Computational Analysis of the Body in Europen Fariy Tales

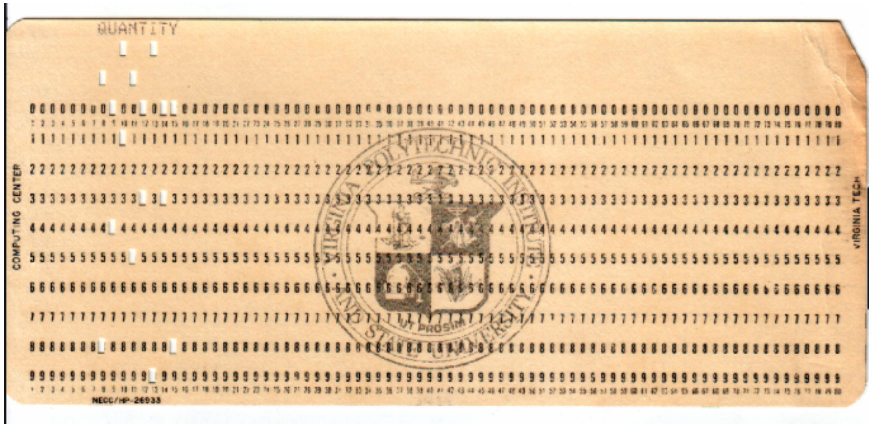
Motivation

We want to use computational analysis to understand the representations and constructions of gender and body in European fairytales

Why Fairy tales?

Previous methods

- Counting words that are suspected of being related to one group of characters
- Punch cards



- Nowadays

The Data Collection

*The data analyzed was from six tale collections

* 223 tales

* some literary and classical

How the data was collected

- The data was inserted to the data base by hand (hand-coded)

- It contained 11,000 entries

- The researchers included 13 points of data about the tales themselves and 14 points of data about every word

- The data unlike in other projects was inserted by one man

Why?

* reintroducing the humanistic importance

* some data needed interpretation

* the research cant really be objective

* allowing to move beyond the traditional ‘count the word’

collocation Extraction

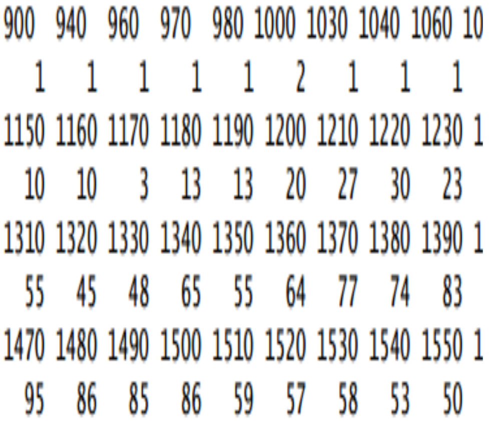
- collocation is a sequence of words
  or terms that co-occur more often than would be expected by chance

So why hand-coded

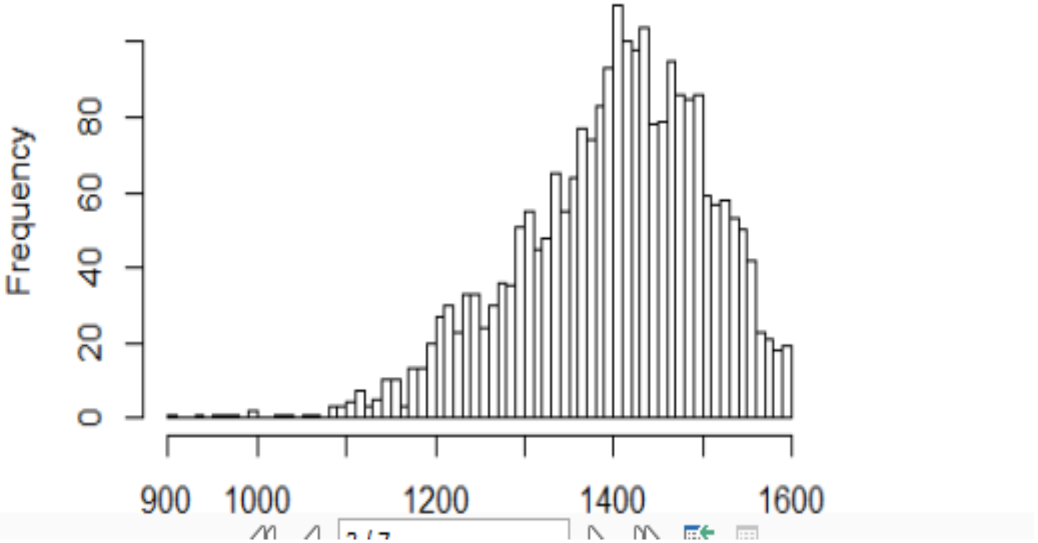
- No test can be objective
- Deeper understanding of the text using the settings surrounding the text

Hand-coding? Sounds too simple

- One may say that by using Hand-coding we don’t need computers because the people are doing all of the work



```
 900  940  960  970  980 1000 1030 1040 1060 10
   1    1    1    1    1    2    1    1    1
1150 1160 1170 1180 1190 1200 1210 1220 1230 1
  10   10    3   13   13   20   27   30   23
1310 1320 1330 1340 1350 1360 1370 1380 1390 1
  55   45   48   65   55   64   77   74   83
1470 1480 1490 1500 1510 1520 1530 1540 1550 1
  95   86   85   86   59   57   58   53   50
```

