0.181 -0.592 -0.504
## iconodules                     -0.009  0.707  0.078 -0.267 -0.456  0.395
## palas                           0.018  0.281 -0.334 -0.720  0.016  0.274
## splinters                       0.512  0.140 -0.206  0.394  0.229  0.216
## nears                           0.158 -0.362 -0.162  0.614 -0.364  0.965
## cashed                          0.090  0.244  0.074  0.549  0.194  0.035
## dimasi                         -1.080 -0.279 -0.897  0.378 -0.473  0.269
## yisra                          -0.338  0.472 -0.805  0.789 -0.300 -0.103
## mistranslated                  -0.555  0.017 -0.027  0.048 -0.408  0.613
## adenocarcinoma                  0.070  0.595 -0.178 -0.216  0.284  0.521
## cheetham                       -0.094 -0.146 -0.410  0.753  0.086 -0.099
## cowbell                         0.287  0.868 -0.072  0.548 -0.178 -0.148
## wallerstein                    -0.150 -0.204 -0.239  0.264  0.368 -0.724
## bluie                          -0.407  0.125  0.220  0.703 -0.293  0.277
##                                 [,49]  [,50]
## kentauros                       0.475 -0.755
## tornatore                      -0.594 -0.324
## phantastica                    -0.355 -0.300
## steinhoff                       0.063 -0.161
## minthe                          0.021  0.062
## jizyah                          0.162  0.049
## pommern                        -0.168 -0.548
## iconodules                      0.006 -0.609
## palas                           0.025  0.497
## splinters                       0.071  0.283
## nears                          -0.192  0.438
## cashed                          0.135  0.006
## dimasi                          0.030 -0.156
## yisra                           0.383  0.142
## mistranslated                   0.560 -0.214
## adenocarcinoma                 -0.495 -0.017
## cheetham                        0.202 -0.011
## cowbell                         0.401  0.498
## wallerstein                     0.461 -1.009
## bluie                          -0.314 -0.174
##  [ reached getOption("max.print") -- omitted 71270 rows ]
```

W tym momencie dysponujemy już w pełni nauczonym modelem, czyli reprezentacją słów w pewnej 50-wymiarowej przestrzeni. Oczywiście najprostsza rzeczą jest po prostu podanie konkretnego słowa i wypisanie podobieństwa do innych wyrazów, korzystając z cosinusa podobieństwa jako miary.

```
# PRZYKŁAD 9.6

query <- W["student", , drop = FALSE]
cos_sim <- sim2(x = W, y = query, method = "cosine", norm = "l2")
head(sort(cos_sim[,1], decreasing = T))
```

```
##       student      students        school      graduate undergraduate
##     1.0000000     0.7621514     0.7508940     0.7284443     0.7018185
##       teacher
##     0.6861288
```

```
# PRZYKŁAD 9.7

query <- W["poland", , drop = FALSE]
cos_sim <- sim2(x = W, y = query, method = "cosine", norm = "l2")
head(sort(cos_sim[,1], decreasing = T))
```

```
##    poland    russia   hungary   finland     spain     italy
## 1.0000000 0.7979293 0.7544698 0.7481203 0.7429340 0.7403539
```

Jak widać, można w ten sposób otrzymać również zależności geograficzne. Ciekawą cechą tego typu modeli jest to, że skoro operujemy w pewnej przestrzni wektorowej, to w pewnien sposób możemy oczekiwać, że będzie tam działać zykła arytmatyka wektorowa. W szczególności dodawanie i odejmowanie może doprowadzić do ciekawych wyników:

