Aufgabe 1

a) Why do randomly initialized, untrained sentence embeddings produce usable results?

The randomly initialized sentence embeddings are used as features for a collection of downstream tasks.

Sentence representations are randomly computed from pre-trained word embeddings

b) How does the Bag of random embedding projections (BOREP) approach create a sentence representation?

First, should a matrix W, where D is the dimension of the projection and d is the dimension of input word embedding, randomly initialize and sampled uniformly.

Second, the sentence representation is obtained through pooling function.

Aufgabe 2

What are the effects of the four parameters size, window, negative, and cbow?

-size <int>

Set size of word vectors; default is 100

-window <int>

Set max skip length between words; default is 5

-negative <int>

Number of negative examples; default is 5, common values are 3 - 10 (0 = not used)

-cbow <int>

Use the continuous bag of words model; default is 1 (use 0 for skip-gram model)

use the distance tool in the word2vec directory and compute:

man: humble, handsome, honest, boy, woman

woman: girl, herself, servant, stranger, pregnant

Aufgabe 3.1

3) Which dimensionality does the average word embedding have? Which dimensionality does the concatenated power mean word embedding have?

len(reviews)*300

len(reviews)*1200

4) Which problem arises when you want to use a power mean with negative parameter p such as the harmonic mean?

$$M_p(x_1, \ldots, x_n) = \left(\frac{1}{n}\sum_i x_i^p\right)^{1/p}$$

If use with negative parameter p, OOC Vector (Zeros) could cause zero-cross problem.

averaged word2vec embedding vector

```
averaged word2vec embedding vector of the first review from the training data set:
[ 0.003906 -0.072474  0.001797 -0.006953 -0.032279 -0.109269 -0.094438
  0.138351  0.010301  0.076515 -0.024344  0.042118 -0.191492 -0.070353
 -0.00983  -0.018117  0.075178  0.019973 -0.03481  -0.062609 -0.12871
  0.037849  0.063594 -0.121558 -0.014127 -0.049616  0.065996  0.133311
 -0.012356  0.005654  0.005724  0.070768  0.085456  0.009663  0.033707
  0.050417 -0.008998 -0.010588  0.110542 -0.052137 -0.013071  0.036003
  0.044056 -0.010821  0.096768 -0.047483 -0.056048  0.016153 -0.055568
 -0.105606 -0.041787  0.020619 -0.020686 -0.053484 -0.073474  0.075004
  0.125047  0.044731  0.003082  0.012419 -0.055997  0.029226 -0.05676
  0.00442  -0.004629  0.104066  0.039893 -0.055487 -0.010656  0.009034
 -0.005925  0.008736  0.007253 -0.088455 -0.064267 -0.025828 -0.086053
  0.147226 -0.066218  0.014482  0.045523 -0.065965  0.09209   0.028901
 -0.022437 -0.018044 -0.004565  0.004845  0.126719  0.022253  0.052865
  0.038622 -0.057582 -0.09721   0.026425  0.063433  0.073158 -0.037558
  0.026572 -0.002013  0.009438  0.096973  0.017496 -0.005421  0.055286
  0.080214 -0.026933  0.014701 -0.035514 -0.032755  0.10567   0.001405
 -0.009928 -0.02501   0.063327  0.1747    0.066264  0.056293  0.050931
  0.001259  0.098566  0.065127  0.017255 -0.070794  0.012565 -0.136988
  0.011489 -0.163693  0.241806  0.014834 -0.021118  0.095115 -0.04436
  0.087821  0.084873 -0.003366  0.001558 -0.004673 -0.019684  0.034017
 -0.038022  0.163994  0.101018 -0.150262 -0.095108 -0.01596  -0.037878
  0.221835 -0.110626 -0.155293 -0.132966  0.038021  0.02601   0.151204
 -0.032189 -0.024151 -0.063164 -0.079911 -0.056237  0.035596  0.066275
 -0.070065  0.031405  0.109534 -0.045883 -0.035996 -0.009307 -0.101369
  0.034511 -0.099051 -0.034848  0.070485 -0.081997 -0.091923 -0.042106
  0.035985 -0.122598 -0.150472  0.168003 -0.130483  0.060355 -0.062201
 -0.054327 -0.136775 -0.107153  0.077523  0.136866  0.1023   -0.066081
  0.080868 -0.044252  0.01951   0.024055  0.019126 -0.025441 -0.074265
  0.065041  0.042743  0.021426  0.037506  0.115686 -0.111431  0.096284
 -0.123615 -0.002214 -0.10838   0.097789 -0.104174  0.112719 -0.003969
  0.058453 -0.084527 -0.116462  0.049964 -0.096054  0.018555  0.041925
  0.094227  0.085466  0.058878 -0.004535 -0.0295    0.135481 -0.003319
 -0.030251  0.024106 -0.109855 -0.005802 -0.102658 -0.00947  -0.036341
 -0.134268  0.022443  0.117643 -0.041114  0.006397  0.083006  0.051482
  0.015505 -0.081057  0.06121   0.063507  0.063881  0.030273  0.048589
  0.133985  0.103512 -0.070621  0.05483  -0.083062 -0.056558  0.102758
  0.08154  -0.054566 -0.131676 -0.017691  0.199143 -0.144693 -0.074197
 -0.035089 -0.014274 -0.061331 -0.042936 -0.037995  0.105598  0.0401
 -0.004298  0.061012  0.047103 -0.006754  0.045459 -0.034147 -0.007187
  0.063194  0.021067 -0.175242 -0.087055  0.207827  0.007727  0.052547
 -0.034999 -0.009544  0.082221 -0.04902  -0.104674  0.029052  0.050228
 -0.043871  0.18602  -0.017957 -0.167837 -0.213638 -0.054337  0.045439
  0.151816 -0.024018 -0.029653  0.050565  0.125748 -0.17679 ]
```

