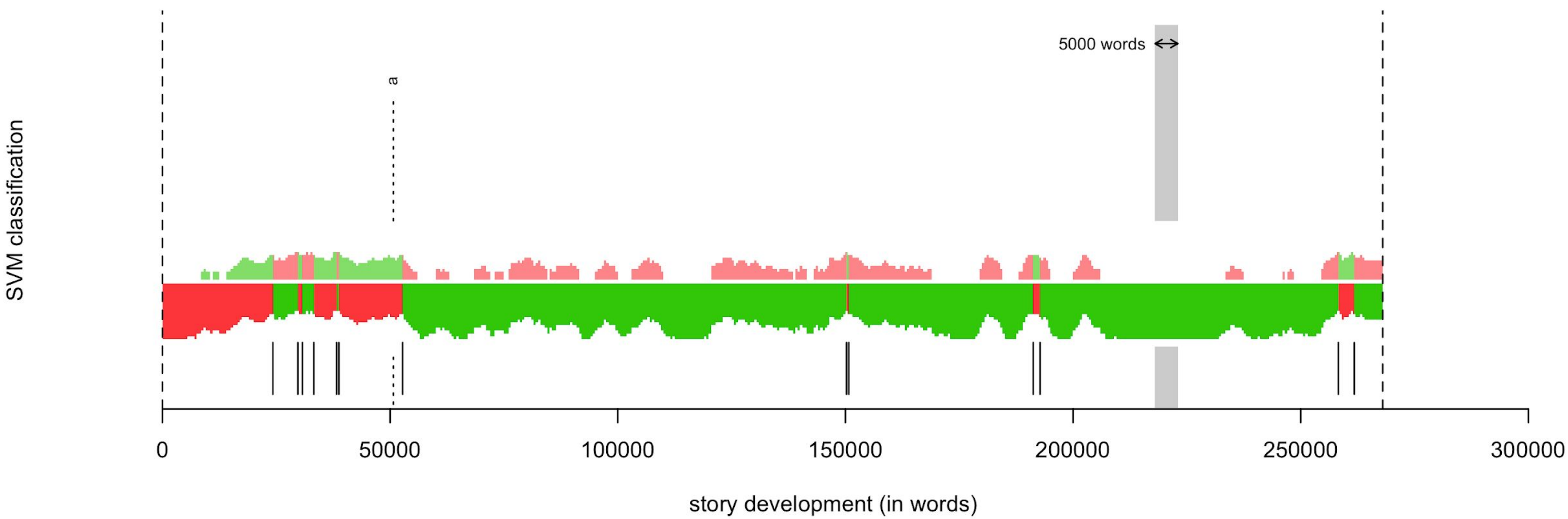
New Advances in Text Mining

or exploring word vectors

Maciej Eder
Institute of Polish Language (Polish Academy of Sciences)



SVM classification
5000 words
a
0
50000
100000
150000
200000
250000
300000
story development (in words)

Motivation

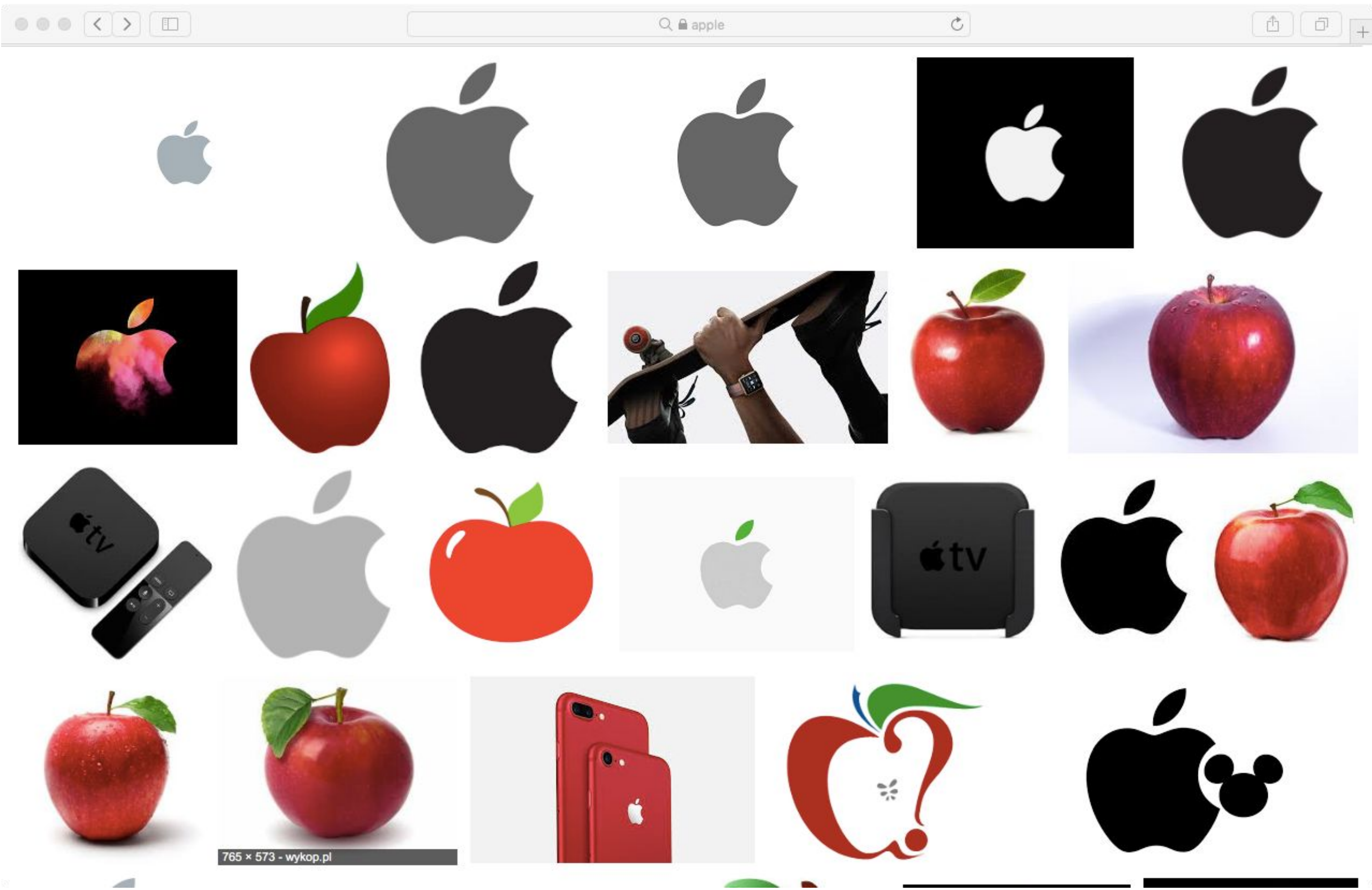
Information retrieval:

- How to “read” a big collection of documents, e.g. an archive?
- How to get relevant websites using search engines?
- How to fine-grain the results for ‘apple’ (1. a fruit, 2. a company)

Linguistics:

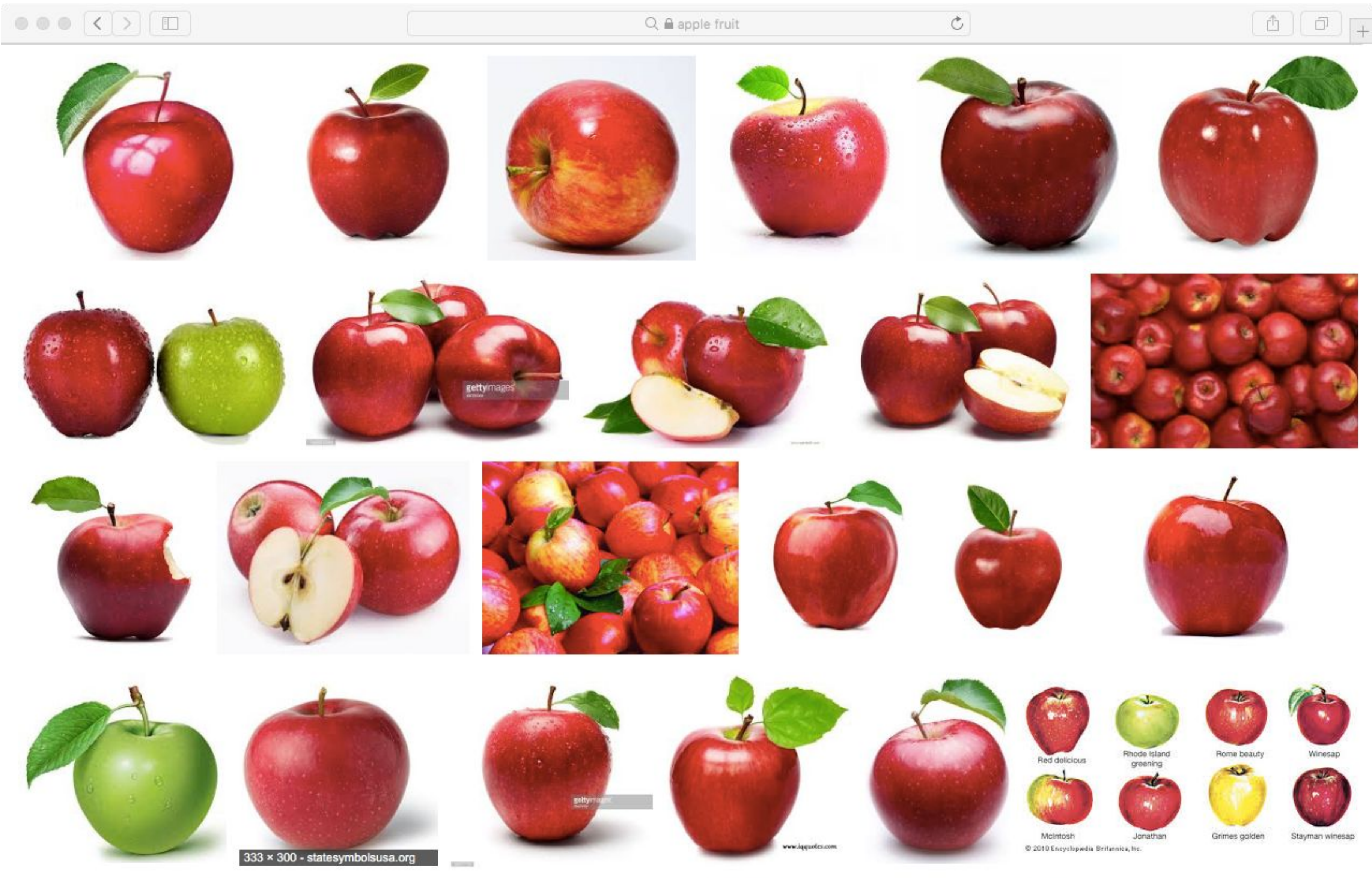
- What is the underlying model for defining word meaning?

