This document is part of the Coordination and Support Action “Preparation and Launch of a Large-scale Action for Quality Translation Technology (QTLaunchPad)”.This projecthasreceivedfundingfromthe European Union’sSeventh Framework Programme forresearch, technologicaldevelopmentanddemonstrationundergrantagreementno.296347.

Deliverable D4.3.1

Multilingual corpus acquisition software

Author(s): Vassilis Papavassiliou, Prokopis Prokopidis, Stelios Piperidis (ILSP)

Dissemination Level: Public

Date: 27.11.2014

Grant agreement no.	296347
Project acronym	QTLaunchPad
Project full title	Preparation and Launch of a Large-scale Action for Quality Translation Technology
Funding scheme	Coordination and Support Action
Coordinator	Prof. Hans Uszkoreit (DFKI)
Start date, duration	1 July 2012, 24 months
Distribution	Public
Contractual date of delivery	December 2013
Actual date of delivery	27 November 2014
Deliverable number	D4.3.1
Deliverable title	Multilingual Corpus Acquisition Software
Type	Report
Status and version	Final v2.0
Number of pages	13
Contributing partners	ILSP, DCU
WP leader	ILSP
Task leader	ILSP
Authors	Vassilis Papavassiliou, Prokopis Prokopidis, Stelios Piperidis
EC project officer	Aleksandra Wesolowska
The partners in QTLaunchPad are:	Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Germany
	Dublin City University (DCU), Ireland
	Institute for Language and Speech Processing, R.C. “Athena” (ILSP/ATHENA RC), Greece
	The University of Sheffield (USFD), United Kingdom

Version History

Version	Date	Comments
1.0	10/01/2014	
2.0	27/11/2014	With respect to the comments in the final review report, the following items have been included: • Information about the performance of the pair detector and its comparison with a similar tool (Section 2.2) • Resources used to bootstrap the crawler and the default configuration file (Appendices)

For copies of reports, updates on project activities and other QTLaunchPad-related information, contact:

DFKI GmbH
QTLaunchPad
Dr.AljoschaBurchardt aljoscha.burchardt@dfki.de
Alt-Moabit 91c Phone: +49 (30) 23895-1838
10559 Berlin, Germany Fax: +49 (30) 23895-1810

Copies of reports and other material can also be accessed via http://www.qt21.eu/launchpad

Contents

1 Introduction 4
2 System architecture 4
2.1 Link Extractor 5
2.2 Pair Detector 5
3 Installation and use 7

1 Introduction

This deliverable documents the open-source ILSP focused crawler (ILSP-FC) that was extended and enhanced in the context of *WP4: Identification, acquisition and discovery of language resources* and was used to acquire the domain-specific multilingual data described in *D4.3.2 Domain-specific and genre-specific multilingual data*. Depending on user-defined configuration, the crawler employs processing workflows for the creation of either domain-specific or general monolingual or bilingual collections. Therefore, the modules of the system are already described in *D4.2.1 Monolingual corpus acquisition software* and this deliverable focuses on link ranking and pair detection. Due to its nature, this deliverable aggregates in Section 2 considerable parts of a conference paper[1] that also discussed this module integrated in the crawler. In addition, this deliverable includes in Section 3 information on how to get, install and use the ILSP-FC.

2 System architecture

The workflow of the ILSP-FC for building bilingual resources is illustrated in Figure 1. The required input from the user consists of a list of terms that describe a domain in the targeted languages and one or more URLs pointing to web sites from which parallel data could be extracted (e.g. a multilingual web site, or web sites like www.site.de and www.site.com).

