

SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information and communication Technologies

PANACEA Project

Grant Agreement no.: 248064

**Platform for Automatic, Normalized Annotation and
Cost-Effective Acquisition**
of Language Resources for Human Language Technologies

D-4.2: Initial functional prototype and documentation describing the initial CAA subsystem and its components

Dissemination Level: Public

Delivery Date: February 1, 2011

Status – Version: v1.0

Author(s) and Affiliation: Prokopis Prokopidis (ILSP), Vassilis Papavassiliou (ILSP),
Antonio Toral (DCU), Victoria Arranz (ELDA), Núria Bel
(LIPE)

Relevant PANACEA Deliverables

- D3.1** Architecture and Design of the Platform (T6)
- D4.1** Technologies and tools for corpus creation, normalization and annotation (T6)
- D4.3** Monolingual corpus acquired in five languages and two domains (due T13)
- D7.1** Criteria for evaluation of resources, technology and integration (T6)

D4.2. Initial functional prototype and documentation describing the initial CAA subsystem and its components

This document is part of technical documentation generated in the PANACEA Project, Platform for Automatic, Normalized Annotation and Cost-Effective Acquisition (Grant Agreement no. 248064).



This document is licensed under a Creative Commons Attribution 3.0 Spain License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/es/>.

Please send feedback and questions on this document to: iulatri@upf.edu

TRL Group (Tecnologies dels Recursos Lingüístics), Institut Universitari de Lingüística Aplicada, Universitat Pompeu Fabra (IULA-UPF)

Table of contents

1	Introduction	2
2	Terminology	2
3	Web services	3
3.1	Focused monolingual crawler	3
3.1.1	Mandatory and optional parameters	3
3.1.2	Crawling for language- and domain-specific data.....	6
3.2	Focused bilingual crawler	10
3.2.1	Mandatory and optional parameters	10
3.2.2	Crawling for bilingual domain-specific data.....	10
3.3	Boilerplate remover.....	14
4	Traveling Object tools.....	15
5	Conclusions and Workplan	16
6	References	17
	Appendix.....	19
A.	An English cesDoc document	19
B.	A cesAlign document pointing to an EN-FR pair of cesDoc documents.....	20
C.	Soaplab2 ACD configuration file for the Focused Monolingual Crawler.....	21
D.	Soaplab2 ACD configuration file for the Focused Bilingual Crawler	23
E.	Soaplab2 ACD configuration file for the Boilerplate Remover.....	25

1 Introduction

PANACEA WP4 targets on the creation of a Corpus Acquisition and Annotation (CAA) subsystem for the acquisition and processing of monolingual and bilingual language resources (LRs) required in the PANACEA context.

This document focuses on the development and integration of the first version of the CAA subsystem in the PANACEA platform¹. This version incorporates a Corpus Acquisition Component (CAC) and a Cleanup and Normalization Component (CNC) as planned in Section 7 of D4.1 *Technologies and tools for corpus creation, normalization and annotation*. The present deliverable, together with D4.3, constitutes the second milestone of WP4.

We present the terminology used in this document in Section 2. The CAC and CNC modules and their deployment as web services are discussed in Section 3. In Section 4, we discuss some tools regarding the generation and presentation of the XML output of the services. Finally, we conclude and sketch future work in Section 5.

2 Terminology

This section defines common terminology used in the rest of this document.

Corpus (or text corpus): a (large) set of texts. In PANACEA, we assume the texts are stored electronically, in a given file format and character encoding, without any formatting information, eventually provided with metadata and/or linguistic annotation. Often, the texts are referred to as documents, in which case the texts are assumed to be topic-coherent.

Monolingual corpus: a corpus of texts in one language.

Bilingual corpus: a corpus of texts in two languages.

Parallel corpus: a bilingual corpus consisting of texts organized in pairs which are translations of each other, i.e. they include the same information (parallel texts). Usually, the pairs are identified at least for documents (parallel documents) and the corpus described as document-aligned parallel corpus. If the translation pairs are identified also for sentences (parallel sentences) we talk about sentence-aligned parallel corpus.

Comparable corpus: a bilingual corpus consisting of texts organized in pairs (comparable documents) which are only approximate translations of each other, i.e. they include similar information.

Web Crawler: a computer program that browses the World Wide Web in a methodical and automated manner in order to copy/store web documents (html pages, pdf documents, etc.) for later processing (e.g. indexing, creating corpora, etc.) In the initial version of the crawlers to be developed in WP4.1, the acquired corpora will consist only of html pages. In the context of this report, web documents, web pages and html pages are synonymous.

Focused web crawler: is a web crawler that downloads html pages that are relevant to a predefined topic in order to build topic-specific web collections. In the context of PANACEA,

¹ Intellectual Property Rights issues are not discussed in this deliverable, as they are currently being handled in the context of WP2 *Dissemination and Exploitation*.

we aim to build a domain-specific crawler that will crawl for data on the automotive, legal and environment domains.

Seed pages: Web pages known to be relevant to a specific domain. A (focused) web crawler will be initialized with these pages.

3 Web services

This section describes the CAC and CNC modules and their deployment as web services ready to be integrated into the PANACEA platform. We present three tools and their corresponding services: the Focused Monolingual Crawler (FMC), the Focused Bilingual Crawler (FBC) and the boilerplate remover (BR). The requirements for the first version of the CAA subsystem, as reported in D7.1, are: i) Req-TEC-0101a (Components accessibility), ii) Req-TEC-0104 (Common interface compliant), iii) Req-TEC-0105 (Metadata description), iv) Req-TEC-0106 (Format compliant), v) Req-TEC-0108 (Error handling), vi) Req-TEC-0109 (Temporary data), vii) Req-TEC-1101 (Input/output proprietary data management) and viii) Req-TEC-0110 (Data Transfer). The crawling web services respect the initial design of the common interfaces in D3.1, in particular Section 9.2.7.2 *Crawling*.

3.1 Focused monolingual crawler

The FMC is the first module in the PANACEA pipeline for building LR by crawling web documents with rich textual content. Its purpose is to adapt an efficient and distributed web crawling methodology that will collect web pages with content belonging to specific languages and predefined domains. The common strategy adopted by a general web crawler is to initialize the crawler by the seed pages, visit these pages and extract the links within them. Then new web pages are visited following the extracted links and so on. In focused crawling, a text to topic classifier is included in order to classify each page as relevant to the domain or not.

3.1.1 Mandatory and optional parameters

The FMC has been deployed as a web service in a Tomcat 6 web server hosting a Soaplab2² web application accessible from <http://sifnos.ilsp.gr:8888/soaplab2-axis/>. The FMC is made available under the name *ilsp_mono_crawl*. The web interface of this tool is presented in fig. 1. It contains four mandatory parameters:

1. The *Domain* parameter corresponds to a descriptive title for the crawler's job (e.g. LAB_EN_TEST_017). At this phase of the project, it also defines the name of a MySQL relational database that is created automatically, and in which the whole data (e.g. downloaded pages, extracted links, relevant scores, logs, etc.) are stored. It is worth mentioning that the *Domain* value may not be a mandatory parameter in the next version of the crawler, since the database name can be equivalent to the job name assigned automatically by the administrative framework of the PANACEA platform.
2. The *LanguagesList* is required for the definition of the targeted language. At this time supported languages are English, French, German, Greek, Italian and Spanish.
3. The *TermList* is a list of triplets (<relevance weight, term, topic-class>) that define the domain, as shown in Figure 1. Terms can be single words or phrases relevant to the domain.

