TIRA: Text based Information Retrieval Architecture

Yunlu Ai, Robert Gerling, Marco Neumann, Christian Nitschke, Patrick Riehmann
yunlu.ai@medien.uni-weimar.de,
robert.gerling@medien.uni-weimar.de,
marco.neumann@medien.uni-weimar.de,
christian.nitschke@medien.uni-weimar.de,
patrick.riehmann@medien.uni-weimar.de

Bauhaus University Weimar
Faculty of Media
Media Systems
D-99421 Weimar, Germany

Abstract We present a framework that offers appropriate methods for managing and browsing the fast-growing flood of information in the research field of Text-based Information Retrieval. The usage of new Web-technologies and a flexible, modular design make this framework a seminal example for accomplishing the challenges raised by actual problems in Information Retrieval.
Key words: Genre Classification, Machine Learning, Information Need, Text-based Information Retrieval, WWW, Data Mining, Search Strategies

1 Introduction and Related Work

The exponential growth of electronically stored data requires qualified strategies to retrieve information. Especially in text-based data mining systems, the demands of search processes change: Instead of retrieving any data, today it is more important to retrieve appropriate information from huge collections of data. To accomplish this task, it is necessary to afford flexible designs of retrieval processes.
An example in the private sector states the Internet search using search engines: Usually a big number of search results is obtained, containing a lot of irrelevant data. To minimize the amount of irrelevant data, Information Retrieval (IR) systems can be employed. Another example is the healthcare sector: The huge amount of medical data requires the use of elaborate data mining systems to ensure good patient care and effective medical treatment.
Data analysis is applied as a step-by-step processing and the use of methods from IR, machine learning and statistics. Even though there are powerful applications available to solve particular retrieval problems, these applications are monolithic solutions that each of which is dedicated to solve a special problem.
The intention of TIRA is to offer a flexible text-based IR-framework that provides technologies to visually define complex IR-processes by connecting different single IR-components as well as to execute them and to show the retrieved information with the help of user-defined styles.
Scalability and reuseability are accomplished by the use of a Web-based client-server

architecture, autonomous, distributed components and XML as data encoding format. TIRA is a modular and self-configuring system providing the possibility to use a standard Web server for the communication between IR-components. Therefore it is simple to use, reliable and easy to extend. There exist other approaches concerning IR:

UIMA (Unstructured Information Management Architecture) is a software architecture and framework for supporting the development, integration and deployment of search and analysis technologies. It implements algorithms from IR, natural language processing and machine learning. [1]

CRISP-DM data mining methodology (SPSS/DaimlerChrysler) is described in terms of a hierarchical process model, consisting of sets of tasks described at four levels of abstraction. Data mining processes are splitted into generic and specialized tasks, that are executed in several process instances. [2]

WEKA is a collection of machine learning algorithms for data mining tasks. It contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. [3]

