

GRUNDLAGEN DER KÜNSTLICHEN INTELLIGENZ

Programming Exercise 4: Probabilistic Models

Moritz Klischat, Franz Rieger, and Lukas Schöbel

(last updated 10th January 2020)

Submission deadline: 6th February 2020, 23:59

Programing Exercise 4: PO(TU)S

This introduction gives an overview on both subproblems, which can be solved independently of each other.

Problem 1: POTUS Tweet generator

Did you ever dream of becoming US president, but are lacking the stable genuity to create daily tweets like a POTUS?

With this exercise you can: Create a tweet generator based on a n -gram model. N -grams are sequences of n consecutive words. For example with $n = 2$ (denoted as bigram) and the given text *will leave florida*, the n -grams would be [will, leave] and [leave, florida]. Provided with @realDonaldTrump's tweets from the past years, the conditional probabilities of the n -grams need to be computed based on their frequency. Subsequently, new tweets are sampled from the conditional probabilities.

More detailed information are provided in the jupyter notebook on Artemis (see below).

Problem 2: POS (Part-of-speech tagging using Viterbi's algorithm)

Assigning grammatical tags like NOUN or VERB to words in sentences is referred to as *part-of-speech tagging* (POS). In this problem, POS tagging needs to be implemented using the Viterbi algorithm. Provided with a dataset of tagged texts, the parameters of a Hidden Markov model (HMM) are learned and subsequently, the Viterbi algorithm is applied to find the most likely sequences of tags for a previously unknown sentence.

Please note: HMMs and the Viterbi algorithm will also be discussed in exercise 9.

More detailed information are provided in the jupyter notebook on Artemis (see below).

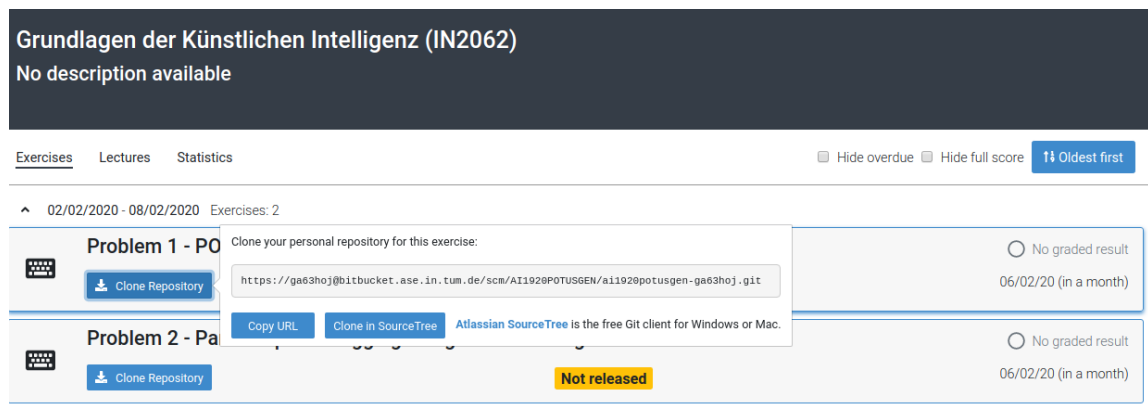
Evaluation on Artemis

From now on, this course will use Artemis <https://artemis.ase.in.tum.de/> to assess programming exercises. Artemis is an *Automatic Assessment Management System for Interactive Learning*, where you can get instant feedback on your solution.

To work on the exercise and submit your solution, perform the following steps:

1. Log in on <https://artemis.ase.in.tum.de/> with your TUM-ID and password.

2. Enter the course "Grundlagen der Künstlichen Intelligenz (IN2062)" and start the programming exercise, where you receive the url of your individual git repository:



3. Clone the git repository onto your local machine e.g. using command line and your individual url:
`git clone https://replace_with_url_of_your_repositroy`
4. Start working: the exercise is divided into two problems with a jupyter notebook each and has to be solved inside the notebook. Start the notebook from the terminal:
`jupyter notebook replace_with_name_of_the_notebook.ipynb`
5. After completing your code, submit your solution to Artemis using git commands
`git commit -a -m "your commit message here"`
`git push`
6. On Artemis, your submission is evaluated instantly after pushing.

Passing criteria

The exercise is passed if the evaluation of all tests for both problems on Artemis is successful. Detailed information on the passing criteria are given in the respective jupyter notebooks.

You can push your solution as many times as you want. For the final assessment, the last commit will be used.

Important: Since the exercise is evaluated automatically, do not change any file name, function/class names or input/output arguments of any function.