² Soaplab2, <http://soaplab.sourceforge.net/soaplab2/>, is a tool that can automatically generate and deploy web services on top of existing command-line analysis programs.

Weights are signed integers and indicate the relevance of the term with respect to the topic-classes. A score of relevance S for a web page is calculated by the following equation:

$$S = \sum_{i=1}^N \sum_{j=1}^4 n_{ij} \cdot w_i \cdot w_j \quad (1)$$

where N is the amount of terms in the topic definition, w_i is the weight of term i , w_j is the weight of location j and n_{ij} denotes the number of occurrences of term i in location j . The four discrete locations in a web page are *title*, *metadata*, *keywords*, and *plain text*. The corresponding weights for these locations are 10, 4, 2, and 1. If the score is greater than a predefined threshold, the web page will be classified as relevant to the specific domain.

For this version of the FMC, the predefined threshold is 90, while a default weight for a relevant term is 100. This implies that just one occurrence of a relevant term in a page results in classifying the page as relevant and, therefore, in guiding the FMC to extract and follow the links of the page. Higher or lower weights indicate more or less relevant terms, respectively. Negative values (e.g. -100) assigned to terms can be used to lower the scores of pages containing these terms.

Topic-classes are sub-categories of the targeted domain. For example, “working conditions” and “wages” could be two sub-classes of the “Labour Legislation” domain.

The topic definition can be provided as a URL which points to an appropriate file with term triplets. Alternatively, it can be provided as direct data by a user who can complete the corresponding field manually or upload an already existing file. The topic definition should contain one triplet per line and be encoded in UTF-8. An extract from a topic definition for “Environment” in English is provided below:

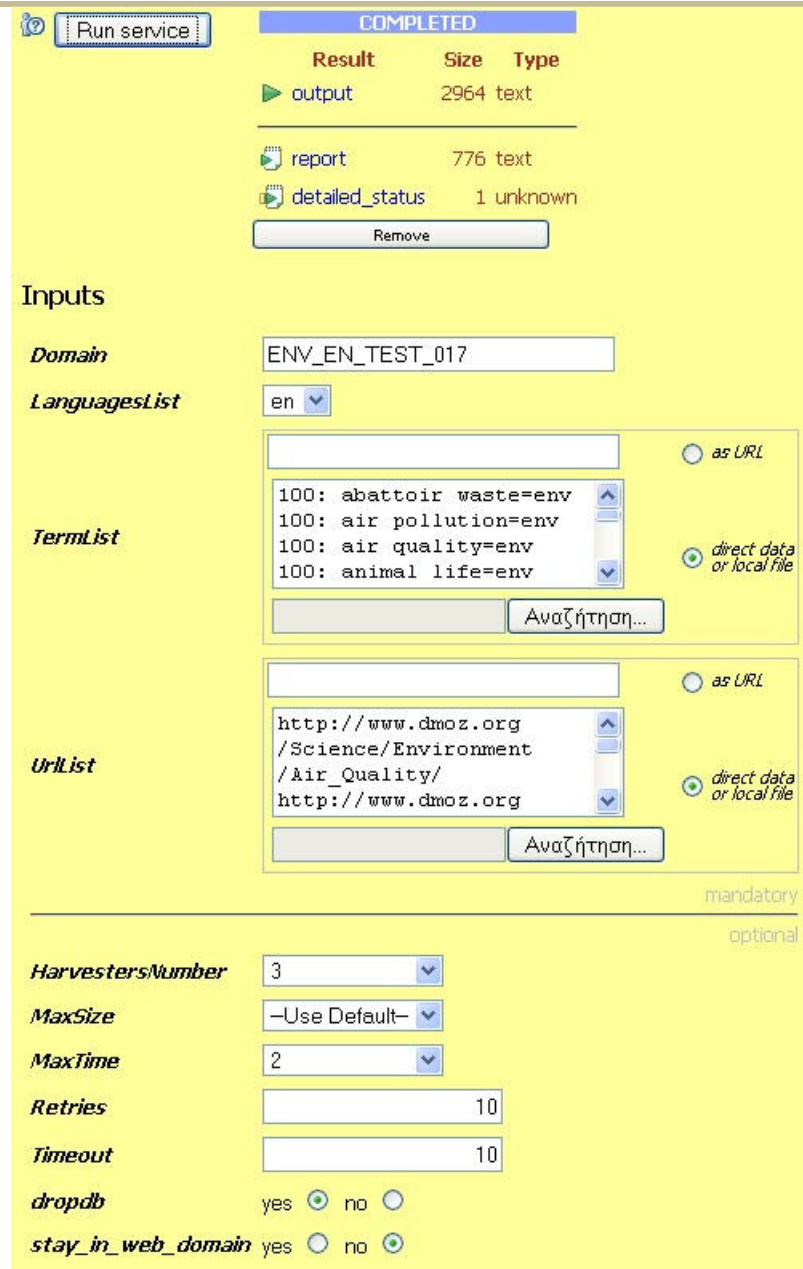
```
100: air pollution=environment_EN
100: biodiversity=environment_EN
100: climate change=environment_EN
```

More details about constructing such topic definitions are reported in Section 4.1 of D4.3. Moreover, some samples of topic definitions for the “Environment” and “Labour Legislation” domains in the languages targeted by PANACEA are included in the monolingual corpora of D4.3.

4. The *UrlList* is a list of seed URLs with which the crawler is initialized. These URLs should be relevant to the domain (i.e. contain positively-weighted terms from the topic). Since the seed pages are assumed to be relevant, the crawler extracts their links and follows these links to visit/retrieve new pages. Similarly to the topic definition, the *UrlList* can be provided as a URL, as direct data or as an existing file upload. The *UrlList* should contain one URL per line. More details about constructing a list of seed URLs are reported in Section 4.2 of D4.3. Again, some samples of seed URLs for the “Environment” and “Labour Legislation” domains in the languages targeted by PANACEA are included in the monolingual corpora of D4.3.

There are also some optional parameters, which can be used to configure the crawl job. At this first version of the FMC, the optional parameters are:

1. *HarvestersNumber*: Defines the number of crawler threads to be used for a specific job. For example, if *HarvestersNumber* is defined to 5, five crawler threads will be working in parallel on the job. This parameter affects the performance of the job, since a simple way of increasing performance is to use more than one crawler for a job.
2. *MaxSize*: Indicates the maximum size of data (in MBs) that a job has to download. Therefore, the crawl job will stop after *MaxSize* MBs have been stored in the database. The default value is 1 MB. This option is disabled for users accessing the service via the Soaplab2 web application.
3. *MaxTime*: Indicates the maximum time (in minutes) that a crawl job has to run. Thus, the crawl job will stop after running for *MaxTime* minutes. The default value is 1. Users accessing the service via the web application can initiate crawl jobs lasting for 10 minutes maximum.
4. *Retries*: Defines the maximum number of http requests that a crawler will make in order to fetch a web document. If there is no response from the server after *Retries* times, the web document will be skipped. The default value is 10. This option is disabled for users accessing the service via the web application.
5. *Timeout*: Indicates the time (in seconds) that a crawler has to wait for a server's response. The default value is 10 seconds. This option is disabled for users accessing the service via the web application.
6. *dropdb*: If set to true, this boolean parameter guides the module to delete the database schema of the crawl job after storing the results of the crawl in the filesystem. If false, the database schema is deleted after two days by a cron job.
7. *stay_in_web_domain*: If set to true, this boolean parameter guides the module to download data only from web domains in the *UrlList*. The true value is used mainly for demo purposes, while the default value is false.