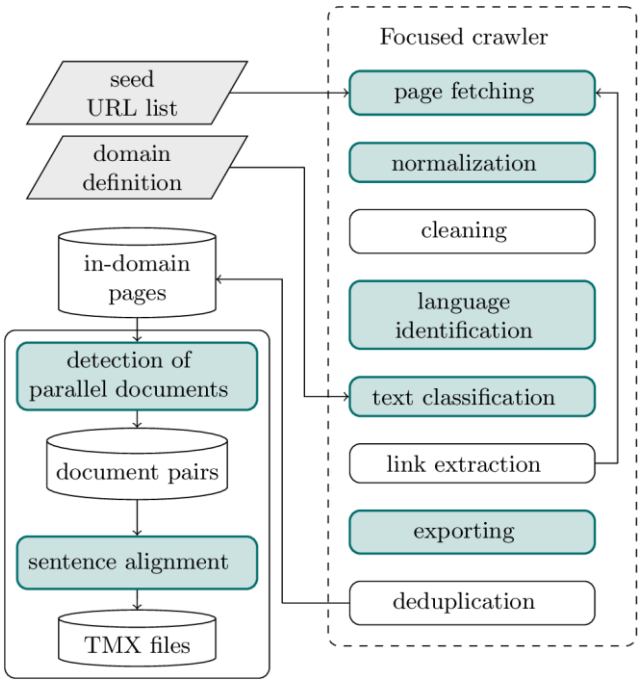

Figure 1. System architecture

[1]Papavassiliou, P. Prokopidis and G. Thurmair. A modular open-source focused crawler for mining monolingual and bilingual corpora from the web.6th Workshop on Building and Using Comparable Corpora. ACL 2013. Sofia

In general, the crawler starts from these URLs and in a spider-like mode visits pages inside these web sites. As described in *D4.2.1 Monolingual corpus acquisition software*, the modules for page fetching, normalization, cleaning language identification, text classification, link extraction, exporting of cesDoc files and (near) de-duplication are applied. The result of these processes is a domain-specific collection that contains web documents in the targeted languages.

2.1 Link Extractor

The main difference between using the ILSP-FC for building monolingual and bilingual collections concerns the selection and prioritization of extracted links. In case the crawler is used for acquiring monolingual data all links are added to the list of the links to be visited and the score link is mainly influenced by the "domainess" of its surrounding text (see subsection 2.6 of *D4.2.1 Monolingual corpus acquisition software*). When ILSP-FC is used for building multilingual collections, only the links that point inside the targeted web sites are selected and the link score is powered by the probability that the link under consideration originates from a web page in L1 and "points" to a web page that is probably in L2. This is the case when, for example, the targeted languages are EN and DE, and the anchor text of a link in an English web page contains strings like "de", "Deutsch", etc. Thus, the crawler is forced to visit candidate translations before following other links.

2.2 Pair Detector

After in-domain pages are downloaded, the Pair Detector module uses three methods to identify pairs of pages that could be considered parallel. The first method compares the URLs of the acquired web pages and considers them candidate translations if the differences only concern special patterns like /l1/ and /l2/, etc. The second method is based on cooccurrences, in two documents, of images with the same filename, while the third takes into account structural similarity.

Each XML file[2] is parsed and the following features are extracted: i) the document language; ii) the depth of the original source page, (e.g. for http://domain.org/d1/d2/d3/page.html, depth is 4); iii) the amount of paragraphs; iv) the length (in terms of tokens) of the clean text; and v) the fingerprint of the main content, which is a sequence of integers that represent the structural information of the page, in a way similar to the approach of Bitextor described by Esplà-Gomis and Forcada[3]. For instance, the fingerprint [-2, 28, 145, -4, 9, -3, 48, -5, 740] denotes that the corresponding document consists of: i) a title (i.e. a <p> element with *title* as value of the type attribute and length of 28 characters, ii) a paragraph with 128 characters, iii) a heading with length of 9 characters, iv) a list element with 48 characters and v) a paragraph which contains at least one term of the user-provided domain definition and with length of 740 characters.

[2] The Exporter module generates an XML file for each stored web document. This XML file contains metadata (e.g. language, domain, URL, etc.) about the corresponding document.
[3] Miquel Esplà-Gomis and Mikel L. Forcada. 2010. Combining Content-Based and URL-Based Heuristics to Harvest Aligned Bitexts from Multilingual Sites with Bitextor. The Prague Bulletin of Mathemathical Lingustics, 93:77--86.

The *language* feature is used to filter out pairs of files that are in the same language. Pages that have a depth difference above 1 are also filtered out, on the assumption that it is very likely that translations are found at the same or neighbouring depths of the web site tree.

Next, we extract the filenames of the images from HTML source and each document is represented as a list of image filenames. Since it is very likely that some images appear in many web pages, we count the occurrence frequency of each image and discard relatively frequent images (i.e. Facebook and Twitter icons, logos etc.) from the lists. In order to classify images into "critical" or "common" we need to calculate a threshold. In principle, one should expect that low/high frequencies correspond to "critical"/"common" images. We employ a nonparametric approach for estimating the probability density function. If the estimated probability density function has a main lobe in the low values, which corresponds to "critical" images, the threshold is chosen to be equal to the minimum just after this lobe. The underlining assumption is that if a web page in L1 contains image(s) then the web page with its translation in L2 will contain more or less the same images. In case this assumption is not valid for a multilingual site (i.e. there are only images that appear in all pages, e.g. template icons), probably all images will be included. To eliminate this, we discard images that exist in more than 10% of the total HTML files.