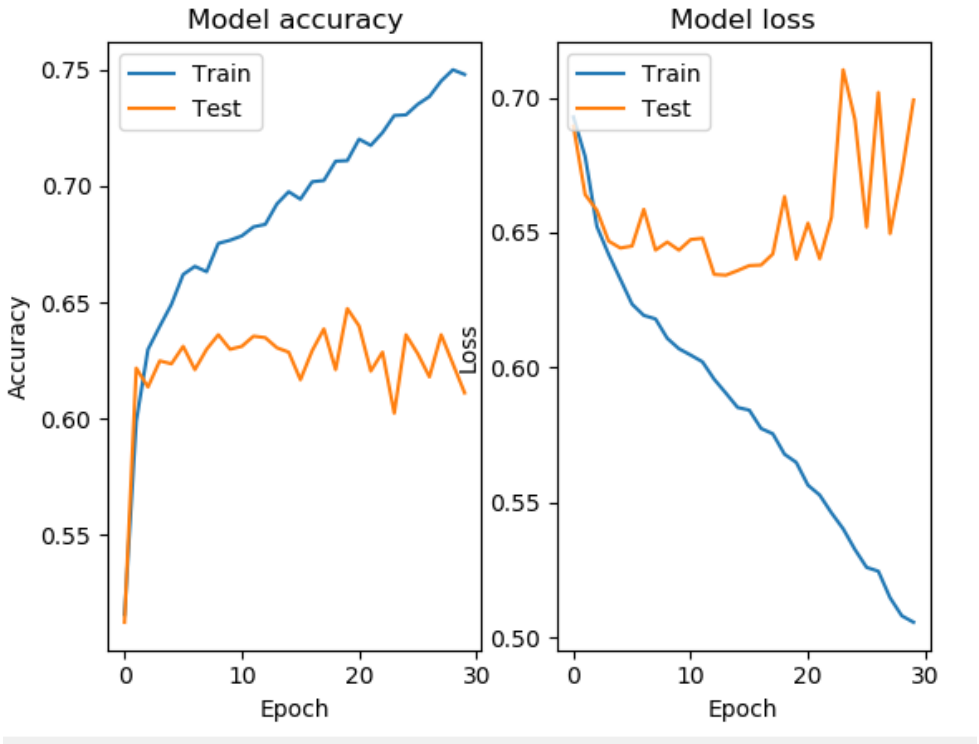
the power mean and the averaged word2vec embedding vector

```
the power mean word2vec embedding vector of the first review from the training data set:
[0.045344 0.034206 0.010218 0.014786 0.02274  0.03414  0.026953 0.041873
 0.012394 0.025487 0.015923 0.018215 0.062605 0.02308  0.016846 0.015206
 0.033897 0.019786 0.013645 0.029949 0.053019 0.019556 0.021799 0.030788
 0.018963 0.011557 0.026203 0.06182  0.01767  0.024847 0.014364 0.025675
 0.038327 0.011648 0.020181 0.026026 0.01607  0.014874 0.050689 0.026907
 0.005578 0.013335 0.023878 0.014563 0.024535 0.032258 0.020893 0.013343
 0.018033 0.022715 0.026563 0.040778 0.023277 0.031341 0.023802 0.030741
 0.052133 0.012182 0.007409 0.009524 0.027358 0.02285  0.024207 0.015832
 0.020383 0.02578  0.018496 0.021549 0.012724 0.029989 0.014337 0.015706
 0.011454 0.043486 0.014686 0.018776 0.022346 0.060037 0.01791  0.02008
 0.023777 0.023367 0.027754 0.016353 0.01814  0.011911 0.014333 0.015146
 0.031492 0.011122 0.026206 0.009009 0.018336 0.035194 0.019174 0.025956
 0.028896 0.029223 0.02701  0.008108 0.0224   0.020537 0.03335  0.012585
 0.030095 0.019546 0.032916 0.027044 0.016806 0.010619 0.031736 0.011352
 0.018336 0.019595 0.022149 0.053046 0.013498 0.013018 0.018697 0.022578
 0.033119 0.029052 0.017465 0.026198 0.014683 0.041207 0.022841 0.046051
 0.089749 0.02279  0.011951 0.028422 0.038987 0.028019 0.021754 0.022836
 0.023922 0.011458 0.014462 0.02449  0.018282 0.061231 0.027615 0.048695
 0.037127 0.009073 0.014037 0.076544 0.03829  0.061676 0.045714 0.020233
 0.01686  0.042487 0.026201 0.009763 0.016069 0.019658 0.030457 0.018934
 0.020597 0.032681 0.023198 0.041248 0.016576 0.015809 0.021037 0.02074
 0.011211 0.018019 0.009368 0.029079 0.019171 0.042116 0.018889 0.01918
 0.029719 0.051824 0.047745 0.063548 0.014339 0.031007 0.022071 0.052023
 0.030872 0.021723 0.035086 0.033732 0.027024 0.032869 0.012897 0.013518
 0.022247 0.027676 0.0221   0.040957 0.013594 0.020997 0.017019 0.017521
 0.032528 0.027413 0.023099 0.046521 0.007407 0.02194  0.029574 0.03759
