# Learning from Data - Week 3

Assignment 3

Mart Busger op Vollenbroek
S2174634

## Assignment 3.1

### 3.1.1

*Russia*: When entered “Russia” in the similarity tool, the first hits were Ukraine, Moscow, Russian, Belarus and Kremlin. I expected that Putin would have had a higher similarity score, but with 0.663687 it scores rather poorly against the highest scoring word (Ukraine with 0.791829). Ukraine of course is easily explained because of the current political situation wherein Russia has an interest.

*Apple*: When I entered “Apple” I expected to find words relating to the company, but also to words relating to the fruit. This was not the case, as all words that were returned were words relating to the company. After that I entered “apple”, which only returned words relating words to the fruit. It seems that using capitals with certain words really changes their meaning.

*bank*: The word bank has more than one meaning. It can be a river bank, but also a place to store and lend money. In this dataset, the only captured meaning is the one relating to the place to store or lend money.

### 3.1.2

*1. What is polysemy, what is homonymy, and what’s the difference?*
Polysemous words are words that have the same way of writing them, but have completely different (disconnected) meanings (e.g. a bank for money and a river bank). Homonymous words are also written the same but have a connected meaning (e.g. a tree as a woody plant and a tree as a decision tree).

*2. Word embeddings are especially suited for word-level classification tasks. One of these tasks, which has the goal to distinguish between different senses of a word, is called word sense disambiguation (WSD). Why are word embeddings (as given in this assignment), not a good feature for this task?*
They are trained on only six categories, and a trained to find the category with which it has the most similarity. WSD needs the context to guess the best meaning using a lexical resource as WordNet.

*3. In the similarity tool, type ‘cookie’, and look at the most similar words. Then, start the same tool, but with another (smaller) set of vectors, using ./distance vectors.bin. Now, again, try ‘cookie’. What do you see? What could have caused this?*
Using ‘cookie' with the Google News vectors data, the words with the highest cosine similarity are words relating to the edible object. Examples are peanut_butter_fudge, oatmail_cookie and cupcake. However, cookie is entered with vectors.bin, the words with highest cosine similarity are words relating to the thing that stores personal information which is used in browsers. The difference between these words is that the system is trained on different data. The vectors.bin probably does not have many data on the edible cookie, as opposed to the Google News vectors, which is larger and thus has a higher possibility to have more meanings for one word.

### 3.1.3

*National sport:*
Netherlands football Canada: hockey (0.566876)
Netherlands football New_Zealand: rugby (0.651627)
Netherlands football USA: basketball (0.547168)

*Countries and their old leaders:*
China Mao Russia: Stalin (0.669819)
Germany Hitler Italy: Mussolini (0.703932)
China Mao Germany: Hitler (0.594724)

*Family relations:*
grandfather father father: son (0.819212)
grandmother mother mother: daughter (0.779377)
father brother mother: sister (0.837864)

## Assignment 3.2

### 3.2.1

*Binary:*
Classification accuracy: 89.41%
Baseline most common class is NON-LOCATION with 66.61%

*Six way:*
Classification accuracy: 81.53%
Baseline most common class is GPE with 31.75%

### 3.2.2

| n_iter | eta0 | Accuracy binary | Accuracy multinomal |
|---|---|---|---|
| 5 | 1 | 89,41% | 81,53% |
| 1 | 1 | 83,46% | 80,14% |
| 10 | 1 | 91,74% | 82,74% |
| 20 | 1 | 93,0% | 73,9% |
| 5 | 0,5 | 89,41% | 81,53% |
| 5 | 2 | 89,41% | 81,53% |
| 5 | 5 | 89,41% | 81,53% |
| 5 | 10 | 89,41% | 81,53% |
| 5 | 20 | 89,41% | 81,53% |
| 20 | 10 | 93,0% | 77,4% |
| 20 | 20 | 93,0% | 77,4% |

*Table 1: Different settings perceptron*

After trying out the settings found in table 1, it became clear that the n_iter parameter is has the most influence on accuracy. For the binary classification, setting the parameter to 20 has the biggest impact, for multinomial it 10. It is worth noting that the eta0 paramater seems to have no effect on the accuracy.

## 3.2.3

Clinton Bush Reagan (PERSON) - Obama: PERSON (TRUE)
yearly monthly daily (DATE) - hourly: GPE (FALSE)
Pennsylvania Texas California (GPE) - Illenois: PERSON (FALSE)

## 3.2.4

```
             precision    recall  f1-score   support

   CARDINAL       0.92      0.88      0.90      1311
       DATE       0.87      0.91      0.89      1017
        GPE       0.86      0.94      0.90      2915
        LOC       0.75      0.65      0.70       177
        ORG       0.73      0.64      0.69      2072
     PERSON       0.77      0.78      0.78      1407

avg / total       0.82      0.83      0.82      8899

[[1152   34   28    2   88    7]
 [  42  921    0    1   52    1]
 [  14   11 2737    2   92   59]
 [   0    0   38  115   21    3]
 [  29   86  341   23 1334  259]
 [  20    6   35   10  232 1104]]
```

Combining the classification report with the confusion matrix it becomes clear that some classes score better than others. For instance the GPE class scores very high with an accuracy of 0.94 but GPE also appears the most with 2915 entries in the test data. The lowest scoring class is ORG with an accuracy of 0.64, closely followed by LOC which has an 0.65 accuracy. ORG however as 2072 entries while LOC only has 177 entries.
ORG seems to be classified often as LOC or PERSON and LOC seems to be classified often as GPE. The latter is easily explained by the similarity between locations and geo-political entities, while the former is more vague. By using word embeddings, the words themselves and their surrounding words all become features in the vector for a word and these features are represented as arrays with scores for every class.