BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Exposé
Design and Implementation of a Benchmarking Framework to Evaluate Information Extraction Quality

Willi Schönborn, 774190
Advisor: Prof. Dr. Stefan Edlich

March 9, 2013

Motivation

Information extraction (IE) in general and natural language processing (NLP), as a field of computer science, in particular have been around for more than half a century already and a lot of scientific research has been done. With the rapid growth of the world wide web in the last years, the task to let machines extract information from unstructured documents written by and for human beings has got a lot of attention lately. What's missing in the current ecosystem, consisting of crawlers, indexers, search engines, data stores and natural language processing systems, is a tool to evaluate and compare the performance and quality of information extractors.

Requirements

The goal of this work is to define a set of gold standards, very much like the Star Schema Benchmark[1] and TCP-H[3] provides them for data warehousing systems, as well as to design and implement a platform that provides the tools necessary to analyze and compare the performance and quality of information extraction systems in a programmatic manner.

Functional requirements

1. Monitored execution
   Thiry party programs, in the form of simple jar files, will be executed in a contained environment. This includes the definition and documentation of a standard command line interface for those third party binaries, which may include names and ordering of optional parameters as well as return codes and supported formats of produced results.

2. Measuring
   The monitored executions will utilize built-in tools of the Java platform, e.g. the Java Management Extensions (JMX), to measure execution time, latency and memory consumption.

3. Computing scores
   After successful executions, the produced results will be analyzed and compared to an expected, predefined set of solutions. The result of this step will be the calculation of the F-score.

4. Statistics
   Statistical functions will be applied to the figures collected during the monitored execution. The results will be key performance indicators like the average memory consumption or the F-score in relation to the execution time.

5. Web UI
   The platform will include a web based user interface which provides HTTP uploads for IE programs, monitoring execution progress as well as statistics and scores in form of diagrams.

Technical requirements

1. Modularity
   The technical main goal is to provide a highly modularized platform based on current best practices in enterprise architecture. To achieve this, the software will be developed using the OSGi[2] Service Platform. Different modules will be defined as OSGi bundles with explicit bundle interfaces.

2. Embeddability
   The diffierent OSGi bundles of the platform may themselves be used as a third party libraries in other projects. The API of all bundles must therefore be well documented and tested.

The platform will be developed in an open-source fashion. That includes the source code being hosted on a public website, e.g. github.com, as well as being released under a corporate-friendly open source license.

References

[1] Pat O'Neil, Betty O'Neil, and Xuedong Chen. *Star Schema Benchmark*. May 2009. URL: `http://www.cs.umb.edu/~poneil/StarSchemaB.PDF` (visited on 07/10/2012).

[2] *OSGi Alliance Specifications*. URL: `http://www.osgi.org/Specifications/HomePage` (visited on 07/10/2012).

[3] *The TPC Benchmark^{TM}H*. URL: `http://www.tpc.org/tpch/` (visited on 07/10/2012).