 0.048937 0.011161 0.022673 0.03131  0.033014 0.019995 0.038404 0.028418
 0.01357  0.026718 0.029845 0.024618 0.01974  0.030728 0.045691 0.016977
 0.008704 0.022074 0.031227 0.014107 0.034775 0.02063  0.020434 0.038306
 0.012809 0.032872 0.020536 0.012937 0.022301 0.028341 0.006897 0.042003
 0.018392 0.024856 0.020283 0.012314 0.017801 0.04054  0.022324 0.022193
 0.013529 0.020197 0.02262  0.030266 0.016896 0.013402 0.040746 0.008107
 0.065022 0.036898 0.027278 0.023057 0.013579 0.023499 0.024158 0.023808
 0.042311 0.024352 0.016954 0.025645 0.054377 0.017125 0.012084 0.011848
 0.008983 0.020861 0.013155 0.048703 0.02353  0.067809 0.004629 0.01886
 0.026335 0.02653  0.024709 0.030558 0.025322 0.01561  0.020734 0.00924
 0.057964 0.025623 0.045566 0.079103 0.014476 0.017364 0.048573 0.026391
 0.009945 0.01376  0.028913 0.0655  ]
```

Aufgabe 3.2

```
Colocations handled automatically by placer.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 300)               360300
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 602
=================================================================
Total params: 360,902
Trainable params: 360,902
Non-trainable params: 0
_________________________________________________________________
```

```
test cross-entropy: 0.6243273927466134
test accuracy: 0.6616635393395582
```



Result from 3. Homework

```
loss : 0.3643300420161409
accuracy : 0.6929330827296712
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 80)                8080
_________________________________________________________________
dense_2 (Dense)              (None, 50)                4050
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 102
=================================================================
Total params: 12,232
Trainable params: 12,232
Non-trainable params: 0
```

Network from 3. Homework has 3 Layers, which mean the more complexity in model. It could get better result. Moreover, the activation function in 3. Homework is ReLu, which has a huge acceleration effect on the convergence of stochastic gradient descent.