Google search: ‘apple’



Google search: ‘apple’ + ‘computer’

Google search: ‘apple’ + ‘fruit’



Distributional hypothesis

Meaning defined by the context

The meaning of words lies in their use.

(Wittgenstein 1953: 80, 109)

You shall know a word by the company it keeps.

(Firth 1962: 11)

Distributional hypothesis

The degree of semantic similarity between two words (or other linguistic units) can be modeled as a function of the degree of overlap among their linguistic contexts.

(Harris 1954, Miller and Charles 1991,
Baroni and Lenci 2010)

Different definitions of the context

a sentence

a paragraph

a document (e.g. a blog post)

n-gram model, e.g. 2-grams:

> ‘it is’, ‘is a’, ‘a truth’, ‘truth universally’, ‘universally acknowledged’, ‘acknowledged that’ ‘that a’

skip-gram model (involving a moving window)

> ‘it is’, ‘it … a’, ‘it … … truth’, ‘it … … … universally’, etc.

Word embeddings

The Hound of the Baskervilles

Mr. Sherlock Holmes, who was usually very late in the mornings, save upon those not infrequent occasions when he was up all night, was seated at the breakfast table. I stood upon the hearth-rug and picked up the stick which our visitor had left behind him the night before. It was a fine, thick piece of wood, bulbous-headed, of the sort which is known as a “Penang lawyer.” Just under the head was a broad silver band nearly an inch across. “To James Mortimer, M.R.C.S., from his friends of the C.C.H.,” was engraved upon it, with the date "1884." It was just such a stick as the old-fashioned family practitioner used to carry – dignified, solid, and reassuring.

Frequencies of n-grams or/and skip-grams

mr sherlock	13
sherlock holmes	33
holmes who	2
who was	15
was usually	1
usually very	1
late mornings	2
…	…

Word cooccurrences (36,869 x 36,869)

	morning	apple	late	breakfast	table
morning	49	5	12	7	6
apple	5	9	1	2	3
late	12	1	39	3	1
breakfast	7	2	3	15	10
table	6	3	1	10	20
tea	3	5	2	12	9
banana	2	6	0	3	6
hour	8	0	4	1	0
night	2	0	4	0	0

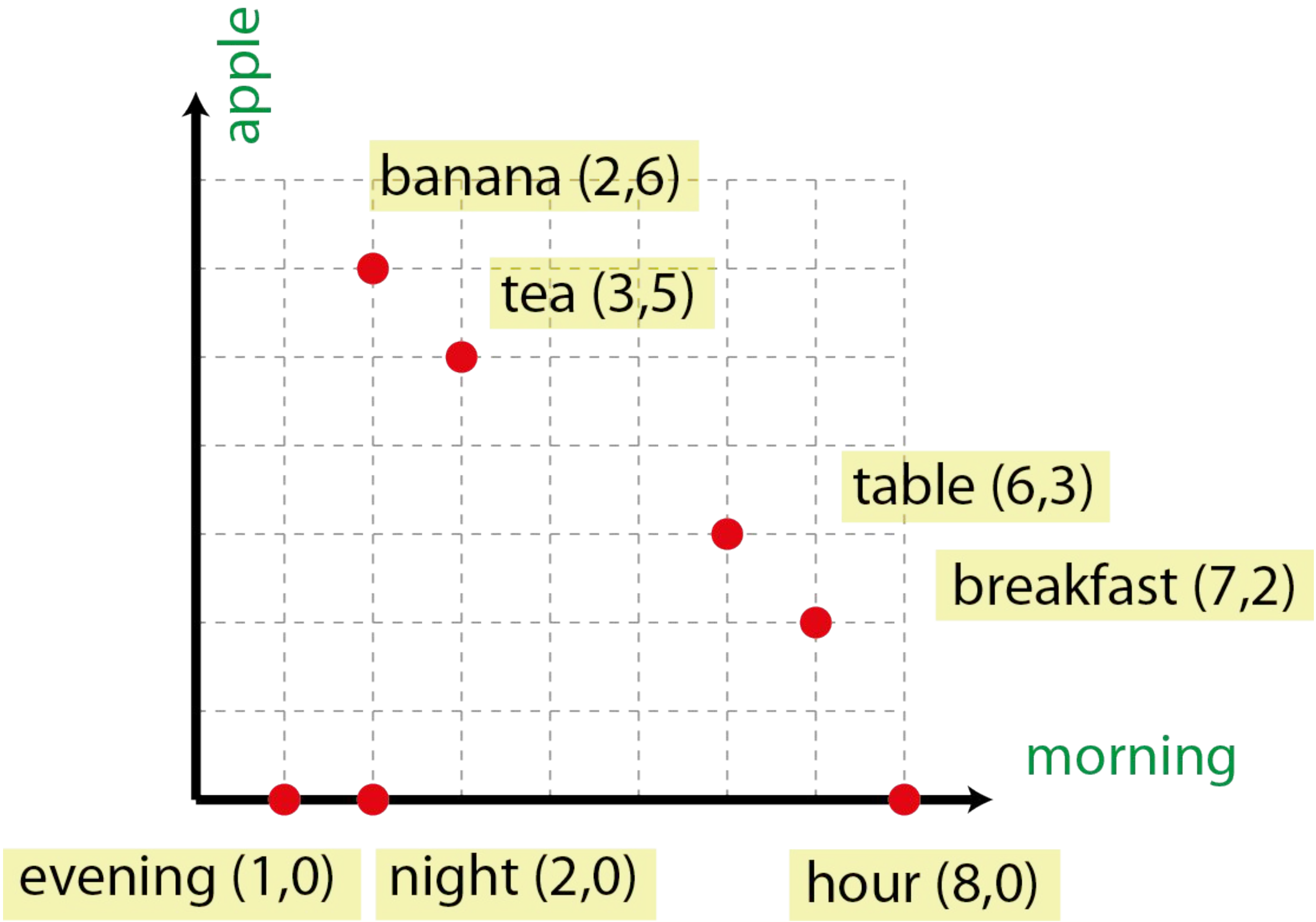
Multidimensionality

1-dimensional space

	morning	apple	late	breakfast	table
morning	49	5	12	7	6
apple	5	9	20	2	3
late	12	20	39	3	1
breakfast	7	2	3	15	10
table	6	3	1	10	20
tea	3	5	2	12	9
banana	2	6	0	3	6
hour	8	0	4	1	0
night	2	0	4	0	0

