Report

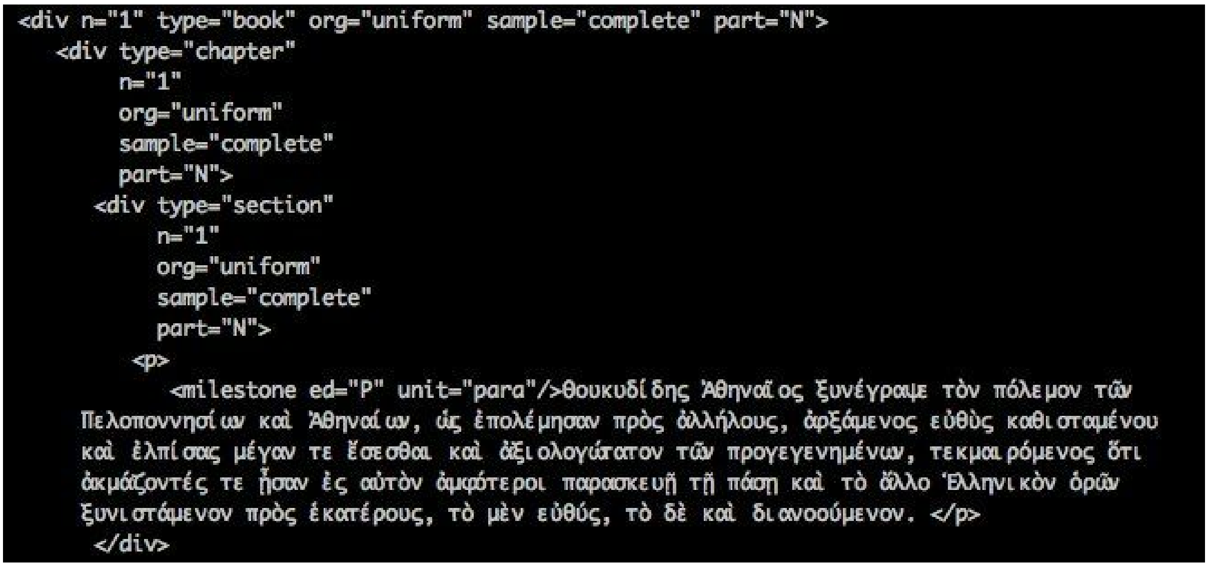
Project: Search and Clustering in the text of the Peloponnesian War

1. Project Summary

The aim of the project is to develop a web service that provides an online search platform for the Greek text of Thucydides’ Peloponnesian War. Additionally, the software presents a visualization of search results with thematic groups, called clusters. Furthermore, it is desirable to deliver an extensible foundation where new features or enhancements should easily be done in the application.

2. Input Data

An xml file, available within the project PerseusDL on GitHub, with the greek text was used as input data. The content is subdivided into books, chapters and sections, as showed below:

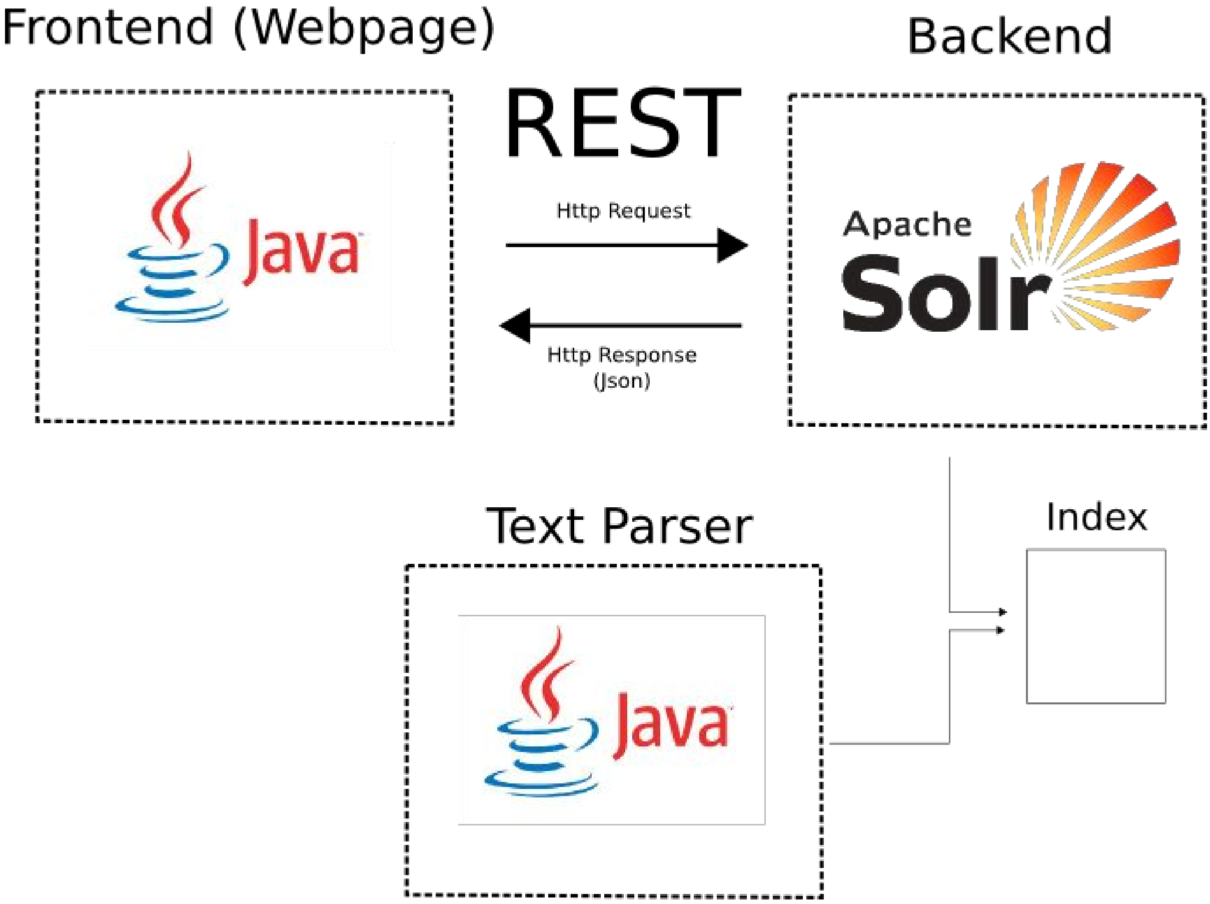


```
<div n="1" type="book" org="uniform" sample="complete" part="N">
   <div type="chapter"
        n="1"
        org="uniform"
        sample="complete"
        part="N">
      <div type="section"
           n="1"
           org="uniform"
           sample="complete"
           part="N">
        <p>
           <milestone ed="P" unit="para"/>θουκυδίδης Ἀθηναῖος ξυνέγραψε τὸν πόλεμον τῶν
    Πελοποννησίων καὶ Ἀθηναίων, ὡς ἐπολέμησαν πρὸς ἀλλήλους, ἀρξάμενος εὐθὺς καθισταμένου
    καὶ ἐλπίσας μέγαν τε ἔσεσθαι καὶ ἀξιολογώτατον τῶν προγεγενημένων, τεκμαιρόμενος ὅτι
    ἀκμάζοντές τε ἦσαν ἐς αὐτὸν ἀμφότεροι παρασκευῇ τῇ πάσῃ καὶ τὸ ἄλλο Ἑλληνικὸν ὁρῶν
    ξυνιστάμενον πρὸς ἑκατέρους, τὸ μὲν εὐθύς, τὸ δὲ καὶ διανοούμενον. </p>
      </div>
```

In this project we consider the section as the searchable unit or document. For example, if the word “war” is entered in the search field, the user is going to receive the sections containing “war” as results.

3. Architecture

The application was decomposed into three collaborating services, as presented below. The backend is responsible for the search and clustering capabilities and it was implemented with Apache Solr. The frontend is the website the user interacts with. It provides the graphical user interface displaying the results and clustering processed in the backend and it is written in Java. Finally, the also java-based text parser is in charge of processing the input xml document and loading the parsed sections into the solr index. The services communicate using synchronous protocols such as HTTP/REST and they are developed and deployed independently of one another.



3.1 Backend

This component is the core of the application. The Apache Solr platform is an open-source search server based on the Lucene Java search library. Its major features include full-text search, hit highlighting, real-time indexing and dynamic clustering.

Its database is called index due to the way data is stored, in order to facilitate fast and accurate information retrieval, usually with inverted index data structures. Other features such as tokenization, trimming are also supported by Solr.

The clustering plugin attempts to automatically discover groups of related search hits (documents) and assign human-readable labels to these groups. The clustering algorithm is applied to the search result of each single query.

3.2 Frontend

This module communicates with the backend and display the results. It is the frontend that decides which search operator is going to be executed. The two options are the “and” and the “or” operators. For example, if the user search for two words, the system will return results, according to the “and” operator, that have both words. The “and” operator is the default one.

3.1.1 Clustering

There are three clustering algorithms out of the box, that can be applied. In this project, the default one, Lingo, was used and its documentation can be found under: http://project.carrot2.org/publications/osinski-2003-lingo.pdf.
The number of results for a single query is likely to have impact on the quality of groups' description. That means, the more results for a search query, the more quality for the clustering labels.

4. Benefits of the Project

The system offers searching features for the greek text of the Peloponnesian War through text results as well as thematic groups. It allows a further perspective of analyzing data from the text of Thucydides. Moreover, it consolidates a base for an infinity of possible feature extensions, such as wild cards search, retrieving results through the lemma form of a word, other search operators and enhancing clustering quality using other algorithms.