Following this step, each document is examined against all others and two documents are considered parallel if a) the ratio of their paragraph amounts (the ratio of their lengths in terms of paragraphs), b) the ratio of their clean text lengths (in terms of tokens), and c) the Jaccard similarity coefficient of their image lists, are higher than empirically predefined thresholds.

More pairs are detected by examining structure similarity. Since the XML files contain information about structure, content (i.e. titles, headings, list items) and domain specificity (i.e. paragraphs with the topic attribute), we use these files instead of examining the similarity of the HTML source.

A 3-dimensional feature vector is constructed for each candidate pair of parallel documents. The first element in this vector is the ratio of their fingerprint lengths, the second is the ratio of their sizes in paragraphs, and the third is the ratio of the edit distance of the fingerprints of the two documents to the maximum fingerprint length. Classification of a pair as parallel is achieved using a soft-margin polynomial Support Vector Machine trained with the positive and negative examples collected during previous work.

Since the Pair Detector does not use any language specific resources (i.e. it examines the structure and contents of the targeted multilingual web sites), we report the rate of accuracy (about 95%) that resulted from the manual evaluation[4] in prior data acquisition tasks. However, manual evaluation was also applied on the multilingual data delivered in QT-Launchpad and the results are presented in D.4.3.2 *Domain-specific and genre-specific multilingual data*.

[4] Papavassiliou, P. Prokopidis and G. Thurmair. A modular open-source focused crawler for mining monolingual and bilingual corpora from the web.6th Workshop on Building and Using Comparable Corpora. ACL 2013. Sofia

Recently, ILSP-FC was compared with Bitextor in automatically harvesting bitexts from multilingual websites[5]. Even though ILSP-FC and Bitextor adopt different strategies/tools in data harvesting (i.e. crawling vs site copying), cleaning (i.e. different methods of Boilerpipe), language identification (i.e. Cybozu vs LangID), de-duplication (near vs exact), the focus of this comparison was on their ability to detect pairs of parallel documents. The manual evaluation of a sample of the identified document pairs that were extracted from 21 multilingual websites showed that the crawlers performed precision rates of 90.76% and 94.79%, respectively. This difference is explained by the fact that ILSP-FC does not use any additional resources during pair detection while Bitextor requires a bilingual lexicon of the targeted languages.

3 Installation and use

Documentation on how to get, setup and run the ILSP-FC for acquiring multilingual data from the Web is available from http://nlp.ilsp.gr/redmine/projects/ilsp-fc/wiki/Documentation. Depending on user's requirements and system setup, there are several ways to get ILSP-FC:

- In case the user wants to run ILSP-FC directly from the command line, he can download the latest executable jar-with-dependencies jar file provided in the Files section of the site.
- In order to integrate ILSP-FC in his own Java project, the suggested way is to use Eclipse and Maven and follow the instructions at the Developer Setup page for further information. Otherwise (i.e. if the user does not want to use Maven), he can use the latest source-with-dependencies.zip file provided in the Files section of this site. It contains the source for ILSP-FC and jars for all third party libraries.

Once an ILSP-FC runnable jar has been built or downloaded, the user can run it by using the following example (see Appendix A for the list of URL seeds, i.e. the targeted websites, and Appendix B for the default configuration file). Note that in this example a topic definition is not used, since it is considered that the targeted website contains only "in-domain" data. In case that this assumption is not valid, a bilingual topic definition (which can be the union of the topic definitions in the targeted languages) should be used.

```
java -jar ilsp-fc-X.Y.Z-jar-with-dependencies.jar crawlandexport -a
qtlp -type p -l1 pt -l2 de -u PT-DE_AUTOMOTIVE_seedurl_list.txt -n 250
-f -k -t 20 -len 5 -mtlen 200 -of output_AUTOMOTIVE_DE-PT.txt -ofh
output_AUTOMOTIVE_DE-PT.txt.html -xslt -oxslt -cfg FBC_config.xml -u_r
"_de;_pt"
```

The following list explains the options of the ILSP-FC runnable jar:

```
-a :    user agent name (required)
```

