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

WP-4.5: Final Report on the Corpus Acquisition & Annotation subsystem and its components

Dissemination Level:	Public
Delivery Date:	October 31, 2012
Status – Version:	V1.2
Author(s) and Affiliation:	Prokopis Prokopidis, Vassilis Papavassiliou (ILSP), Antonio Toral (DCU), Marc Poch Riera (UPF), Francesca Frontini, Francesco Rubino (CNR), Gregor Thurmair (LG)

Relevant PANACEA Deliverables

D3.1	Architecture and Design of the Platform (T6)
D4.1	Technologies and tools for corpus creation, normalization and annotation (T6)
D4.2	Initial functional prototype and documentation (T13)
D4.3	Monolingual corpus acquired in five languages and two domains (T13)
D4.4	Revised functional prototype and documentation (T22)
D5.3	English-French and English-Greek parallel corpus for the Environment and Labour Legislation domains (T22)
D7.2	First Evaluation Report (T14)
D7.3	Second evaluation report (T23)

This document is part of technical documentation generated in the PANACEA Project, **P**latform for **A**utomatic, **N**ormalized **A**nnotation and **C**ost-**E**ffective **A**cquisition (Grant Agreement no. 248064).

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

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

Table of Contents

Document History

1.0	Pre-final version delivered to the Consortium
1.1	Correction of typos in Section 2.2.1
1.2	Integration of comments from partners

1 Introduction

PANACEA WP4 targets the creation of a Corpus Acquisition and Annotation (CAA) subsystem for the acquisition and processing of monolingual and bilingual language resources (LRs). The CAA subsystem consists of tools that have been integrated as web services in the PANACEA platform of LR production. *D4.2 Initial functional prototype and documentation* in T13 and *D4.4 Report on the revised Corpus Acquisition & Annotation subsystem and its components* in T23 provided initial and updated documentation on this subsystem, while this deliverable presents the final documentation of the subsystem as it evolved after the third development cycle of the project.

The deliverable is structured as follows. The Corpus Acquisition Component (i.e. the Focused Monolingual and Bilingual Crawlers (FMC/FBC)) is described in section 2. The final list of tools for corpus normalization (cleaning and de-duplication) is detailed in section 3. Section 4 provides documentation on all NLP tools included in the subsystem.

Due to its nature, this deliverable aggregates considerable parts of all previous WP4 deliverables. The main new additions include a) new functionalities for, among others, crawling strategy, de-duplication, and detection of parallel document pairs; and b) new NLP tools for syntactic analysis, named entity recognition, tweet processing and anonymization.

2 Corpus Acquisition Component

This section describes the final versions of the tools (FMC and FBC) developed in the PANACEA context for the acquisition of monolingual and bilingual language resources (LRs) from the Web. It also presents their deployment as web services integrated into the PANACEA platform. The use of these tools to provide domain-specific monolingual and bilingual resources for training language models and translation models in Statistical Machine Translation (in combination with WP5), is discussed in two conference papers (Pecina et al. 2011,2012) and in the attached paper (to be submitted to the *Language Resources and Evaluation* journal). Both the FMC and FBC tools are also available as an open-source Java project named ILSP Focused Crawler (ilsp-fc) from http://nlp.ilsp.gr/redmine/projects/ilsp-fc.

The FMC, available as a web service[1] in the PANACEA platform, is used for building monolingual domain-specific LRs by crawling web documents with rich textual content. This tool integrates modules for fetching and parsing HTML web pages, text classification, boilerplate removal, de-duplication and exporting of acquired documents in a variant of the cesDOC Corpus Encoding Standard[2], described in *D3.1 Architecture and Design of the Platform, Sec. 6.1.2* as Travelling Object 1 (TO1). The output of the FMC is a list of links pointing to TO1 documents. See http://nlp.ilsp.gr/nlp/examples/2547.xml for an example in French for the Environment domain.

The FBC, available as web service[3] is the first module in the PANACEA pipeline for building parallel, domain-specific LRs from the web. It aims to harvest multilingual web sites, download web documents that are relevant to a predefined domain and in the targeted languages and to identify pairs of parallel documents in the collection of stored documents. To this end, the FBC integrates all processing modules for monolingual data acquisition (i.e. normalization, language identification, cleaning, and text classification) and, in addition, a component for detection of pairs of parallel web pages. The final output of the FBC is a list of links to XML files following the cesAlign Corpus Encoding Standard for linking (parts of) documents. This example http://nlp.ilsp.gr/panacea/xces-xslt/202_225.xml serves as a link between a pair of documents in English and Greek.

2.1 Focused Monolingual Crawler

This section describes the main modules integrated in the FMC. It also documents the use of the corresponding web service. On-line documentation for this web service is also available at http://registry.elda.org/services/160.

The FMC is a focused/topical crawler that aspires to build domain-specific web collections (Qin and Chen 2005) in a targeted language, by extracting links of already fetched web pages, adding them to the list of pages to be visited and selecting web documents that are relevant to the targeted domain. In order to ensure the crawler's scalability, FMC adopts a distributed computing architecture based on Bixo[4], an open source web mining toolkit that runs on top of Hadoop[5] (http://hadoop.apache.org), a well-known framework for distributed data processing.

[1] http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_fmc_row
[2] http://www.xces.org/
[3] http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_bilingual_crawl_row
[4] http://openbixo.org/
[5] http://hadoop.apache.org/

In addition, Bixo also depends on the Heritrix[6] web crawler and makes use of ideas developed in the Nutch[7] web-search software project, two open source frameworks for mining data from the web.

The common strategy adopted for a general web crawl is initializing the crawler with a set of seed pages, visiting these pages and extracting the links within them. New web pages are visited following the extracted links and the procedure is repeated until a predefined termination criterion is met. Focused monolingual crawling is an iterative procedure that includes additional steps for content processing (e.g. text to topic classification) of visited web pages. A typical workflow for acquiring monolingual domain-specific data is illustrated in Figure 1.

The schedule of the FMC is called the Frontier and includes the URLs to be fetched in each iteration. At the beginning of the process, the Frontier is initialized with a list of seed URLs provided by the user (see the *urlList* parameter in subsection 2.1.10). If these URLs point to web pages relevant to the targeted domain, corpus construction starts from the first iteration. In the opposite case, the FMC will travel the Web in order to discover relevant pages following the Tunneling strategy (Bergmark et al 2002) as described in subsection 2.1.7.

