Beginner practical TCrules

Rule-Based Translation with TC-ST Data

Task and Requirements

- Goal: rule-based c2c translator based on TC-ST parallel corpus
    - Subtask 1: create a pattern/rule database
    - Subtask 2: query translation using DB

Plan

- parallel corpus:
  - from TC-ST data (GeeksforGeeks)
  - sort out / regenerate functions and generalize them

- preprocessing:
  - remove unnecessary spaces and tabs
  - delete extra lines from input code

- tokenization:
  - split input to smaller units
  - map similar tokens to the same value
  - study similarity and make use of the overlap of syntax, keywords and constructs

Plan

- syntax tree:
  - generalize each input, and take the corresponding value in the target language

- hand-crafted rules:
  - expected difficulties by defining rules for translations between C++ to Python or Python to Java
    → start with simple cases

- other challenges:
  - code generalization
  - keys
  - joining partial code

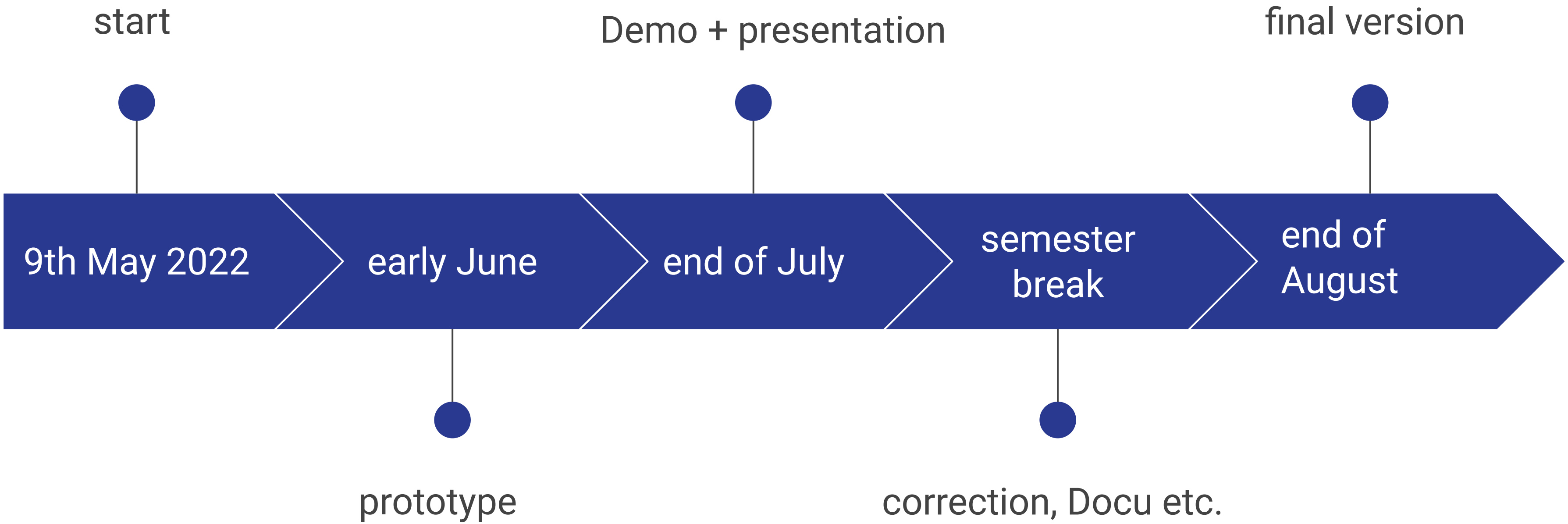
Architecture

- **Problem: Translation from A → B**
- Idea: “similarity matching”, dictionary of handcrafted-rules, AI
- input: file with source code in programming language A
- translate source code line by line by means of the rules and AI / ML
- if no fit / corresponding rule: translate manually & update DB
- output: file with source code in programming language B
- evaluate correctness

Technologies and Frameworks

- PyTorch (machine learning library)
- PyUnit (unit testing)
- Pylint (PEP8 coding style)
- Sphinx (documentation tool)
- DeepCode (static code analysis)
- CodiMD / trello (working state tracking)
- …

Preliminary schedule



9th May 2022 – start

early June – prototype

end of July – Demo + presentation

semester break – correction, Docu etc.

end of August – final version