Chapter 1

Software requirement specification

1.1 Introduction

1.1.1 Purpose

The software requirement specification should provide all needed information to develop the context extraction framework and define all delivery objects. All interfaces to external components, input and output data, deployment considerations and quality attribute should be well defined within this document.

1.1.2 Scope

The context extraction framework will perform automated text extraction on a set of HTML test data with two to three different text extraction algorithms. The performance of each algorithm is measured and an output file with the measured results is generated.

1.2 General description

1.2.1 Operating Environment

see travis ci

1.2.2 Design and Implementation Constraints

1.3 System Features

In this chapter, each system feature is specified.

Name	Read configuration
Feature id	f1
Description	The text extraction framework is configurable with an external text file. The configuration file will contain following items: • Path to folder with html files • Path to folder with text files • Path to folder with output files • Configuration for algorithms • etc. The configuration file location is defined as a relative path to the source directory. The configuration file is structured in a key value list: `key:value;` `key:value;` `key:value;`
Relevance	needed
Related stories	tbd

Name	Create test
Feature id	f2
Description	A test contains two input files which are a html file and a text file. They are located in the defined directories by the configuration. As soon as the test framework finds a html and a text file with the same name, a new test is created and the files are read.
Relevance	needed
Related stories	tbd

Name	Run test
Feature id	f3
Description	A test is run as defined in the configuration file. The configuration file defines which algorithms are tested. The output of a test is a text file which contains the results.
Relevance	needed
Related stories	tbd

Name	Integrate Justext algorithm
Feature id	f4
Description	If the justext algorithm is activated in the configuration file and a test is run, the HTML file is extracted with justext.
Relevance	needed
Related stories	tbd

Name	Integrate Boilerpipe algorithm
Feature id	f5
Description	If the Boilerpipe algorithm is activated in the configuration file and a test is run, the HTML file is extracted with Boilerpipe.
Relevance	needed
Related stories	tbd

Name	Integrate RSS feed algorithm
Feature id	f6
Description	If the RSS feed algorithm algorithm is activated in the configuration file and a test is run, the HTML file is extracted with the RSS feed algorithm.
Relevance	nice to have
Related stories	tbd

Name	Comparison of extracted text files
Feature id	f6
Description	Each output file from the different algorithm needs to be compared to the text file with the actual content. • Split HTML document into blocks separated by HTML tags • Define which blocks are content and which are boilerplate based on the text file which defines the content • Define which blocks are content and which are boilerplate based on the output file of each text extraction algorithm • Compare the results and categorize all blocks as true negative or false positive • Put the results into an output text file (structure output file: tbd)
Relevance	nice to have
Related stories	tbd

Name	Analize data
Feature id	f7
Description	The generated output data is used to perform some further calculations. Possible values to calculate are: • Presicion: $\frac{TP}{TP+FP}$ • Recall/True positive rate (TPR): $\frac{TP}{TP+FN}$ • false positive rate (FPR: $\frac{FP}{FP+TN}$ • F-measure: $2 * \frac{presicion*recall}{presicion+recall}$ • Reciever Operation Characteristics (ROC): $TPR = f(FPR)$ Presicion / Recall / ROC / AUC
Relevance	needed
Related stories	tbd

1.4 Data Requirements

1.5 External Interface Requirements

1.5.1 Boilerpipe

The boilerpipe algorithm is already implemented in Java so it is easy to integrate. The API can be found under following link. `https://code.google.com/p/boilerpipe/`

Other useful links:

Getting started: `http://code.google.com/p/boilerpipe/wiki/QuickStart`

javadoc extractor: `http://boilerpipe.googlecode.com/svn/trunk/boilerpipe-core/javadoc/1.0/de/l3s/boilerpipe/extractors/ExtractorBase.html`

1.5.2 justext

The justext algorithem is implemented in python and it is not yet defined how it will be integrated into the text extraction framework. See risk analysis. The documentation can be found under following link:

`https://code.google.com/p/justext/`

jython: http://www.jython.org/

1.6 Quality Attributes