MINISTRY OF EDUCATION, YOUTH AND SPORTS

norway grants

The Research Council of Norway

D1.1.1: System specifications
Overall system design definitions

Authors:

Björn Gambäck
NTNU
gamback@idi.ntnu.no

Aleš Horák
Masaryk U.
hales@fi.muni.cz

Pavel Rychlý
Masaryk U.
pary@fi.muni.cz

Grant Agreement Number	7F14047
Project Acronym	HABiT
Project Title	Harvesting big text data for under-resourced languages
Deliverable Title	**D1.1.1: System specifications** **Overall system design definitions**
Responsible Partner	**Björn Gambäck, NTNU**
Dissemination Level	Public
Due Delivery Date	31 December 2016 (+30 days)
Actual Delivery Date	17 January 2017
Status	Final, v2.0

Principal Investigator	**Karel Pala**
Project Promoter	**Masaryk University**
Tel	+420 549 49 5616
E-mail	pala@fi.muni.cz
Project Website Address	www.habit-project.eu

Table of Contents

1 Introduction

The goals of WP1 ("System integration") are to

- To define the overall system specifications
- To integrate and test all software modules produced in WP4–WP6 in an iterative way
- To create a demo website to illustrate the project to end users and potential stakeholders
- To create the overall integrated system

This document addresses the first of those goals, or specifically Task T1.1: "System and modules requirements specification", which is concerned with the overall system specifications and the definition of the functional and non-functional requirements. This entails the design plans for the basic HABIT system framework and the representations and interfaces between different system components. This includes design of the functionality of the modules as well as the specification of the control structure and the coordination of information flow among the modules. However, the specifications of the individual modules will be included in separate deliverables and only touched on briefly here.

2 Overall system design

The basic HABIT architecture will be implemented and installed at all participating institutions. The main contributors to the implementation are Masaryk and NTNU, but the associate partner Addis Ababa University will also contribute with feedback on the system design and in particular on the strategies for collecting and annotating Ethiopian language corpora and on building tools for those. The HABIT system will consist of a set of re-usable components that will provide overall interaction coordination between different modules and shared knowledge (information) storage mechanism. Documentation of the architecture will be written and distributed to other partners using the project www-pages. The basic system modules will be loosely coupled through the HABIT architecture. The very general HABIT application structure will not contain any actual functionality except for invoking the system modules.

The system specification document is intended to be updated and modified during the course of the project, since the HABIT development approach will be iterative, concerning both individual system modules and the system as a whole. This approach entails the creation of intermediate system prototypes, that will incorporate the results of the project's validation and evaluation activities, allowing the project consortium to effectively address any critical issues that may emerge during development. Two development phases are planned, each resulting in a system prototype, due at the end of 2016 and at the end 2017, respectively. These prototypes and the associated respective versions of the software modules will be subsequently validated/evaluated in terms of performance and quality. The testing results will then be fed back into the module development process to support the system improvement as it proceeds towards next prototypes. The third (and pre-final) system prototype will be used for usability studies and final system evaluation. The usability studies will finish at the end of the project lifetime, when the final system will also be released.

3 System modules

The HABiT system will thus be composed of mostly independent modules. Different tasks could be solved by combining several tools together. Each module could be developed and tested independently or by using only a small number of other modules. Most of the tools will have two interfaces:

1. **unix command line**
   A unix command (or script) uses data from the standard input and produces results to the standard output
2. **web API**
   The module is run as a web service using standard HTTP protocol and JSON format.

In essence, there will be four types of modules in the HABiT system: corpus harvesting, corpus annotation, corpus searching, and corpus application and exploitation modules.

3.1 Corpus harvesting and corpora building modules

For web harvesting and corpora building, the tools developed by Masaryk (in cooperation with Lexical Computing Ltd.) will be adapted and adopted, including the following:

Spiderling Crawler: a web spider for linguistics. It can crawl text-rich parts of the web and collect a lot of data suitable for text corpora.

JusText: a web page HTML boilerplate removal tool. It can strip navigation links, headers, footers, etc. from HTML pages and leave just regular text containing full sentences.

Onion (ONe Instance ONly): a tool for removal of exact duplicates and near duplicates from large collections of texts. It can measure the similarity of paragraphs or whole documents and drop duplicate based on user set threshold levels.

Chared: a tool for detecting the character encoding of a text in a known language. It contains models for a wide range of languages.

Unitok: a universal text tokeniser with specific settings for many languages. It can turn plain text into a sequence of newline-separated tokens ("vertical" format), while preserving XML-like tags containing metadata.

czaccent: adding diacritics into Czech texts without diacritics.

Most of these methods have already been successfully used for various languages (English, German, Italian, Chinese, Arabic). During the project attention will be paid to further development and improvement of the indicated techniques.

3.2 Corpus annotation modules

Annotation tools (e.g., part-of-speech taggers) will take advantage of the existing instruments based both on rule and statistical approaches, such as the **Oslo-Bergen Tagger** for Norwegian developed by the associate partner Oslo University.

For Ethiopian languages new tools will have to be developed (taggers and parsers), although we will investigate the possibilities to re-use the few resources that already are available for these languages, in particular Michael Gasser's **HornMorpho** (University of Indiana), which provides some morphological processing for Amharic, Tigrinya and Afaan Oromo.

For corpus indexing, Masaryk's **compilecorp** tool will be used. It creates supplemental data like indexes and frequency tables for faster corpus querying and exploitation.

3.3 Corpus searching modules

Shallow processing techniques and tools such as the **Sketch Engine**, **freqs** and **lscngr** will be used. These will provide detailed information about the grammatical and collocation behaviour of the individual words of language in the form of tables, and can thus be used in various lexicographical applications such as compiling electronic dictionaries, machine translation, and extracting terminology. The Word Sketch Engine is also capable of indexing and fast searching in parallel corpora.

3.4 Corpus exploitation modules

Creating large textual corpora is a very important issue and the main aim of the project. However, actually using and exploiting the corpora in language processing tasks is equally important, overall. The HaBiT corpus exploitation modules will include a range of different tools for creating models for vector-spaced word meaning representation and word embeddings, for named entity recognition, for keyword extraction, and for word sense disambiguation.