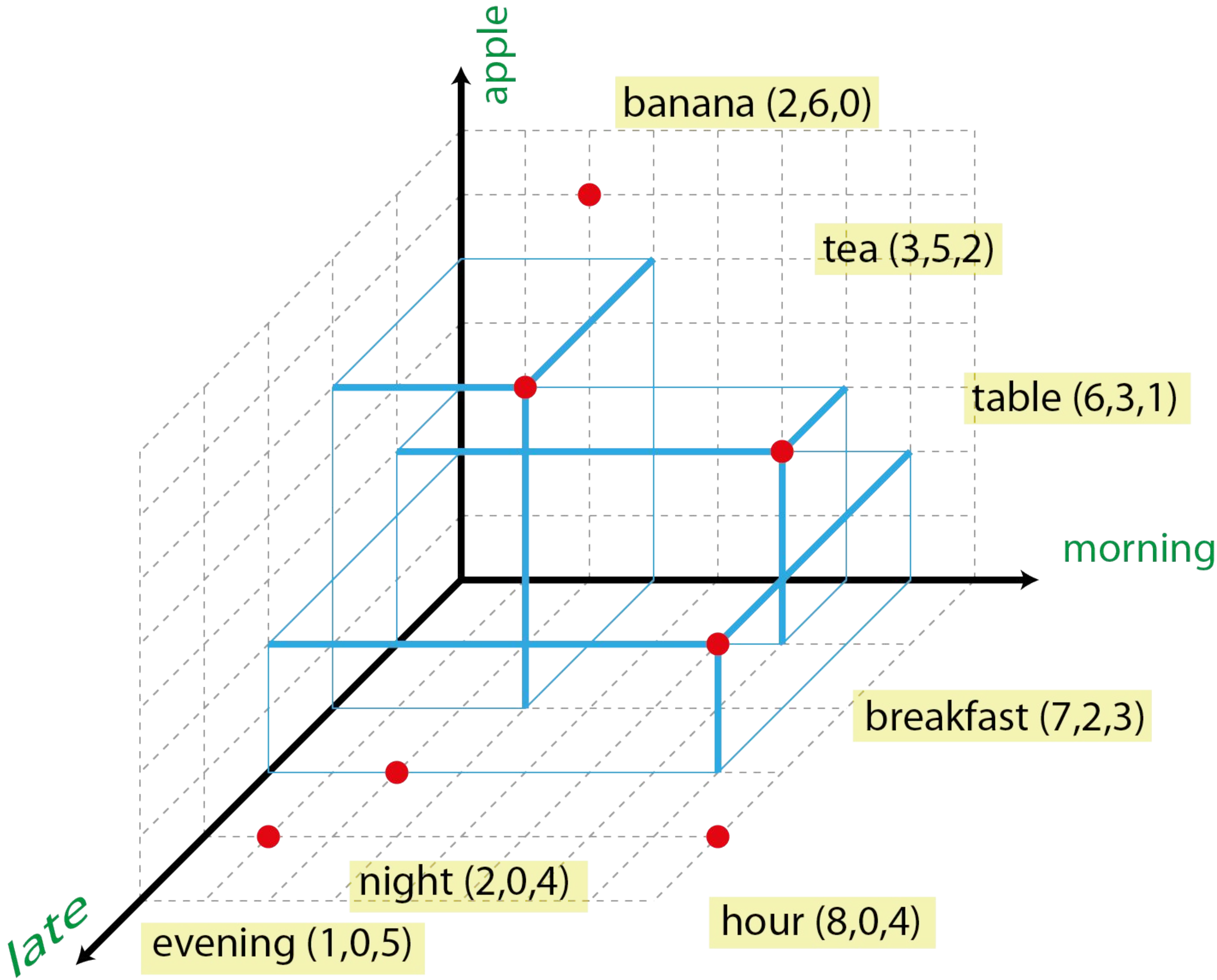
2-dimensional space

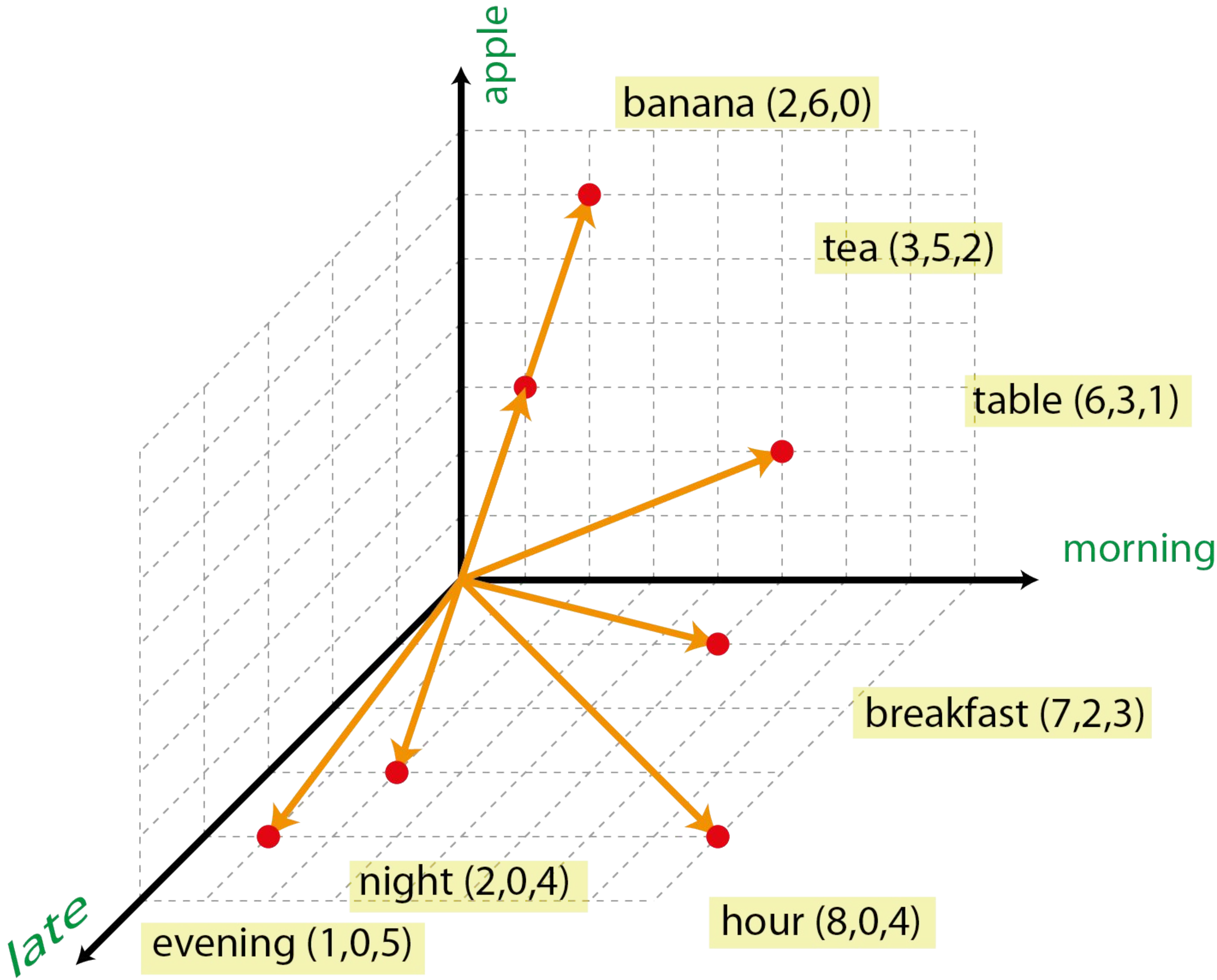
	morning	apple	late	breakfast	table
morning	49	5	12	7	6
apple	5	9	20	2	3
late	12	20	39	3	1
breakfast	7	2	3	15	10
table	6	3	1	10	20
tea	3	5	2	12	9
banana	2	6	0	3	6
hour	8	0	4	1	0
night	2	0	4	0	0


apple
banana (2,6)
tea (3,5)
table (6,3)
breakfast (7,2)
morning
evening (1,0)
night (2,0)
hour (8,0)

3-dimensional space

	morning	apple	late	breakfast	table
morning	49	5	12	7	6
apple	5	9	20	2	3
late	12	20	39	3	1
breakfast	7	2	3	15	10
table	6	3	1	10	20
tea	3	5	2	12	9
banana	2	6	0	3	6
hour	8	0	4	1	0
night	2	0	4	0	0


apple
banana (2,6,0)
tea (3,5,2)
table (6,3,1)
morning
breakfast (7,2,3)
late
night (2,0,4)
hour (8,0,4)
evening (1,0,5)


apple
banana (2,6,0)
tea (3,5,2)
table (6,3,1)
morning
breakfast (7,2,3)
night (2,0,4)
hour (8,0,4)
late
evening (1,0,5)

4-dimensional space

	morning	apple	late	breakfast	table
morning	49	5	12	7	6
apple	5	9	20	2	3
late	12	20	39	3	1
breakfast	7	2	3	15	10
table	6	3	1	10	20
tea	3	5	2	12	9
banana	2	6	0	3	6
hour	8	0	4	1	0
night	2	0	4	0	0

Multi-dimensional space (without plotting it!)

2 dimensions

table = (6, 3)
tea = (3, 5)
banana = (2, 6)

3 dimensions

table = (6, 3, 1)
tea = (3, 5, 2)
banana = (2, 6, 0)

4 dimensions

table = (6, 3, 1, 10)
tea = (3, 5, 2, 12)
banana = (2, 6, 0, 3)

n dimensions

table = (6, 3, 1, 10, 20, ...)
tea = (3, 5, 2, 12, 9, ...)
banana = (2, 6, 0, 3, 6, ...)

Building a space of 36,869 dimensions?

The corpus of 100 English novels: 36,869 unique words.

A reasonably large corpus needs at least 100,000 dimensions.

A corpus of 1,000,000,000 words: more than 500,000 dimensions.

Very easy to build.

Very difficult to manipulate.

Large amounts of RAM needed.

Computations are time consuming.

The hell of too many dimensions

The idea of dimensionality reduction:

- selecting a subset of “best” dimensions

- using neural networks to extract high-order information from particular dimensions
  - word2vec (Mikolov et al. 2013)

- using matrix factorization to reduce dimensions
  - PCA, t-SNE, GloVe, …

WhyR? Because there’s ‘text2vec’!