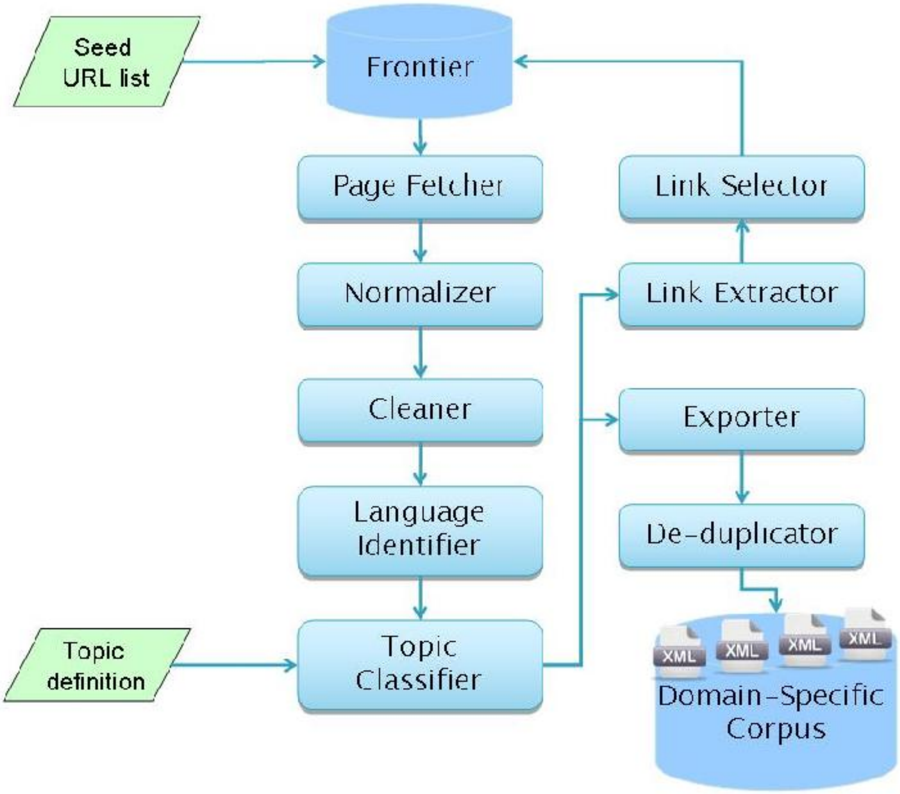


Figure 1. Workflow for acquiring monolingual domain-specific data

[6] http://crawler.archive.org
[7] http://nutch.apache.org/about.html

2.1.1 Page Fetcher

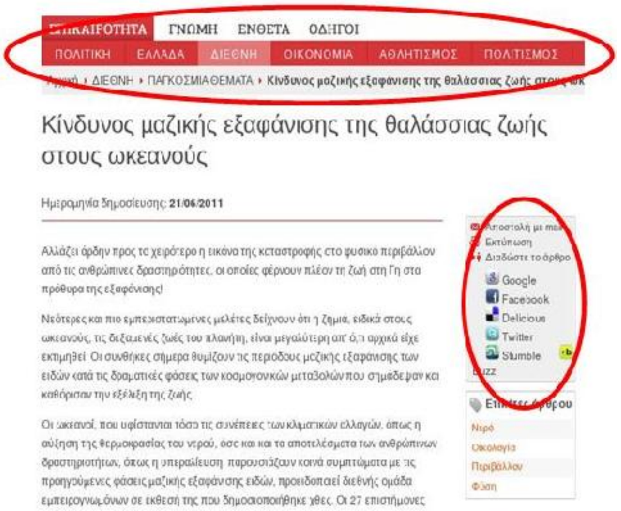
The first module of the FMC concerns page fetching. A multithreaded crawling implementation has been adopted in order to ensure concurrent visiting of more than one page. The user of the web service can define the number of harvesters to be used as discussed in subsection 2.1.10. Besides this parameter, several settings that determine the fetching process can be configured before starting the crawling process (see Appendix 7.1). Such settings concern document type filtering (i.e. in the current implementation of the FMC, only HTML web pages are accepted to be downloaded) and crawling politeness and evolution (e.g. respect of robots.txt, time intervals for revisiting URLs from the same web domain, maximum content size in bytes for downloading a web page, maximum number of URLs to be visited per iteration, maximum number of URLs from a specific host per iteration, maximum number of attempts to fetch a web page before giving up, maximum number of redirects to follow, etc).

2.1.2 Normalizer

The Normalizer module uses the Apache Tika[8] toolkit to parse the structure of each fetched web page and extract its metadata. Extracted metadata are exported at a later stage (see subsection 2.1.8) if the web document is considered relevant for the collection to be constructed. The text encoding of the web page is also detected based on the *HTTP Content-Encoding* header and the *charset* part of the *Content-Type* header, and if needed, the content is converted into UTF-8. Besides default conversion, special care is taken for normalization of specific characters like *no break space, narrow no-break space, three-per-em space,* etc.

2.1.3 Cleaner

Apart from its textual content, a typical web page also contains certain "noise" elements including navigation links, advertisements, disclaimers, etc. (often called boilerplate) of only limited or no use for linguistic purposes. Such irrelevant parts should be removed or marked as such to ensure the production of good-quality language resources. For this task FMC uses a modified version of Boilerpipe[9] (Kohlschütter et al, 2010) that also extracts structural information like *title*, *heading* and *list item*. It also segments text in paragraphs exploiting the presence of specific HTML tags like `<p>`, `</br>` and `<li>`. Paragraphs judged to be boilerplate and/or detected as titles, etc. are properly annotated (see subsection 2.1.8)



Κίνδυνος μαζικής εξαφάνισης της θαλάσσιας ζωής στους ωκεανούς

Ημερομηνία δημοσίευσης: **21/06/2011**

Αλλάζει άρδην προς το χειρότερο η εικόνα της καταστροφής στο φυσικό περιβάλλον από τις ανθρώπινες δραστηριότητες, οι οποίες φέρνουν πλέον τη ζωή στη Γη στα πρόθυρα της εξαφάνισης!

Νεότερες και πιο εμπεριστατωμένες μελέτες δείχνουν ότι η ζημιά, ειδικά στους ωκεανούς, τις δεξαμενές ζωής του πλανήτη, είναι μεγαλύτερη απ' ό,τι αρχικά είχε εκτιμηθεί. Οι συνθήκες σήμερα θυμίζουν τις περιόδους μαζικής εξαφάνισης των ειδών κατά τις δραματικές φάσεις των κοσμογονικών μεταβολών που σημάδεψαν και καθόρισαν την εξέλιξη της ζωής.

Οι ωκεανοί, που υφίστανται τόσο τις συνέπειες των κλιματικών αλλαγών, όπως η αύξηση της θερμοκρασίας του νερού, όσο και και τα αποτελέσματα των ανθρώπινων δραστηριοτήτων, όπως η υπεραλίευση, παρουσιάζουν κοινά συμπτώματα με τις προηγούμενες φάσεις μαζικής εξαφάνισης ειδών, προειδοποιεί διεθνής ομάδα εμπειρογνωμόνων σε έκθεσή της που δημοσιοποιήθηκε χθες. Οι 27 επιστήμονες

2.1.4 Language Identifier

The next processing step concerns language identification. The FMC uses the Cybozu[10] language identification library that considers n-grams as features and exploits a Naive Bayes classifier for language identification. If the document is not in the targeted language (see the

[8] http://tika.apache.org

[9] http://code.google.com/p/boilerpipe/ . Our modified version is available as an open source dependency of the ilsp-fc.

[10] http://code.google.com/p/language-detection/

language parameter in subsection 2.1.10), the web page is excluded from the next processing step and only its links are extracted (see subsection 2.1.6).

2.1.5 Topic Classifier

The aim of this module is to identify if a page that is normalized and in the targeted language contains data relevant to the targeted domain. To this end, the content of the page is compared to the domain definition provided by the user (see parameter *termList* in subsection 2.1.10), following a string-matching method adopted by the Combine web crawler[11]. A naive stemmer included in the org.apache.lucene library is used to stem user-provided terms and document content. Based on the number of terms' occurrences, their location in the web page (i.e. in the title, keywords, and/or body) and the weights of found terms, a page relevance score *p* is calculated as follows:

$$p = \sum_{i=1}^{N} \sum_{j=1}^{4} n_{ij} \cdot w_i^t \cdot w_j^l$$

where *N* is the amount of terms in the domain definition, w_i^t is the weight of term *i*, w_j^l 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.

In addition, the amount of unique terms that exist in the main content (i.e. cleaned text) of the page, *m*, is calculated. Then, the values *p* and *m* are compared with two predefined thresholds (*t1* and *t2*) and if both values are higher than the thresholds, the web page is categorized as relevant to the domain and stored.

It is worth mentioning that the user can affect the strictness of the classifier by setting the values of both thresholds in a configuration file (see Appendix 7.1). The t_1 threshold is a multiple of the median value of the term weights provided by the user. The factor of multiplication is defined in the *min_content_terms* parameter of the configuration file. Similarly, t_2 is defined in *min_unique_content_terms*.

Besides the classification of a document as relevant to the targeted domain or not, the Topic Classifier also categorizes in-domain pages into one or more of the sub-domains that can be defined in the domain definition provided by the user (see parameter *termList* in subsection 2.1.10). The identified sub-domain(s) is/are stored in the <`subdomain`> element of the XML file exported from each relevant web page (see 2.1.8). Therefore, a user can easily calculate how many documents/tokens belong to each sub-domain (or in a combination of sub-domains) and use this statistical information as evidence of the distribution of the sub-domains in the acquired data.

2.1.6 Links Extractor

Even when a web page is not to be stored (because it was deemed irrelevant to the domain, or not in the targeted language), its links are extracted and added to the list of links scheduled to be visited. Since the crawling strategy is a critical issue for a focused crawler, a score *sl* is calculated for each link as follows:

[11] Software package for general and focused Web-crawling, http://combine.it.lth.se/

$$sl = p / L + \sum_{i=1}^{N} n_i \cdot w_i$$

where *p* is the relevance score of the source page, *L* is the amount of links originating from the source page, *N* is the amount of terms in the topic definition, n_i denotes the number of occurrences of the *i-th* term in the surrounding text and w_i is the weight of the *i-th* term. According to this approach, the link score is influenced by the source web page relevance score (see 2.1.5) and the estimated relevance of the link's anchor text.

