St Petersburg University

Toolkit for automatic checking of programming assignments completion with code quality analysis and plagiarism search

Ван Тяньцзин

Научный руководитель: доцент кафедры СП, к.ф.-м.н. Д.В. Луцив

Introduction

- Plagiarism in academia is researched actively
- Plagiarism in source code is also popular research subject
- Plagiarism in academical source code is not a typical problem… but:
  - Still is a problem when working with junior students
  - Adds monotonous work to the teacher
  - Is seldomly a subject of research

The idea for this research is to create a toolkit for the teacher/trainer to automate:

- Programming assignment plagiarism detection (2nd term)
- Programming assignment code quality analysis (later)

The goal of the research

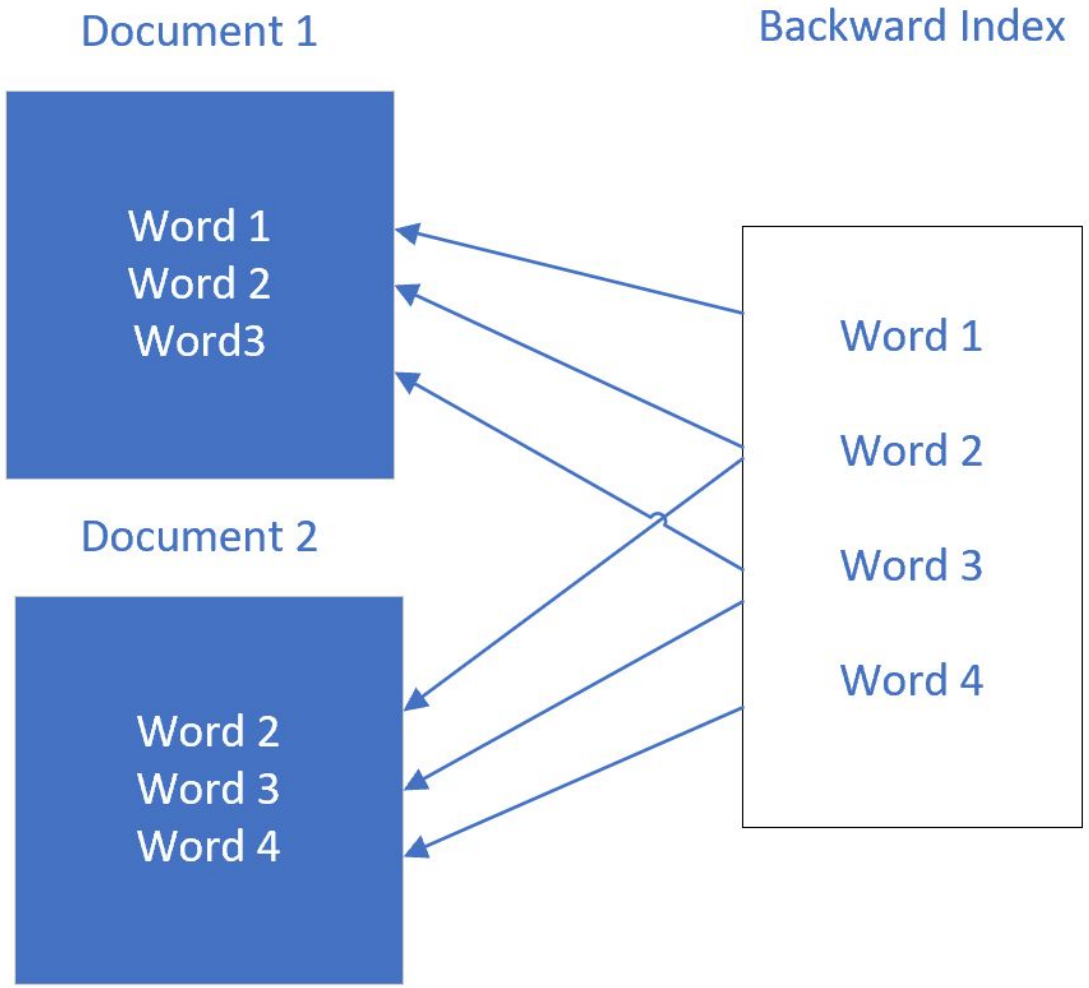
The goal of the research is to design and implement a toolkit for plagiarism detection. To achieve it, the following tasks are formulated:

- Overview existing tools and methods of code plagiarism detection and code search
  - Research question: use existing search tools and methods or implement custom one?
- Design an architecture of plagiarism detection tool
- How to implement plagiarism detections tool according to the architecture

Indexing & searching

- Indexing & searching mechanisms
  - Fuzzy indexing like SimHash & MinHash
- Existing search engines
  - ElasticSearch

Indexing & searching



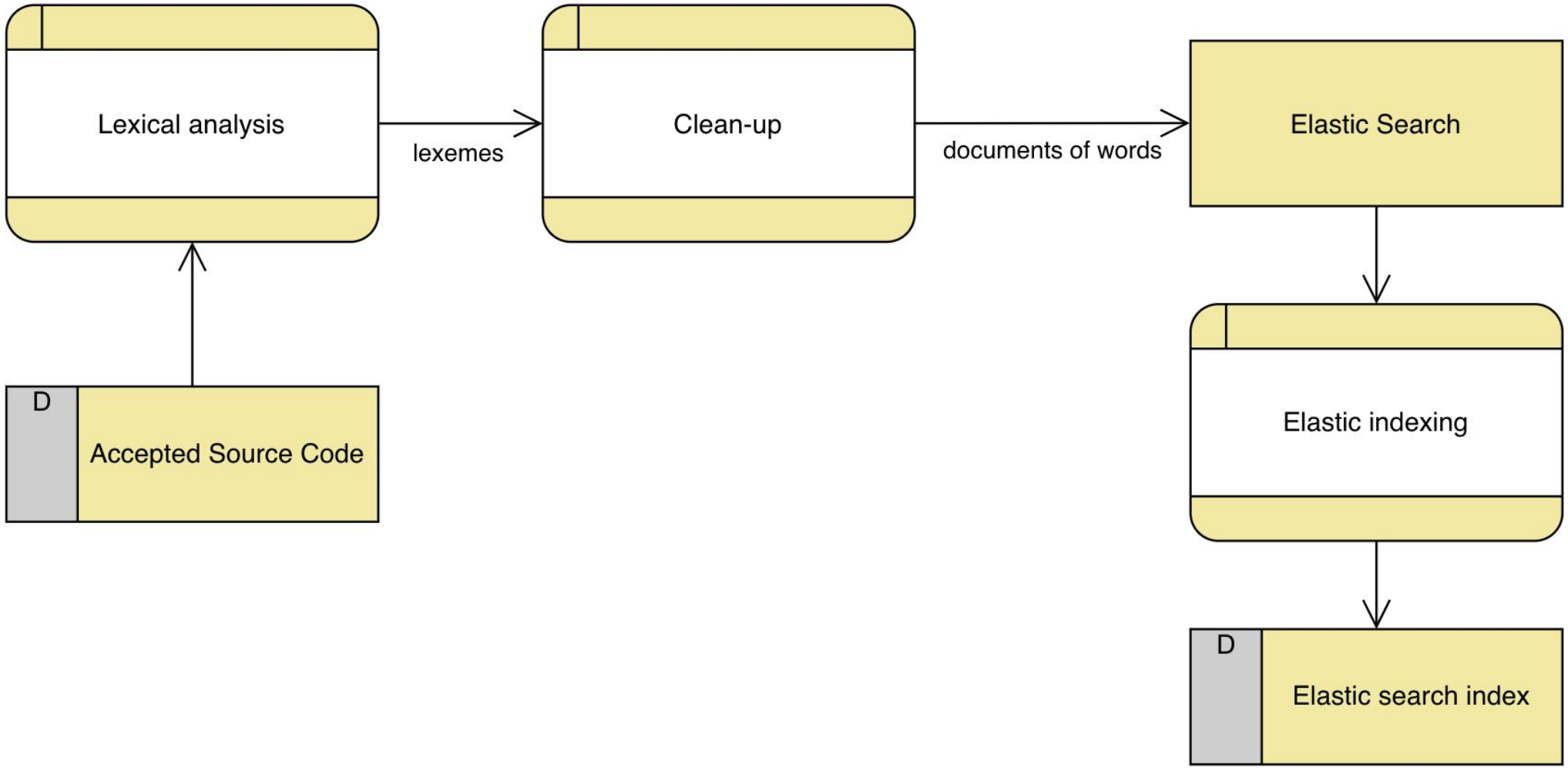
Homework validation tool architecture :: use cases

