# Machine Learning for NLP Assignment 3

# Deadline: December 4th 2020

---

The theoretical part of this assignment involves finding information yourself on one of the most intuitive machine learning approaches: K-nearest neighbors. In the practical part, you will round up your system and carry out a feature ablation and error analysis.

You can extend the script you used for Assignment 2. You can find a script with an illustration of how to combine embeddings with other features under code/assignment3 in the github repository (use git pull for obtaining the last version): https://github.com/cltl/ma-ml4nlp-labs

During the first week of the module, one or two more advanced models will be added to the github repository. You can train and run these models during the second week (you will not need to program anything yourself for these models).

**Please indicate how much time you spent on each component in your submission. This is meant to ensure the workload of the course is appropriate (you get as much out of it as you can without becoming overworked). You can also provide a complete overview at the end of your submission.**

## 1 Theoretical Component

Find out what a K-nearest neigbor classifier is and provide a brief description of how it works.

## 2 Rounding up your system

If you have not done so yet, make sure your named entity recognizer includes some more advanced features (i.e. features that require a brief explanation, preprocessing or slightly more complex code to be used, such as a well-thought out way of capturing various forms of capitalization, previous and/or following tokens, etc.).

Investigate how your SVM with embeddings as features responds to adding some of the other features you have been using.

### 2.1 Feature Ablation Analysis

Feature ablation analysis is meant to establish the contribution of individual features. The idea is that you run experiments with different combinations of features and identify which features contribute the most (and which hardly contribute anything at all or even harm results).

Carry out an extensive feature ablation analysis for at least one system. Apply a selection of tests to the others.

## 2.2 Error Analysis

The purpose of error analysis is to gain deeper insight into what goes wrong and what works well. First insights can be gained by looking at a confusion matrix: e.g. which classes does a system find difficult to distinguish? Which classes does it find difficult to identify as names in the first place? Which are never confused?

The confusion matrix is just a first step. In the next step, you try and identify why a system has difficulties: is there a correlation between feature values and cases that go wrong? Are there other observations on frequent mistakes? In your analysis, make sure to pay attention that:

1. Your analysis is systematic (go beyond simply describing a handful of examples)
2. That you do not focus blindly on mistakes only: are there also similar cases where the system did behave correctly?

Make sure that you provide an advanced error analysis for at least one system and motivate your choice for the system.

**For your optional submission, it is recommended that you finished your feature ablation analysis and started on the error analysis.**

## 2.3 Running more advanced models

During the first week of this module, one or more models that are more advanced will be added to github. Train these models and evaluate their output.

## 2.4 Updating Your Report

Describe the work you did during this module as part of your system report:

1. Update the short task description, related work, preprocessing and feature description (from Assignments 1 and 2) if necessary.
2. Update the results sections with your complete systems (if applicable) and the outcome of the new models.
3. Add the outcome of your feature ablation analysis (with comment) and the outcome of your error analysis.
4. Add a discussion and conclusion.

## 3 Submission Assignment 3

By the end of the module, submit the following (this submission is **obligatory**):

1. Theoretical component. A pdf file including:
    - The description of how a K-nearest neighbor model works
    - Your updated system report (as outlined above)
    - An overview of the time spent on individual components of the course during these first two weeks
2. Practical component. A .zip or .tar.gz file of a folder with your student id as its name. This folder should contain any code you have worked with and written for these three assignments, with a readme explaining how to run things, if necessary. **The language model you used for obtaining the embeddings should NOT be included in your submission.**