Experiments W2V VS BERT

```
## Warning: package 'data.table' was built under R version 3.4.4

## Warning: package 'ggplot2' was built under R version 3.4.4

## Warning in py_to_r.pandas.core.frame.DataFrame(result): index contains
## duplicated values: row names not set

## Warning in py_to_r.pandas.core.frame.DataFrame(result): index contains
## duplicated values: row names not set
```

test 1/2: W2V vs BERT - word to word (context)

DATA

dataset with mannual annotation of similaraties between pairs of words (wordsim353);

```
head(data_sim)
```

```
##          w1       w2 score
## 1:     love      sex  6.77
## 2:    tiger      cat  7.35
## 3:    tiger    tiger 10.00
## 4:     book    paper  7.46
## 5: computer keyboard  7.62
## 6: computer internet  7.58
```

and dataset with vector representations (w2v) of the set of words appeared (we show only the first two dimensions);

```
head(data_w2v_words[ , c(review_cols, "dim_1", "dim_2"), with = FALSE])
```

```
##    id     w id_token token       dim_1       dim_2
## 1:  1  love        1     i -0.22558594 -0.01953125
## 2:  1  love        2  love  0.10302734 -0.15234375
## 3:  1  love        3   you  0.20410156  0.01318359
## 4:  2 tiger        1     i -0.22558594 -0.01953125
## 5:  2 tiger        2   saw  0.09423828  0.20117188
## 6:  2 tiger        3 tiger -0.06835938  0.18261719
```

we select rows with equal w and token

NOTE: w2v is context-free but we could test it.

Also, we have our BERT representations (free-context) in this way,

```
head(data_bert_words[ , c(review_cols, "dim_w_1", "dim_w_2"), with = FALSE])
```

```
##    id        w id_token    token     dim_w_1     dim_w_2
## 1:  1     love        1     love  0.38649017  0.36187920
## 2:  2    tiger        1    tiger -0.30712840 -0.31644982
## 3:  3     book        1     book  0.41267234 -0.00370523
## 4:  4 computer        1 computer -0.67297429 -0.10599531
## 5:  5    plane        1    plane  0.01826328 -0.21225268
## 6:  6    train        1    train -0.52878708 -1.14276505

## [1] 428
```

```
## [1] 428
```

Firstly, we remove words that we haven't in both datasets (4 words are lost in w2v dataset because are out-of-vocabulary). Now, we can compute (cosine) similarities between vector representations (w2v and BERT) pairs of words in "data_sim" and compare with the mannual scoring.

```
##          w1       w2 score w2v_cosine bert_cosine
## 1:     love      sex  6.77  0.2639377   0.6756448
## 2:    tiger      cat  7.35  0.5172962   0.7996021
## 3:    tiger    tiger 10.00  1.0000000   1.0000000
## 4:     book    paper  7.46  0.3634626   0.5779281
## 5: computer keyboard  7.62  0.3963916   0.8194257
## 6: computer internet  7.58  0.4068623   0.5341467
```

With this dataset we can compare mannual similarity with cosine metric for w2v and BERT representations.

ANALYSIS

We clean NA puntuations, and we get 328 complete rows.