GloVe: compressing 36,869 dimensions into 50

word_embedding_models — R — 80×24

```
>
> word_vectors[c("morning","late","breakfast","table", "tea", "hour", "night"),
, drop = FALSE]
                [,1]      [,2]      [,3]        [,4]       [,5]        [,6]
morning   -0.4998386 0.8350450 0.5873120 -0.09019544 -0.5345772 -0.77275923
late      -0.9732213 0.1103356 0.4902478 -0.04141034 -0.2574947 -0.50231665
breakfast -0.6918233 0.7649647 0.1739464  0.12478728 -0.7184420  0.33174518
table      0.4921054 0.9053893 0.5450380  0.65241125 -0.8131843 -0.05642531
tea       -0.4009631 0.5501283 0.8121779  0.57247782  0.1606661  0.86464387
hour      -0.8599038 0.5155548 0.2559546  0.62165272 -0.7260329 -0.99435511
night     -0.7714264 0.9512208 0.9479608  0.25956929 -0.3607481 -0.53961708
                 [,7]        [,8]       [,9]       [,10]      [,11]      [,12]
morning    0.35772985  0.03997092  0.3435241 -0.11813826  0.0484014 -0.8525132
late      -0.07495511 -0.07403972 -0.1661168  0.14339211 -0.4475114 -0.2785054
breakfast  0.19370743 -0.32543486  0.5572639 -0.07352333 -0.2561855 -1.3351534
table      0.65669778  0.29875891  1.0548203 -0.14451154  0.2591997 -1.8601638
tea       -0.01855537  0.66382118  0.4017728 -0.07146503 -0.4062743 -1.5018901
hour      -0.29693695 -0.87763810  0.7813829  0.37892531  0.3057022 -0.5688118
night      0.46871918 -0.08623601  0.2940462  0.06773485  0.4148599 -0.4228552
               [,13]      [,14]      [,15]       [,16]       [,17]      [,18]
morning    1.1133878 -0.3312990 0.26591632  0.08771919 -0.50270061  0.4488931
late       0.9875614 -0.7558785 0.36087240  0.24651729 -0.55823122 -0.2315132
breakfast  0.3826097 -0.4766384 0.24586234 -0.07293794 -0.34987316 -0.3140625
table     -0.2147395 -0.3609527 0.60534249 -0.10966569 -0.09097221 -0.8749285
```

The fun part: comparing words!

Nearest neighbor = semantic similarity

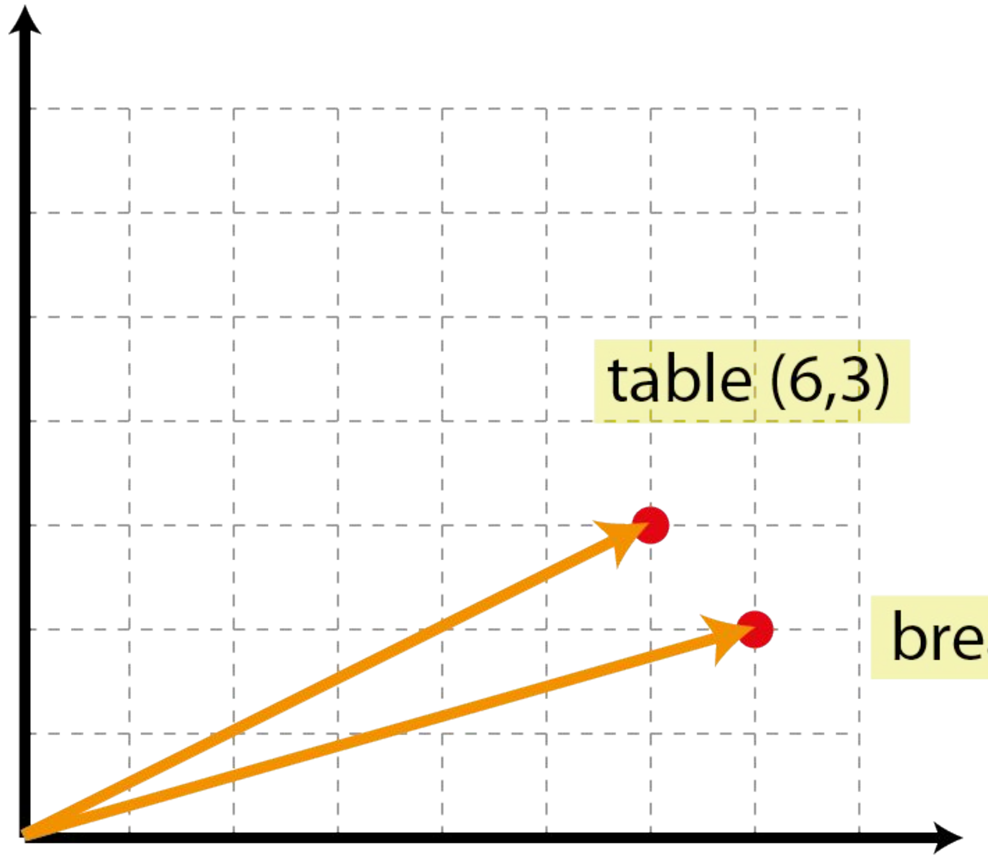
Nearest neighbors to the word ‘sailor’

word_embedding_models — R — 80×24

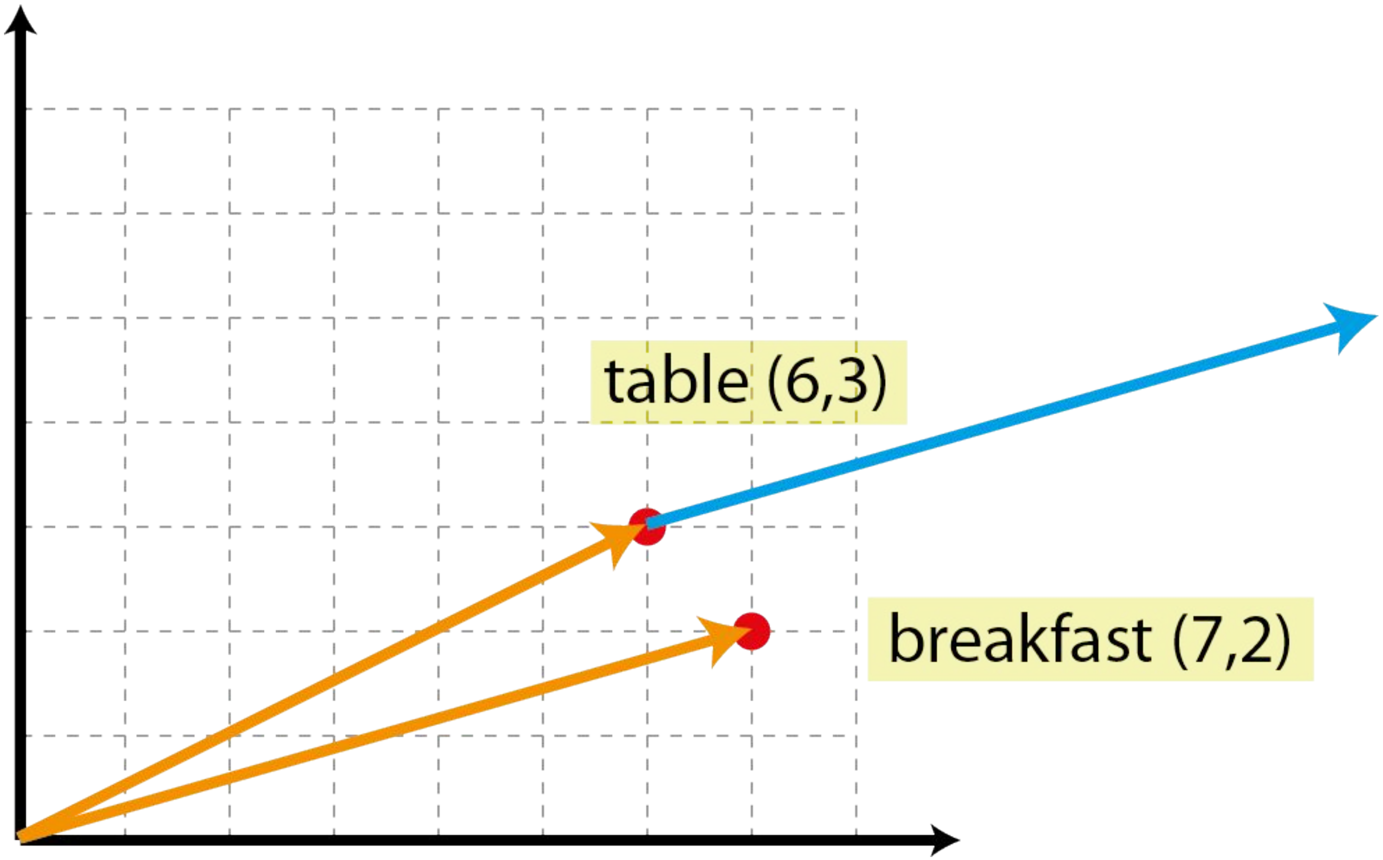
```
>
>
>
>
> my_vector = word_vectors["sailor", , drop = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")
>
> head(sort(cos_sim[,1], decreasing = TRUE), 25)
    sailor     soldier   traveller   clergyman      seaman     officer
 1.0000000   0.7601082   0.7530898   0.7463107   0.6785353   0.6539179
   surgeon  englishman philosopher     veteran     peasant      rascal
 0.6435175   0.6418434   0.6304873   0.6287759   0.6276460   0.6172152
 frenchman      sinner    horseman    sculptor     actress       actor
 0.6149195   0.6133865   0.6108677   0.6089718   0.6075764   0.6071017
 barrister     gallant      fellow     student      spider     scholar
 0.6036215   0.5897630   0.5878295   0.5818883   0.5815894   0.5703828
      poet
 0.5680710
>
>
>
>
>
>
```