2.1.7 Links Selector

As shown by Cho et al (1998), using a similarity metric that takes into account the content of anchor texts leads to improvements in differentiation among out-links. Based on this conclusion, we adopt the Best-First algorithm, where links are ranked by their score and the first *N* links are selected to be fetched in the next iteration. The user can define parameter *N* in the (*fetch_buffer_size* in Appendix 7.1) configuration file. Since the link score models the likelihood that the link under consideration points to a web page relevant to the target domain, the proposed method forces the crawler to visit relevant web pages earlier.

However, it is often the case that relevant pages cannot be found without first visiting less relevant web pages. This fact implies that the restricted selection of links originating from relevant pages to be followed may choke the crawler. To overcome this shortcoming, we adopted the "Tunneling" algorithm, according to which the crawler will not give up probing a direction immediately after it encounters an irrelevant page, but will continue searching in that direction for a pre-defined number of steps. This allows the focused crawler to travel from one relevant web cluster to another when the gap (number of irrelevant pages) between them is within a limit. The value of this limit is a parameter defined in the configuration file as *max_depth*.

2.1.8 Exporter

The Exporter module is applied for the generation of an XML file in the TO1/cesDoc format for each stored web document. The XML files contain the textual content converted into UTF-8 and segmented in paragraphs. Moreover, each XML file contains metadata about the corresponding document inside a `<cesHeader>` element.

The first element of the header, the `<fileDesc>` element, includes general information about the document. Specifically, the `<titleStmt>` sub-element contains the title of the document (`<title>` container) and the PANACEA partner responsible for these operations on this particular document. The `<publicationStmt>` sub-element holds information about the status (i.e. distributor and its e-address, availability and publication date) of the document. The `<sourceDesc>` sub-element groups bibliographical information for the document such as the title, the author, the publisher, the date downloaded and the URL it was downloaded from.

The second element of the header, the `<profileDesc>` includes information about the content of the document. In particular, the `<langUsage>` sub-element reports the language of the document and the `<textClass>` holds the key terms of the document, the sub-domain as identified by the Topic Classifier (see 2.1.5). It is worth mentioning that the key terms included inside the `<keywords>` sub-element of `<textClass>` are the keywords extracted from the metadata of the web document. Therefore, these terms should not be confused with the terms detected in this particular document during comparison with the domain definition. The <annotations> sub-element of <profileDesc> is used for storing links to other documents relevant to this basic version. After the exporting phase, there is only one <annotation> which points to the original HTML document.

The <body> element contains the content of the document segmented in paragraphs. Besides the normalized text, each paragraph element <p> is enriched with attributes providing more information about the process outcome. Specifically, (<p>) elements in the XML files may contain the following attributes:

1. *crawlinfo* with possible values:

    a. *boilerplate*, meaning that the paragraph has been considered boilerplate by the Cleaner module (see subsection 2.1.3) as shown in the following example:

```
<p id="p1" crawlinfo="boilerplate">Home</p>
<p id="p2" crawlinfo="boilerplate">Partners</p>
<p id="p3" crawlinfo="boilerplate">Main Menu</p>
<p id="p4" crawlinfo="boilerplate">Home</p>
<p id="p5" crawlinfo="boilerplate">Background</p>
<p id="p6" crawlinfo="boilerplate">The Theme for 2011</p>
<p id="p7" crawlinfo="boilerplate">How can you participate?</p>
<p id="p8" crawlinfo="boilerplate">Register your Activity</p>
<p id="p9" crawlinfo="boilerplate">WMBD Around the World</p>
<p id="p10" crawlinfo="boilerplate">WMBD Community</p>
<p id="p11" crawlinfo="boilerplate">Press / Materials</p>
<p id="p12" crawlinfo="boilerplate">Related Links</p>
<p id="p13" crawlinfo="boilerplate">Partners</p>
<p id="p14" crawlinfo="boilerplate">Translate this Site:</p>
<p id="p15" crawlinfo="boilerplate">Partners &amp; Sponsors</p>
<p id="p16" crawlinfo="ooi-length">WMBD Partners:</p>
<p id="p17" topic="sustainable development">United Nations
Environment Programme (UNEP) is the voice for the environment
in the United Nations system. It is an advocate, educator,
catalyst and facilitator, promoting the wise use of the
planet's natural assets for sustainable development.</p>
```

    b. *ooi-lang*, denoting that the paragraph is not in the targeted language. One of the results of manual evaluation in the first evaluation cycle, reported in D7.2 *First evaluation report. Evaluation of PANACEA v1 and produced resources* was that about 5% of the acquired documents contained at least one paragraph not in the targeted language. Therefore, the Exporter applies the embedded language identifier (see subsection 2.1.4) at paragraph level as well. If a paragraph is not in the targeted language, the attribute *crawlinfo* takes the value *ooi-lang*. As an example, notice `p63` paragraph in the listing below.

```
<p id="p61" topic="delta;marsh">The waters of the Danube, which
flow into the Black Sea, form the largest and best preserved of
Europe's deltas. The Danube delta hosts over 300 species of
birds as well as 45 freshwater fish species in its numerous
lakes and marshes.</p>
```

```
<p id="p62" crawlinfo="ooi-length">Delta du Danube</p>
<p id="p63" crawlinfo="ooi-lang">Les eaux du Danube se jettent
dans la mer Noire en formant le plus vaste et le mieux préservé
des deltas européens. Ses innombrables lacs et marais abritent
plus de 300 espèces d'oiseaux ainsi que 45 espèces de poissons
d'eau douce.</p>
```

c. *ooi-length*, denoting that this paragraph is so short that either it is not useful, or it can confuse the language identifier. Another finding from the first evaluation cycle was that a very large proportion of the documents (approx. 80%) contained at least one short paragraph of only limited or no use. To eliminate this, the Exporter compares the length of each paragraph (in terms of tokes) with a predefined threshold provided by the user (see parameter *minimumLength* in 2.1.10) and classifies short paragraphs as out of interest (i.e. adds the value *ooi-length* to the *crawlinfo* attribute). For an example, see *p41* and *p43* paragraphs in the listing below (and *p62* in the listing above).

```
<p id="p40" type="listitem" topic="forest;nature
reserve">National Trust membership gives you access to green
space and helps fund conservation. The trust manages 250,000
hectares of land, including forest, woods, nature reserves,
farmland and moorland, as well as 707 miles of coastline in
England, Wales and Northern Ireland.</p>
<p id="p41" crawlinfo="ooi-length">Plantlife</p>
<p id="p42">Plantlife works to protect wild plants and their
habitats. Activities include rescuing wild plants from the
brink of extinction, and ensuring that common plants don't
become rare in the wild. It actively campaigns on a number of
issues affecting wild plants and fungi. The Plantlife website
has a wealth of downloadable information about wild plants and
plant conservation. Find out how you can support the
organisation here .</p>
<p id="p43" crawlinfo="ooi-length">Buglife - The Invertebrate
Conservation Trust</p>
```

2. *type* with possible values: *title*, *heading* and *listitem* as identified by the Cleaner module (see 2.1.3).

3. *topic* with a string value including all terms from the domain definition detected in this paragraph.

2.1.9 De-duplicator

The Web contains many duplicate (parts of) pages. For instance, Baroni et al. (2009) reported that during building of the Wacky corpora the amount of documents was reduced by more than 50% after de-duplication. Ignoring this phenomenon and including duplicate documents could have a negative effect in creating a representative corpus. Therefore, the De-duplicator examines the main content of the stored documents in order to detect and remove near-duplicates. This module employs the de-duplication strategy[12] included in the Nutch framework, which involves the construction of a text profile based on quantized word frequencies, and an MD5 hash for each page (see section 3.2).

[12] http://svn.apache.org/repos/asf/nutch/trunk/src/java/org/apache/nutch/crawl/

An additional step has been integrated into the final version of FMC for detection and removal of (near) duplicates. Each document is represented as a list with size equal to the number of paragraphs (without *crawlinfo* attribute) of the document. The elements of the list are the MD5 hashes of the paragraphs. Then, each list is checked against all other lists. For each candidate pair, the intersection of the lists is calculated. If the ratio of the intersection cardinality with the cardinality of the shortest list is over a predefined threshold, the documents are considered near-duplicates and the shortest is discarded.