Plagiarism detection

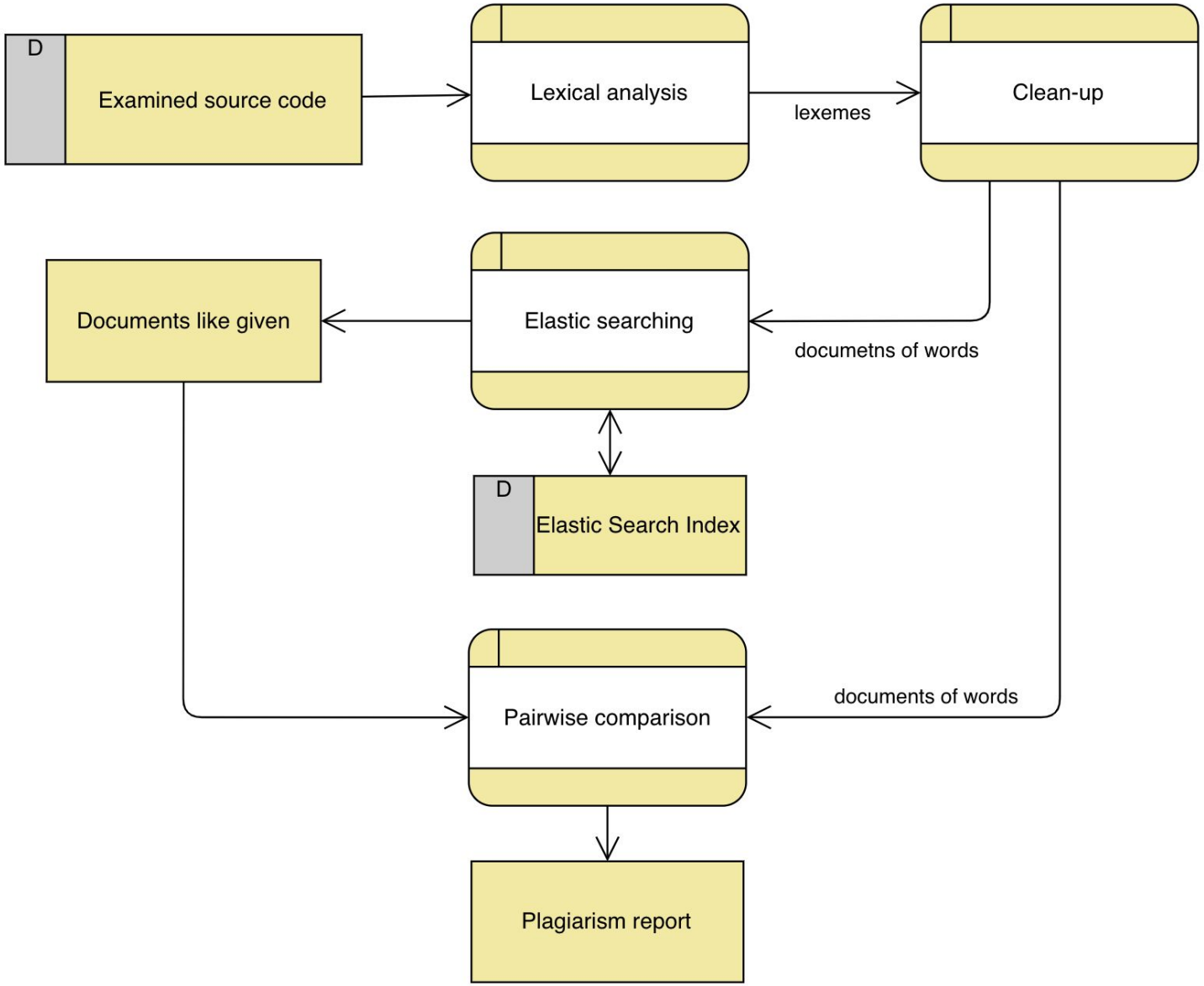
1. Accepted solutions are indexed
2. New posted solution is analyzed
3. Plagiarism report is generated