Run service

COMPLETED

Result	Size	Type
output	2964	text
report	776	text
detailed_status	1	unknown

Inputs

Domain ENV_EN_TEST_017

LanguagesList en

TermList

100: abattoir waste=env
100: air pollution=env
100: air quality=env
100: animal life=env

UrlList

http://www.dmoz.org/Science/Environment/Air_Quality/
http://www.dmoz.org

optional

HarvestersNumber 3

MaxSize -Use Default-

MaxTime 2

Retries 10

Timeout 10

dropdb yes ☒ no ☐

stay_in_web_domain yes ☐ no ☒

Figure 1 The web interface of the focused monolingual crawler

3.1.2 Crawling for language- and domain-specific data

Following the solution path reported in Section 7.1 of D4.1, we adopted and modified the Combine crawler [Ardo, 2005] for this task. Combine³ incorporates modules which are required in focused crawling. Moreover, it is a modular and open-source tool that allows monitoring of the crawl progress by logging its actions in a relational database.

The first element in the pipeline of the crawl concerns the format detection and the conversion of character encoding to UTF-8. Based on the *content_type* header of the HTTP response, Combine detects the format of each visited web document. In this initial version of FMC, the crawler targets HTML pages only and discards documents in other formats. The next version of

³ <http://combine.it.lth.se/documentation/>

FMC will incorporate modules for converting other formats into a unified format (plain text). For encoding identification, Combine utilizes the *content_charset* header of the HTTP response, which indicates the character encoding of a visited web document. If needed, Combine employs the Encode Perl library⁴ to convert the encoding in UTF-8.

The second module in the pipeline is a language identifier. *Lingua:Identify*⁵, an open-source and flexible language identifier based on n-grams and implemented in Perl, is used for this task. As a result, documents that are not in the targeted language are discarded. Since *Lingua:Identify* did not support Greek, we contributed data to the tool developer, who generated a language fingerprint for Greek.

A significant element of the FMC is a text to topic classifier based on key terms or word n-grams. Such algorithms are widely used in web page classification [Qi and Davison, 2009]. Combine exploits a string-to-string matching algorithm for calculating a score of relevance for each visited page [Golub and Ardo, 2005]. In FMC, the score is calculated as described in 3.1.1 above. If this score overcomes a predefined threshold, the web page is classified as relevant to the domain and the HTML content of the web page is stored in the relational database. Then, the links within the stored page are extracted and the score of this page is assigned to these links. The crawler uses these links to visit new pages.

Combine, in its default implementation, extracts links only from pages that have been classified as relevant. In our initial experiments with FMC and in order to collect a lot of links for the process of the crawl, we set the threshold equal to 90, a relatively low value. Given that the weight of each term in our term lists was 100 (see Section 3.1.1), the selected value of the threshold was so low that even when a page contained just one relevant term, the page was considered relevant. In the next version of the FMC, we plan to combine the Best-First algorithm with the Tunnelling technique [D. Bergmark et al., 2002], as described in section 7 of D4.1. According to this technique, links will be extracted from both relevant and non-relevant pages, while only relevant pages will be stored.

One critical issue is the fact that Combine, in its current implementation, does not follow the most promising URLs (i.e. links within pages with high relevance to the domain), but considers the list of extracted URLs as a First-In First-Out (FIFO) queue. In other words, Combine employs the breadth-first algorithm [Pinkerton, 1994], which sorts the URLs to be visited with respect to the time they were extracted. It was proved [Dorado, 2008, Menczer et al., 2004] that Best-First [Cho et al., 1998] and InfoSpiders [Menczer and Belew, 2000] seem to be the most effective methods for focused crawling. Since InfoSpiders displays a disadvantage at the early stage of the crawling process (when the neural networks are not trained yet), we decided to exploit the Best-First algorithm and we modified Combine accordingly. The Best-First algorithm sorts the URLs with respect to their scores and selects a predefined number of the most promising URLs. In the next period of the project, we aim to introduce a method for estimating the number of the promising links instead of using a predefined number. Specifically, we plan to investigate (in cooperation with the ILSP team of the FP7 ICT ACCURAT project⁶) if the adoption of an unsupervised machine learning method (like, for example, K-means

⁴ <http://search.cpan.org/dist/Encode/>

⁵ <http://search.cpan.org/~ambs/Lingua-Identify-0.26/README>

⁶ <http://www accurat-project.eu/>

[MacQueen, 1967]) could be beneficial for classifying, the candidate URLs as “promising” or not.

The next processing step in the FMC pipeline concerns web page cleaning. For this task we integrated the Boilerplate tool (see also Section 3.3).

In order to improve the quality of the LRs delivered by the CAC to downstream tools and applications, a crawler should incorporate a stage for duplicate detection. Following the PANACEA Description of Work document, we will include a duplicate detection module in the second version of FMC (D4.4 due T21). In addition, we aim to integrate a dedicated web service for duplicate detection into the PANACEA platform.

The functionalities of the FMC are made available in the Soaplab2 server via a Soaplab2 ACD configuration file. The file, via which all mandatory and optional parameters mentioned above are defined, can be examined in Appendix C.

A workflow for the FMC has been developed and tested in the Taverna workflow management system⁷. Figure 2 presents the workflow with input ports for all mandatory and two optional (HarvestersNumber and MaxTime) parameters. The output of the FMC web service and workflow is a text file pointing to *CesDoc* documents with text segmented in paragraphs (c.f. Appendix A) and basic metadata as described in Section 6.1.3 of D3.1.

⁷ <http://www.taverna.org.uk/>

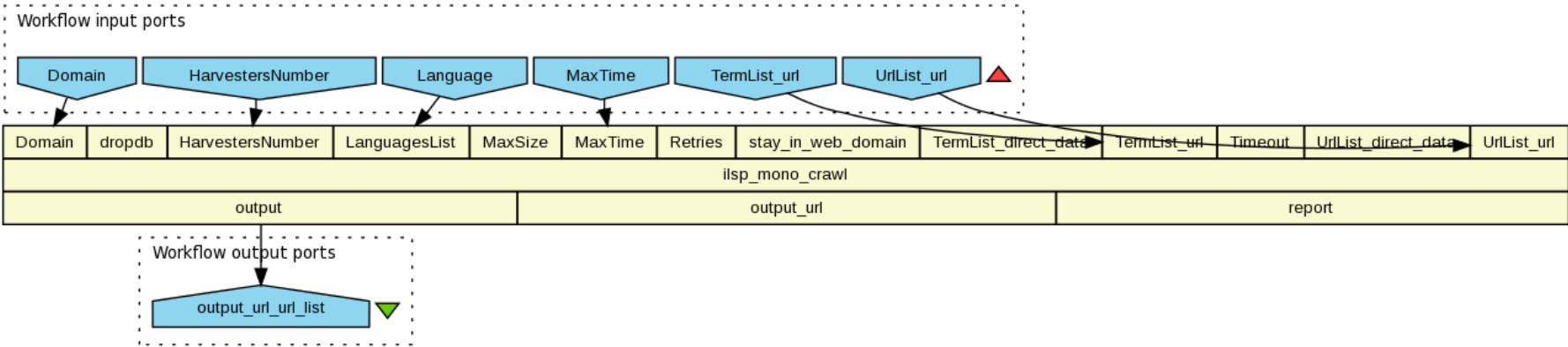


Figure 2 Focused monolingual crawler in Taverna

3.2 Focused bilingual crawler

The FBC is the first module in the PANACEA pipeline for building parallel LR from the web. It aims to find multilingual websites and downloads web documents that are relevant to a predefined domain and in the targeted languages. At this initial version, the FBC starts from an initial URL, and in a spider-like mode finds the links within these pages pointing to pages inside the same web site, visits the new pages and so on. Therefore, FBC can be considered as a combination of a website copier (i.e. a tool that locally mirrors the structure of a web site), a text to topic classifier, and a module that extracts pairs of web documents in different languages.