2.1.10 FMC as Web service

The web-service is available at http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_fmc_row and documentation of this web service is available at http://registry.elda.org/services/160. It uses three mandatory parameters:

1. The *language* parameter denotes the targeted language (see 2.1.4 and 2.1.8). Currently supported languages are English, French, German, Greek, Italian and Spanish. Each downloaded web page is analyzed by the embedded language identifier. If the document is not in the targeted language the document is discarded. In addition, the language identifier is applied at paragraph level of stored documents (i.e. relevant to the targeted domain) and paragraphs in a language other than the main document language are marked as such.

2. The *termList* is a list of term triplets (<relevance,term,subtopic>) that describe a domain and, optionally, subcategories of this domain (see 2.1.5). More details about constructing such domain definitions are reported in section 4.1 of *D4.3 Monolingual corpus acquired in five languages and two domains*. An example domain definition can be found at http://nlp.ilsp.gr/panacea/testinput/monolingual/ENV_topics/ENV_EN_topic.txt for the *Environment* domain in English and an extract is provided below:

```
80:chemical waste=deterioration of the environment
25:civil liability=environmental policy
70:classified forest=environmental policy
50:clean industry=environmental policy
50:clean technology=environmental policy
70:clearing of land=cultivation of agricultural land;deterioration
of the environment
100:climate change=deterioration of the environment;natural
environment
```

3. The *urlList* parameter is a list of seed URLs with which the crawler is initialized (see 2.1.1). An example seed URL list for Environment in English can be found at http://nlp.ilsp.gr/panacea/testinput/monolingual/ENV_seeds/ENV_EN_seeds.txt.

The web service uses four optional parameters that allow the user to configure the crawl:

1. The *maxTime* parameter guides the crawler to stop after *maxTime* 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.

2. The *threadsNumber* parameter sets the number of harvesters that will be used to fetch web pages in parallel.

3. The *minimumLength* parameter sets the minimum number of tokens that an acceptable paragraph should include. Paragraphs will fewer tokens than *minimumLength* will be assigned an *ooi-length* value for the attribute *crawlinfo*.

4. The *insert_xslt* parameter concerns the introduction of a stylesheet in the XML file for better rendering of the contents in a web browser.

2.1.11 Acquired Corpora

The initial version of the FMC was used to acquire documents in the PANACEA languages English, Spanish, Italian, French and Greek for the Environment and the Labour Legislation domains (named MCv1 in PANACEA context). Details about these collections are reported in *D4.3 Monolingual corpus acquired in five languages and two domains* and *D7.2 First evaluation report. Evaluation of PANACEA v1 and produced resources*. In addition, the effect of using the English, French and Greek collections on training domain-adapted language models and using them in Statistical Machine Translation is discussed in Pecina et al. (2011).

Following the PANACEA timetable and the comments of the first annual review report, the FMC was used to construct augmented collections (named MCv2 in PANACEA context) in the above mentioned language and domain combinations. The size of the produced MCv2 corpora[13] ranges from 13K to 28K web pages (26M to 70M tokens) depending on the selected domain (ENV or LAB) and the targeted language (EL, EN, ES, FR, IT). The only exception concerns the Greek data in the Labour Legislation domain, where only ~7K web pages were acquired. However, this collection amounts to ~21M tokens, since it consists mainly of large legal documents or lengthy discussions/arguments about Labour Legislation. Details regarding the preparation of the required input (i.e. domain definitions and seed URLs), the quantity of the acquired data and the distribution of sub-domains in MCv2 are provided in *D7.3 Second evaluation report.* The new collections in EN, FR and EL were used for domain adaptation of an SMT system again, and the results are reported in Pecina et al. (2012).

In addition, and after suggestions during the second review, the FMC was also used to acquire documents in Italian and German for Health & Safety / Arbeitsschutz / Sicurezza sul lavoro, with a focus on the sub-domain of construction industry. The acquired data consist of 9591 IT documents (15.2 M tokens) and 4786 DE documents (12.36 M tokens). Moreover, FMC was employed to construct collections in English and German for the Automotive domain with focus on transmission/gearboxes sub-domain. The delivered corpora contain 7191 DE documents (6.6 Mt) and 13351 EN documents (17.5 Mt). The numbers of tokens were computed with a “naive” string tokenizer applied to the text of paragraphs without the *crawlinfo* attribute. Details about the quality of these collections will be reported in the forthcoming deliverable *D8.3 Task-based evaluation.*

2.2 Focused Bilingual Crawler

The Focused Bilingual Crawler (FBC) integrates the Focused Monolingual Crawler (i.e. all modules discussed in section 2.1) and a module for detecting pairs of parallel documents from domain-specific collections acquired from the web. A typical FBC workflow is presented in Figure 2.

[13] MCv2 was delivered internally to project partners in T20 and was augmented with automatic morphosyntactic annotations as described in Appendix 7.2.

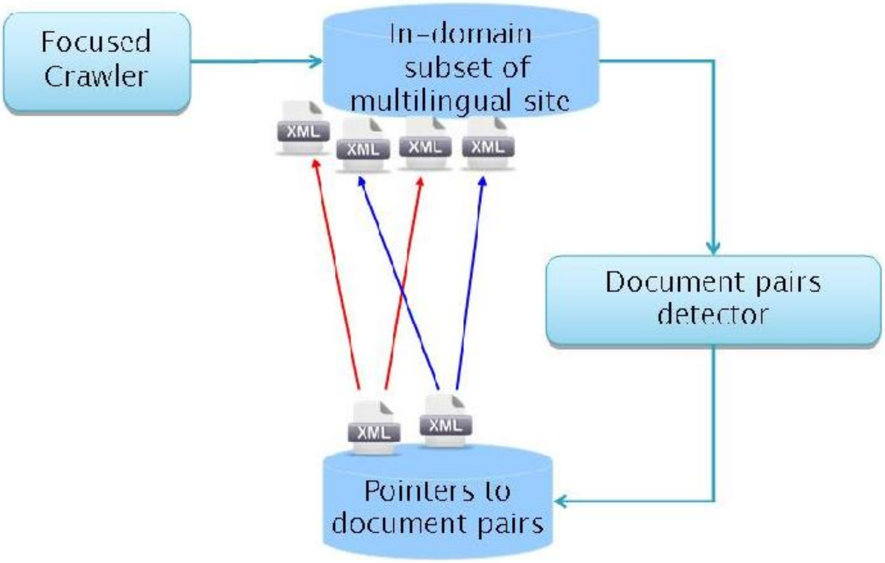

Focused Crawler
In-domain subset of multilingual site
XML
Document pairs detector
Pointers to document pairs

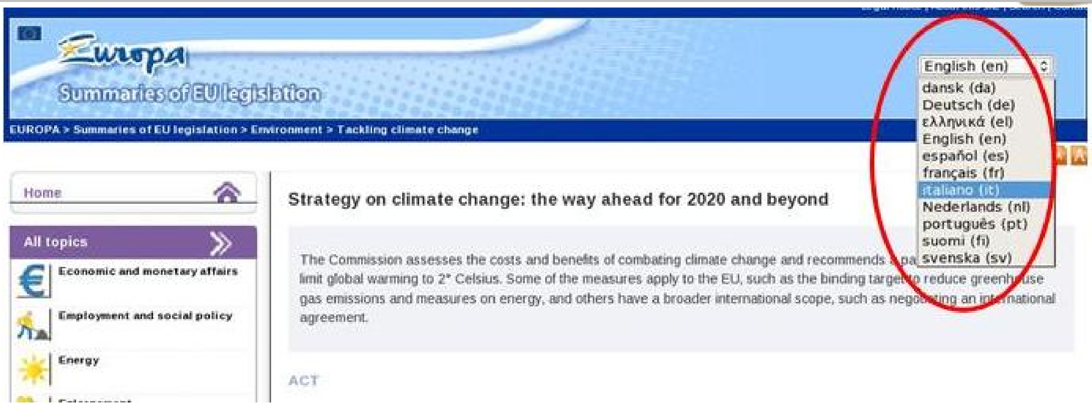
Figure 2. Workflow for acquiring parallel domain-specific data

The required input from the user consists of a list of terms that describe a domain in two languages and a URL pointing to a multilingual web site (see parameters *termList* and *urlList* in sub section 2.2.2). The FBC starts from this URL and in a spider-like mode extracts links to pages inside the same web site. Extracted links are prioritized according to a) the probability that they point to a translation of the web page they originated from b) the relevance-to-the-domain score of their surrounding text and c) the relevance-to-the-domain score of the web page they were extracted from. As in the case of FMC, FBC follows most promising links (see 2.1.7), visits new pages and continues crawling the web site until no more links can be extracted, or crawl time expires.

