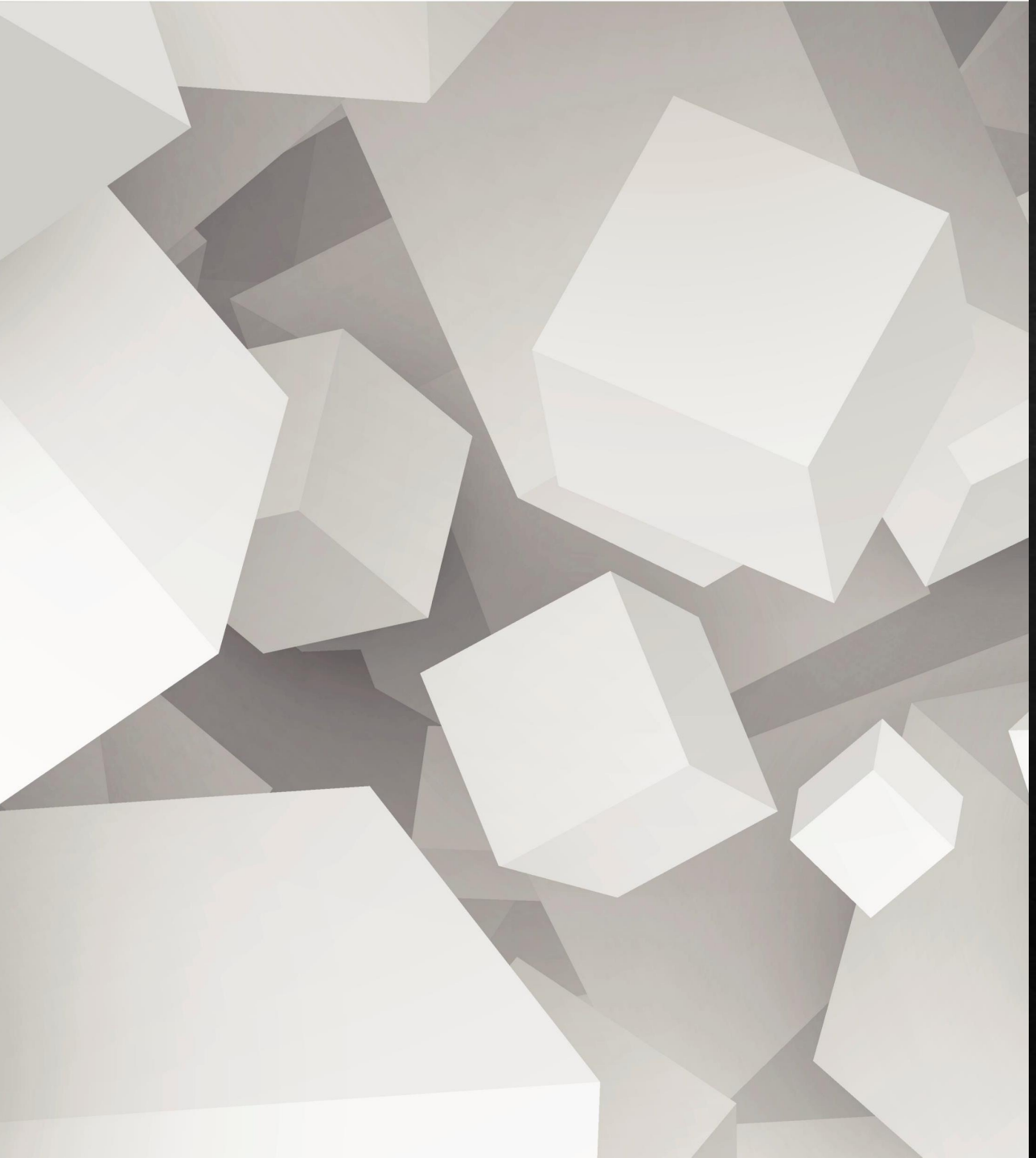
A Paralleled Text Analyzer

Zhang, Chenbo

Zhang, Yiwei

Yang, Shizhe

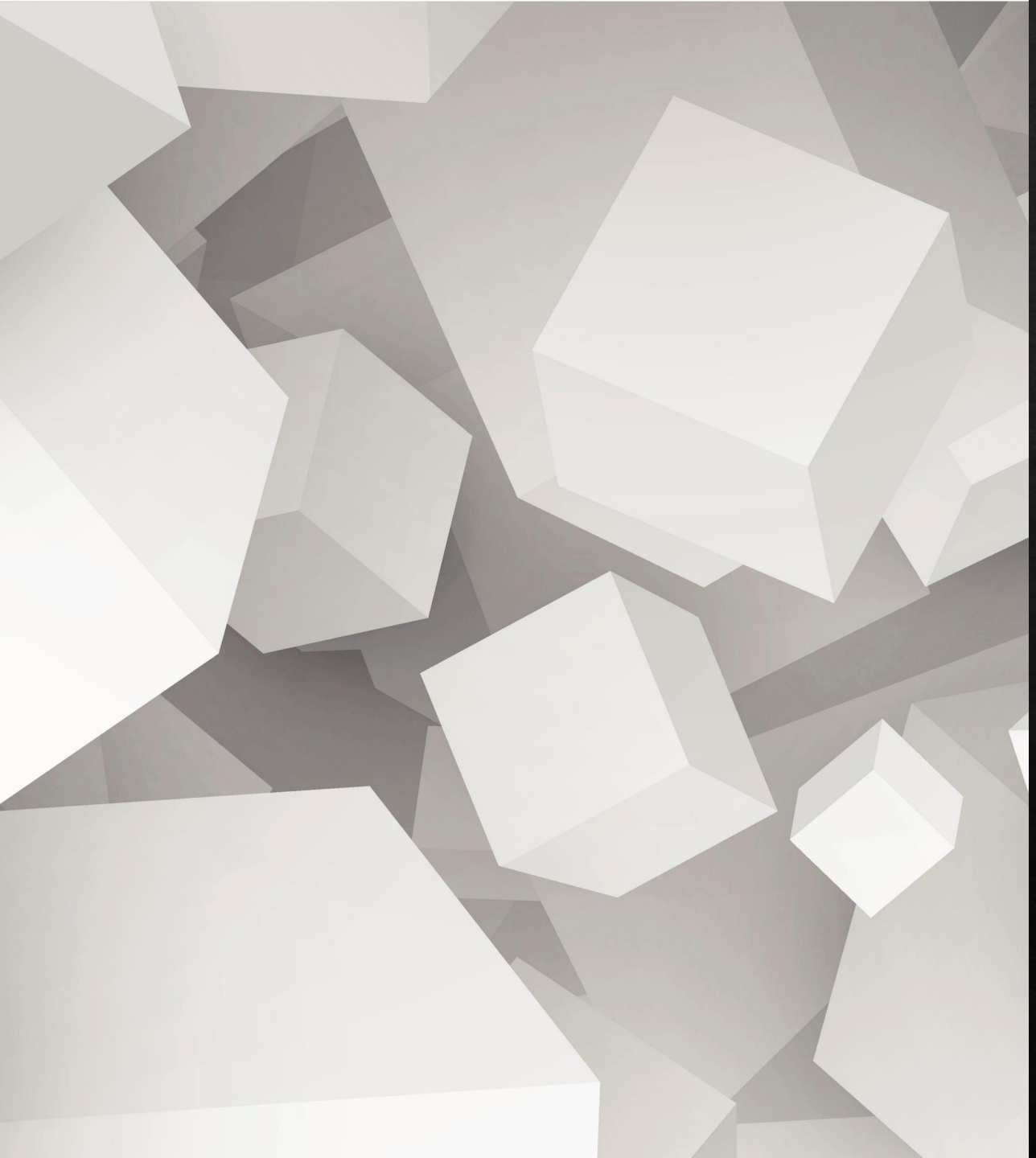


Situation

It starts with a full stack experience. We have a Restful API server of information management deposit at http://34.238.84.218:8000/.

Currently it can handle restful API request.

We want to add a module to generate tag and focus of text efficiently, which will need a paralleled algorithm to analyze the text data.



Situation

Our first thought is to use ranking method to generate tags and focus, see https://github.com/MarianaEel/Word_Analyzer.

However, this method will create a **Gigantic** Hashmap if the text information is too large. Also daily expression with no connection with the dialog will appear many times and will become the tag mistakenly, even with NTFL words removed.

So, not this method.

Difficulties

- The comparison and selection of paralleled algorithm.
- The implementation of algorithm
- Configure the environment on server.
- The deployment of algorithm on server.
- The integration of the algorithm with other part of the system.

Agile Development

- Sprint 1 (3.31 - 4.6) :
  - Research about word ranking algorithm and tag generator.
  - Discuss the availability of paralleled each algorithm and pick one.
  - Prepare the environment to implement the algorithm.
- Sprint 2 (4.7 – 4.13) :

  Implement the algorithm without parallel

Sprint 3 (4.14 - 4.22):

Using MPI or OpenACC to parallelize the algorithm, using plot to compare speed.

Sprint 4 :

Implement with the system.

Goals

Faster search speed on large text dataset

Multi-key search