# word-BERT-rep VS def-BERT-rep (I)

## Objective

Explore similarities between BERT-vector representatin ("vector representation" abbreviated as VR) of a word and BERT-vector representation for the word definition. We use average and sum of BERT-VR as baseline for a later comparison with another composition functions.

## Exploring data

We treat with two dataset:

- the first contains (as rows, where name-row is one word) the first 5 words (top 5) with definitions similariest to
- the second have the similarities in before dataset

```
head(data$ranking)
```

```
##             first second  third    fourth     fifth
## love            1  death credit discovery      fuck
## tiger     hundred  seven   five   exhibit    record
## book      hundred  seven   five    record    school
## computer    seven   five viewer   hundred fertility
## plane     hundred  seven   five   exhibit    record
## train     hundred  seven   five   rooster   exhibit
```

```
dim(data$ranking)
```

```
## [1] 202   5
```

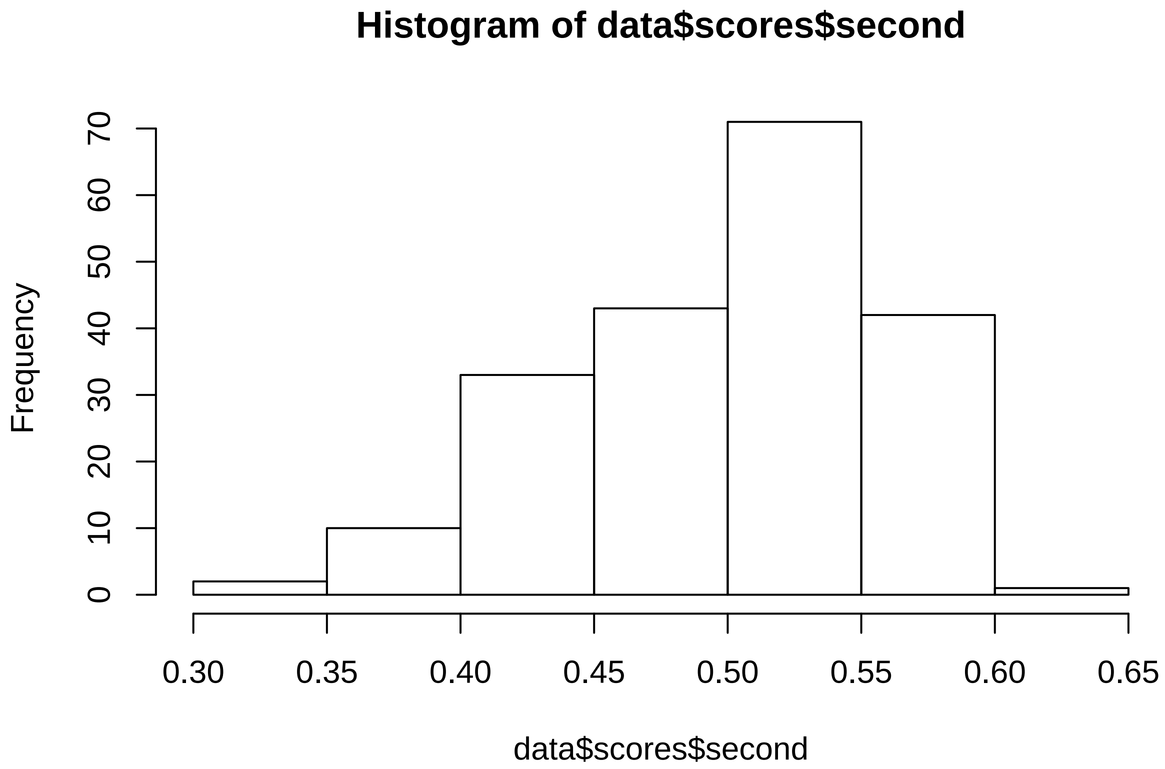
```
head(data$scores)
```

```
##               first    second     third    fourth     fifth
## love      0.4715205 0.4648822 0.4648412 0.4621409 0.4612962
## tiger     0.5680336 0.5422061 0.5059615 0.4415017 0.4293665
## book      0.5449572 0.5376849 0.5355507 0.5160842 0.5083381
## computer  0.5275755 0.5247760 0.5021460 0.4875561 0.4855717
## plane     0.5822837 0.5603127 0.5447720 0.4809039 0.4573219
## train     0.4515287 0.4304824 0.4210125 0.3784637 0.3531368
```

```
dim(data$scores)
```