The only difference with the approach followed by FMC is that the link score is influenced by the likelihood that the link under consideration points to a translation of the web page it is extracted from. To model this probability we introduced simple heuristics that boost the link score if the source web page is in *langugel* and the anchor text of the link under consideration contains specific words/tokens/phrases, which imply that the link points to a web page in *language2*. The formulation of the score link is slightly differentiated from the one adopted in FMC by introducing the parameter *c* as follows:

$$sl = c + p / L + \sum_{i=1}^{N} n_i \cdot w_i$$

where *c* gets a high positive value if the link "points" to a web page in *language2* and 0 otherwise. For instance, let us suppose that FBC aims to detect pairs of parallel web documents for the Environment domain in EN and IT. If the EN web page in Figure 3 is fetched, the selected link (in the red ellipse) will get a high score since the anchor text "italiano (it)" implies that this link points to the Italian translation of the English page. It is worth mentioning that the current version of FBC uses this method with the aim of visiting candidate translations before following other links.



EUROPA > Summaries of EU legislation > Environment > Tackling climate change

Strategy on climate change: the way ahead for 2020 and beyond

The Commission assesses the costs and benefits of combating climate change and recommends a pa limit global warming to 2° Celsius. Some of the measures apply to the EU, such as the binding target to reduce greenhouse gas emissions and measures on energy, and others have a broader international scope, such as negotiating an international agreement.

ACT

Figure 3. An English web page that contains links which are very likely to point to translations of the current page

2.2.1 Pair Detector

After downloading in-domain pages from the selected web sites, a Pair Detector module is employed to identify pairs of pages that could be considered parallel. The final version of Pair Detector integrated in the FBC, adopts two methods with the aim of detecting pairs of candidate parallel documents. The first method is based on co-occurrences, in two documents, of images with the same filename; the second takes into account structural similarity.

As a first step, the XML files exported after crawling are parsed and the following features are extracted: i) the document language (reported in the <langUsage> sub-element); ii) the depth of the page, (e.g. for http://domain.org/dir1/dir2/dir3/page.html the depth is 4); iii) the amount of paragraphs (i.e. amount of <p> elements); iv) the length (in terms of tokens) of the clean text; and v) the fingerprint of the <body> element (i.e. a sequence of integers that "form" the structural information of the page, similarly to the approach described by Esplà-Gomis and Forcada, 2010). For instance, the fingerprint of the following extract is [-2, 28, 145, -4, 9, -3, 48, -5, 740, -2, 35] where boilerplate paragraphs are ignored; -2, -3 and -4 denote that the attribute *type* of corresponding <p> elements has a value *title*, *heading* and *listitem* respectively; -5 denotes the existence of attribute *topic* in a <p>; and positive integers are the lengths (in terms of characters) of the paragraphs.

```
<p id="p171" type="title">Strategia degli investimenti</p> <!-- -2, 28--!>

<p id="p172">I ricavi degli investimenti sono un elemento essenziale per
finanziare le rendite e mantenere il potere d'acquisto dei beneficiari delle
rendite.</p> <!-- 145 -->

<p id="p173" type="listitem">Document:</p> <!-- -4, 9 -->

<p id="p174" crawlinfo="boilerplate" type="listitem">Factsheet «La strategia
d'investimento della Suva in sintesi»(Il link viene aperto in una nuova
finestra) </p> <!-- ignored -->

<p id="p175" type="heading">Perché la Suva effettua investimenti
finanziari?</p> <!-- -3, 48, etc... -->

<p id="p176" topic="prevenzione;prevenzione degli infortuni;infortunio sul
lavoro">Nonostante i molti sforzi compiuti nella prevenzione degli infortuni
sul lavoro e nel tempo libero ogni anno accadono oltre 2500 infortuni con
conseguenze invalidanti o mortali. In questi casi si versa una rendita per
invalidità agli infortunati oppure una rendita per orfani o vedovile ai
superstiti. Nello stesso anno in cui attribuisce una rendita, la Suva provvede
ad accantonare i mezzi necessari a pagare le rendite future. La maggior parte
del patrimonio investito dalla Suva è rappresentato proprio da questi mezzi,
ossia dal capitale di copertura delle rendite. La restante parte del
```

```
patrimonio è costituta da accantonamenti per prestazioni assicurative a breve
termine come le spese di cura, le indennità giornaliere e le riserve.</p>
<p id="p177" type="heading">Come viene investito il patrimonio?</p>
```

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 as candidate pairs, on the assumption that it is very likely that translations are in the same or neighbouring depths in the web site tree.

Next, we extract the filenames of the images from the HTML source of each stored web page and each file is represented as a list of image filenames. Since it is very likely that some images are illustrated in many web pages, we count the occurrence frequency of each image and discard relatively frequent images (i.e. images, like Facebook and Twitter icons, that exist in more than 10% of the total XML files) from the lists. Then, 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 relatively high (i.e. over predefined thresholds respectively).

More pairs are detected by examining structure similarity. For each candidate pair of parallel documents a 3-dimensional feature vector is constructed. The first element is the ratio of their fingerprint lengths, the second is the ratio of their paragraph amounts and the third is the ratio between the edit distance of the fingerprints of the two documents and 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 developing the initial version of the FBC.

2.2.2 FBC as Web service

The web-service is available at http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_bilingual_crawl_row and a documentation of this web service is available at http://registry.elda.org/services/127. It uses five mandatory parameters:

1. The *domain* parameter is a descriptive title for the crawler's job.
2. The *language1* parameter denotes the first targeted language (see parameter *language* in section 2.1.10).
3. The *language2* parameter denotes the second targeted language. The values of this and the previous parameter define the targeted language pair.
4. The *termList* parameter is a bilingual list of terms that define a domain. Each visited web page is compared with this list and classified as relevant or not. The format is:

   ```
   100:term1=subdomain1;subdomain2>lang1
   100:multiword term2=subdomain2>lang2
   ```

   An example domain definition for the EN-FR pair for Environment can be found at http://nlp.ilsp.gr/panacea/testinput/bilingual/ENV_topics/ENV_EN_FR_topic.txt.
5. The *urlList* parameter is a seed URL list which should contain only one URL (e.g. http://europa.eu/legislation_summaries/environment/index_en.htm). The crawler will visit only links pointing to pages inside versions of the top domain of the URL (e.g. http://www.fifa.com/, http://es.fifa.com/ , etc.)

The web service uses five optional parameters that allow the user to configure the crawl:

1. The *MaxTime* parameter denotes that the crawler will stop crawling after *MaxTime* minutes (default is 10). The crawl job evolves in sequential cycles, where each cycle involves a) selection of urls to be fetched, b) fetching, c) classification and d) extraction of new links. If time expires, the crawl job will stop after finishing the current cycle. Then pairs will be detected and links to these pairs will be constructed. Thus the whole process will run for more than *MaxTime* minutes.

2. The *filter* parameter is a string to filter out URLs which do not contain this string. The use of this filter forces the crawler to focus on a part of the multilingual web domain. Note that if this filter is used, the seed URL should contain this string. For example, a valid string for the seed URL mentioned above is "/legislation_summaries/environment". This filter should be best used for demo purposes and narrow crawls.

3. The *insert_xslt* parameter concerns the introduction of a stylesheet in the XML file for better rendering of the contents of both files of a pair, in a browser.

4. The *minimumLength* parameter sets the minimum number of tokens that an acceptable paragraph should include. Paragraphs will fewer tokens than *minimumLength* will be assigned an *ooi-length* value for the attribute *crawlinfo*. Each paragraph with length (in terms of tokens) lower than this value will be marked as 'out-of-interest' (*ooi-length*).

5. The *threadsNumber* parameter sets the number of harvesters that will be used to fetch web pages in parallel.

2.2.3 Acquired Corpora

The initial version of FBC was used to construct domain-specific parallel corpora (aligned on document level) in the EN-FR and EN-EL language combinations for the Environment the Labour Legislation domains. Details about these collections are reported in *D5.3 English-French and English-Greek parallel corpus for the Environment and Labour Legislation domains*, two conference papers (Pecina et al., 2011; Pecina et al., 2012) and the attached draft paper.

The FBC was also used to acquire pairs of parallel documents in Italian and German for Health & Safety / Arbeitsschutz / Sicurezza sul lavoro, with a focus on the sub-domain of construction industry. The acquired data consist of 807 pairs of documents containing 1.4M tokens in Italian, and 1.21M tokens in German. Moreover, the FBC was employed to construct an EN-DE collection for the Automotive domain with focus on the transmission/gearboxes sub-domain. The delivered corpus consists of 1161 pairs of documents containing 0.71 and 0.58 M tokens for EN and DE respectively. The numbers of tokens were computed with a “naive” string tokenizer applied to the text of paragraphs without the *crawlinfo* attribute.