Adding vectors

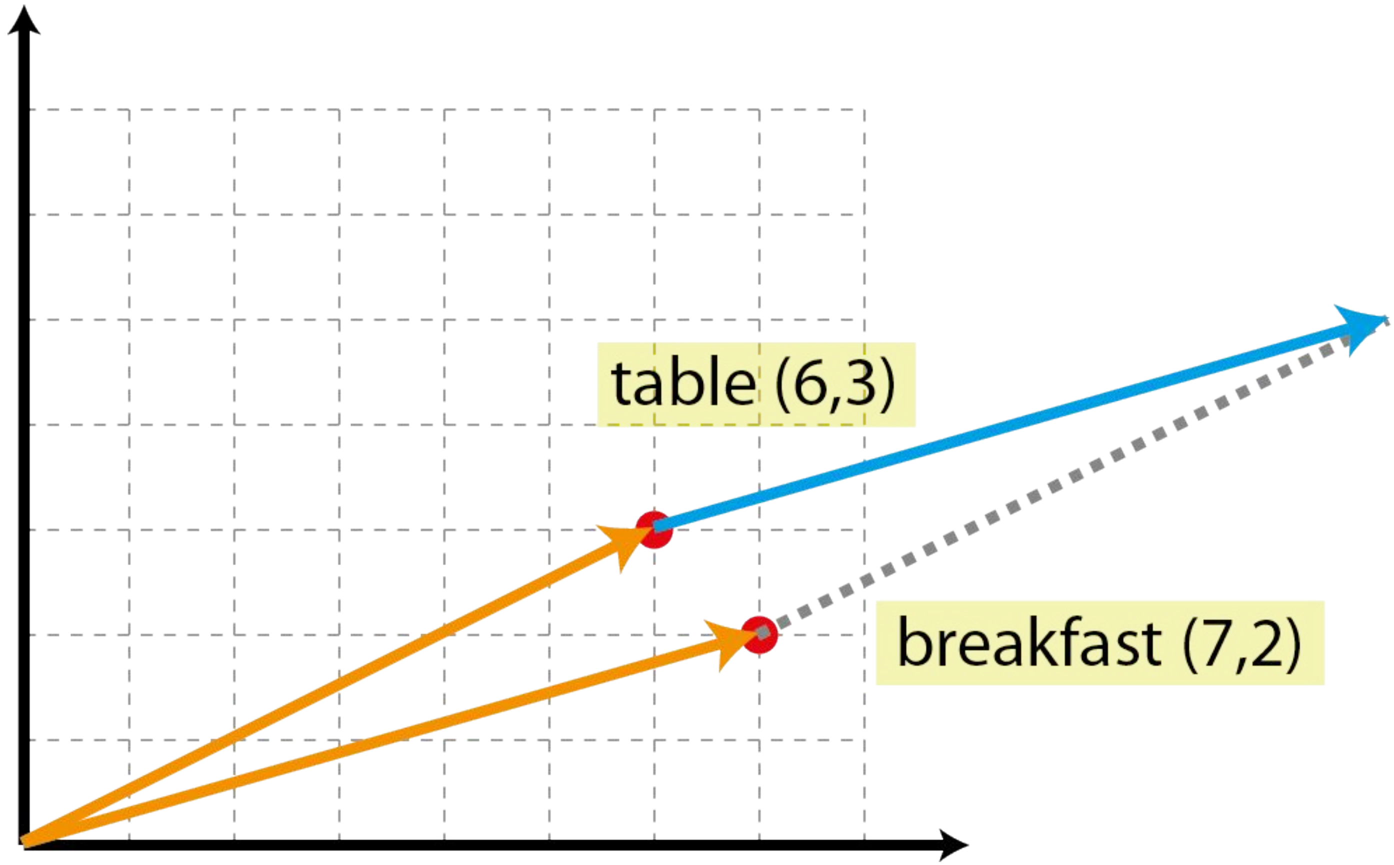
‘table’ (6,3) + ‘breakfast’ (7,2)



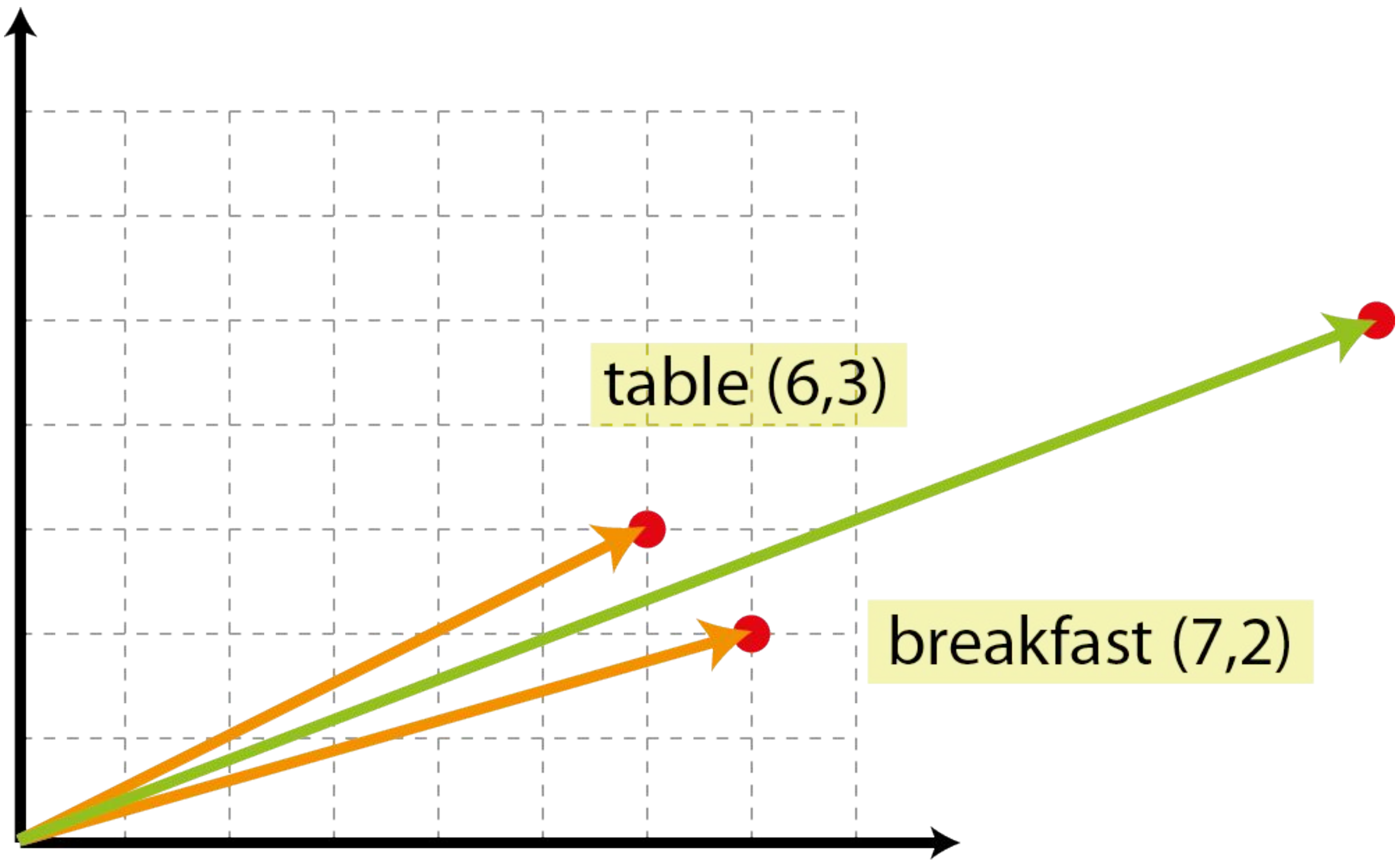
‘table’ (6,3) + ‘breakfast’ (7,2)



‘table’ (6,3) + ‘breakfast’ (7,2)



‘table’ (6,3) + ‘breakfast’ (7,2)



Adding vectors: is it that easy? really?

‘table’	= (6, 3)
‘breakfast’	= (7, 2)
$v_1 + v_2$	= (13, 5)

‘table’	= (6, 3, 1, 10, 20, ...)
‘breakfast’	= (7, 2, 3, 15, 10, ...)
$v_1 + v_2$	= (13, 5, 4, 25, 30, ...)

Polysemy and homonymy

Polysemy is the capacity for a word to have multiple meanings.

‘wood’ – a piece of a tree
– a geographical area with many trees

Homonymy is the situation when two words of different meanings have the same spelling.

‘bank’ – the land alongside a river or lake
– a financial institution

‘fly’ – (verb) move through the air using wings
– (noun) a flying insect

Nearest neighbors to the word ‘fly’

word_embedding_models — R — 80×24

```
>
>
>
>
> my_vector = word_vectors["fly", , drop = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")
>
> head(sort(cos_sim[,1], decreasing = TRUE), 25)
      fly     throw     fling      leaf     break     steal       run     flies
1.0000000 0.6467766 0.6299961 0.6246969 0.6176590 0.6134873 0.6005637 0.5879509
     drag       off      away      flew      bird   rushed      melt      fade
0.5653130 0.5615228 0.5598641 0.5597132 0.5591125 0.5571684 0.5535182 0.5519729
  swallow       bay       out      move      from       win     horse      jump
0.5442571 0.5441752 0.5408818 0.5384753 0.5353849 0.5344742 0.5318074 0.5311232
     leap
0.5274384
>
>
>
>
>
>
>
>
```

Nearest neighbors to the meta-vector ‘fly’ + ‘flying’

word_embedding_models — R — 80×24

```
>
>
>
>
> # adding vectors 'fly' and 'flying'
>
> my_vector = word_vectors["fly", , drop = FALSE] + word_vectors["flying", , dro
p = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")
>
> head(sort(cos_sim[,1], decreasing = TRUE), 25)
      fly    flying     flies     birds       off      bird       run      away
0.8653735 0.8181322 0.6448537 0.6360863 0.6323331 0.6307387 0.6261180 0.6187517
     flew      fade     wings      from      shot      wind       sea     break
0.6181654 0.6002336 0.5868018 0.5835881 0.5817891 0.5702739 0.5700460 0.5696137
   rushed     drove       out   running   blowing   rattled   through     horse
0.5694868 0.5653354 0.5649315 0.5597399 0.5596051 0.5484567 0.5479156 0.5476267
   leaves
0.5457169
>
>
>
>
>
```