```
# PRZYKŁAD 9.8

query <- W["paris", , drop = FALSE] - W["france", , drop = FALSE] + W["germany", , dro
p = FALSE]
cos_sim <- sim2(x = W, y = query, method = "cosine", norm = "l2")
head(sort(cos_sim[,1], decreasing = T))
```

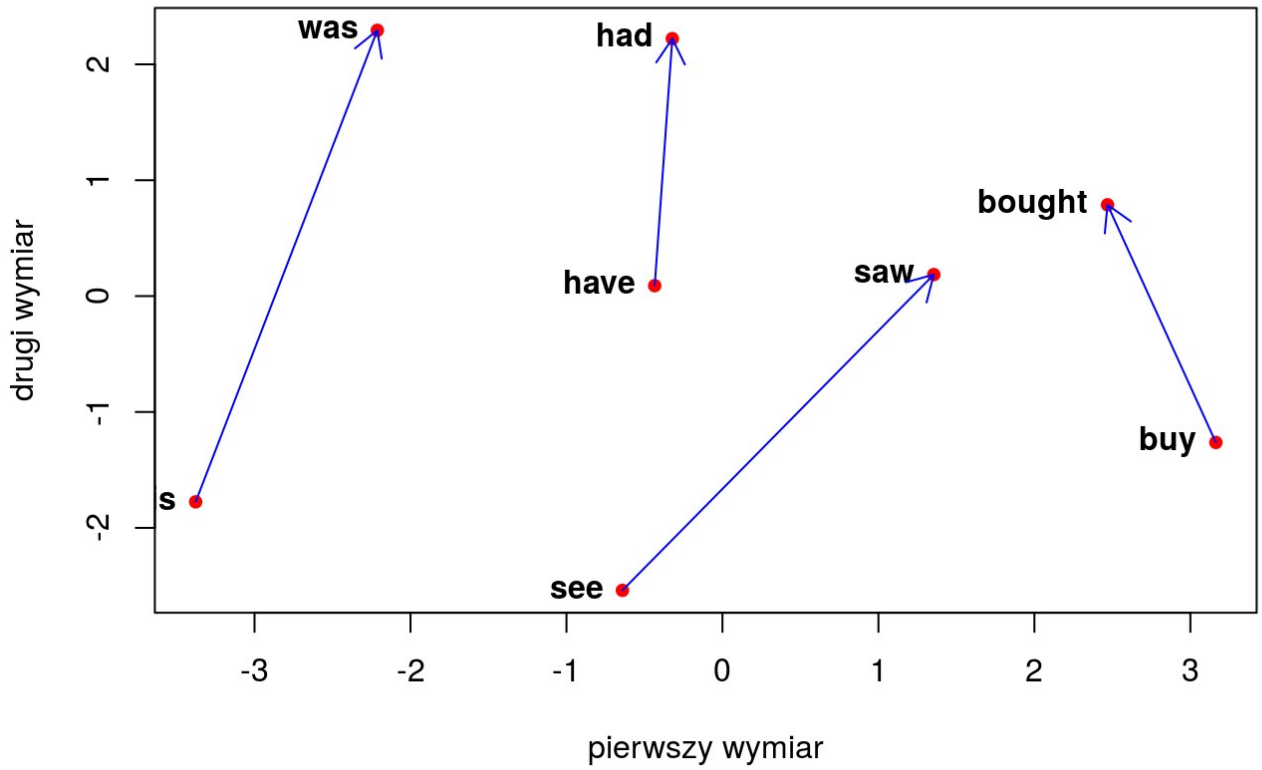
```
##    berlin     paris    munich   germany   leipzig   dresden
## 0.7578314 0.7458624 0.6459617 0.6394657 0.5938925 0.5883877
```