Details about the quality of these collections will be reported in the forthcoming deliverable *D8.3 Task-based evaluation*. In addition, details about the performance of the pair detector will also be reported in *D8.2 Tool-based evaluation*.

3 Corpus normalization tools and services

This section describes two PANACEA web services concerned with removal of boilerplate and detection of duplicate documents.

3.1 Cleaner

The Cleaner module described in 2.1.3 is also available as a standalone web service accessible from http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_cleaner_row. The service has one mandatory parameter:

1. The *input* parameter is the URL of a web document to be cleaned.

The Cleaner also uses five optional parameters:

1. The *outputType* parameter sets the type of the output. It can be: i) a text file containing only the clean text, ii) an XML file containing metadata of the web document and the clean text only, and iii) an XML file containing metadata of the web document and the content of the web document annotated as boilerplate or text. Users can select the type of output according to their needs. For example, the first type might be useful for somebody who has already downloaded web documents and would like to apply de-duplication on document level by using only the clean text of the downloaded web documents. The second type could be useful for someone who would like to extract metadata from the source web documents and keep only the clean text from these sources. If the user is interested in both boilerplate and clean text, the third type should be selected. It is worth mentioning that both the second and third types provide structural information about the web document, by using the attribute *type* and the values *title*, *heading* or *listitem*.

2. The *methodsList* parameter sets the method for removing boilerplate. Boilerpipe provides six methods: ArticleExtractor, ArticleSentencesExtractor, DefaultExtractor, KeepEverythingExtractor, LargestContentExtractor, and NumWordsRulesExtractor (default). Short descriptions of the methods are reported at http://boilerpipe.googlecode.com/ svn/trunk/boilerpipe-core/ javadoc/1.0/index.html. The attribute *crawlinfo* with value *boilerplate* will be added to every paragraph of the web document which has been classified as boilerplate. Remaining paragraphs constitute the clean text.

3. The *minimumLength* parameter defines the minimum accepted length in terms of tokens for each paragraph of the clean text. Users not interested in short paragraphs can set the value of this parameter accordingly. The attribute *crawlinfo* with value *ooi-length* will be added to every paragraph of the clean text with length less than *minimumLength*. The default value is 10.

4. The *language* parameter sets the targeted language. The current list of ISO 639 codes for supported languages includes en, el, es, fr, it and de. Selecting one of these languages implies that the user is only interested in content in this language. Therefore, the embedded language identifier will be applied on each "accepted" paragraph (i.e. each paragraph that has not been classified as *boilerplate* and has length over the *minimumLength*), and a *crawlinfo* attribute with value *ooi-lang* will be added to every paragraph that is not in the targeted language. If there is no targeted language (default), the embedded language identifier will be applied on the main content (clean text) of the web document, and the

ISO code of the identified language code will fill the element <*language*>.

5. The *termList* is a list of triplets (<*relevance weight, term, topic-class*>) that define the domain, or the sub-domains. This parameter can be provided by uploading an already existing file with a list of terms as described in section 2.1.10 above. The embedded text to topic classifier will be applied on the document and, if the document is classified as relevant to a sub-domain, the <*subdomain*> container will be filled accordingly. In addition, the *Cleaner* will search for these terms in each "accepted" paragraph. If one or more terms are found in a paragraph, the attribute *topic* will be added to this paragraph, and found terms will be stored as the attribute value.

3.2 De-duplicator

The De-duplicator module described in 2.1.9 is also available as a standalone web service accessible from http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_deduplicatormd5_row. The service has two mandatory parameters:

1. The *input* denotes a file containing a list with URLs to the files to be de-duplicated.

2. The *inputType* denotes the type of the files to be de-duplicated. These files could be text or TO1 XML files similar to the ones generated by the FMC.

The service also has two optional parameters:

1. *minimumTokenLength* During the calculation of the page profile, all tokens equal or shorter than this value are discarded. The default value is 2.

2. *quantValue*. Tokens with frequency (after quantization) below this value are discarded. The default value is 3.

The output is a text file containing a list with URLs pointing to the files that have remained after de-duplication.

4 NLP tools and services

This section catalogues and describes the NLP tools introduced to the PANACEA platform as web services. The final set of services includes services that provide sentence splitting, tokenization, POS tagging, lemmatization, syntactic analysis functionalities for all languages targeted by PANACEA. Additional services for some languages provide named entity recognition, tweet processing and anonymization functionalities.

In the following subsections, we provide information on the modus operandi and the performance of selected tools behind the services. Most importantly, we point to the web pages and WSDL URLs via which the services can be accessed, tested, and integrated. When applicable, we also link to Taverna[14] workflows integrating the services in larger processing pipelines.

As prescribed in *D3.1 Architecture and Design of the Platform,* the NLP functionalities relevant to this deliverable share two mandatory parameters, *input* and *language***.** When applicable, we

[14] The workflows can be used in the Taverna Workflow Management System http://www.taverna.org.uk/. See Appendix 7.5 for some example workflows for Greek and German.

document additional, tool-specific parameters. Another prerequisite for integrating a tool in the PANACEA platform is that it can process input and generate output in the common encoding format documented in *D3.1, Section 6.1.3*. To achieve this goal, PANACEA partners investigated two approaches. UPF, DCU and CNR built specific web services[15] to perform I/O conversions from and to their tools. ILSP adapted its NLP tools by integrating importers and exporters from and to the common encoding format. ILSP also provided a converter from UIMA Common Analysis Structure files to PANACEA's Travelling Object 2 format encoded in GrAF.

Finally, for each service, we provide links to entries in the PANACEA registry, where (updated) documentation and access information will be provided during and after the project's timeline, thus ensuring the sustainability of the PANACEA platform.

[15] See, among others, the UPF, DCU, and CNR converters at Appendix 7.4, Web services for the CAA subsystem in the PANACEA platform

4.1 Tools for English and French hosted by DCU

4.1.1 Europarl Tools: Sentence splitting, tokenization and lowercasing

The Europarl tools[16] were developed to process the proceedings of the European Parliament, in order to derive parallel corpora suitable for training Statistical Machine Translation systems.

The tools that have been integrated in PANACEA are the sentence-splitter, the tokeniser and the lowercaser. The sentence-splitter and the tokeniser are based on a set of regular expressions (independent of the language) and use optionally a list of language-dependent abbreviations. The lowercaser uses Perl's lc function.

These webservices can be accessed and integrated via the information from Table 1, Table 2, and Table 3.

URL	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.europarl_sentence_splitter_row
WSDL	http://www.cngl.ie/panacea-soaplab2-axis/typed/services/panacea.europarl_sentence_splitter?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/76
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/7

Table 1 WS Details for Europarl sentence-splitter

URL	http://www.cngl.ie/panacea-soaplab2-axis//#panacea.europarl_tokeniser_row
WSDL	http://www.cngl.ie/panacea-soaplab2-axis/typed/services/panacea.europarl_tokeniser?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/77
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/7

Table 2 WS Details for Europarl tokeniser

URL	http://www.cngl.ie/panacea-soaplab2-axis//#panacea.europarl_lowercase_row
WSDL	http://www.cngl.ie/panacea-soaplab2-axis/typed/services/panacea.europarl_lowercase?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/75

Table 3 WS Details for Europarl lowercaser

[16] http://www.statmt.org/europarl/

4.1.2 Berkeley tagger

Berkeley tagger is a web service that wraps the Berkeley Parser (Petrov et al., 2006) and outputs the PoS information. Apart from handling English and French, it is also available for German. The tool has one optional parameter, *tokenize*, which, if activated, guides the tool to tokenize the text before tagging it. The Berkeley tagger can be accessed and integrated via the information from Table 4.

URL	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.berkeley_tagger_row
WSDL	http://www.cngl.ie/panacea-soaplab2-axis/typed/services/panacea.berkeley_tagger?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/72

Table 4 WS Details for berkeley_tagger

4.1.3 Treetagger

This is a web service that wraps the TreeTagger (Schmidt, 1994), and outputs PoS tags and lemmas. It is available for a number of languages, including English and French, for which it has been used to tag the datasets crawled in the project for the environment and labour legislation domains. TreeTagger can be accessed and integrated via the information from Table 5.