The FBC has been deployed as a web-service accessible from the <http://sifnos.ilsp.gr:8888/soaplab2-axis/> Soaplab2 web application under the name *ilsp_bilingual_crawl*. The Soaplab2 web interface of this tool is presented in Figure 3.

3.2.1 Mandatory and optional parameters

The FBC has common mandatory and optional parameters with the FMC (cf. Section 3.1.1 of this document). The main differences are that i) two languages are required (parameters *Language1* and *Language2*), ii) the *TermList* has to include terms in both languages and iii) the *UrlList* must contain only one URL. This user-provided URL should point to a multilingual site.

3.2.2 Crawling for bilingual domain-specific data

The current version of FBC incorporates the same processing stages with FMC (c.f. Section 3.1.2). The FBC follows links originating from the URL provided by the user, and it stores web pages relevant to the bilingual topic definition provided again by the user.

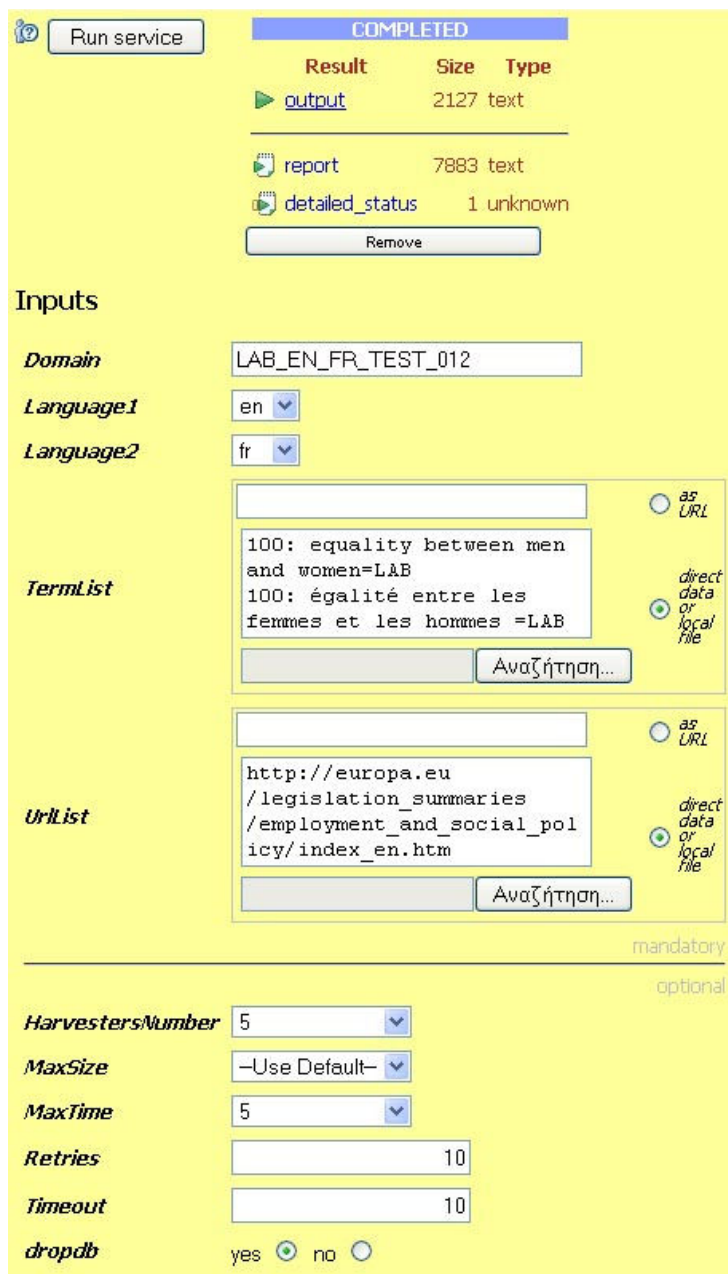
The bilingual topic definition should contain one triplet (< relevance weight, term, topic-class>) per line and be encoded in UTF-8. An extract from a bilingual topic definition for “Labour Legislation” in English and French is provided below:

```
100: collective agreement=LAB_EN_FR
100: work contract=LAB_EN_FR
100: trade union rights=LAB_EN_FR
100: trade union confederation=LAB_EN_FR
100: trade union=LAB_EN_FR
100: continuous working day=LAB_EN_FR
100: convention collective=LAB_EN_FR
100: contrat de travail=LAB_EN_FR
100: droits syndicaux=LAB_EN_FR
100: confédération syndicale=LAB_EN_FR
100: syndicat=LAB_EN_FR
100: journée continue=LAB_EN_FR
```

The following step of the FBC concerns examining the pool of stored HTML pages and deciding which pages can be considered as pairs from which parallel sentences can be extracted. Following the solution path reported in Section 7.2 of D4.1 we employed Bitextor⁸, for this task. Bitextor detects such pairs by exploiting a set of measures like: a) relative difference in file size, b) relative difference in length of plain text, c) edit distance of the HTML structures and d) edit

⁸ <http://sourceforge.net/projects/bitextor/>

distance of the lists of numbers contained in the stored pages [Esplà-Gomis and Forcada, 2010]. In the next FBC version (D4.4 in T21) we aim to add measures concerning the relations of the terms found in the documents (i.e. examining if the terms found in a document are translations of the terms found in another document).



Run service

COMPLETED

Result	Size	Type
output	2127	text
report	7883	text
detailed_status	1	unknown

[Remove](#)

Inputs

Domain

Language1

Language2

TermList

☐ as URL
☒ direct data or local file

[Αναζήτηση...](#)

UrlList

☐ as URL
☒ direct data or local file

[Αναζήτηση...](#)

mandatory

optional

HarvestersNumber

MaxSize

MaxTime

Retries

Timeout

dropdb ☒ yes ☐ no

Figure 3 The web interface of the focused bilingual crawler

The functionalities of the FMC are made available in the Soapplab2 server via a Soapplab2 ACD configuration file. The file, via which all mandatory and optional parameters are defined, can be examined in Appendix D.

A workflow for the FBC has been developed and tested in the Taverna workflow management system⁹. Figure 4 presents the workflow with input ports for all mandatory and two optional (HarvestersNumber and MaxTime) parameters. The output of the FBC and its corresponding web service and workflow is a URL containing a list of links to *cesAlign* documents (see Appendix B for an example). Each *cesAlign* document points to a pair of *cesDoc* documents in the targeted languages.