Nearest neighbors to the meta-vector ‘fly’ + ‘insect’

word_embedding_models — R — 80×24

```
>
>
>
>
> # adding vectors 'fly' and 'insect'
>
> my_vector = word_vectors["fly", , drop = FALSE] + word_vectors["insect", , dro
p = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")
>
> head(sort(cos_sim[,1], decreasing = TRUE), 25)
      fly   insect      bird     flies       dog      leaf       egg       owl
0.8108235 0.7193251 0.7080360 0.5570992 0.5433246 0.5369178 0.5311136 0.5294823
      pie    squint     sheep     eagle     beast     shell       bay      turk
0.5199201 0.5101511 0.5093800 0.5056634 0.5036850 0.5001367 0.4988964 0.4928782
    flesh     snake     birds      cart       cow     tiger     aspen  electric
0.4912413 0.4869208 0.4844103 0.4803065 0.4800823 0.4769653 0.4754988 0.4734288
    stray
0.4731378
>
>
>
>
>
```

Subtracting vectors

Subtracting vectors: hmm… it's simple, too!

‘table’	= (6, 3)
‘breakfast’	= (7, 2)
$v_1 + v_2$	= (–1, 1)

‘table’	= (6, 3, 1, 10, 20, ...)
‘breakfast’	= (7, 2, 3, 15, 10, ...)
$v_1 + v_2$	= (–1, 1, –2, –5, 10, ...)

‘man’ – ‘woman’ = a male subspace?

word_embedding_models — R — 80×24

```
>
>
>
>
> # subtracting the vectors 'woman' from the vector 'man'
>
> my_vector = word_vectors["man", , drop = FALSE] - word_vectors["woman", , drop
 = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")
>
> head(sort(cos_sim[,1], decreasing = TRUE), 25)
    saxon  regiment    lawyer     guard     sword  fighting    fought    troops
0.5771056 0.5740115 0.5680868 0.5429482 0.5429394 0.5314613 0.5081317 0.4921744
     main   drained    labour     hatch       leg   cutting      monk    mutton
0.4907867 0.4851692 0.4808583 0.4808496 0.4802772 0.4729971 0.4717324 0.4683175
    enemy    vholes  pocketed     swore      vote followers promotion     horse
0.4680031 0.4669026 0.4638248 0.4618321 0.4616109 0.4603224 0.4596896 0.4595477
  bargain
0.4594418
>
>
>
>
> 
```

‘woman’ – ‘man’ = a female subspace?

word_embedding_models — R — 80×24

```
>
>
>
>
> # subtracting the vectors 'man' from the vector 'woman'
>
> my_vector = word_vectors["woman", , drop = FALSE] - word_vectors["man", , drop
 = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")
>
> head(sort(cos_sim[,1], decreasing = TRUE), 25)
    ironing    foreboded commonplaces     girlhood      dimples        ignis
  0.5939901    0.5803606    0.5746604    0.5722333    0.5625328    0.5518884
 coquettish    georgette       armida        janee      calypso        decks
  0.5510084    0.5503963    0.5451703    0.5443584    0.5428159    0.5414840
  scapulary       pouted     snowdrop       finery        abbot       paints
  0.5382216    0.5342270    0.5331425    0.5307851    0.5288597    0.5283636
      album          zen       gibing      pyjamas       ainley journalistic
  0.5279419    0.5251386    0.5241894    0.5227228    0.5219709    0.5193048
    girlish
  0.5190120
>
>
> 
```

Adding & subtracting

‘woman’ – ‘man’ + ‘king’ = ???

‘woman’ – ‘man’ + ‘king’ = ‘queen’

word_embedding_models — R — 80×24

```
>
>
>
>
> # the famous 'queen' example
>
> my_vector = word_vectors["woman", , drop = FALSE] - word_vectors["man", , drop
 = FALSE] + word_vectors["king", , drop = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")
>
> head(sort(cos_sim[,1], decreasing = TRUE), 25)
      king      queen        son       duke      reign    england        god
 0.8075043  0.6938168  0.6046449  0.5784283  0.5736905  0.5696248  0.5646961
     north    solomon   daughter    blanche    charles margravine    majesty
 0.5550476  0.5511255  0.5483749  0.5477272  0.5457612  0.5451942  0.5418842
  monmouth     heaven     sister    brother      twala      james      elder
 0.5370609  0.5325900  0.5325644  0.5323389  0.5274534  0.5243814  0.5097400
    father     prince       earl       born
 0.5096470  0.5093563  0.5034941  0.5034924
>
>
>
>
>
```

Grammatical subspaces

A subspace for plural forms

‘dogs’ – ‘dog’ + ‘man’ = [expected: ‘men’]

Taking advantage of adding vectors

(‘dogs’ – ‘dog’) + (‘cats’ – ‘cat’) + ‘man’ = [expected: ‘men’]

A subspace for past tense

(‘could’ – ‘can’) + (‘did’ – ‘do’) + ‘is’ = [expected: ‘was’] ‘woman’ – ‘man’ + ‘trousers’ = ???

‘woman’ – ‘man’ + ‘trousers’ = ‘gowns’

word_embedding_models — R — 80×24

```
>
>
>
>
> # female equivalent to 'trousers'
>
> my_vector = word_vectors["woman", , drop = FALSE] - word_vectors["man", , drop
 = FALSE] + word_vectors["trousers", , drop = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")

> head(sort(cos_sim[,1], decreasing = TRUE), 25)
    trousers        gowns      flannel       cotton     sealskin         silk
   0.7692682    0.7013703    0.6927098    0.6729204    0.6702147    0.6494591
       frock        plaid        linen        apron         robe       muslin
   0.6376247    0.6276649    0.6181248    0.6172595    0.6059950    0.6015045
     closets          pea        satin         gown        shawl        smock
   0.5995790    0.5908928    0.5811999    0.5776781    0.5758523    0.5753766
handkerchief     clothing       duffle      woollen       frocks       bonnet
   0.5735857    0.5694255    0.5685790    0.5625112    0.5566792    0.5550128
     wrapper
   0.5533696
>
>
>
```

‘woman’ – ‘man’ + ‘wealthy’ = ???

'woman' – 'man' + 'wealthy' = 'buxom' (!)

word_embedding_models — R — 80×24

```
>
>
>
>
> # female similarities to 'wealthy'
>
> my_vector = word_vectors["wealthy", , drop = FALSE] - word_vectors["man", , dr
op = FALSE] + word_vectors["woman", , drop = FALSE]
> cos_sim = sim2(x = word_vectors, y = my_vector, method = "cosine", norm ="l2")
>
> head(sort(cos_sim[,1], decreasing = TRUE), 25)
      wealthy         buxom       goddess   scapegrace inexperienced
    0.7059662     0.6611004     0.6403990    0.6387685     0.5855287
   handsomest       amiable        devout      fairest         deity
    0.5800985     0.5761505     0.5729314    0.5700947     0.5681629
     oleander       elegant       elderly  restitution        shroud
    0.5668195     0.5514754     0.5479551    0.5380491     0.5378170
     agonized        yankee          boer        abbot      blooming
    0.5349051     0.5345736     0.5342318    0.5330943     0.5324355
       kaffir     miscreant        cowled      saintly    uneducated
    0.5316644     0.5231922     0.5211477    0.5195454     0.5170840
>
>
>
```

Positioning words on a plane

Defining, say, 150 words similar to a meta-vector

1. First, compute a meta-vector by adding vectors of *n* seed words:

   emma, john, joseph, mary, jane, elisabeth, anthony

2. Then, get *m* words (here, 150 words) similar to the meta-vector:

   esther, lucy, sophy, joe, jane, martin, jemima, helen, mary, poyser, garth, arthur, jude, richard, ruth, margaret, sue, lettie, ada, aunt, bathsheba, sally, milly, emma, robert, philip, janet, leila, riccabocca, tess, leonard, lionel, caroline, roderick, graham, adrian, caddy, antony, alicia, john, sara, ayesha, mordecai, katharine, zara, hammond, laura, leslie, lambert, uncle, said, pip, fagin, ralph, shirley, mirah, dear, heriot, estella, clare, eleanor, lorna, pendennis, adam, felix, troy, leo, kenelm, boldwood, ...