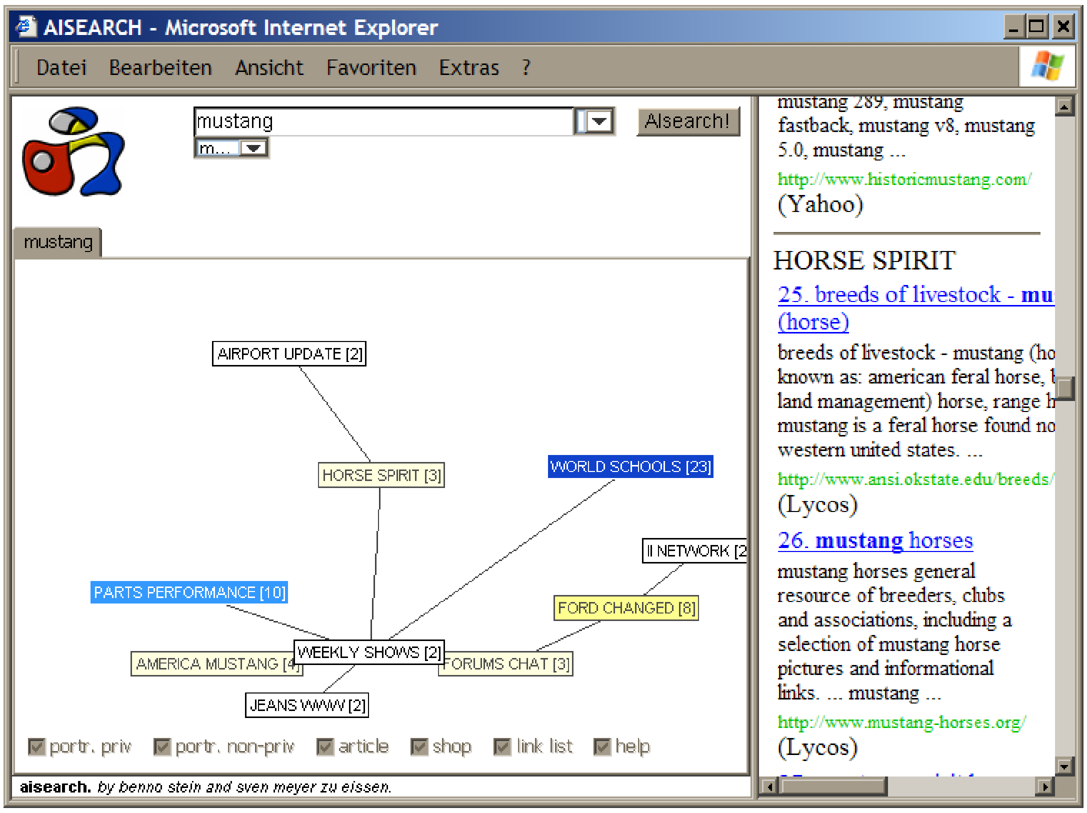
Nuggets is a proprietary desktop data mining software that uses machine learning techiques to explore data and to generate if-then rules. The goal is to reveal relationships between different variables. [4]

The following sections introduce the concepts and the architecture of TIRA. In addition to the separate components of the framework, the main features and technologies are explained.

2 Concepts

A fairly complex IR-process is required for getting a properly structured visualization of the huge amount of results of a search query. Such a visualization should be more descriptive and intuitive than common list views provided by standard search sites. For example, the visualization of a labeled topic structure graph in the *AIsearch* application (cf. Figure 1) requires the combination of several algorithms. Some common IR-algorithms like stemming, indexing and clustering as well as newly developed algorithms have been combined to construct such a complex process.

Like *AIsearch*, many of IR and data mining processes are built as monolithic applications, but during the design phase in an early step of the development they often



mustang 289, mustang fastback, mustang v8, mustang 5.0, mustang ...
http://www.historicmustang.com/
(Yahoo)

HORSE SPIRIT
25. breeds of livestock - mu
(horse)
breeds of livestock - mustang (ho
known as: american feral horse, l
land management) horse, range h
mustang is a feral horse found no
western united states. ...
http://www.ansi.okstate.edu/breeds/
(Lycos)

26. mustang horses
mustang horses general resource of breeders, clubs and associations, including a selection of mustang horse pictures and informational links. ... mustang ...
http://www.mustang-horses.org/
(Lycos)

Figure 1. The labeled structure graph of *AIsearch* shows search results in a more convenient view.

have been described as UML-activity diagram. With TIRA we propose an approach for a framework providing the ability to connect single IR-algorithms and -actions for building executable and customized retrieval processes without having a fixed or tightly coupled implementation.

TIRA uses UML-activity diagrams for defining such processes like the example shown in Figure 2.

A single action in such a UML-activity diagram describes a single runnable IR-process component called *processor module* that transforms real input data into real output data. The functionality to execute the UML-activity diagram that models an IR-process is provided by the server application of TIRA.

According to the requirements of modern software engineering like flexibility, reusability, scalability and modularization, the server is implemented as a platform independent application that invokes the autonomous processor modules for the execution of IR-processes. The TIRA approach is characterized by the following concepts.