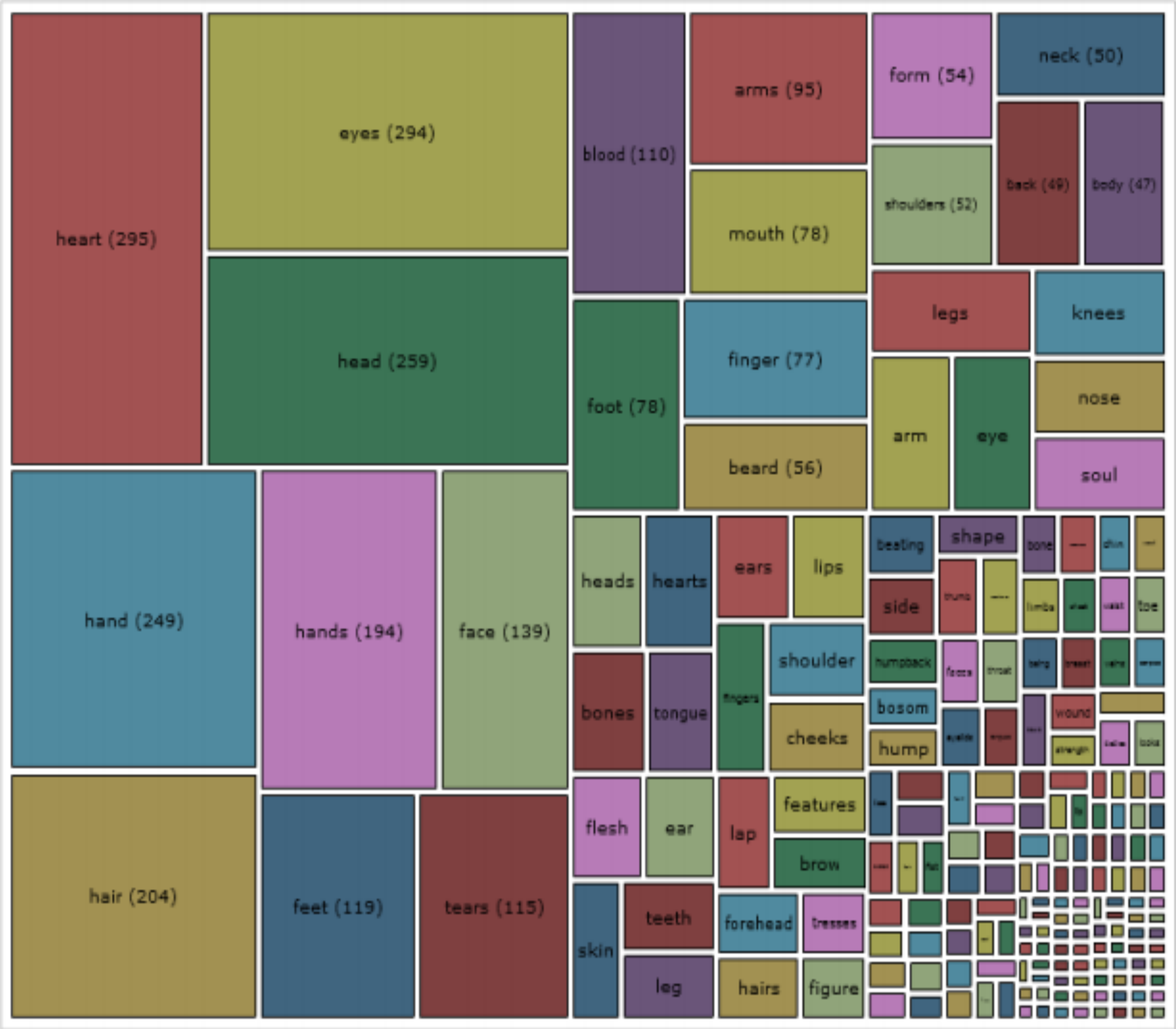


Data Analysis

- After the insertion of all data successfully we want to analyze it. the first step is to extract all the basic statistics.

