# Exercise 1

## Part a

*How is the result compared to using the full brown tagset in the introduction?  
Why do you think one of the tagsets yields higher scores than the other one?*

Result from the run, using the Universal Tagset instead of the Brown Tagset: **0.8689**

Result from the previous run with the given Brown Tagset: 0.7915

The large difference in the scores achieved by using two different tagsets might be a result of the different tags applied and their number, too. The Brown tagset uses 87 different tags to classify the parts of speech it encounters and therefore it is a very elaborate tagset. On the other hand, the Universal Tagset only applies 12 tags, which makes it a more reduced tool to use. However, this reduction to 12 “core” tags for classifying parts of speech might lead to higher accuracies in classification tasks, since the tagger is then less prone to wrong tagging of words, or – to put it differently – the probability that the tagger tags a word correctly is higher when there are only 12 tags then when there are 87 tags to choose from.

## Part b

# Exercise 2

## Part a

*Train the ScikitConsecutivePosTagger on the \*news\_train\* set and test on the \*news\_dev\_test\* set with the \*pos\_features\*. Do you get the same result as with the same data and features and the NLTK code in exercise 1a?*

Results from running the code on the ScikitConsecutivePosTagger: 0.857

The result when running the same data and features with the ScikitConsecutivePosTagger differ slightly from the ones achieved with the NLTK code. To be more precise, the tagger implemented in exercise 2a achieves a slightly lower accuracy than the one implemented in exercise 1a.

## Part b

With the best choice of alpha, do you get the same results as with the NLTK code in exercise 1a, worse results or better results?

|  |  |
| --- | --- |
| **Value of alpha-parameter with BernoulliNB** | **Result** |
| 1 | 0.857 |
| 0.5 | 0.8749 |
| 0.1 | 0.8695 |
| 0.01 | 0.8683 |
| 0.001 | 0.8651 |
| 0.0001 | 0.8631 |

The best choice of the alpha-parameter is alpha=0.5, which yields an accuracy of 0.8749. This is slightly better than what the NLTK-code gives as a result. Moreover, the parameter alpha=0.1 yields a better result as well, while all the other parameters fall behind the NLTK-result from exercise 1a.

## Part c