⁹ <http://www.taverna.org.uk/>

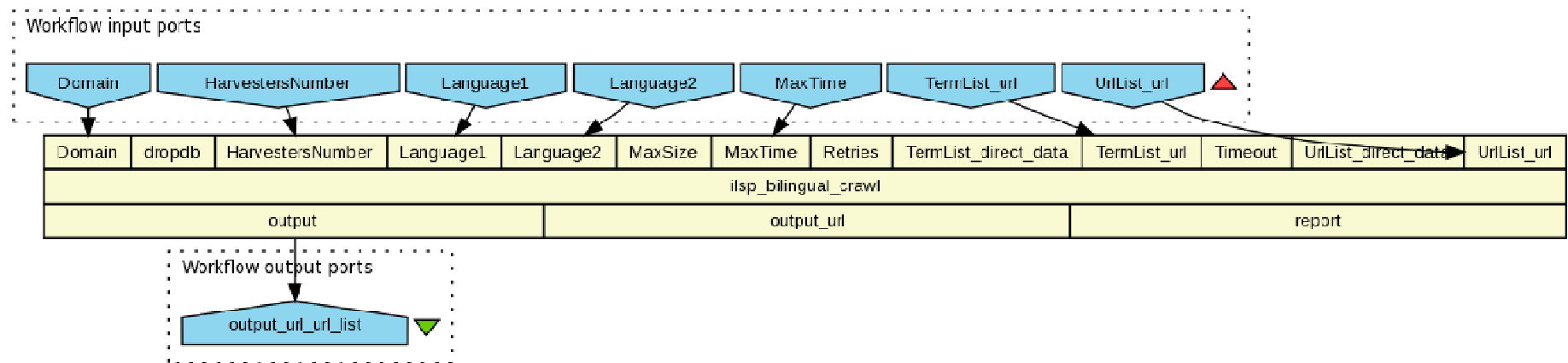
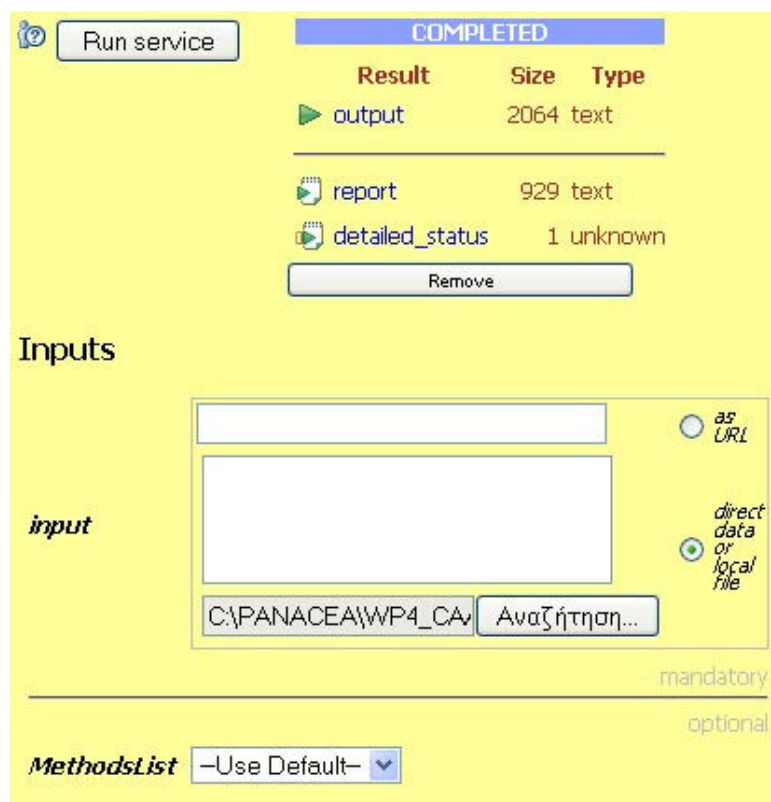


Figure 4 Focused bilingual crawler in Taverna

3.3 Boilerplate remover

The BR aims to detect and remove boilerplate text that typically is not related to the main content (e.g. navigation links, advertisements, disclaimers, etc.) from a web document. For this task we employed the Boilerpipe¹⁰ tool which uses only a small set of shallow text features (e.g. number of words and link density) for classifying the individual text elements in a Web page [Kohlschütter et al., 2010]. Boilerpipe provides six methods for removing boilerplate (ArticleExtractor, ArticleSentencesExtractor, DefaultExtractor, KeepEverythingExtractor, LargestContentExtractor, NumWordsRulesExtractor). These methods either use different features or exploit different classification algorithms. Short descriptions of the methods are reported on <http://boilerpipe.googlecode.com/svn/trunk/boilerpipe-core/javadoc/1.0/index.html>.

The web-service of the BR is available in <http://sifnos.ilsp.gr:8888/soaplab2-axis/> under the name *boilerplate_remover*. The web interface of this tool is illustrated in Figure 5. The input of the BR is a web document to be cleaned. The user can provide input as a URL or as an already stored file in HTML format. A method can be selected from *MethodsList* combo box. The default method is the third (DefaultExtractor). The output of the BR is the cleaned text file.



Result	Size	Type
output	2064	text
report	929	text
detailed_status	1	unknown

Remove

Inputs

input

as URL

direct data or local file

mandatory

optional

MethodsList -Use Default-

Figure 5 The web interface of the boilerplate remover

The functionalities of the BR are made available in the Soaplab2 server via a Soaplab2 ACD configuration file, which can be examined in Appendix E.

¹⁰ <http://code.google.com/p/boilerpipe/>

4 Traveling Object tools

This section discusses some tools regarding the generation and presentation of the XML output of the crawling services, which in PANACEA lingo is also called the Traveling Object. ILSP has developed a Java tool called FormatConverter that can convert data from and to the Traveling Object formats defined in D3.1. The FormatConverter tool uses the Factory Object-Oriented design pattern¹¹, and consists of two interfaces to classes that can be sub-classed to cater for new input and output formats.

FormatConverter has been distributed among PANACEA partners, who have already extended and adapted it to their tools and web services (see, for example, D5.2 Section 3). For example, the tool has been integrated in the monolingual and bilingual pipelines, to take care of converting content and metadata extracted from crawled pages, to the “basic” version of the Traveling Object (D3.1, Section 6.1.2). The sample in Appendix A is an example of such a conversion.

ILSP has also developed an XSLT¹² stylesheet for easy viewing of “basic” Traveling Object files. When such a file is loaded on a web browser, the current version of the stylesheet allows a user to examine the document’s metadata in a separate page (Figure 6), and copy the text of whole paragraphs to the clipboard (Figure 7).