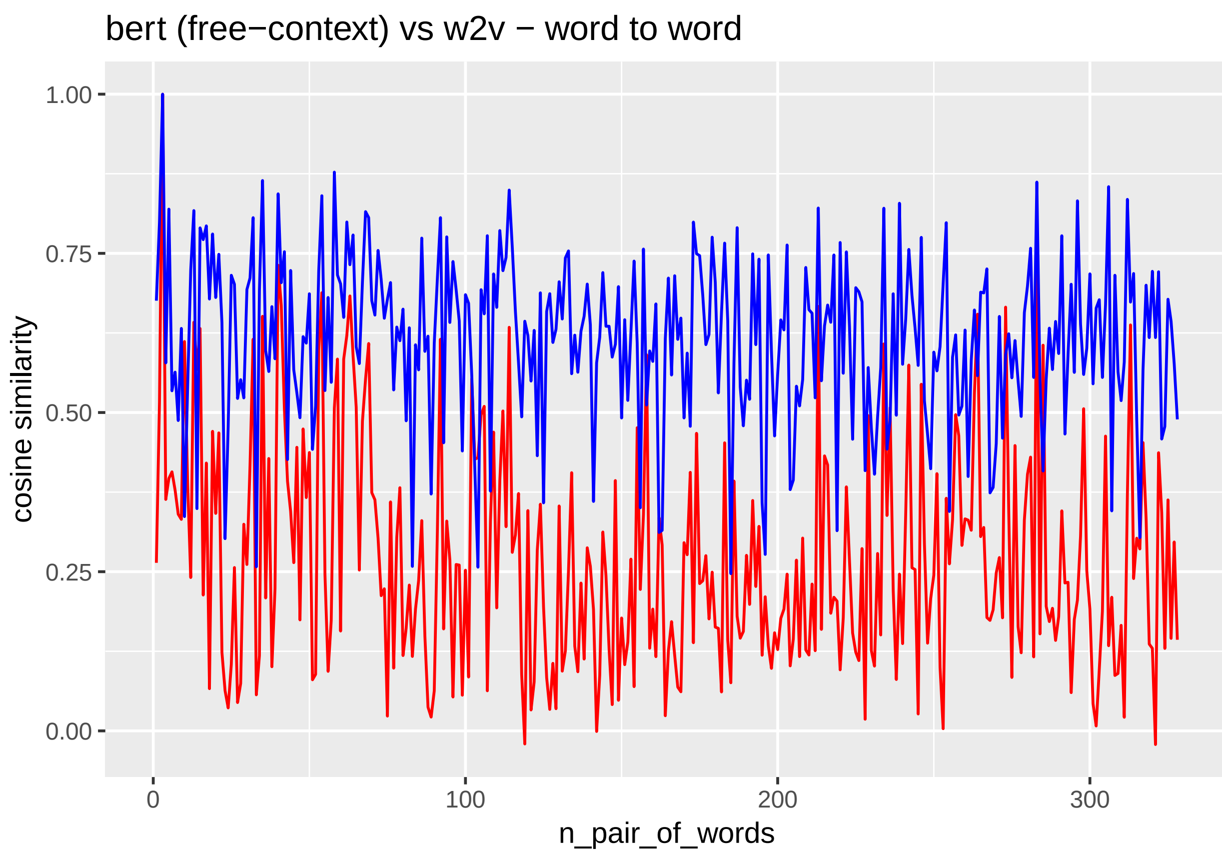
NOTE: we observe 3 values of cosine (w2v) < 0

```
##              w1        w2 score    w2v_cosine bert_cosine
## 1:    precedent cognition  2.81 -0.0205696013   0.6433847
## 2:         mars     water  2.94 -0.0009222099   0.5781125
## 3: preservation     world  6.19 -0.0214706240   0.6177243
```

```
## Empty data.table (0 rows) of 5 cols: w1,w2,score,w2v_cosine,bert_cosine
```

We can plot similarities scores,

```
ggplot() + geom_line(aes(x = seq(1:length(data_sim$w1)), y = data_sim$w2v_cosine),color='red') +
           geom_line(aes(x = seq(1:length(data_sim$w1)), y = data_sim$bert_cosine),color='blue') +
           ylab('cosine similarity') + xlab('n_pair_of_words') + ggtitle("bert (free-context) vs w2v - w
```



We can observe the best results for BERT representations.

Also we can compute the Pearson coefficient in both case respect to the mannual scores in dataset. For w2vec similarities,

```
cor(data_sim$score, data_sim$w2v_cosine, method = c("pearson"))
```

```
## [1] 0.655213
```

and for BERT cosine similarities,

```
cor(data_sim$score, data_sim$bert_cosine, method = c("pearson"))
```

```
## [1] 0.2467143
```

We can observe highest similarity metric for BERT representations, but has a worst correlation with mannual scoring.

And corrlation between both vector representations scoring is,

```
cor(data_sim$w2v_cosine, data_sim$bert_cosine, method = c("pearson"))
```

```
## [1] 0.3178288
```

BERT non-free-context

Now, we can use the BERT vector representation of the same word got from word in a phrase (context).

In the next dataset we have a phrase containing the word and we have the vector representation got in this case,

```
## Warning in py_to_r.pandas.core.frame.DataFrame(result): index contains
```

```
## duplicated values: row names not set

##    id     w id_token token dim_context_1 dim_context_2
## 1:  1  love        1     i     0.2335791    0.24898028
## 2:  1  love        2  love     0.9315368    0.91580886
## 3:  1  love        3   you    -0.1724851   -0.77048999
## 4:  2 tiger        1     i     0.1126512   -0.53610945
## 5:  2 tiger        2   saw    -0.1356604   -0.20776120
## 6:  2 tiger        3     a    -0.2600941   -0.07189066
```