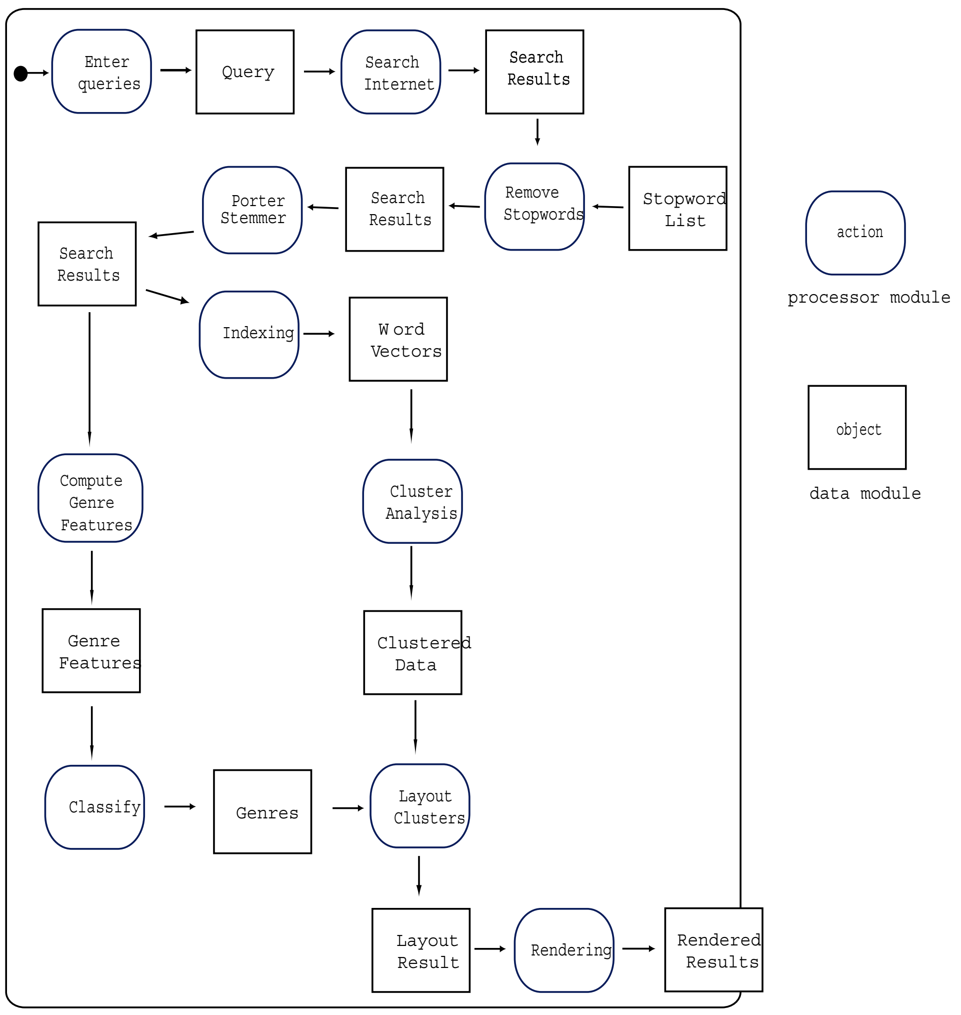


Figure 2. The UML-activity diagram shows all necessary actions and objects for the specific visualization of search results in Figure 1.

Custom IR-Processes TIRA provides a simple and efficient way to define IR-Processes that match the special needs of IR-problems by providing the ability to connect single IR-process components in a graphical user interface.

UML-Activity Diagram Syntax UML is used to provide a well known and widely accepted modeling technique for representing IR-Processes.

Petri Net Representation of IR-processes The designed IR-processes are internally represented as Petri nets.

Distributed Design The components of TIRA are loosely coupled. Every component of our system can be deployed on separate machines.