Code style checking **(future)**

Plagiarism detection tool architecture :: Workflow of indexing



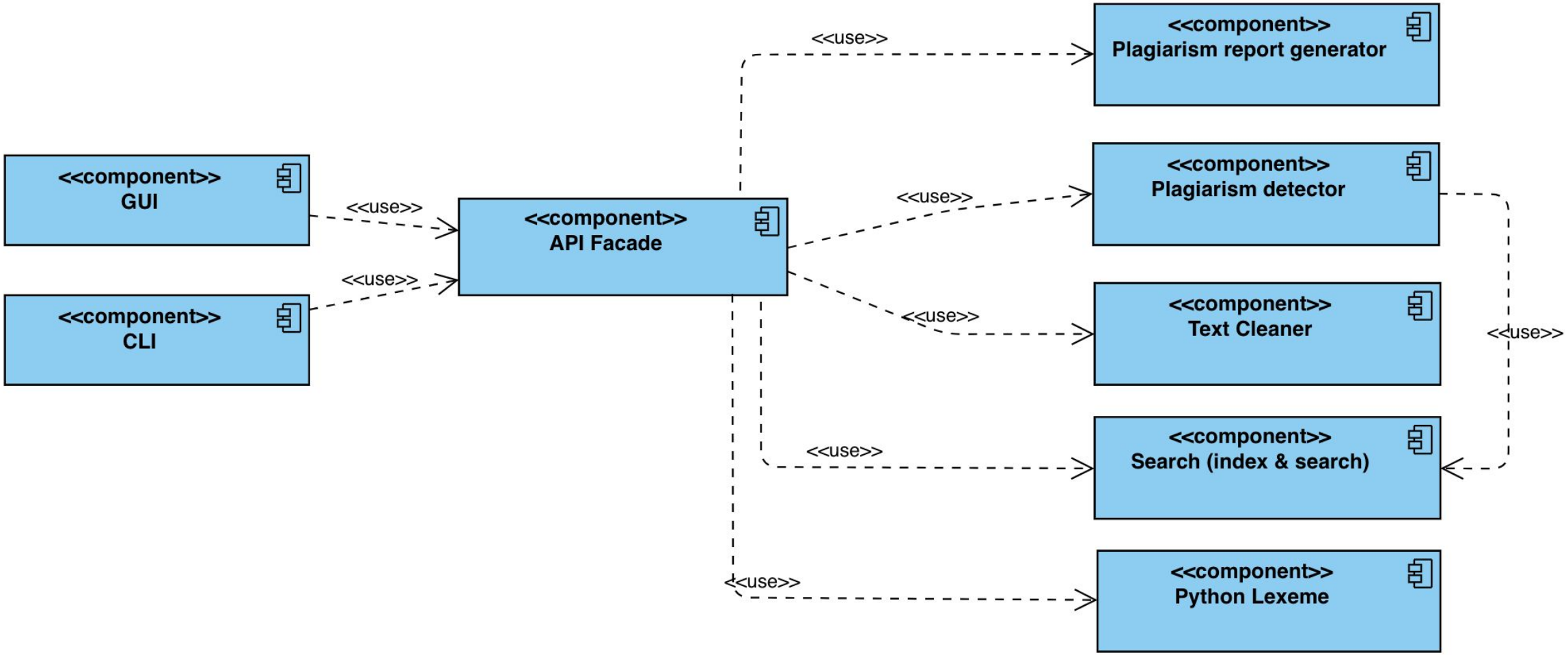
Plagiarism detection tool architecture :: Workflow of search



Pairwise comparison

1. Already implemented in a another plagiarism detection tool
2. Uses string slow string matching algorithms
3. Therefore, in this research, the pairwise comparison algorithm will not be considered in detail

Plagiarism detection tool architecture :: main components



The conclusion

For now, the following is done

- Overview existing tools and methods of code plagiarism detection and code search
  - Source code representation, source code similarity
  - Indexing and searching source code
  - Selection of indexing method and picking existing or implementing new tool
- Design an architecture of plagiarism detection tool
  - Use cases identified
  - Data flow is designed
  - Main components are distinguished
- Implement plagiarism detections tool **(at early beginning)**