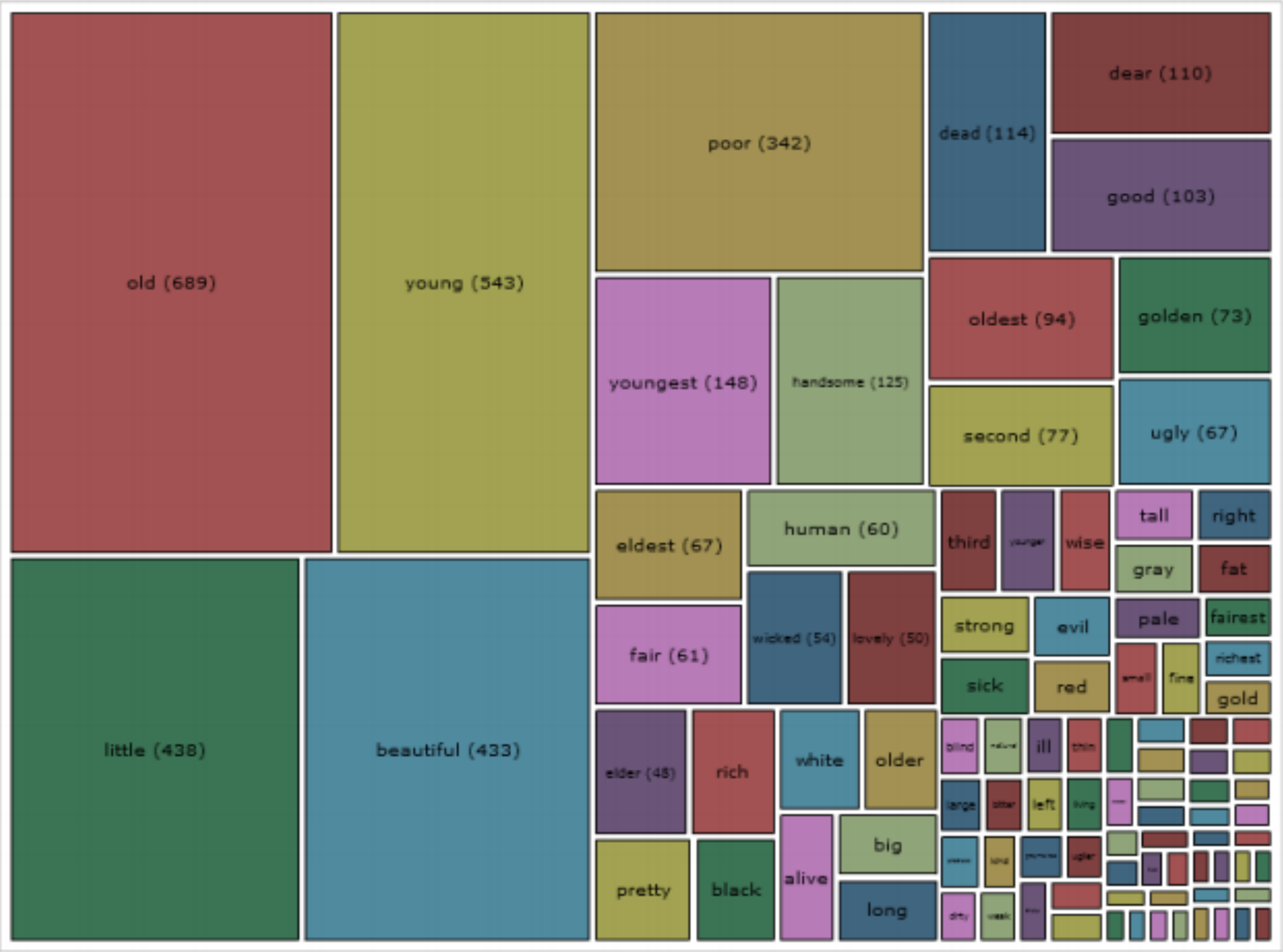
The results

- 139 different body parts were mentioned
- heart (295 mentions), eyes (294), head (259), hand (249), hair (204), hands (194), face (139), feet (119), tears (115), and blood (110)
- arms (95), foot (78), mouth (78), finger (77), beard (56), form (54), shoulders (52), neck (50), back (49), and body (47)


heart (295)
eyes (294)
head (259)
blood (110)
arms (95)
mouth (78)
form (54)
shoulders (52)
neck (50)
back (49)
body (47)
hand (249)
hands (194)
face (139)
foot (78)
finger (77)
beard (56)
legs
knees
arm
eye
nose
soul
hair (204)
feet (119)
tears (115)
heads
hearts
ears
lips
bones
tongue
fingers
shoulder
cheeks
beating
shape
bone
chin
side
thumb
limbs
waist
toe
humpback
faces
throat
being
breast
bosom
hump
eyelids
wound
strength
locks
flesh
ear
lap
features
brow
skin
teeth
leg
forehead
tresses
hairs
figure

Adjectival description

- old (689 mentions), young (543), little (438), beautiful (433), poor (342), youngest (148), handsome (125), dead (114), dear (110), and good (103)
- oldest (94 mentions), second (77), golden (73), eldest (67), ugly (67), fair (61), human (60), wicked (54), lovely (50), and elder (48)

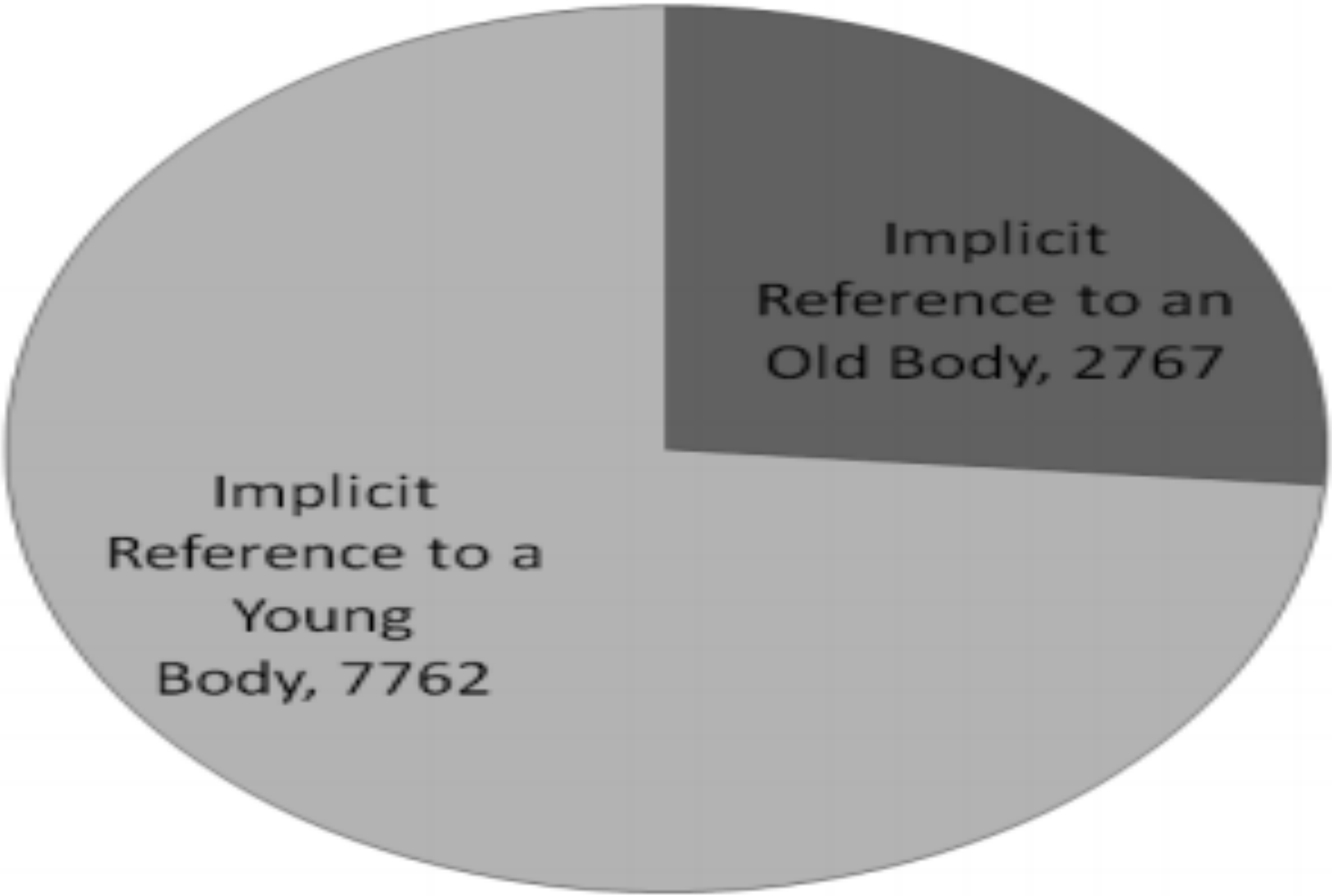

old (689)
young (543)
poor (342)
dead (114)
dear (110)
good (103)
youngest (148)
handsome (125)
oldest (94)
golden (73)
second (77)
ugly (67)
little (438)
beautiful (433)
eldest (67)
human (60)
third
younger
wise
tall
right
gray
fat
fair (61)
wicked (54)
lovely (50)
strong
evil
pale
fairest
richest
sick
red
small
fine
gold
elder (48)
rich
white
older
blind
natural
ill
thin
large
bitter
left
living
pretty
black
alive
big
long
dirty
weak