URL	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.treetagger_row
WSDL	http://www.cngl.ie/panacea-soaplab2-axis/services/panacea.treetagger?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/276

Table 5 WS Details for treetagger

4.2 Tools for Spanish hosted by UPF

4.2.1 IULA Preprocess and IULA Tokenizer

The IULA Preprocess and the IULA Tokenizer services provide preprocessing functionalities for Spanish. IULA Preprocess segments text into minor structural units (titles, paragraphs, sentences, etc.); detects entities usually not found in dictionaries (numbers, abbreviations, URLs, emails, proper nouns, etc.); and makes sure that sequences of two or more words (in dates, phrases, proper nouns, etc.) are kept together in a single block. The IULA Tokenizer service delivers the same results vertically tokenized, one word per line. The two services accept input and output encoded in UTF-8 or ISO-8859-1/-15.

Both services employ the IULA Processing Tool (IPT), developed by Martínez and Vivaldi (2010). IPT is based on rules that depend on a series of resources to improve obtained results: a grammatical phrase list, a foreign expression list, a follow-up abbreviation list, a word-form lexical database (which is also used by the IULA POS-tagger described in the following subsection), and a stop-list to increase lexical-lookup efficiency. IPT has been evaluated against a hand-tagged corpus used as a Gold Standard, divided in two domain specific topics (Press and Genomics). Accuracies of 99.39% and 91.55% are reported by Martínez et al. (2010) for sentence splitting in the two collections. Respective results for NER are 95.43% and 99.76%.

Web form	http://kurwenal.upf.edu/soaplab2-axis/#chunking_segmentation.iula_preprocess_row, http://kurwenal.upf.edu/soaplab2-axis/#tokenization.iula_tokenizer_row
WSDL	http://kurwenal.upf.edu/soaplab2-axis/services/chunking_segmentation.iula_preprocess?wsdl , http://kurwenal.upf.edu/soaplab2-axis/services/tokenization.iula_tokenizer?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/124, http://registry.elda.org/services/119

Table 6 WS Details for IULA Preprocess and IULA Tokenizer

4.2.2 IULA Tagger

The IULA Tagger web service provides functionalities for **PoS tagging and Lemmatization** of Spanish. The service uses the IULA PoS Tagger (Vivaldi, 2009), an adaptation of the TreeTagger (Schmidt, 1994) that integrates a lemmatizer and uses the IULA tagset for Spanish. The accuracy for both tagging and lemmatization is 98% tested against a 100K words test set.

URL	http://kurwenal.upf.edu/soaplab2-axis/#morphosintactic_tagging.iula_tagger_row
WSDL	http://kurwenal.upf.edu/soaplab2-axis/services/morphosintactic_tagging.iula_tagger?wsdl,
PANACEA Catalogue Entry	http://registry.elda.org/services/118
PANACEA MyExperiment	http://myexperiment.elda.org/workflows/5, http://myexperiment.elda.org/workflows/22,

Workflow(s) using the WS	http://myexperiment.elda.org/workflows/23

Table 7 WS Details for IULA Tagger

4.2.3 Freeling

Freeling is an open source language analysis tool suite, developed by the TALP Research Center of the Universitat Politècnica de Catalunya and released under the GPL.

The freeling_tagging web service makes use of Freeling for annotating Spanish[17] texts with PAROLE[18] compatible morphosyntactic descriptions. Since Freeling is a comprehensive tool offering many functionalities it has also been used in services for tokenization and parsing (freeling_tokenizer and freeling_dependency, respectively).

URL	Tokenizer: http://ws04.iula.upf.edu/soaplab2-axis/#tokenization.freeling_tokenizer_row PoS tagging: http://ws04.iula.upf.edu/soaplab2-axis/#morphosintactic_tagging.freeling_tagging_row Dependency parsing: http://ws04.iula.upf.edu/soaplab2-axis/#syntactic_tagging.freeling_dependency_row
WSDL	(wsdls for each service can be found on the Panacea Catalogue)
PANACEA Catalogue Entry	Tokenizer: http://registry.elda.org/services/101 ;PoS Tagging: http://registry.elda.org/services/99 ;Dependency parsing: http://registry.elda.org/services/105
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/5 , http://myexperiment.elda.org/workflows/22 , http://myexperiment.elda.org/workflows/23

Table 8 WS Details for Freeling

The Freeling services accept a set of optional parameters, which is briefly described in Table 9. Additional documentation can be found at the project's site: http://nlp.lsi.upc.edu/freeling/ doc/userman/html/node74.html.

Parameter name	**Semantics**
flush	Consider each newline as a sentence end
ner	Type of NE recognition is to be performed (basic, bio, none)
noafx	Whether to perform affix analysis
nodate	Whether to detect dates and time expressions

[17] The Freeling service hosted by UPF can be used for POS tagging and lemmatisation of, among others, English and Catalan.

[18] http://nlp.lsi.upc.edu/freeling/doc/userman/parole-es.html,

nodict	Whether to perform dictionary search
noloc	Whether to perform multiword detection
nonumb	Whether to perform number detection
noprob	Whether to perform probability assignment
nopunt	Whether to perform punctuation detection
noquant	Whether to perform quantities detection

Table 9 Optional parameters for Freeling services

4.2.4 Freeling 3

Freeling 3 is the new version of Freeling and it has been deployed as different web services following the criteria used to deploy the previous version. Some of the new features of Freeling are:

- Full UTF-8 support.
- New languages: Russian and ancient Spanish (XII-XVI).
- Reorganized ML components. Inclusion of SVM models thaks to libsvm.
- Simpler installation: External dependencies only from out-of-the-box libboost packages.
- Compilable in Linux, MacOSX, and Windows (with MSVC).
- Improved server mode.

Freeling performs different functions which can be used in PANACEA thanks to the web services: tokenization, sentence splitting, PoS tagging, morphosyntactic tagging, chunking, dependency parsing and NER.

URL	Tokenizer: http://ws04.iula.upf.edu/soaplab2-axis/#tokenization.freeling3_tokenizer_row Sentence Splitter: http://ws04.iula.upf.edu/soaplab2-axis/#segmentation.freeling3_sentence_splitter_row PoS tagging: http://ws04.iula.upf.edu/soaplab2-axis/#morphosintactic_tagging.freeling3_tagging_row Dependency parsing: http://ws04.iula.upf.edu/soaplab2-axis/#syntactic_tagging.freeling3_dependency_row
WSDL	(wsdls for each service can be found on the Panacea Catalogue)
PANACEA Catalogue Entry	Tokenizer: http://registry.elda.org/services/238 Sentence Splitter: http://registry.elda.org/services/239 PoS Tagging: http://registry.elda.org/services/237 Dependency parsing: http://registry.elda.org/services/240
PANACEA MyExperiment	http://myexperiment.elda.org/workflows/75

Workflow(s) using the WS	

Table 10 WS Details for Freeling 3

4.2.5 MALT parser

An instance of Malt parser (http://www.maltparser.org/) for Spanish trained with the Iula treebank developed in the Metanet4you project.

The tool performs PoS tagging with FreeLing and then performs the dependency parsing using Malt parser. The output follows the CoNLL format.

Detailed documentation:
http://ws02.iula.upf.edu/panacea/documentation/ws04/WS_malt_parser.pdf

URL	http://ws04.iula.upf.edu/soaplab2-axis/#syntactic_tagging.malt_parser_row
WSDL	http://ws04.iula.upf.edu/soaplab2-axis/services/syntactic_tagging.malt_parser?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/249

Table 11 WS Details for MaltParser

4.2.6 Twitter NLP

A Java-based tokenizer and part-of-speech tagger for Twitter English, its training data of manually labelled POS annotated tweets, a web-based annotation tool, and hierarchical word clusters from unlabeled tweets.

The tool was developed by Noah's ARK group is Noah Smith's research group at the Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

URL	http://ws04.iula.upf.edu/soaplab2-axis/#morphosintactic_tagging.twitter_nlp_row
WSDL	http://ws04.iula.upf.edu/soaplab2-axis/services/morphosintactic_tagging.twitter_nlp?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/261

Table 12 WS Details for Twitter NLP

4.2.7 Anonymizer

It is a tool based on the NER tool of the Freeling3 web service that can detect and substitute named entities by codes. It can be used to anonymize texts.