Task Distribution Every processor module has to compute only one action of the UML-activity diagram.

Autonomous Processor Modules According to the client-server paradigm, every single processor module has been designed as an autonomous and stand-alone service that provides the ability to process the assigned input data and generate output data.

General Access For Web-wide access, processor and data modules are represented and controllable by URLs.

Transport Protocol To avoid firewall problems, HTTP is applied as data transport protocol.

Platform and Machine Independence Processor and data modules may operate on any system that provides a Web server.

Programming Language Independence Every programming language that is executable by the hosting HTTP server can be used. Possible technologies for invocation include CGI-scripts, HTTP Server modules or JAVA servlets.

Simple Extension of Functionality To integrate additional processor modules it is sufficient to provide their URLs.

Flexible Data Representation A flexible and extensible data representation is required for the exchange and the storage of the data. Therefore XML as a widely accepted markup language is used. Furthermore XSLT is applied for the conversion of XML data into layouted XHTML data for visualization.

3 Architecture

In detail TIRA consists of four main components. The *client* provides the ability to define a customized IR-process as a UML-activity diagram. The *server* is responsible for managing the execution of defined IR-processes and for providing informations about the available processor and data modules. Every *processor module* provides a particular service functionality. These processor modules are separately invoked by the server for processing single steps in the context of the whole IR-process. The *data modules* hold input/output data for the processor modules and are also responsible for generating visualizations of their associated data content.

3.1 Client

The client, which is implemented as an Applet lets a user define an IR-process. While the Applet starts it requests a list of available processor and data modules from the

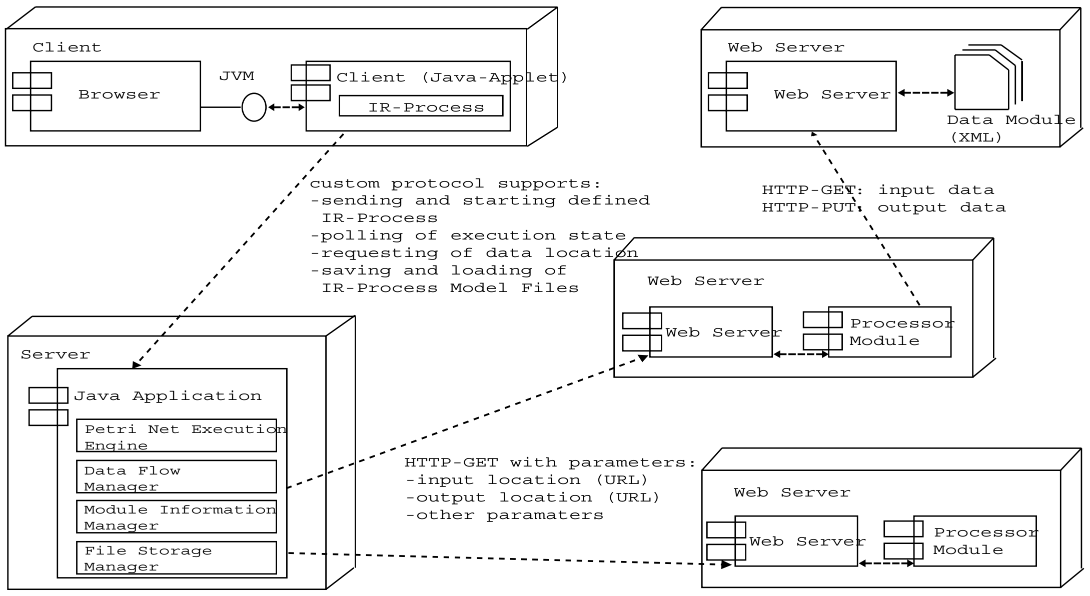


Figure 3. The UML-deployment diagram gives an overview of the different components and their cooperation during the definition and execution of a custom IR-process: The client transmits the IR-process model to the server. The server executes the IR-process by invoking single processor modules and generating process related URLs to manage the data flow.

server. For building a complex IR-process, the different processor modules can be connected with respect to the accepted input and output data modules. The model of the defined IR-process is transmitted to the server for execution. During the execution of the IR-process, the client continually polls the processing state from the server. After the execution has finished, the client is able to request the data that was generated by each component of the IR-process. To view the data, a standard Web browser is sufficient. The server manages URLs that are associated with the data modules; the data itself are stored in XHTML files.