and we select the corresponding vector,

```
##    id        w dim_context_1 dim_context_2
## 1:  1     love     0.9315368     0.9158089
## 2:  2    tiger     0.3108854     0.1306733
## 3:  3     book     0.1590683    -0.4085730
## 4:  4 computer    -0.4819463     0.2318698
## 5:  5    plane     1.0386617    -0.6260117
## 6:  6    train     1.0346285    -0.8108625

##          w1       w2 score w2v_cosine bert_cosine bert_context_cosine
## 1:     love      sex  6.77  0.2639377   0.6756448           0.2499513
## 2:    tiger      cat  7.35  0.5172962   0.7996021           0.5943350
## 3:    tiger    tiger 10.00  1.0000000   1.0000000           1.0000000
## 4:     book    paper  7.46  0.3634626   0.5779281           0.6570200
## 5: computer keyboard  7.62  0.3963916   0.8194257           0.4377427
## 6: computer internet  7.58  0.4068623   0.5341467           0.4450932
```

in this case the before results are,

```r
cor(data_sim$score, data_sim$bert_context_cosine, method = c("pearson"))
```

```
## [1] 0.3516758
```

we can observe a bit improve respect to results with BERT free-context.

BERT VS W2V

```
##             w1            w2 score w2v_cosine bert_cosine
##  1:       love           sex  6.77 0.26393773   0.6756448
##  2:      tiger           cat  7.35 0.51729619   0.7996021
##  3:      tiger         tiger 10.00 1.00000000   1.0000000
##  4:       book         paper  7.46 0.36346261   0.5779281
##  5:   computer      keyboard  7.62 0.39639163   0.8194257
##  6:   computer      internet  7.58 0.40686231   0.5341467
##  7:      plane           car  5.77 0.37796983   0.5634068
##  8:      train           car  6.31 0.34025611   0.4874269
##  9:  telephone communication  7.50 0.33218451   0.6321077
## 10: television         radio  6.77 0.61149707   0.3365614
## 11:      media         radio  7.42 0.38991608   0.5381966
## 12:       drug         abuse  6.85 0.24085768   0.7239172
## 13:      bread        butter  6.19 0.64172602   0.8171788
## 14:   cucumber        potato  5.92 0.56785624   0.3490837
## 15:     doctor         nurse  7.00 0.63195230   0.7900290
## 16:  professor        doctor  6.62 0.21336083   0.7715104
## 17:    student     professor  6.81 0.42066182   0.7931704
## 18:      smart       student  4.62 0.06630216   0.6782536
## 19:      smart        stupid  5.81 0.47047193   0.7803929
```

```
## 20:    company         stock  7.08 0.34156865   0.6806516
## 21:      stock        market  8.08 0.46805560   0.7485273
## 22:      stock         phone  1.62 0.12326756   0.6411072
## 23:      stock            cd  1.31 0.06321469   0.3019016
## 24:      stock        jaguar  0.92 0.03606690   0.4773706
## 25:      stock           egg  1.81 0.10417768   0.7155421
## 26:  fertility           egg  6.69 0.25652347   0.7014832
## 27:      stock          live  3.73 0.04447177   0.5221197
## 28:      stock          life  0.92 0.07456468   0.5515331
## 29:       book       library  7.46 0.32453122   0.5229406
## 30:       bank         money  8.12 0.26132065   0.6932225
##            w1            w2 score w2v_cosine bert_cosine
##     bert_context_cosine
##  1:           0.2499513
##  2:           0.5943350
##  3:           1.0000000
##  4:           0.6570200
##  5:           0.4377427
##  6:           0.4450932
##  7:           0.3358731
##  8:           0.3567408
##  9:           0.4262896
## 10:           0.4234216
## 11:           0.4373691
## 12:           0.4857973
## 13:           0.7890271
## 14:           0.6133671
## 15:           0.5549122
## 16:           0.5273623
## 17:           0.2166793
## 18:           0.2158561
## 19:           0.3837771
## 20:           0.3215411
## 21:           0.3491502
## 22:           0.2345345
## 23:           0.3275711
## 24:           0.1591660
## 25:           0.3019592
## 26:           0.3869548
## 27:           0.2509690
## 28:           0.3395007
## 29:           0.4023845
## 30:           0.4221412
##     bert_context_cosine
```

```
## [1] 0.9695122
```

In the 96 % of rows BERT win to W2V

```
sum(ifelse(data_sim$bert_context_cosine > data_sim$w2v_cosine, 1, 0))/length(data_sim$bert_context_cosi
```

```
## [1] 0.7621951
```

WARNING!!! BERT free-context better than BERT with context???