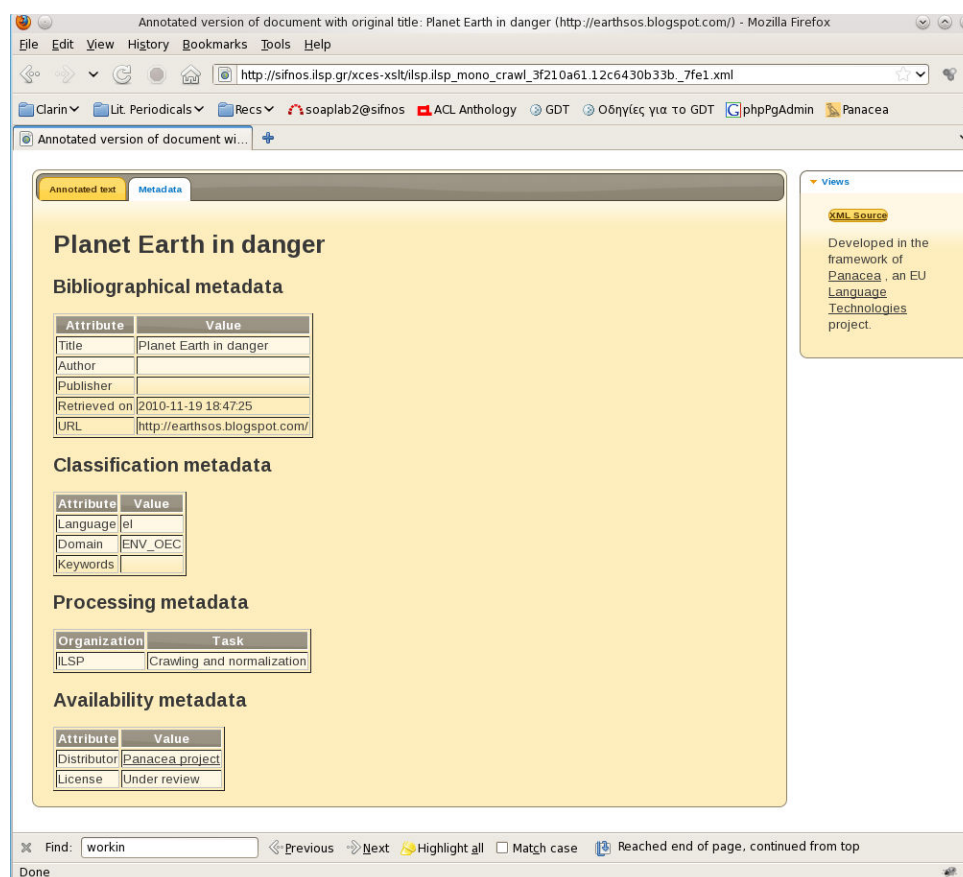


Figure 6 Examining the metadata of a basic TO file in a web browser

¹¹ http://en.wikipedia.org/wiki/Factory_method_pattern

¹² <http://www.w3.org/Style/XSL/>

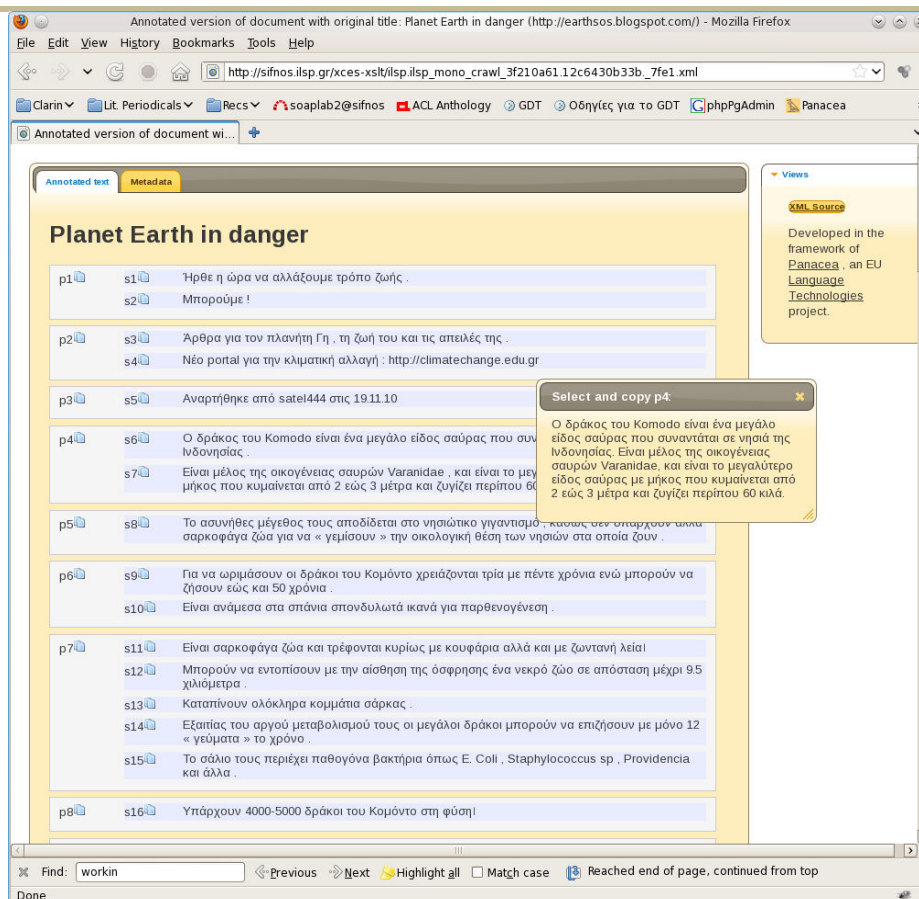


Figure 7 Examining the contents of a basic TO file in a web browser

5 Conclusions and Workplan

In this deliverable we have documented the initial functional prototype of the CAA subsystem. This initial prototype consists of a web service for monolingual and a web service for bilingual focused web crawling. The two web services include modules for language identification and boilerplate removal. A third web service caters for boilerplate removal. The integration of these web services into the PANACEA platform complies with the PANACEA Description of Work document and the solution path detailed in D4.1.

The web services will be evaluated in WP7.2 *Evaluation of the Integration of components*. Based on the evaluation results we aim to implement a revised version of the CAA subsystem. This new version will integrate a web service for duplicate detection and the new version of the web service for boilerplate removal.

The PANACEA platform will also include NLP modules for all languages targeted by the project. In the context of the WP4.3 task, partners have already started adapting existing NLP tools and making them available as web services according to the functionalities and the timetable in D4.1, Table 8.

In more detail, the workplan for the CAA development in the context of WP4 will include the tasks sketched below:

- T18: **Internal deliverable.** All partners adapt NLP tools focusing on sentence splitting/tokenization and POS tagging/lemmatization for EN, DE, EL, ES, IT, FR. The I/O of all tools will be conformant with the common encoding format documented in *D3.1 Architecture and Design of the Platform*.
- T21: **D4.4.** 2nd version of the prototype and documentation. The revised prototype will integrate dedicated web services for normalization (including boilerplate removal and duplicate document detection), as required for *D7.3 Second evaluation report* (T22).
- T22: **Internal deliverable.** Partners adapt NLP tools focusing on parsing and/or chunking for DE, EN, EL, ES, IT, FR. The I/O of all tools will be conformant with the common encoding format documented in *D3.1 Architecture and Design of the Platform*. These tools will be part of the final version of the CAA subsystem.
- T29: **D4.5.** Final version of the prototype and documentation. The revised prototype will integrate NLP tools for a) sentence splitting/tokenization b) POS tagging/lemmatization and c) parsing for EN, DE, EL, ES, IT, FR, as required for *D7.4 Third evaluation report* (T30).

6 References

- Ardo, A. 2005. Combine web crawler, Software package for general and focused Web-crawling, <http://combine.it.lth.se/>.
- Ardo, A., and Golub, K. 2007. Documentation for the Combine (focused) crawling system, <http://combine.it.lth.se/documentation/DocMain/>
- Baroni, M., and Bernardini, S. 2004. BootCaT: Bootstrapping corpora and terms from the web. In Proceedings of LREC 2004. 1313-1316.
- Baroni, M., Kilgarriff, A., Pomikalek J., and Rychly. P. 2006. WebBootCaT: a web tool for instant corpora. In Proceedings of Euralex 2006. 123-132.
- Bergmark D., Lagoze C. and Sbityakov A. 2002. Focused crawls, tunneling, and digital libraries. In Proc. of the 6th European Conference on Research and Advanced Technology for Digital Libraries, pp. 91-106.
- Cho, J., Garcia-Molina, H., and Page, L. 1998. Efficient crawling through URL ordering, Computer Networks and ISDN Systems. 30, 1-7, 161-172.
- Dorado, I. G. 2008. Focused Crawling: algorithm survey and new approaches with a manual analysis. Master thesis.
- Espla-Gomis, M., and Forcada, M.L. 2010. Combining Content-Based and URL-Based Heuristics to Harvest Aligned Bitexts from Multilingual Sites with Bitextor. The Prague Bulletin of Mathematical Linguistics. 93, 77-86.
- Golub, K. and Ardo, A. 2005. Importance of HTML structural elements and metadata in automated subject classification. In Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL). Lecture Notes in Computer Science, vol. 3652. Springer, Berlin, pp. 368-378.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. 2010. Boilerplate Detection using Shallow Text Features. The Third ACM International Conference on Web Search and Data Mining

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In L. M. LeCam and J. Neyman (Eds.), Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume I, pp. 281-297. University of California Press.