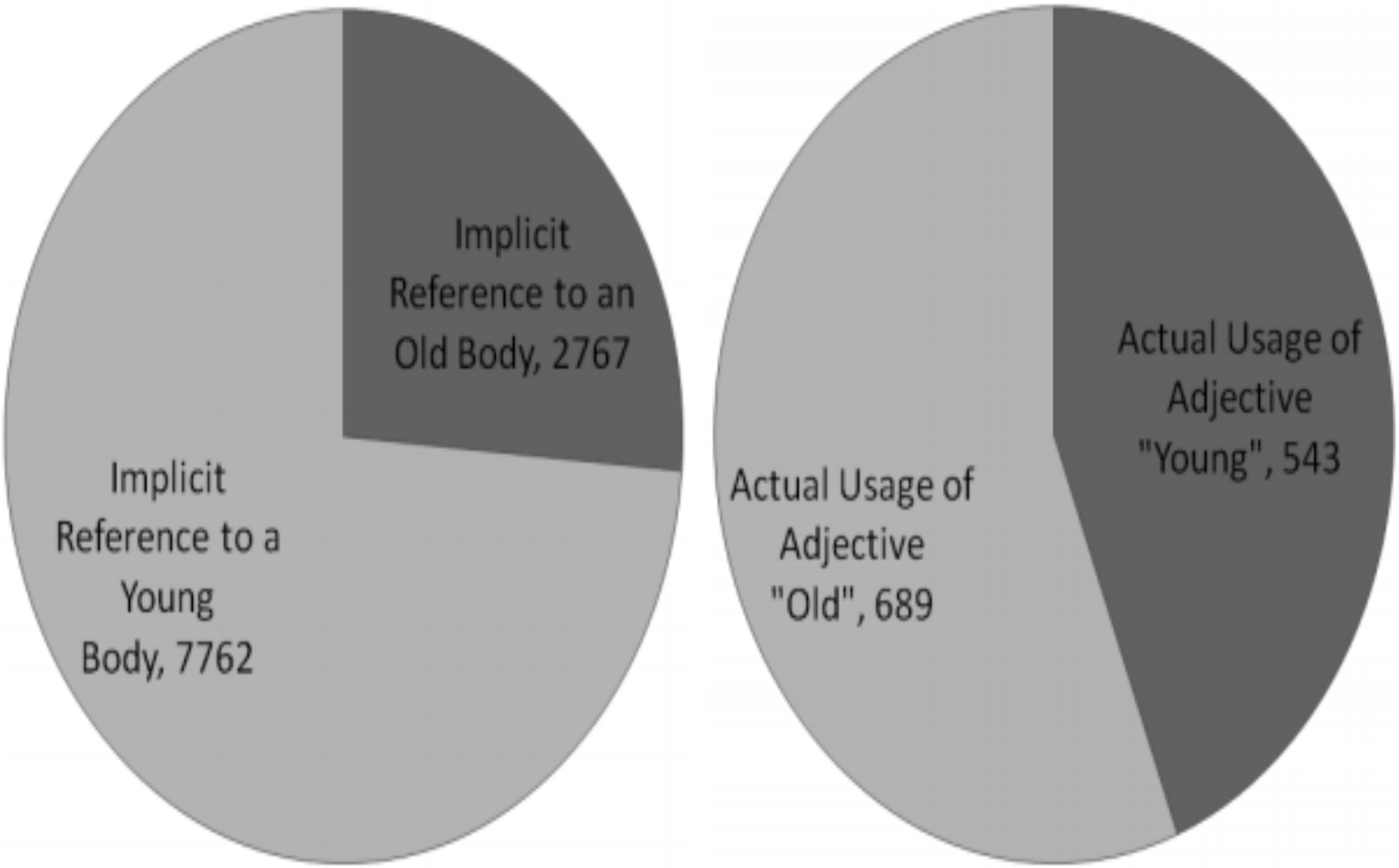
The Simpson Paradox

- Simpson's paradox for quantitative data: a positive trend (,) appears for two separate groups, whereas a negative trend ( ) appears when the groups are combined.Simpson's paradox, or the Yule–Simpson effect, is a phenomenon in probability and statistics, in which a trend appears in several different groups of data but disappears or reverses when these groups are combined. It is sometimes given the descriptive title of “reversal paradox” or ”amalgamation paradox”.