[5] Esplà-Gomis, et. al (2014). Comparing Two Acquisition Systems for Automatically Building an English-Croatian Parallel Corpus from Multilingual Websites. In Proceedings of the Ninth International Conference on Language Resources and Evaluation. Reykjavik, Iceland.

```
-type :the type of crawling, which should be set to "p" for bilingual
crawls (required).

-cfg : the configuration file that will be used instead of the
default. A typical customized example is FBC_config.xml.

-c :   the crawl duration in minutes. Since the crawler runs in cycles
(during which links stored at the top of the crawler's frontier are
extracted and new links are examined) it is very likely that the
defined time will expire during a cycle run. Then, the crawler will
stop only after the end of the running cycle. The default value is 10
minutes.

-n :   the crawl duration in cycles. The default is 1.

-t :   the number of threads that will be used to fetch web pages in
parallel.

-f :   forces the crawler to start a new job (required).

-lang :the targeted language in case of monolingual crawling
(required).

-u :   the text file that contains the seed URLs (one URL per
textline) that will initialize the crawler.

-tc :  domain definition (a text file that contains a list of term
triplets that describe the targeted domain/subdomains). In case the
crawler is used for acquiring multilingual data, the list is the union
of the monolingual topic definitions.

If omitted, the crawl will be a "general" one (i.e. the module for
text-to-domain classification will not be used).

-k :   Forces the crawler to annotate boilerplate content in parsed
text.

-filter :    A regular expression to filter out URLs which do NOT
match this regex. Note that if this filter is used, only the seed URLs
that match this regex will be fetched.

-u_r : the replacements that might help crawler to detect pattern
matches when comparing the URLs.

-mtlen:Minimum number of tokens in cleaned document. If the length (in
terms of tokens) of the cleaned text is less than this value, the
document will not be stored.

-len:  Minimum number of tokens in a paragraph. If the length (in
terms of tokens) of a paragraph is less than this value, the paragraph
will be considered "out –of interest" and will be annotated properly.

-xslt :Insert a stylesheet for rendering xml results as html.

-oxslt:Export crawl results with the help of an xslt file for better
examination of results.

-dom:  Title of the targeted domain (required when domain definition,
i.e. tc parameter, is used).

-dest: the directory in which the results (i.e. the crawled data) will
be stored instead of the current directory.

-of:   a text file containing a list with the exported XML files.

-ofh:  an HTML file containing a list with the rendered XML files.
```

Appendix A

The targeted multilingual websites used for the construction of a DE-PT collection in the "Automotive" domain.

```
#http://www.castrol.com/pt_pt/portugal/sports/cars.html
#http://www.castrol.com/de_de/germany/sports/football.html
#http://www.volvocars.com/de/all-cars/volvo-s80/specifications/Pages/features.aspx
#http://www.volvocars.com/pt/all-cars/volvo-s80/specifications/Pages/features.aspx
#http://www.audi.de/de/brand/de.html
#http://www.audi.pt/pt/brand/pt.html
#http://www.benteler.de/
#http://www.benteler.pt/
#http://www.chevrolet.de/chevrolet-erleben/playliste-fuer-die-weihnachtszeit.html
#http://www.chevrolet.pt/descubra-a-chevrolet/lista-de-musicas-de-natal.html
#http://www.dunlop.eu/dunlop_dede/
#http://www.dunlop.eu/dunlop_ptpt/
#http://www.august-friedberg.com/produkte/automotive_por.asp
#http://www.august-friedberg.com/produkte/automotive_d.asp
#https://www.fulda.com/fulda_de_de/tires/passenger/ecocontrol-hp/index.jsp?from=browse
#https://www.fulda.com/fulda_pt_pt/tires/passenger/ecocontrol-hp/index.jsp?from=browse
#http://ec.europa.eu/transport/road_safety/topics/vehicles/index_pt.htm
#http://ec.europa.eu/transport/road_safety/topics/vehicles/index_de.htm
#http://ec.europa.eu/enterprise/sectors/automotive/index_de.htm
#http://ec.europa.eu/enterprise/sectors/automotive/index_pt.htm
#http://www.sava-tires.com/sava/pt/services/maintenance/
#http://www.sava-tires.com/sava/de/services/maintenance/
#http://www.sava-tires.com/sava/pt/
#http://www.sava-tires.com/sava/de/
#http://pt.euronews.com/2014/03/12/aumento-salarial-no-japao/
#http://pt.euronews.com/2014/03/12/justica-norte-americana-investiga-gm/
http://new.eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:92011E010341
http://new.eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=CELEX:92009E000553
http://new.eur-lex.europa.eu/legal-content/PT/TXT/HTML/?uri=CELEX:92011E010341
http://new.eur-lex.europa.eu/legal-content/PT/TXT/HTML/?uri=CELEX:92009E000553
```