Menczer, F., Pant, G. and Srinivasan, P. 2004. Topical Web Crawlers: Evaluating Adaptive Algorithms, ACM Transactions on Internet Technology, Vol. 4, No. 4, pp. 378–419.

Pinkerton, B. 1994. Finding what people want: Experiences with the Web Crawler. In Proceedings of the 2nd International World Wide Web Conference.

Theobald, M., Siddharth, J., and Paepcke, A. 2008. SpotSigs: Robust and Efficient Near Duplicate Detection in Large Web Collections. In: 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2008).

Appendix

A. An English cesDoc document

```
<?xml version='1.0' encoding='UTF-8'?>
<cesDoc version="0.4">
  <cesHeader version="0.4">
    <fileDesc>
      <titleStmt>
        <title>Glacial Retreat</title>
        <respStmt>
          <resp>
            <type>Crawling and normalization</type>
            <name>ILSP</name>
          </resp>
        </respStmt>
      </titleStmt>
      <publicationStmt>
        <distributor>Panacea project</distributor>
        <eAddress type="web">http://www.panacea-lr.eu</eAddress>
        <availability>Under review</availability>
        <pubDate>2012</pubDate>
      </publicationStmt>
      <sourceDesc>
        <biblStruct>
          <monogr>
            <title>Glacial Retreat</title>
            <author></author>
            <imprint>
              <publisher></publisher>
              <pubDate>2010-06-26 11:03:44.0</pubDate>
              <eAddress>http://www.global-greenhouse-
warming.com/glacial-retreat.html</eAddress>
            </imprint>
          </monogr>
        </biblStruct>
      </sourceDesc>
    </fileDesc>
    <profileDesc>
      <langUsage>
        <language iso639="en"/>
      </langUsage>
      <textClass>
        <keywords>
          <keyTerm>glacial retreat</keyTerm>
          <keyTerm>glacier</keyTerm>
          <keyTerm>ice</keyTerm>
          <keyTerm>thinning</keyTerm>
          <keyTerm>water</keyTerm>
        </keywords>
        <domain></domain>
        <subdomain/>
        <subject/>
      </textClass>
      <annotations>
        <annotation>http://sifnos.ilsp.gr/panacea/D4.3/data/20101230/ENV_EN/493.h
tml</annotation>
      </annotations>
    </profileDesc>
  </cesHeader>
  <text>
    <body>
      <p id="p1">Glacial Retreat</p>
      <p id="p2">Glacial retreat: "With few exceptions, all the alpine glaciers
of the world are losing mass and it is predicted that this trend will continue
as global warming progresses. Glaciers in alpine areas act as buffers. During
the rainy season, water is stored in the glaciers and the melt water helps
maintain river systems during dry periods. An estimated 1.5 to 2 billion people

```

```

in Asia (Himalayan region) and in Europe (The Alps) and the Americas (Andes and
Rocky Mountains) depend on river systems with glaciers inside their catchment
areas. In areas where the glaciers are melting, river runoff will increase for a
period before a sharp decline in runoff. Without the water from mountain
glaciers, serious problems are inevitable and the UN's Millennium Development
Goals for fighting poverty and improving access to clean water will be
jeopardized" United Nations Environment Programme, 2007 Global Outlook for Ice
and Snow.</p>
<p id="p3">Glacial retreat since 1850 has been worldwide and rapid,
affecting the availability of fresh water for irrigation and domestic use,
mountain recreation, animals and plants. These all depend on glacier-melt, and
in the longer term and to some extent so does the level of the oceans.</p>
<p id="p4">Studied by glaciologists, the coincidence of glacial retreat
with the measured increase of atmospheric greenhouse gases is evidence
underpinning anthropogenic climate change. Mid-latitude mountain ranges such as
the Himalayas, Alps, Rocky Mountains, Cascade Range, Glacier National Park, and
the southern Andes. This is also occurring in tropical glacier summits such as
Mount Kilimanjaro in Africa, and Chacaltaya Glacier in Bolivia which are showing
some of the largest proportionate glacial loss.</p>
<p id="p5">The Little Ice Age was a period from about 1550 to 1850 when
the world experienced relatively cool temperatures compared to the present.
Subsequently, until about 1940 glaciers around the world retreated as the
climate warmed. Glacial retreat slowed and even reversed, in many cases, between
1950 and 1980 as a slight global cooling occurred. However, since 1980 a
significant global warming has led to glacier retreat becoming increasingly
rapid and ubiquitous, so much so that many glaciers have disappeared and the
existence of a great number of the remaining glaciers of the world is
threatened.</p>
<p id="p6">In locations such as the Andes of South America and Himalayas
in Asia, the demise of glaciers in these regions will have potential impact on
water supplies, and flooding from 'mountain tsunamis' . The retreat of mountain
glaciers, notably in western North America, Asia, the Alps, Indonesia and
Africa, and tropical and subtropical regions of South America, has been used to
provide qualitative evidence for the rise in global temperatures since the late
19th century. The recent substantial retreat and an acceleration of the rate of
retreat since 1995 of a number of key outlet glaciers of the Greenland and West
Antarctic ice sheets foreshadow a rise in sea level, having a potentially
dramatic effect on coastal regions worldwide.</p>
<p id="p7">The continued glacial retreat will have a number of different
quantitative impacts. In areas that are heavily dependent on water runoff from
glaciers that melt during the warmer summer months, a continuation of the
current retreat will eventually deplete the glacial ice and substantially reduce
or eliminate runoff. A reduction in runoff will affect the ability to irrigate
crops and will reduce summer stream flows necessary to keep dams and reservoirs
replenished. This situation is particularly acute for irrigation in South
America, where numerous artificial lakes are filled almost exclusively by
glacial melt.</p>
<p id="p8">Central Asian countries have also been historically dependent
on the seasonal glacier melt water for irrigation and drinking supplies. In
Norway, the Alps, and the Pacific Northwest of North America, glacier runoff is
important for hydropower.</p>
<p id="p9">The potential for major sea level rise depends mostly on a
significant melting of the polar ice caps of Greenland and Antarctica, as this
is where the vast majority of glacial ice is located.</p>
<p id="p10">See Also</p>
</body>
</text>
</cesDoc>

```

B. A cesAlign document pointing to an EN-FR pair of cesDoc documents

```

<cesAlign version="1.0">
  <cesHeader version="1.0">
    <profileDesc>
      <translations>
        <translation trans.loc="http://sifnos.ilsp.gr:8888/soaplab2-
results//ilsp.ilsp_bilingual_crawl_3fbd1831.12d9e90013f._7ff8/4.xml"
lang="en" wsd="UTF-8" n="1"/>

```

```

    <translation trans.loc="http://sifnos.ilsp.gr:8888/soaplab2-
results//ilsp.ilsp_bilingual_crawl_3fbd1831.12d9e90013f._7ff8/35.xml"
lang="fr" wsd="UTF-8" n="2"/>
  </translations>
</profileDesc>
</cesHeader>
</cesAlign>