  -

  Wikipedia


Implicit Reference to an Old Body, 2767
Implicit Reference to a Young Body, 7762

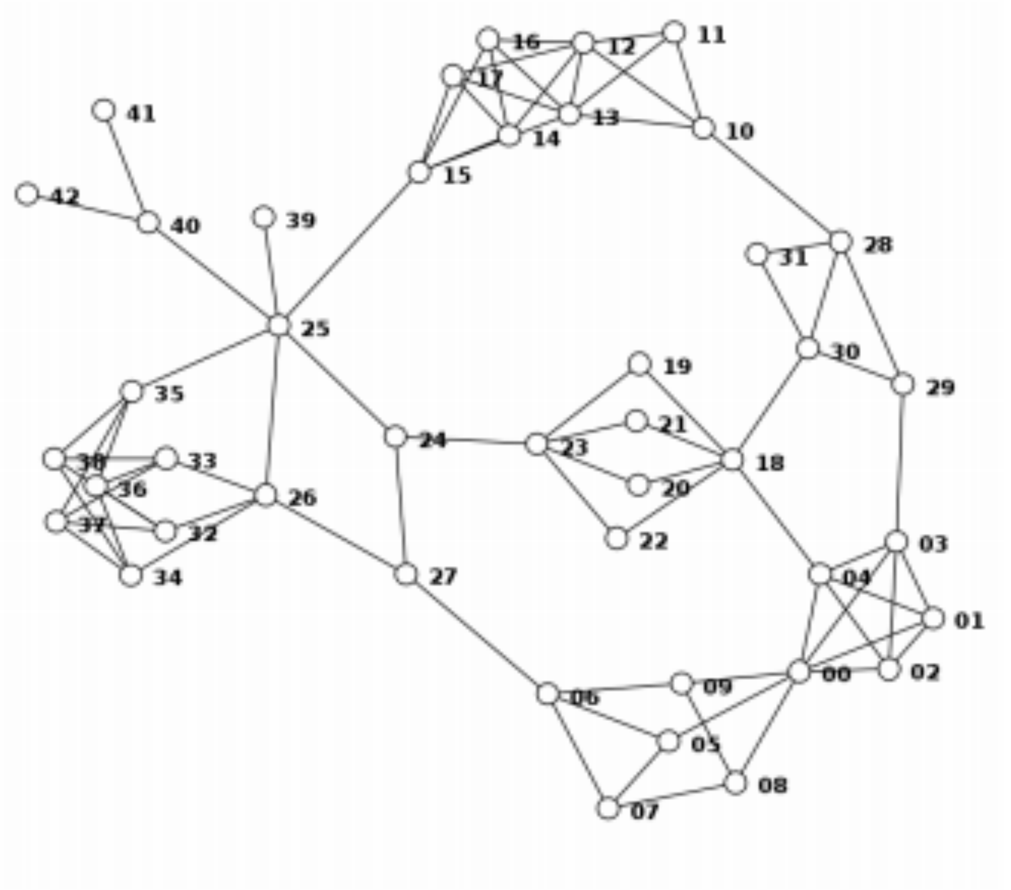

Implicit Reference to an Old Body, 2767
Implicit Reference to a Young Body, 7762
Actual Usage of Adjective "Young", 543
Actual Usage of Adjective "Old", 689

Data correlation

- How often are abstract nouns used in reference to each gender

- Word co-occurrence statics

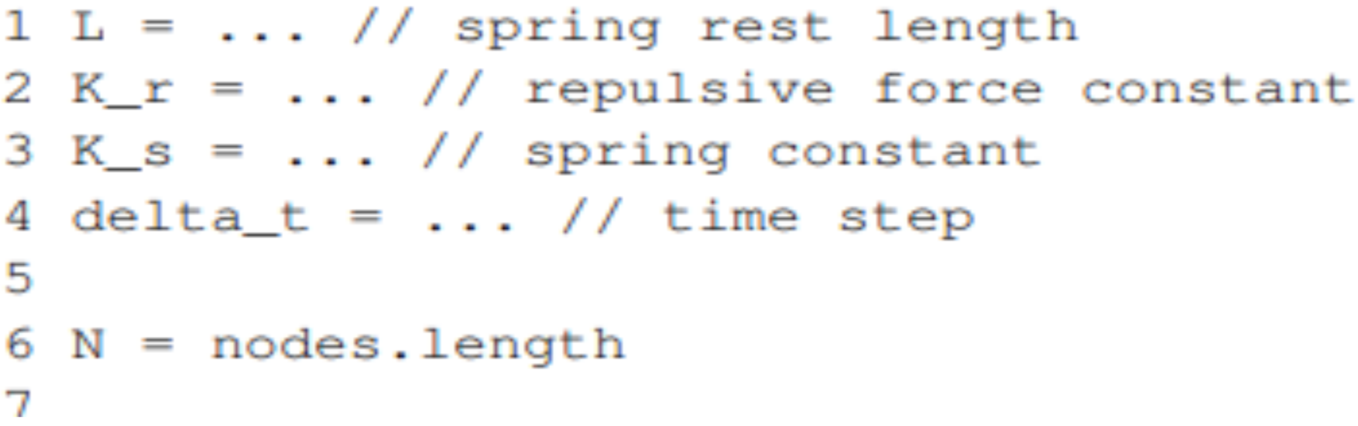
Networks visualization



Definitions

- Repulsive force $F_r = K_r/d^2$

- Spring force $F_s = K_s(d - L)$

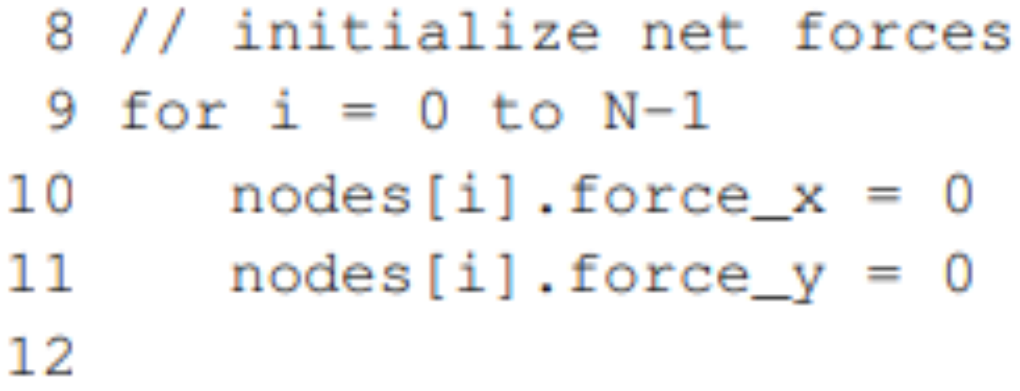


```
L = ... // spring rest length
K_r = ... // repulsive force constant
K_s = ... // spring constant
delta_t = ... // time step

N = nodes.length

```

- nodes – array of the nodes in the net
- We assume that every node has x and y position
- Every node has force_x and force_y

