AP = Anfängerpraktikum
FP = Fortgeschrittenenpraktikum

Initials:
AA = ayser.armiti@informatik.uni-heidelberg.de
CS = sengstock@informatik.uni-heidelberg.de
JS = stroetgen@informatik.uni-heidelberg.de
TB = thomas.boegel@informatik.uni-heidelberg.de
MG = gertz@informatik.uni-heidelberg.de

FP	Integration of additional languages in HeidelTime
JS	Given data: training/evaluation data for the specific language Algorithm: HeidelTime Programming: Java, regular expressions Goal: Additional language for HeidelTime Remark: languages of interest are, e.g., Latin, Romanian, Portuguese, Korean. Student should be familiar with the specific language and have (basic) knowledge of linguistics

AP/FP	Analyzing Quality Differences between Languages in HeidelTime using Parallel Corpora
JS	Given data: parallel corpora, HeidelTime Algorithm: Programming: java, perl/python Goal (AP): Analysis of language differences to discover weaknesses in single languages Goal (FP): + automated rule development for missing rules and expressions in single languages and automated improvement of rules extracting spurious expressions Remark:

FP	Linking German Wikipedia articles
TB	Given: German Wikipedia dump Problem: create a link structure for Wikipedia articles based on temporal expressions, mentioned persons and keywords Algorithm: Existing linking algorithm based on an UIMA pipeline Programming: Java (UIMA), Python

FP	Visualization of temporal news graphs
TB	Given: graph structure of news articles Problem: real-time visualization of chronologically ordered graph structures with user-adjustable parameters and settings Algorithm: existing visualization libraries (e.g., D3, Raphaël) Programming: Python, Java, JavaScript

AP/FP	Geospatial mean-shift vs DBScan vs K-Means clustering
CS	AP: Implementing mean-shift, DBScan, and K-Means clustering and evaluating the clustering results on geospatial data by visualization FP: Implementing adapted DBScan from given paper Data: Geo-referenced Twitter records (10M) Programming: Python or C++, Numpy/matplotlib for visualization

AP/FP	Geospatial visualization of similar tweet clusters
CS	AP: Implementing tweet similarity measures on the basis of jaccard, cosine, euclidean distance Clustering and visualizing the tweets to get prototype topic distributions using DBScan FP: Clustering on the basis of Locality Sensitive Hashing (LSH) Data: Geo-referenced Twitter records (10M) Programming: Python or C++, Numpy/matplotlib for visualization

FP	Social Network Graph Generator
MG	Given: graph parameters, distribution functions Problem: generate realistic social network graph from input parameters. Algorithm: existing graph generators Progamming: Python or Java Remarks: the approach should be scalable up to 10M+ nodes; graph should optionally be stored in widely used graph database

AP	Graph Matching Visualization
AA	Given: graph dataset Problem: generate an interface that can call a C++ back end engine with some parameters, collect the output, and use Gephi to visualize the result. Programming: free to choose, java prefered Remark: Gephi is a java open source framework to visualize graphs