Defining two semantic subspaces

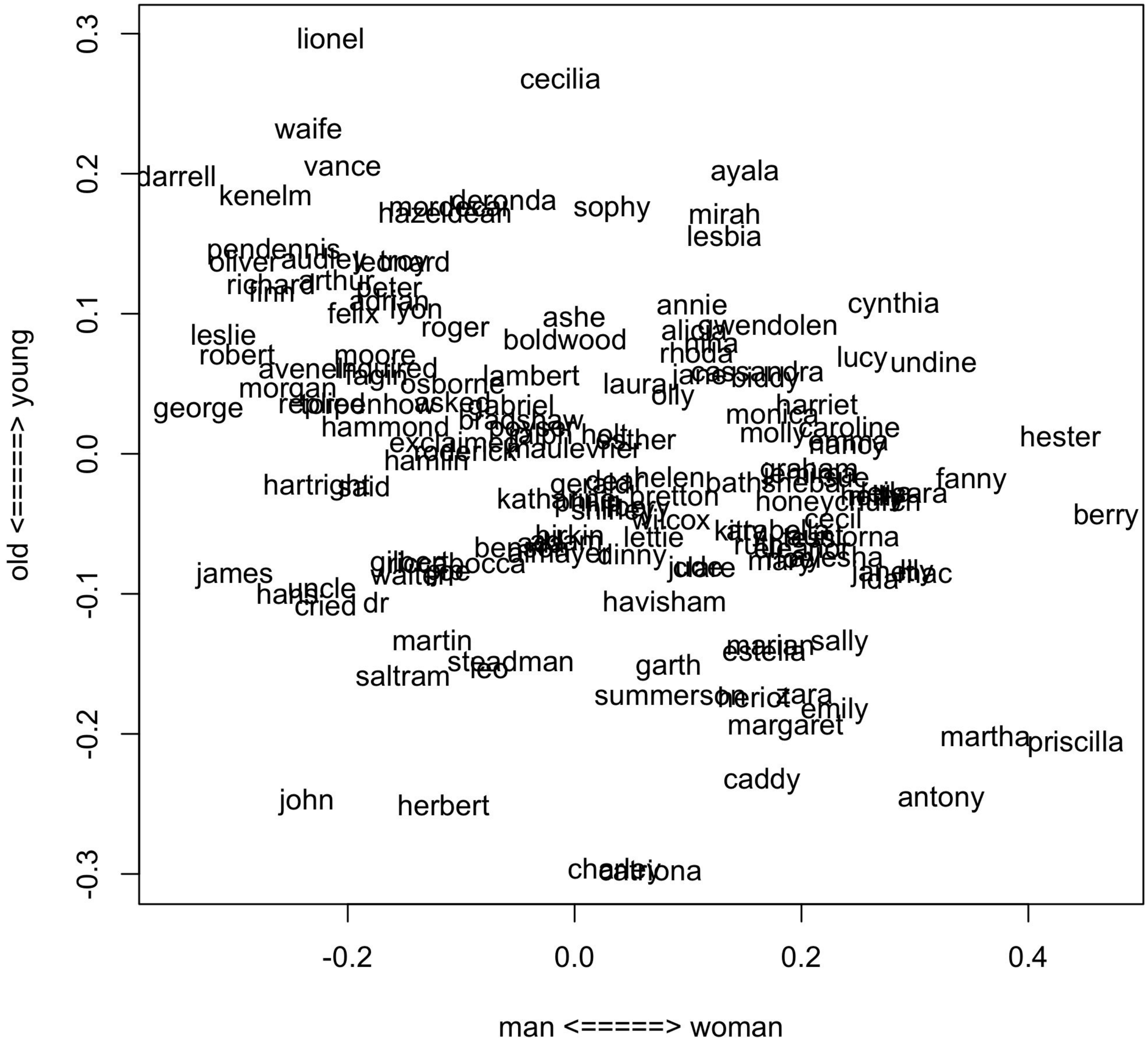
3. Define a gender-related space by subtracting vectors:

    ‘woman’ – ‘man’ = meta-vector 1

4. Define an age-related space by subtracting vectors:

    ‘young’ – ‘old’ = meta-vector 2

5. Then, plot the 150 chosen words against two axes representing the two meta-vectors


lionel
cecilia
waife
vance
darrell
ayala
kenelm
deronda
sophy
mirah
hazeldean
lesbia
pendennis
oliver
leonard
richard
arthur
peter
finn
adrian
cynthia
annie
felix
lyon
ashe
roger
gwendolen
leslie
boldwood
alicia
rhoda
robert
moore
lucy
undine
lambert
cassandra
morgan
osborne
laura
biddy
george
olly
harriet
hammond
monica
gabriel
caroline
hester
molly
esther
nancy
hamlin
graham
fanny
hartright
said
helen
bretton
katharine
honeychurch
berry
wilcox
cecil
lettie
lorna
dinny
james
uncle
hans
cried
dr
havisham
martin
sally
estella
steadman
saltram
garth
summerson
heriot
zara
emily
margaret
martha
priscilla
caddy
antony
john
herbert
catriona
-0.2
0.0
0.2
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
man <=====> woman
old <=====> young