Appendix B

The default configuration of the ILSP-FC in case of acquisition of bilingual data:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
 <agent>
   <email>yourmail@mail.com</email>
   <web_address>www.youraddress.com</web_address>
 </agent>
 <classifier>
   <min_content_terms>
     <value>2</value>
     <description>Minimum number of terms that must exist in clean
      content of each web page in order to be stored.
      This number is multiplied with the median value of
      the terms'weights and the result is the threshold
      for the absolute relevance score.
     </description>
   </min_content_terms>
   <min_unique_content_terms>
     <value>2</value>
     <description>Minimum unique terms that must exist in clean
      content
     </description>
   </min_unique_content_terms>
   <relative_relevance_threshold>
     <value>0.2</value>
     <description>The absolute relevance score is divided by the
      length (in terms of tokens) of the clean content of a document
      and the calculated relative relevance score is compared with
      this value
     </description>
   </relative_relevance_threshold>
   <max_depth>
     <value>10</value>
     <description>Maximum depth to crawl before abandoning a specific
      path. Depth is increased every time a link is extracted from a
      non-relevant web page.
     </description>
   </max_depth>
 </classifier>

<!-- The module for segment alignment was not integrated into the
acquisition workflow when the data acquistion was obtained.
 <aligner>
   <win_align_path>
     <value>hunalign-1.1/win/hunalign.exe</value>
     <description>relative path to executable of hunalign for
      windows. The main hugnalign directory is supposed to be next
      to the crawler's jar
     </description>
   </win_align_path>
   <lin_align_path>
```

```
      <value>hunalign-1.1/linux/src/hunalign/hunalign</value>
      <description>relative path to executable of hunalign for linux.
       The main hugnalign directory is supposed to be next to the
       crawler's jar
      </description>
    </lin_align_path>
    <align_dict>
      <value>hunalign-1.1/dict</value>
      <description>relative path to the dictionaries of hunalign.
       The main hugnalign directory is supposed to be next to the
       crawler's jar
      </description>
    </align_dict>
  </aligner>
-->

  <fetcher>
    <fetch_buffer_size>
      <description>Max number of urls to fetch per run</description>
      <value>512</value>
    </fetch_buffer_size>
    <socket_timeout>
      <value>10000</value>
      <description>Socket timeout in milliseconds(per
       URL)
      </description>
    </socket_timeout>
    <connection_timeout>
      <value>10000</value>
      <description>Connection timeout in milliseconds(per
       URL)
      </description>
    </connection_timeout>
    <max_retry_count>
      <value>2</value>
      <description>Max number of attempts to fetch a Web page before
       giving up
      </description>
    </max_retry_count>
    <min_response_rate>
      <value>0</value>
      <description>Min bytes-per-seconds for fetching a web page
      </description>
    </min_response_rate>
    <valid_mime_types>
      <mime_type value="text/html" />
      <mime_type value="text/plain" />
      <mime_type value="application/xhtml+xml" />
      <!--<mime_type value="application/pdf" />
      <mime_type value="application/x-pdf" /> -->
      <description>Accepted mime types</description>
    </valid_mime_types>
```

```
    <crawl_delay>
      <value>1500</value>
      <description>delay in milliseconds between requests
      </description>
    </crawl_delay>
    <max_content_size>
      <value>531072</value>
      <description>Max content size (bytes) for downloading a web
       page
      </description>
    </max_content_size>
    <max_requests_per_run>
      <value>512</value>
      <description>Max fetch set size per run (Sets are made by URLs
       from the same host)
      </description>
    </max_requests_per_run>
    <max_requests_per_host_per_run>
      <value>512</value>
      <description>Max URLs from a specific host per run</description>
    </max_requests_per_host_per_run>
    <max_connections_per_host>
      <value>100</value>
      <description>Max number of fetching threads for each host
      </description>
    </max_connections_per_host>
    <max_fetched_per_host>
      <value>10000000</value>
      <description>Max web pages to fetch per host</description>
    </max_fetched_per_host>
    <max_redirects>
      <value>5</value>
      <descriptions>Max number of redirects</descriptions>
    </max_redirects>
    <request_timeout>
      <value>600000</value>
      <description>Max time to wait for Fetcher to get all URLs in a
       run
      </description>
    </request_timeout>
  </fetcher>
</configuration>
```