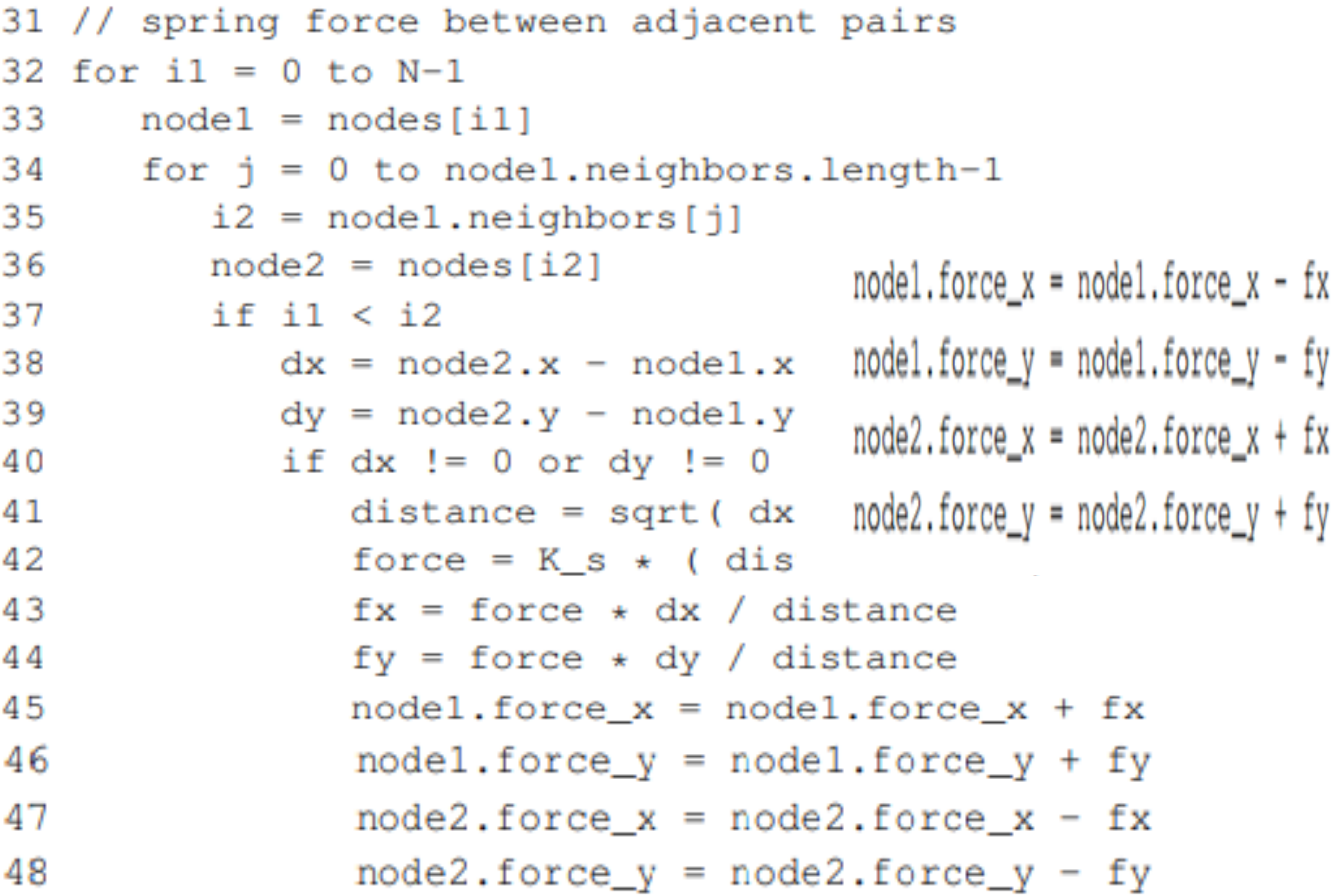


```
// initialize net forces
for i = 0 to N-1
    nodes[i].force_x = 0
    nodes[i].force_y = 0

```

```
// repulsion between all pairs
for i1 = 0 to N-2
   node1 = nodes[i1]
   for i2 = i1+1 to N-1
      node2 = nodes[i2]
      dx = node2.x - node1.x
      dy = node2.y - node1.y
      if dx != 0 or dy != 0
         distanceSquared = dx*dx + dy*dy
         distance = sqrt( distanceSquared )
         force = K_r / distanceSquared
         fx = force * dx / distance
         fy = force * dy / distance
         node1.force_x = node1.force_x - fx
         node1.force_y = node1.force_y - fy
         node2.force_x = node2.force_x + fx
         node2.force_y = node2.force_y + fy