3.2 Server

The server supports the management of the information about all available processor and data modules, the execution of the IR-process models as well as the storage of IR-process models. It constitutes the central unit of TIRA.

Management of Information about the available Processor Modules All processor modules are provided in a predefined directory structure. The server is able to gather information about all available modules that could be used by the client to define IR-process models. Modules can be added to the structure without the need of restarting the server. If a client posts a request for information about the processor modules, the respective path is scanned and the results are compiled as XML. Thus, one has the ability to extend the overall functionality by developing new modules.

Execution of Process Models When the client starts, it requests the server for the information about the available processor modules. This information is used for the definition of IR-process models as UML-activity diagrams in XML. The server is capable of executing such models, internally represented as Petri nets. During the execution the input and output data is stored on a HTTP server. Therefore the appropriate URLs for accessing the data on the HTTP server are assigned dynamically. After this the execution of the process is started. The client is provided with a unique ID that is needed for requesting the state of the execution and the URLs of the output data. The implementation of the Petri net execution engine supports concurrent processing of the modules.

Storage of Process Models When a process model is defined it can be transmitted to the server for storage and reuse. The client is able to request a list of the filenames of all stored files and to choose a concrete file for loading, saving or deleting.

3.3 Processor Modules

Every single processor module is represented and accessible by a URL. This approach depends on a running HTTP server and provides the following advantages: The implementation of the processor modules as HTTP-based services helps to avoid firewall problems and guarantees platform and programming language independence. Every processor module is an autonomous processing unit and is independent of other TIRA components.
Since a processor module is reachable through a HTTP-GET request, one has to find possibilities to access the input and output data. Due to the fact that it is impossible to encode a huge amount of data directly into a URL, the data is stored on an HTTP server, and the data is also represented as URL. For invoking the processor modules the data URLs serve as parameters for the processor modules. Consequently, a processor module is not only a service for processing data, but also a client for accessing the inputs via HTTP-GET and distributing the outputs via HTTP-PUT.

3.4 Data Modules

Data Exchange The input and output data of every single processor module is described and stored in XML. TIRA uses wrapper structures that contain the data at the client side for defining IR-processes and at the server side for preparing the data for visualization.

Data Presentation Due to the fact that pure XML data can not be visualized in a human-readable way directly, special data modules for layouting purposes are employed. On the one hand XSLT modules provide the ability to transform XML data into an XHTML representation by transformations using the Extensible Stylesheet Language (XSL) stylesheets and XALAN as processor. The XHTML data can be visualized by common browsers. On the other hand, special-purpose Applets for the visualization of more complex data structures like graphs are provided by the data modules.

4 Conclusion And Future Work

The main focus will be the extension of TIRA with new modules, to increase the variety of the possible IR-processes. Another aspect is the improvement of the IR-process execution to achieve a speed-up by using techniques like load balancing. Furthermore we like to advance the client application, especially the graphical user interface and the user guide to simplify the procedure of defining an IR-processes. Another issue concerns the usability of the client application to provide adequate feedback about the progress of an IR-process execution. Therefore a good heuristic for the time estimation of the execution progress has to be developed.

Bibliography

[1] *Unstructured Information Management Architecture SDK*.
http://www.alphaworks.ibm.com/tech/uima

[2] Gerber, C. et al. *CRISP-DM 1.0*. August 2000. www.crisp-dm.org

[3] *Weka 3: Data Mining Software*. http://www.cs.waikato.ac.nz/ml/weka

[4] Gilman, M. *Nuggets and Data Mining*. June 2004. http://www.data-mine.com