```

C. Soaplab2 ACD configuration file for the Focused Monolingual Crawler

```

appl: ilsp-mono-crawler [
  documentation: "Monolingual focused crawler. The output is a text file pointing to
    XCES documents with text segmented in paragraphs."
  groups: "ILSP"
  nonemboss: "Y"
  executable: "/usr/bin/java"
]

string: jar [
  additional: "Y"
  default: "/usr/local/soaplab2_execs/bin/combinepw.jar"
  comment: "defaults"
  comment: "display false"
]

string: Domain [
  standard: "Y"
  qualifier: d
  prompt: "A descriptive title for the crawler's job."
]

infile: TermList [
  standard: "Y"
  qualifier: t
  prompt: "A file with a list of terms that define the topic. The format is<br/>
100: term1=Domain<br/>
100: multiword term2=Domain
"
]

infile: UrlList [
  standard: "Y"
  qualifier: u
  prompt: "A seed URL list. The crawler starts from these URLs, finds the links
    within these pages, visits the new pages and so on..."
]

outfile: output [
  standard: "Y"
  qualifier: o
  extension: txt
  prompt: "prompt message: The output of the crawler. "
  information: "information: The output of the crawler. "
  help: "help: The output of the crawler. "
]

list: LanguagesList [
  standard: "Y"
  qualifier: l
  values: "en; el; es; fr; it; de"
  delimiter: ";"
  prompt: "The language has to be compatible with the term list."
  min: 1
  max: 1
]

boolean: stay in web domain [
  additional: "Y"
  default: false
  comment: "defaults"
]

```

```

    qualifier: w
    prompt: "If true, the crawler will download data only from web domains in the
    UrlList. Used mainly for demo purposes. Default is false."
]

boolean: dropdb [
    additional: "Y"
    default: true
    comment: "defaults"
    qualifier: dr
    prompt: "If true, the crawler will delete the database schema of the crawl job,
    after storing the results of the crawl on the server."
]

list: HarvestersNumber [
    standard: "Y"
    qualifier: h
    default: "2"
    values: "1; 2; 3; 4; 5"
    delimiter: ";"
    prompt: "Number of harvesters to use."
    min: 1
    max: 1
]

list: MaxSize [
    standard: "Y"
    qualifier: ms
    default: "1"
    values: "1; 2; 3; 4; 5"
    delimiter: ";"
    prompt: "The crawl job will stop after MaxSize MB have been stored."
    min: 1
    max: 1
]

list: MaxTime [
    standard: "Y"
    qualifier: mt
    default: "1"
    values: "1; 2; 3; 4; 5; 6; 7; 8; 9; 10"
    delimiter: ";"
    prompt: "The crawl job will stop after MaxTime minutes. "
    min: 1
    max: 1
]

integer: Timeout [
    additional: "Y"
    qualifier: to
    default: 10
    prompt: "Wait that many seconds for a server to respond."
]

integer: Retries [
    additional: "Y"
    qualifier: rt
    default: 10
    prompt: "Stop trying to download a web page after that many retries."
]

```


D. Soaplab2 ACD configuration file for the Focused Bilingual Crawler

```

appl: ilsp-bilingual-crawler [
  documentation: "Bilingual focused crawler. The output is a text file pointing to
    CES align documents with links to pairs of cesDoc documents with text segmented
    in paragraphs."
  groups: "ILSP"
  nonemboss: "Y"
  executable: "/usr/bin/java"
]

string: jar [
  additional: "Y"
  default: "/usr/local/soaplab2_execs/bin/bilingualcrawler.jar"
  comment: "defaults"
  comment: "display false"
]

string: bc [
  default: "/usr/local/soaplab2_execs/bin/config.xml"
  qualifier: bc
  comment: "defaults"
  comment: "display false"
]

string: Domain [
  standard: "Y"
  qualifier: d
  prompt: "A descriptive title for the crawler's job."
]

infile: TermList [
  standard: "Y"
  qualifier: t
  prompt: "A file with a bilingual list of terms that define the topic. The format
    is<br/>
    100: concepto en un lenguaje=Domain<br/>
    100: multilingual term in another language=Domain
    "
]

infile: UrlList [
  standard: "Y"
  qualifier: u
  prompt: "A seed URL list with (currently) only one URL. The crawler starts from
    this URL, and in a spider-like mode finds the links within these pages pointing
    to pages inside the same web domain, visits the new pages and so on..."
]

outfile: output [
  standard: "Y"
  qualifier: o
  extension: txt
  prompt: "prompt message: The output of the crawler, which is an cesAlign document
    pointing to pairs of cesDoc documents."
]

list: Language1 [
  standard: "Y"
  qualifier: l1
  values: "en; el; es; fr; it; de"
  delimiter: ";"
  prompt: "The crawler will download documents in this language relevant to the
    topic defined in the term list."
  min: 1
  max: 1
]

list: Language2 [

```

```

    standard: "Y"
    qualifier: l2
    values: "en; el; es; fr; it; de"
    delimiter: ";"
    prompt: "The crawler will download documents in this language relevant to the
    topic defined in the term list."
    min: 1
    max: 1
]

list: HarvestersNumber [
    standard: "Y"
    qualifier: h
    default: "2"
    values: "1; 2; 3; 4; 5"
    delimiter: ";"
    prompt: "Number of harvesters to use."
    min: 1
    max: 1
]

list: MaxSize [
    standard: "Y"
    qualifier: ms
    default: "1"
    values: "1; 2; 3; 4; 5"
    delimiter: ";"
    prompt: "The crawl job will stop after MaxSize MB have been stored."
    min: 1
    max: 1
]

list: MaxTime [
    standard: "Y"
    qualifier: mt
    default: "1"
    values: "0; 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 20; 60; 100"
    delimiter: ";"
    prompt: "The crawl job will stop after MaxTime minutes."
    min: 1
    max: 1
]

integer: Timeout [
    additional: "Y"
    qualifier: to
    default: 10
    prompt: "Wait that many seconds for a server to respond."
]

integer: Retries [
    additional: "Y"
    qualifier: rt
    default: 10
    prompt: "Stop trying to download a web page after that many retries."
]

boolean: dropdb [
    additional: "Y"
    default: true
    comment: "defaults"
    qualifier: dr
    prompt: "If true, the crawler will delete the database schema of the crawl job,
    after storing the results of the crawl on the server."
]

```

E. Soaplab2 ACD configuration file for the Boilerplate Remover

```
appl: boilerplate-remover [
  documentation: "A web service that extracts text and removes boilerplate from HTML
    documents. Boilerplate is removed using the boilerpipe library
    http://code.google.com/p/boilerpipe/)."
  groups: "ILSP"
  nonemboss: "Y"
  executable: "/usr/bin/java"
]

string: jar [
  additional: "Y"
  default: "/usr/local/soaplab2_execs/bin/ilsp-nlp-boilerplate-remover-0.0.1-
    SNAPSHOT.jar"
  comment: "defaults"
  comment: "display false"
]

infile: input [
  standard: "Y"
  qualifier: if
  prompt: "An HTML document."
]

outfile: output [
  standard: "Y"
  qualifier: of
  prompt: "A text file with the main content of the input file (i.e, with
    boilerplate removed."
]

list: MethodsList [
  standard: "Y"
  qualifier: m
  values: "1; 2; 3; 4; 5; 6"
  delimiter: ";"
  default: "3"
  prompt: "The method to use for boilerplate removal.<br/> 1 is for the
    ArticleExtractor,<br/> 2 is for ArticleSentencesExtractor,<br/> 3 (default) is
    for DefaultExtractor,<br/> 4 is for KeepEverythingExtractor,<br/> 5 is for
    LargestContentExtractor,<br/> 6 is for NumWordsRulesExtractor ."
  min: 1
  max: 1
]
```