```

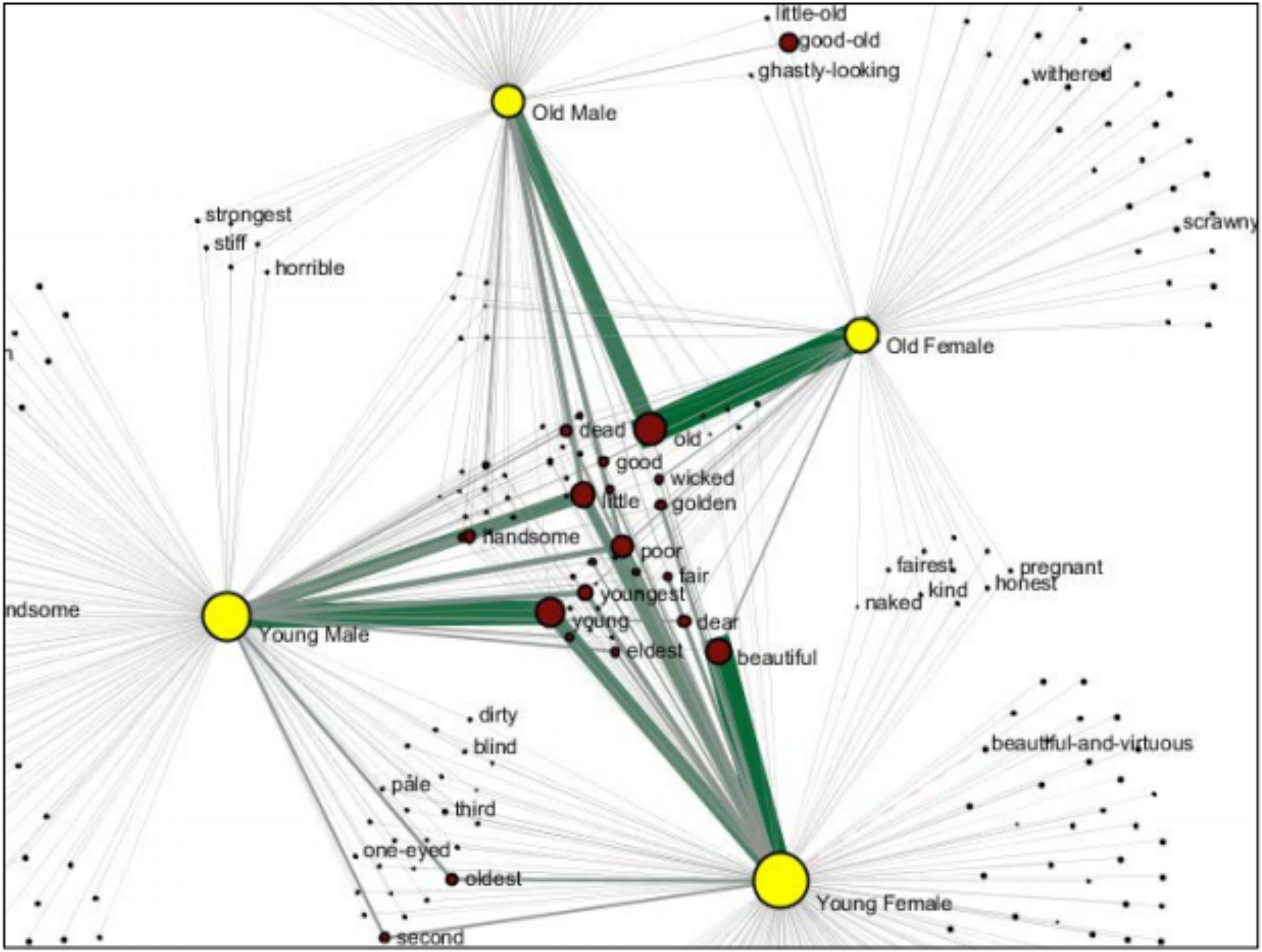


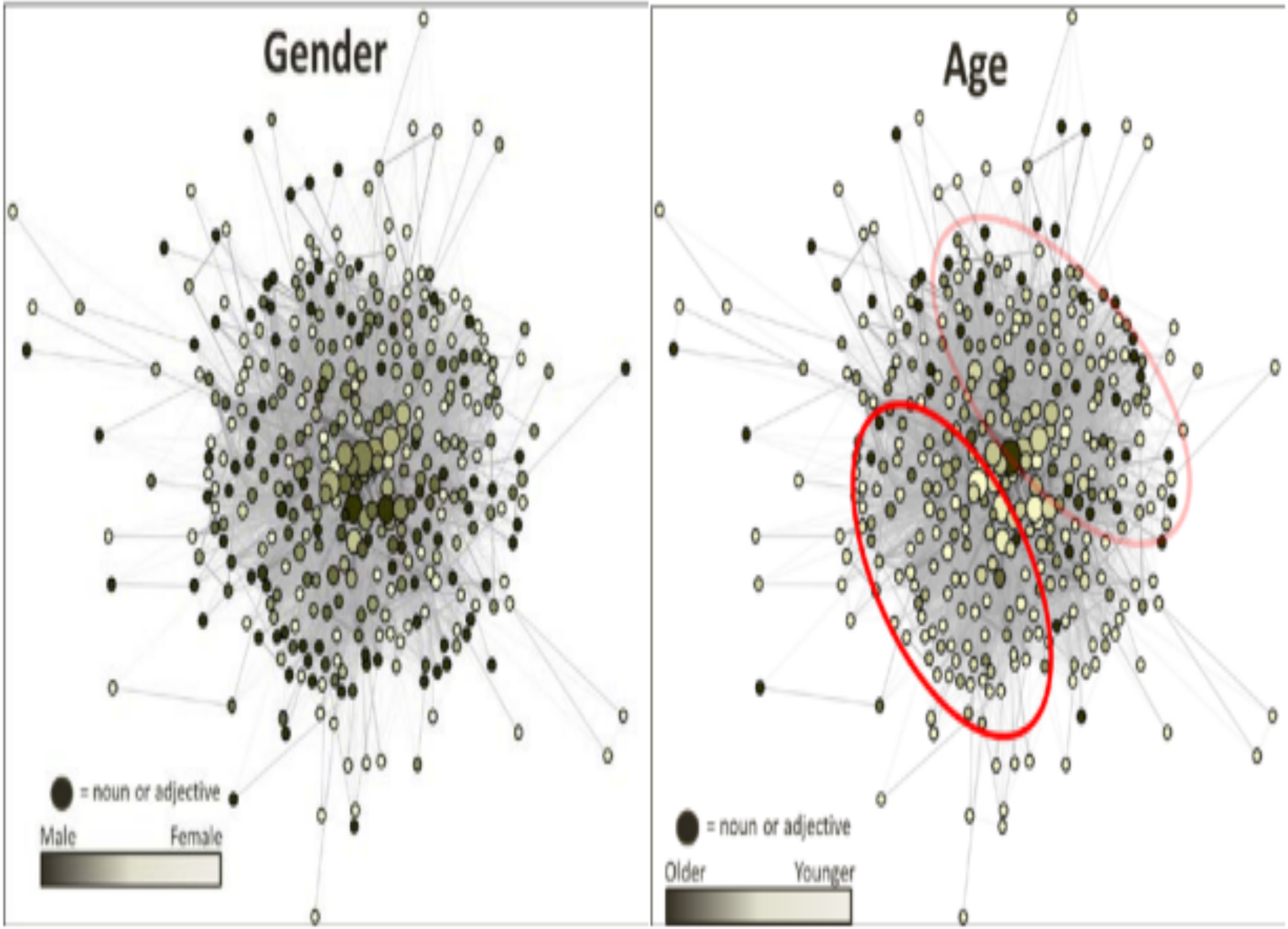
```
// spring force between adjacent pairs
for i1 = 0 to N-1
   node1 = nodes[i1]
   for j = 0 to node1.neighbors.length-1
      i2 = node1.neighbors[j]
      node2 = nodes[i2]
      if i1 < i2
         dx = node2.x - node1.x
         dy = node2.y - node1.y
         if dx != 0 or dy != 0
            distance = sqrt( dx
            force = K_s * ( dis
            fx = force * dx / distance
            fy = force * dy / distance
            node1.force_x = node1.force_x + fx
            node1.force_y = node1.force_y + fy
            node2.force_x = node2.force_x - fx
            node2.force_y = node2.force_y - fy
```