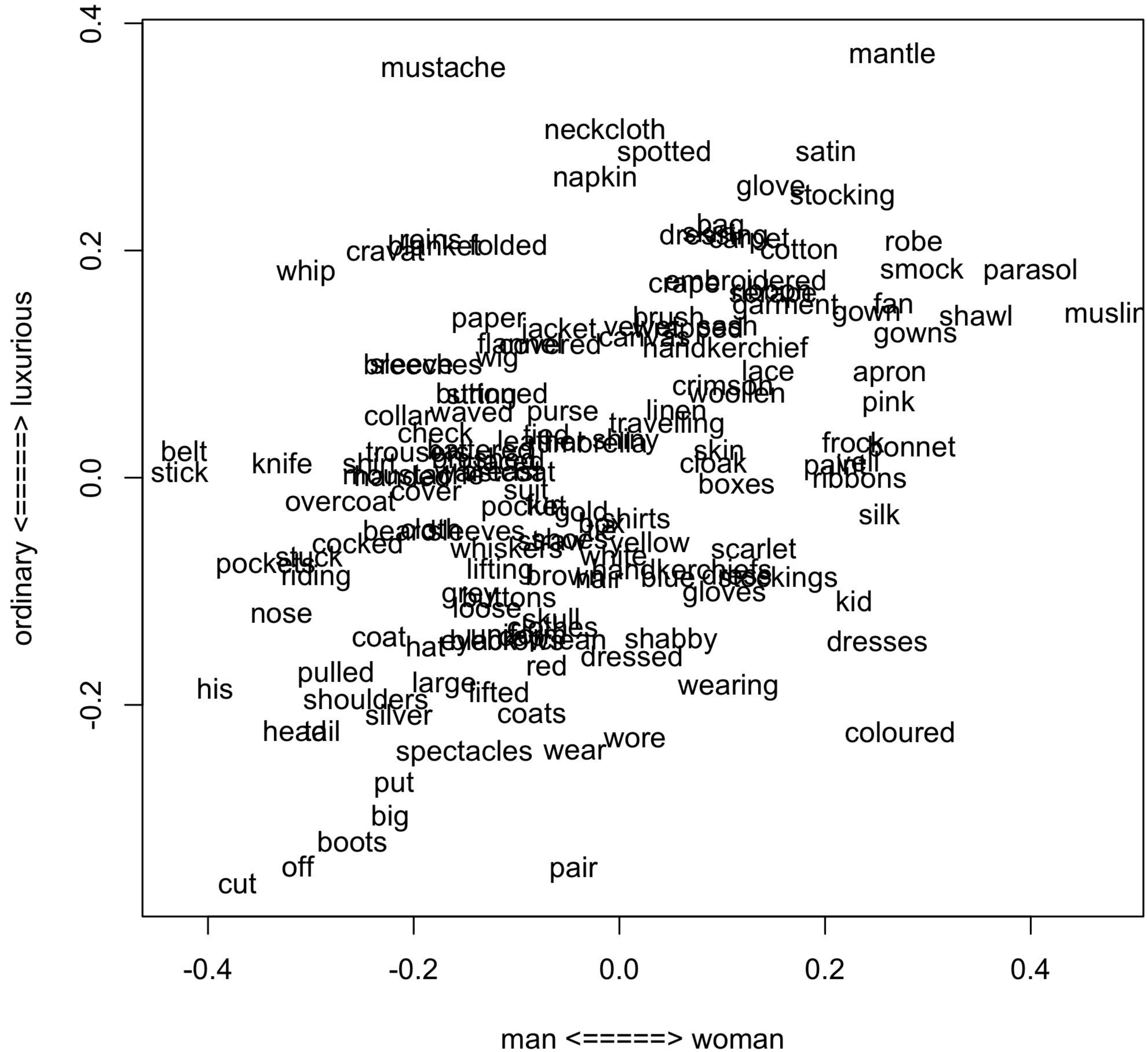
Since it works for names…

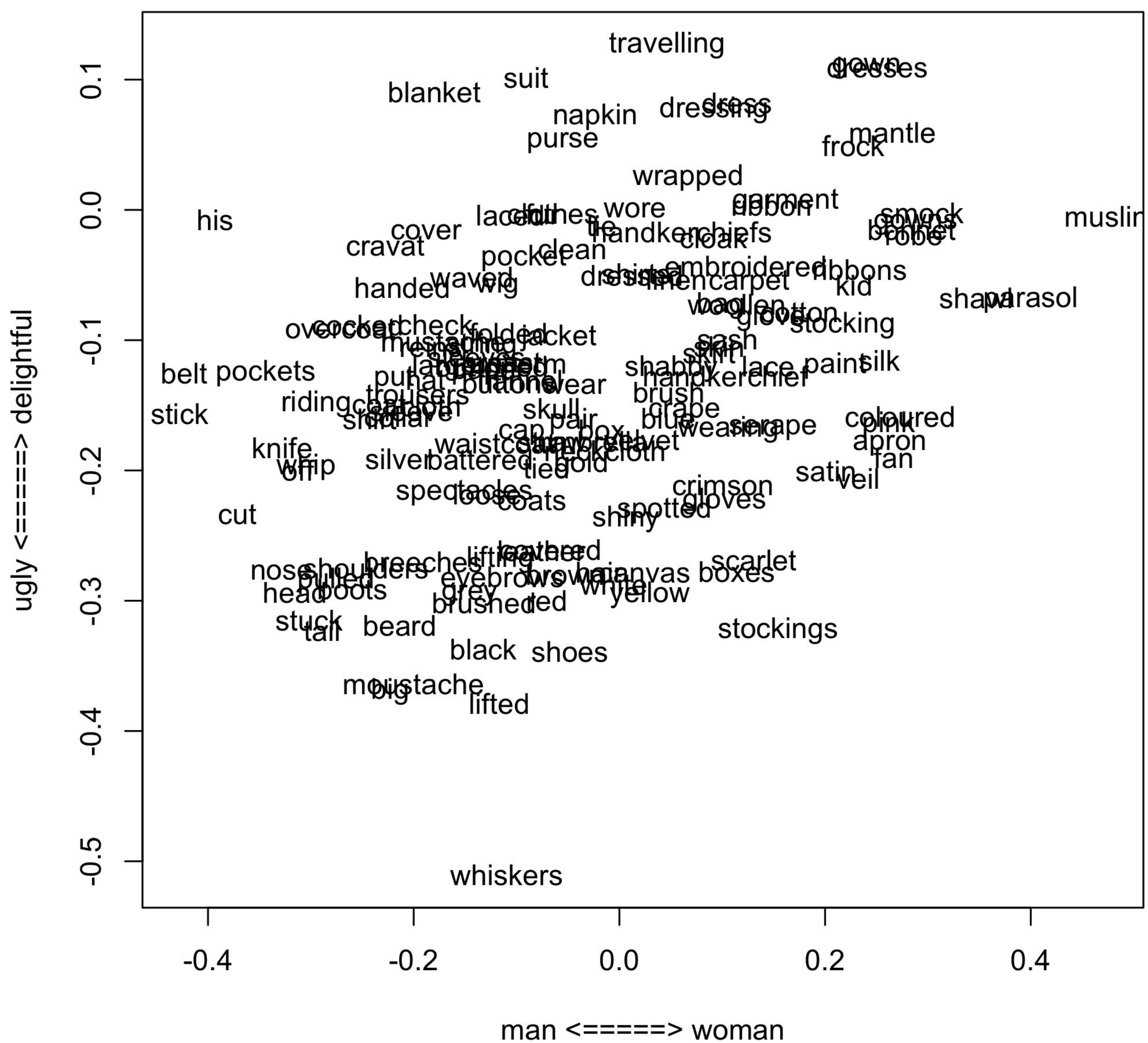
Seed words:

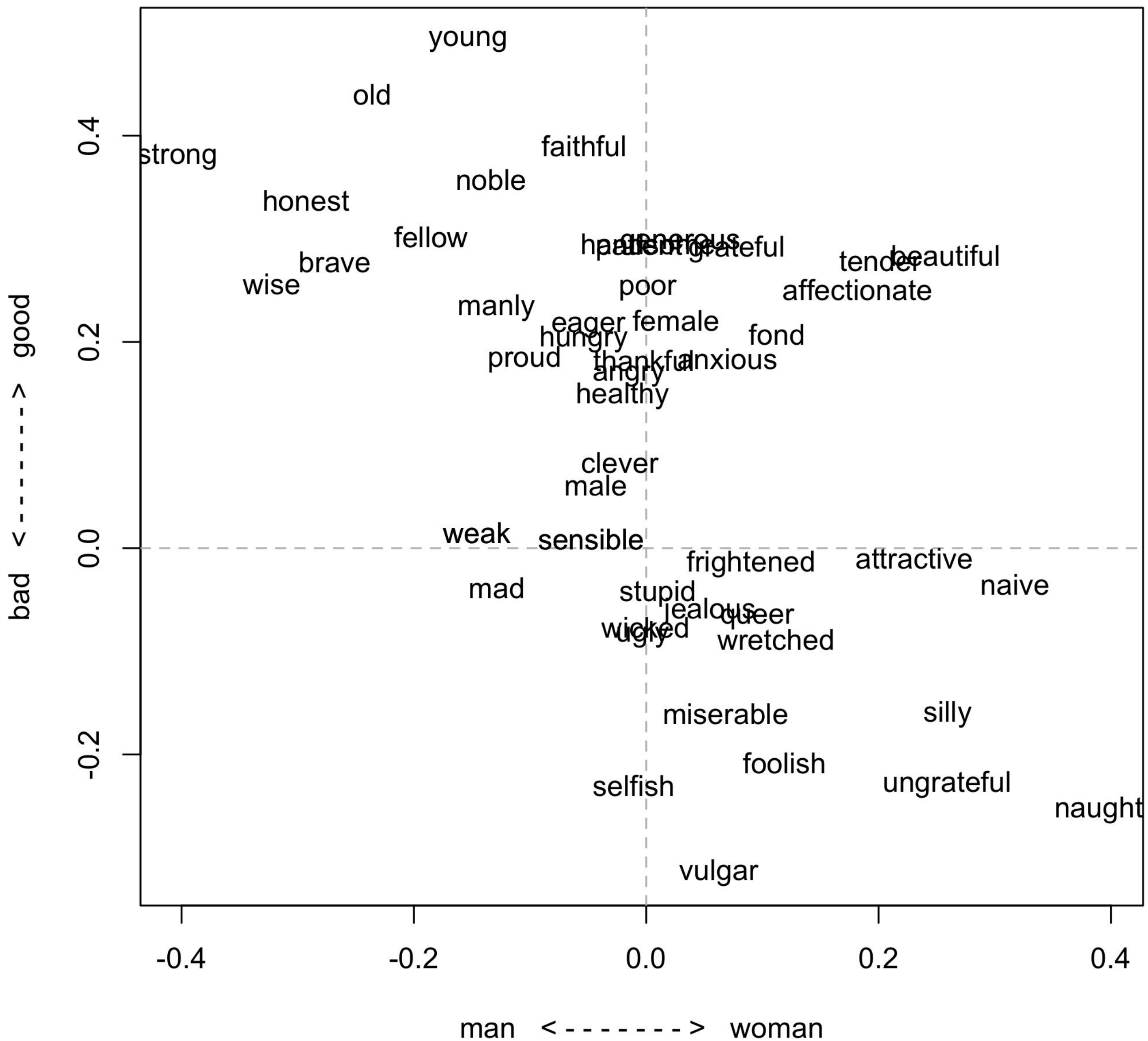
hat, trousers, shirt, coat, waistcoat, cap, umbrella, dress, gloves

150 neighboring words to the above meta-vector:

coat, hat, cap, waistcoat, frock, trousers, handkerchief, silk, cloak, umbrella, gloves, shirt, pocket, dress, wearing, cloth, clothes, black, velvet, wore, gown, jacket, white, flannel, sleeve, apron, bonnet, linen, boots, sleeves, moustache, spectacles, collar, fur, tail, buttoned, wig, kid, red, satin, pulled, shawl, beard, lace, tied, skirt, bag, clean, scarlet, yellow, stockings, breeches, suit, blue, uniform, crape, pink, dressing, skin, coats, overcoat, blanket, cover, leather, brown, put, ribbon, cotton, ...


0.4
0.2
0.0
-0.2
-0.4
-0.2
0.0
0.2
0.4
ordinary <=====> luxurious
man <=====> woman
mustache
mantle
neckcloth
spotted
satin
napkin
glove
stocking
robe
cotton
whip
smock
parasol
folded
shawl
gowns
handkerchief
lace
apron
crimson
woollen
pink
collar
waved
purse
linen
travelling
check
belt
knife
stick
frock
bonnet
skin
cloak
boxes
ribbons
overcoat
cover
suit
silk
shirts
yellow
scarlet
pockets
riding
gloves
kid
nose
shabby
coat
dresses
dressed
red
pulled
wearing
his
large
lifted
shoulders
silver
coats
coloured
wore
spectacles
wear
put
big
boots
off
pair
cut


man <=====> woman
ugly <=====> delightful
-0.4
-0.2
0.0
0.2
0.4
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
travelling
gown
dresses
suit
blanket
napkin
dressing
dress
purse
mantle
frock
wrapped
garment
ribbon
his
wore
smock
muslin
clothes
cover
handkerchiefs
cloak
bonnet
robe
cravat
pocket
clean
embroidered
ribbons
handed
waved
wig
linen
carpet
kid
shawl
parasol
cotton
stocking
overcoat
check
folded
jacket
sash
skirt
belt
pockets
shabby
lace
paint
silk
handkerchief
wear
brush
riding
trousers
stick
skull
crape
blue
pair
box
coloured
wearing
serape
pink
cap
knife
apron
whip
silver
battered
tied
gold
fan
satin
veil
crimson
spectacles
loose
coats
gloves
spotted
shiny
cut
leather
scarlet
nose
shoulders
breeches
canvas
boxes
eyebrows
white
yellow
head
boots
grey
brushed
red
stuck
beard
tall
stockings
black
shoes
moustache
big
lifted
whiskers


young
old
strong
faithful
noble
honest
fellow
brave
wise
grateful
tender
beautiful
poor
affectionate
manly
eager
female
hungry
fond
proud
thankful
anxious
angry
healthy
clever
male
weak
sensible
frightened
attractive
mad
naive
stupid
jealous
queer
wicked
ugly
wretched
miserable
silly
foolish
selfish
ungrateful
naught
vulgar
0.4
0.2
0.0
-0.2
-0.4
bad <- - - - - - - -> good
man <- - - - - - - -> woman

Conclusions

Several tasks:

- similarity
- analogy
- refining word meaning
- distributional semantics hypothesis

Several applications:

- Machine translation (two models compared)
- Named Entity Recognition
- Tracing change of word meaning over time