We review cases with BERT with context better than BERT-free-context,

```r
data_sim[bert_cosine < bert_context_cosine]
```

```
##               w1           w2 score w2v_cosine bert_cosine
##  1:         book        paper  7.46 0.36346261   0.5779281
##  2:   television        radio  6.77 0.61149707   0.3365614
##  3:     cucumber       potato  5.92 0.56785624   0.3490837
##  4:        stock           cd  1.31 0.06321469   0.3019016
##  5:       tennis       racket  7.56 0.39200801   0.4261302
##  6:        space    chemistry  4.88 0.08025177   0.4420664
##  7:          car   automobile  8.94 0.58383676   0.6493923
##  8:     magician       wizard  9.02 0.48634962   0.7047806
##  9:        shore     woodland  3.08 0.11690946   0.2583192
## 10:        tiger       jaguar  8.00 0.55286842   0.5598985
## 11:        tiger       feline  8.00 0.42671448   0.4095752
## 12:        tiger    carnivore  7.08 0.42893752   0.2571064
## 13:        tiger        fauna  5.62 0.32975670   0.3765793
## 14:          cup    tableware  6.85 0.19486345   0.3581415
## 15:      century         year  7.59 0.33483712   0.3116927
## 16:      century       nation  3.16 0.29191471   0.3149015
## 17:       reason hypertension  2.31 0.07555150   0.2469450
## 18:         opec      country  5.63 0.21063469   0.2771165
## 19: impartiality     interest  5.16 0.20375371   0.3143537
## 20:     currency       market  7.50 0.33829964   0.4423884
## 21:         game       series  6.19 0.29130784   0.5099704
## 22:        seven       series  3.56 0.33100622   0.3994922
## 23:      seafood      lobster  8.70 0.65440797   0.5578444
## 24: championship   tournament  8.36 0.66553167   0.5906379
## 25:       summer       nature  5.63 0.12260760   0.4938555
## 26:       murder manslaughter  8.53 0.60576504   0.4081859
##               w1           w2 score w2v_cosine bert_cosine
##     bert_context_cosine
##  1:           0.6570200
##  2:           0.4234216
##  3:           0.6133671
##  4:           0.3275711
##  5:           0.4847074
##  6:           0.4465372
##  7:           0.7566668
##  8:           0.7616620
##  9:           0.7178296
## 10:           0.5718994
## 11:           0.4877220
## 12:           0.3671080
## 13:           0.4179296
## 14:           0.5045422
## 15:           0.4660226
## 16:           0.3539015
## 17:           0.2882232
## 18:           0.2797994
## 19:           0.3716148
## 20:           0.5489230
## 21:           0.5110280
## 22:           0.5542144
## 23:           0.6545407
```

```
## 24:           0.6113276
## 25:           0.5786855
## 26:           0.4892358
##     bert_context_cosine
```

We review the context (the length, in example) that we use with this words,

```r
words <- unique(data_sim[bert_cosine < bert_context_cosine]$w1, data_sim[bert_cosine < bert_context_cos
words
```

```
##  [1] "book"         "television"   "cucumber"     "stock"
##  [5] "tennis"       "space"        "car"          "magician"
##  [9] "shore"        "tiger"        "cup"          "century"
## [13] "reason"       "opec"         "impartiality" "currency"
## [17] "game"         "seven"        "seafood"      "championship"
## [21] "summer"       "murder"
```

```r
data_bert_words_context <- setDT(read_pickle_file(FILE_PKL_TO_READ_BERT_CONTEXT))
```

```
## Warning in py_to_r.pandas.core.frame.DataFrame(result): index contains
## duplicated values: row names not set
```

```r
data_bert_words_context <- data_bert_words_context[w %in% words]
table(data_bert_words_context[ , .(n = max(.SD$id_token)), by = .(w)]$n)
```

```
##
##  3  5  6  7  8  9 10 12
##  1  3  7  3  5  1  1  1
```

distribution with median (and more concentrated) between 6 - 8 words for context length. For all context used in tdataset, the length distribution is,

```r
data_bert_words_context <- setDT(read_pickle_file(FILE_PKL_TO_READ_BERT_CONTEXT))
```

```
## Warning in py_to_r.pandas.core.frame.DataFrame(result): index contains
## duplicated values: row names not set
```

```r
data_bert_words_context <- data_bert_words_context[ , .(n = max(.SD$id_token)), by = .(w)]$n
table(data_bert_words_context)
```

```
## data_bert_words_context
##  3  4  5  6  7  8  9 10 11 12 13 14 15 16 18 19 20 23
##  5 11 41 67 65 74 60 42 15 21 11  5  4  2  2  1  1  1
```

```r
median(data_bert_words_context)
```

```
## [1] 8
```

It looks like words with bert-with-context vector representation is not too much associated to its context length.

WARNING!!! Neither assocciated to its context length????