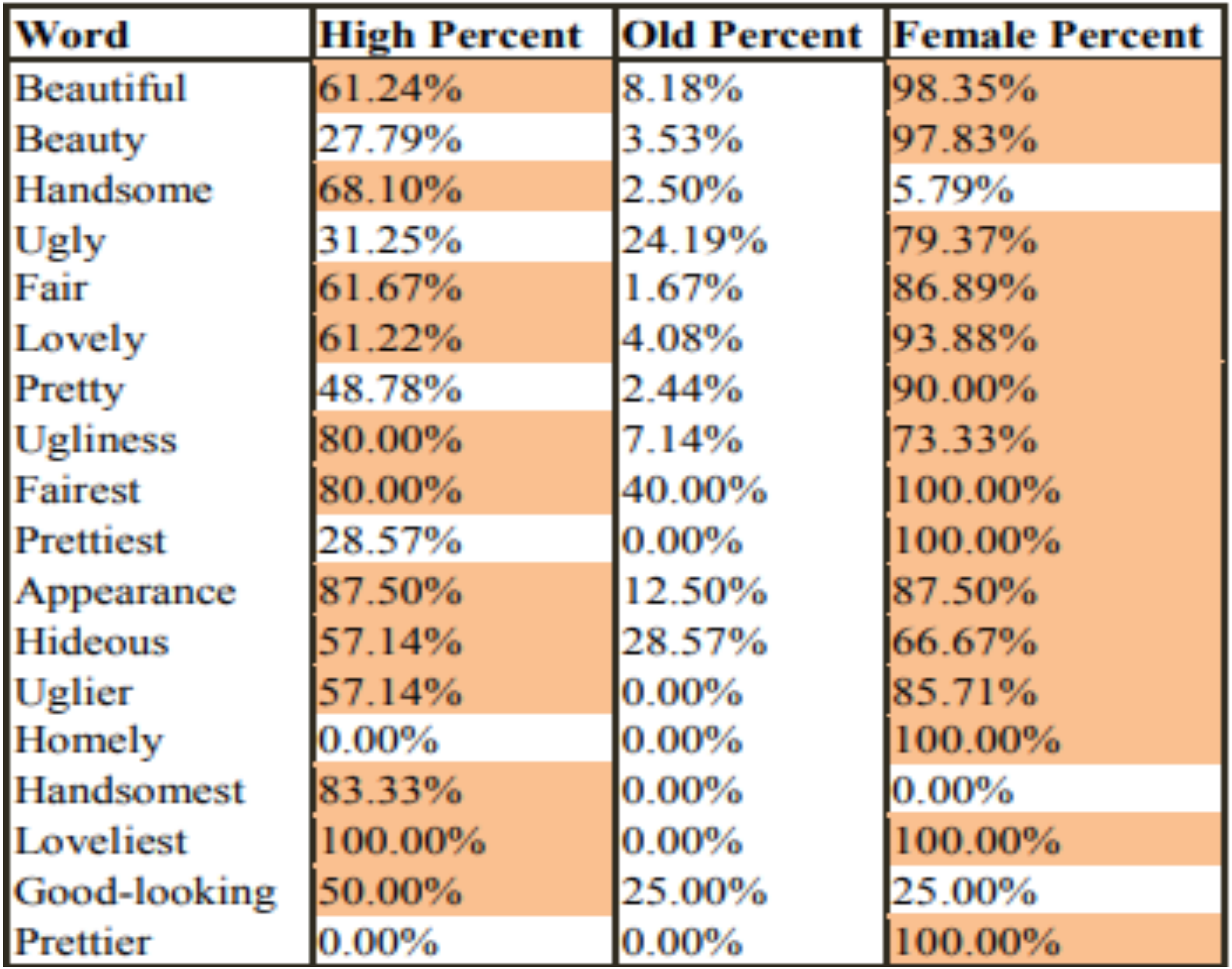
```
node1.force_x = node1.force_x - fx
node1.force_y = node1.force_y - fy
node2.force_x = node2.force_x + fx
node2.force_y = node2.force_y + fy
```

```
for i = 0 to N-1
    node = nodes[i]
    dx = delta_t * node.force_x
    dy = delta_t * node.force_y
    displacementSquared = dx*dx + dy*dy
    if ( displacementSquared
                 > MAX_DISPLACEMENT_SQUARED )
        s = sqrt( MAX_DISPLACEMENT_SQUARED
                 / displacementSquared )
        dx = dx * s
        dy = dy * s
    node.x = node.x + dx
    node.y = node.y + dy
```

Why do we need netwroks?


little-old
good-old
ghastly-looking
withered
Old Male
scrawny
strongest
stiff
horrible
Old Female
n
dead
old
good
wicked
little
golden
handsome
poor
fair
fairest
pregnant
kind
honest
naked
ndsome
youngest
Young Male
young
dear
eldest
beautiful
dirty
blind
beautiful-and-virtuous
pale
third
one-eyed
oldest
Young Female
second


Gender
● = noun or adjective
Male
Female
Age
● = noun or adjective
Older
Younger



Word	High Percent	Old Percent	Female Percent
Beautiful	61.24%	8.18%	98.35%
Beauty	27.79%	3.53%	97.83%
Handsome	68.10%	2.50%	5.79%
Ugly	31.25%	24.19%	79.37%
Fair	61.67%	1.67%	86.89%
Lovely	61.22%	4.08%	93.88%
Pretty	48.78%	2.44%	90.00%
Ugliness	80.00%	7.14%	73.33%
Fairest	80.00%	40.00%	100.00%
Prettiest	28.57%	0.00%	100.00%
Appearance	87.50%	12.50%	87.50%
Hideous	57.14%	28.57%	66.67%
Uglier	57.14%	0.00%	85.71%
Homely	0.00%	0.00%	100.00%
Handsomest	83.33%	0.00%	0.00%
Loveliest	100.00%	0.00%	100.00%
Good-looking	50.00%	25.00%	25.00%
Prettier	0.00%	0.00%	100.00%

Conclusions

- A small amount of adjectives makes up more than half of the total words used to describe body parts
- Male and females are generally described in similar ways
- Old and young bodies tend to be very polarized in their description