```
# PRZYKŁAD 9.9

query <- W["man", , drop = FALSE] - W["he", , drop = FALSE] + W["she", , drop = FALSE]
cos_sim <- sim2(x = W, y = query, method = "cosine", norm = "l2")
head(sort(cos_sim[,1], decreasing = T))
```

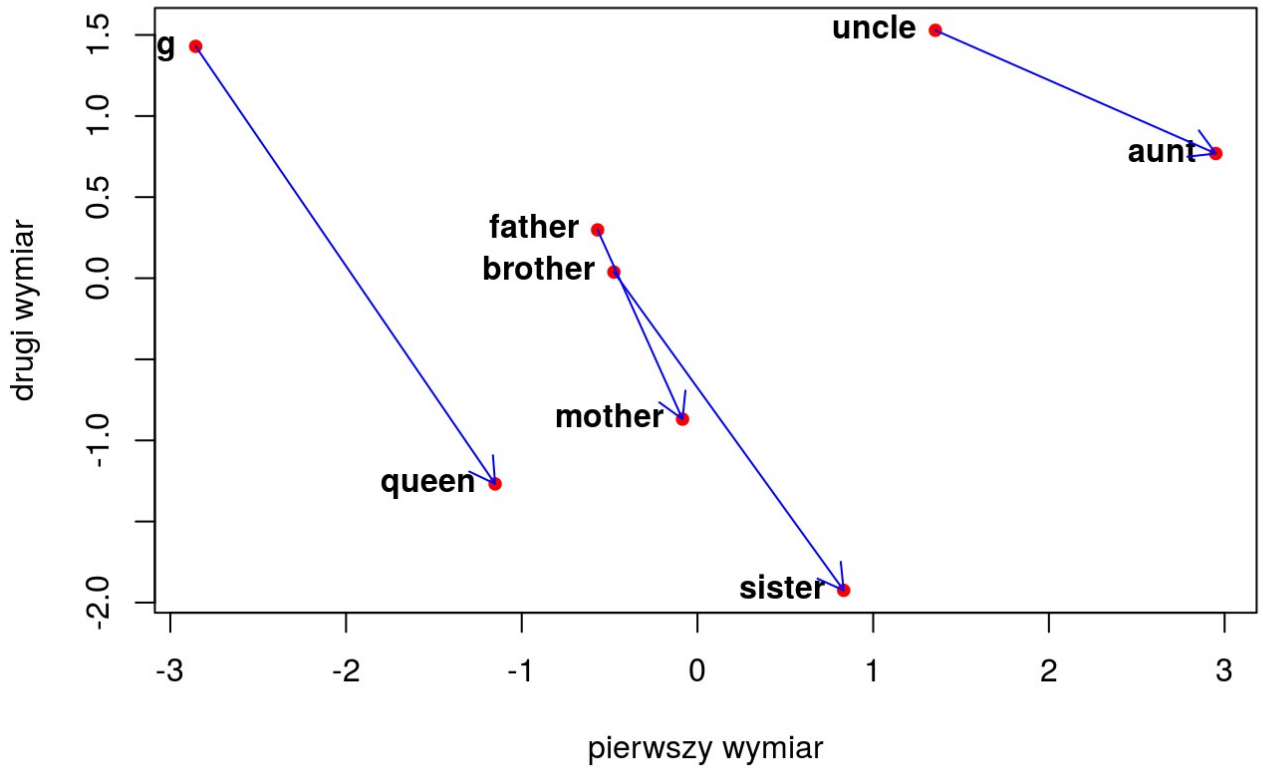
```
##       man     woman beautiful        my      girl     child
## 0.8457696 0.8144908 0.6523653 0.6342506 0.6299519 0.6100867
```

I wreszcie chyba najciekawsza cecha, tzw. regularność lingwistyczna. Okazuje się, że tego typu modele zanurzeniowe mogą służyć do wizualizacji pewnych reguł gramatycznych, np. związanych z częściami mowy etc. Ponieważ pracujemy na 50-wymiarowej przestrzeni, aby wykonac sensowne wykresy musimy wykorzystać metodę skalowania wielowymiarowego.

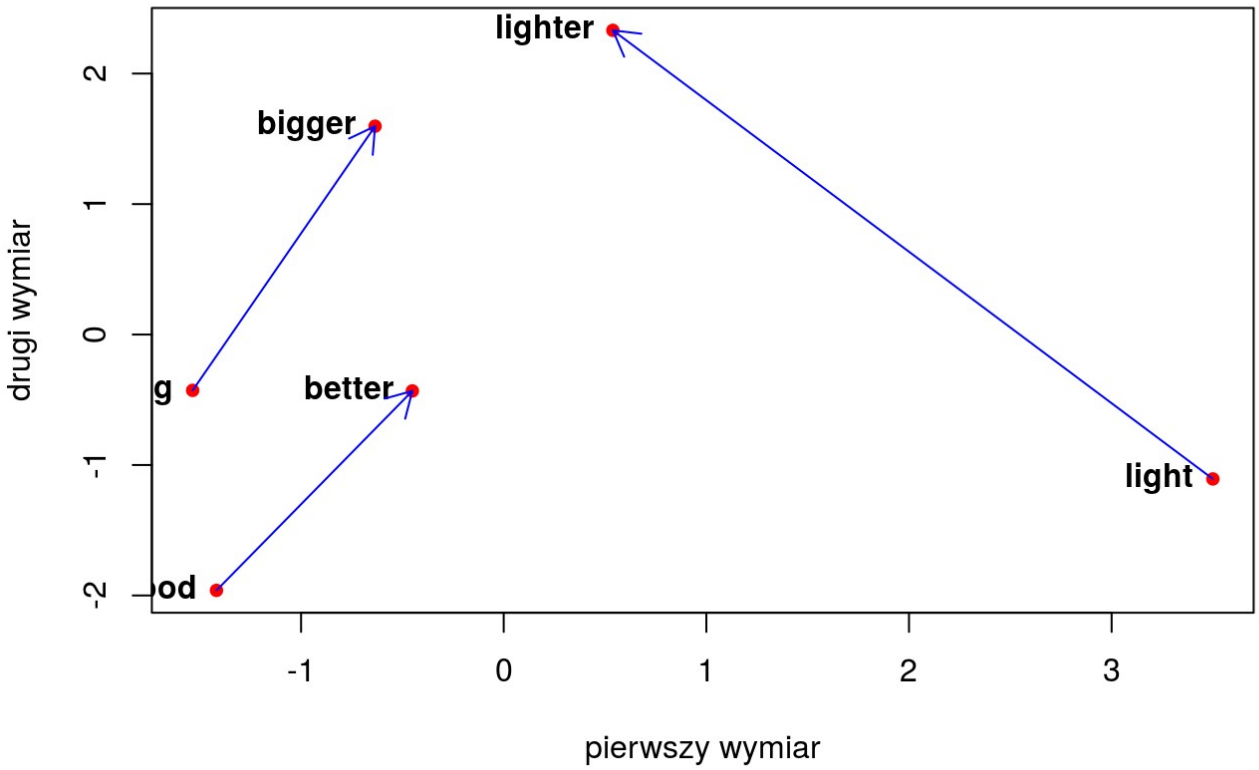
```
# PRZYKŁAD 9.10