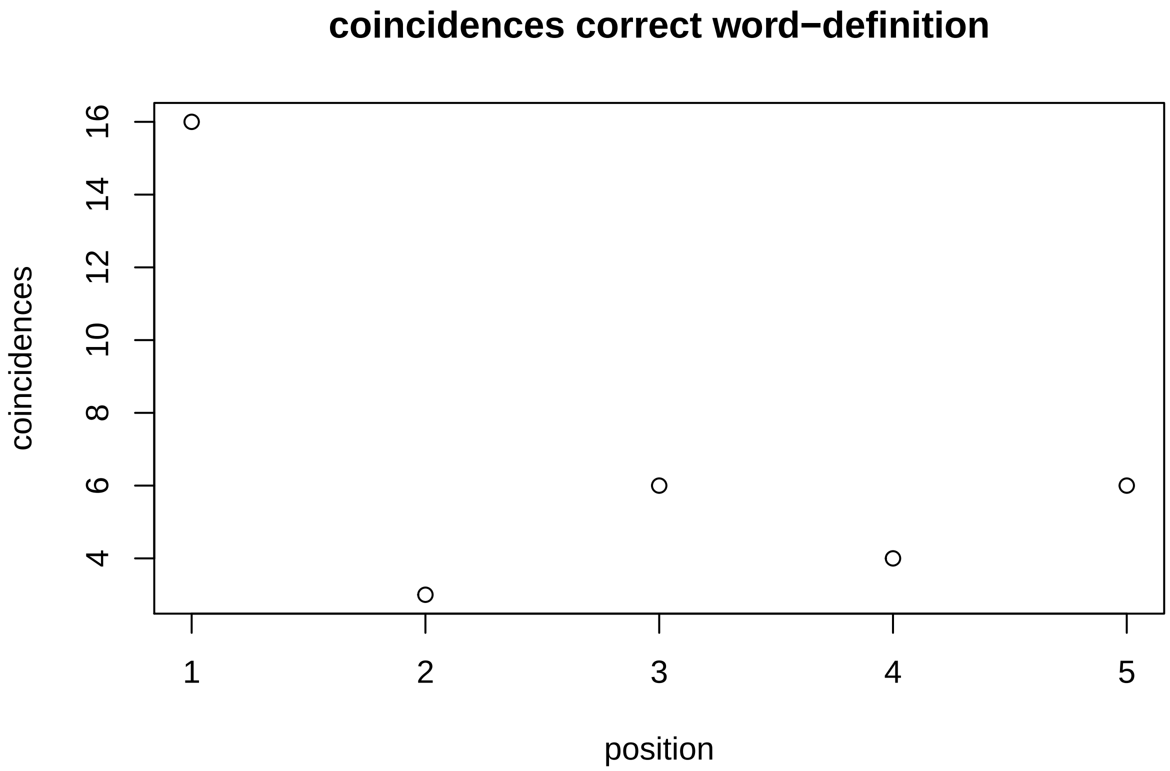
```
## [1] 202   5
```

We review the distribution for each position in the ranking



In the five position we can observe the same curve profile (peak in the right). It looks like similarities have a trend to highest similarity values.

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that gen-



erated the plot.

Not so much definitions corresponding with word.

We review the words that we have with more and less similarity (> or < than median in before histogram)

```
##    first second third fourth fifth
## 1:     0      0     0      0     1
## 2:     1      1     1      0     0
## 3:     1      1     1      1     1
## 4:     1      1     1      1     1
## 5:     1      1     1      1     1
## 6:     0      0     0      0     0
```

and we can observe that too much times the five first positions are in the same range (high or low)

```
#### we compute the number of words with top-5 definitions > and < than similarity median
freq_h <- which(c(apply(data$median, 1, sum) == 5))
freq_l <- which(c(apply(data$median, 1, sum) == 0))

dim(data$median)
```

```
## [1] 202   5
```

```
#### percentages of words with top-5 similariest definitions more (or no more) than median
dim(data$median[freq_h, ])[1]/dim(data$median)[1]
```

```
## [1] 0.3316832
```

```
dim(data$median[freq_l, ])[1]/dim(data$median)[1]
```

```
## [1] 0.3465347
```

trend to most (or no most) similarity with ANY other definition.