q <- c("see", "saw", "have", "had", "is", "was", "buy", "bought")
z <- cmdscale(dist(W[q,]))
plot(z, pch = 19, cex = 0.8, col = "red", xlab = "pierwszy wymiar", ylab = "drugi wymi
ar")
text(z, labels = q, pos = 2, font = 2)
invisible(sapply(seq(1, nrow(z) - 1, 2), function(i) arrows(z[i,1],z[i,2],z[i+1,1],z[i
+1,2], length = 0.15, col = "blue")))
```




```
# PRZYKŁAD 9.11

q <- c("king", "queen", "father", "mother", "brother", "sister", "uncle", "aunt")
z <- cmdscale(dist(W[q,]))
plot(z, pch = 19, cex = 0.8, col = "red", xlab = "pierwszy wymiar", ylab = "drugi wymi
ar")
text(z, labels = q, pos = 2, font = 2)
invisible(sapply(seq(1, nrow(z) - 1, 2), function(i) arrows(z[i,1],z[i,2],z[i+1,1],z[i
+1,2], length = 0.15, col = "blue")))
```




```
# PRZYKŁAD 9.12

q <- c("good", "better", "big", "bigger", "light", "lighter")
z <- cmdscale(dist(W[q,]))
plot(z, pch = 19, cex = 0.8, col = "red", xlab = "pierwszy wymiar", ylab = "drugi wymi
ar")
text(z, labels = q, pos = 2, font = 2)
invisible(sapply(seq(1, nrow(z) - 1, 2), function(i) arrows(z[i,1],z[i,2],z[i+1,1],z[i
+1,2], length = 0.15, col = "blue")))
```


lighter
bigger
g
better
od
light
drugi wymiar
pierwszy wymiar
2
1
0
-1
-2
-1
0
1
2
3