URL	http://ws04.iula.upf.edu/soaplab2-axis/#named_entity_recognition.anonymizer_row
WSDL	http://ws04.iula.upf.edu/soaplab2-axis/services/named_entity_recognition.anonymizer?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/252

Table 13 WS Details for Anonymizer

4.3 Tools for Italian hosted by CNR

4.3.1 Freeling Italian

The freeling_it web service hosted by CNR provides functionalities for **POS tagging and Lemmatization** using the Italian version of FreeLing. The FreeLing project was created and is currently led by Lluís Padró; the tools were developed at the TALP Research Center of the Universitat Politècnica de Catalunya. The package consists of several language analysis libraries. The *analyzer* library contains a complete pipeline for the tokenization, sentence splitting, lemmatization, tagging and morphological analysis of text in several languages, including Italian. FreeLing reads from standard input and produces results to standard output. The input format is plain text (UTF-8 or ISO) and the output is a tabbed file where sentences are separated by an empty line. Each token is stored in a separate line, with lemma and POS information added to the token and separated by tabs. For further details, see Atserias et al. (2006), Padró et al. (2010) and the Freeling page at http://nlp.lsi.upc.edu/freeling.

Sentence splitting and tokenization are rule-based. Lemmatization is based on an Italian dictionary that is extracted from the Morph-it! lexicon developed at the University of Bologna. The lexicon contains over 360,000 forms corresponding to more than 40,000 lemma-POS combinations. POS disambiguation is performed using an HMM tagger, which, in the case of Italian, was trained on a manually annotated corpus of 300,000 words. The declared accuracy for Italian is 97% (Atserias et al. 2006). POS tags are represented by alphanumeric values that encode the EAGLES tagset. Although no documentation of the Italian tagset is provided by TALP, the tagset is similar to the one for Spanish found at http://nlp.lsi.upc.edu/freeling/doc/userman/parole-es.html

URL	http://wiki2.ilc.cnr.it:8080/soaplab2-axis/#panacea.freeling_it_row
WSDL	http://wiki2.ilc.cnr.it:8080/soaplab2-axis/typed/services/panacea.freeling_it?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/139
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/24

Table 14 WS Details for freeling_it

The freeling_it service accepts a set of optional parameters regarding multiword detection, named-entity and output-format. These parameters are briefly described in Table 15.

Parameter name	**Semantics**
multiword	Enables/disables multiwords detection (*yes/no*)
ner	Type of NE recognition is to be performed (*none/basic*, default is *none*)

output-format	Level of analysis to display in the results (*token/splitted/tagged,* default is *tagged*)

Table 15 Optional parameters for freeling_it

4.3.2 Dependency Parser for Italian

The TPC_Desr_dependencyparser is a service for shallow dependency parsing of Italian. It implements the DESR statistical parser, an open source shift-reduce dependency parser developed at the University of Pisa (Attardi 2006, Attardi et al. 2007)[19], trained for Italian. As such it expects POS tags to follow the TANL tagset.

The input format is CoNLL (with TANL tagset[20]) for tokenization, lemmatisation, and morphological analysis, and the output format is CoNLL.

URL	http://langtech3.ilc.cnr.it:8080/soaplab2-axis/#panacea.desr
WSDL	http://langtech3.ilc.cnr.it:8080/soaplab2-axis/services/panacea.desr?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/210
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/53 http://myexperiment.elda.org/workflows/58

Table 16 WS Details for TPC_Desr_dependencyparser_it

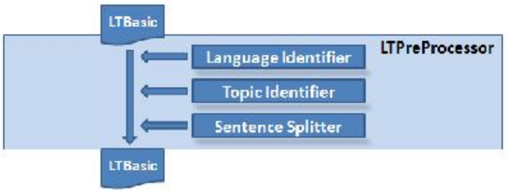
Parameter name	**Semantics**
input	CoNNL annotated text up to pos tag and morphological analysis, or list of URLs to such files
language	The parameter is fixed and must be = IT

[19] https://sites.google.com/site/desrparser/

[20] http://medialab.di.unipi.it/wiki/POS_Tagset
http://medialab.di.unipi.it/wiki/Lexical_Morphological_Annotations

4.4 Tools for German hosted by Linguatec

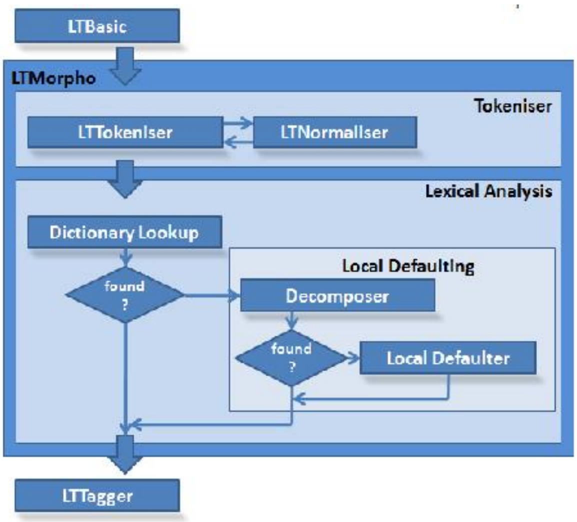
Linguatec has made available two groups of tools: The first one belongs to the **LTPreProc** component, and sets attributes and markups into the XML structure. Members of this group are the Language Identifier (not needed for PANACEA), the Topic Identifier, and the SentenceSplitter.



The second group of tools belongs to the **LTMorpho** component. It consists of the tools for tokenisation and lemmatisation; Tokenisation includes a normalisation component, and lemmatisation consists of the lexical analysis, decomposer and defaulter tools.

While decomposer and defaulter are part of the lexical analysis, they can also be used independently, operating on file input. This would be interesting in PANACEA as it enables the creation of dictionary entry annotations (like part-of-speech and lemma), based on local defaulting, i.e. without any context provided.

The **LTTagger** component disambiguates ambiguous tokens, using a rule-based approach and delivers results in the XCES format.



4.4.1 Topic Identification

The task of the topic identifier is to assign a topic to a document.

It uses two main language resources: a taxonomy of topics, and the features for each topic.

- The taxonomy of the topic identifier consists of about 40 topics, organized as a hierarchy. Examples are 'art', 'technology', 'wood processing' as subtopic of 'material' etc. The taxonomy has been used in Linguatec's MT products.
- Features: Each node in the taxonomy is described by a set of weighted features; features are given as lemmata instead of text forms, and they can contain multiwords; using multiwords improves results significantly, especially for English. The size of the feature file is about 190.000 for German, and about 40.000 for English (lemmata).

Modus Operandi: For each incoming document (part), the key features are identified; then, on document and/or paragraph level, the topic with the highest weight, and above a certain threshold, is selected. In case topics are very close, more than one topic is assigned. The system tries to only assign a topic if there is enough evidence for it; otherwise the text is left in the general domain.

URL	
WSDL will be:	http://80.190.143.163:8080/panaceaV2/services/LTTopicIdentifier?wsdl

Table 17 WS Details for LT Topic Identifier

4.4.2 Sentence Splitting

The task of the LTSentenceSplitter is to detect sentence boundaries and insert <s> …. </s> markups in the input text.

The sentence splitter uses the following language resources for each language:

- lists of startwords. Startwords are words that indicate a sentence beginning if capitalised (like '*The*').
- lists of endwords. Endwords are words that frequently occur before a sentence-final punctuation (i.e. they indicate that a following period is really a sentence-end)
- lists of abbreviations. Abbreviations are further subcategorised into those that mostly occur in final position (like 'etc.'), those that occur nearly always in non-final position (like 'Dr.'), and others that occur that can be used both ways.

The startwords and endwords have been collected from a corpus analysis of the WACky corpus, and manually corrected. They comprise about 12.000 entries per language.

Modus operandi: The SentenceSplitter identifies patterns which indicate a sentence boundary, checking contexts around punctuations in a variable-length window.

WSDL	http://80.190.143.163:8080/panaceaV2/services/SentenceSplitter?wsdl

Table 18 WS Details for LT Sentence Splitter

4.4.3 Tokeniser / Normaliser

In LTMorpho, tokens are basically defined as units which can be looked up in a dictionary, or can be given a linguistic description (by defaulting etc.). So the tokeniser prepares the lexical analysis.

This is why some normalisation is required here as well. If the dictionary contains entries like '*normalisation*' or '*fließtext*', then input like 'normalization' or '*Fliesstext*' would not match. Normalisation is therefore a component which is required to increase the chances of a token to be found in the dictionary.

The tokeniser uses normalisation resources for German and English. These are simple replacement table, replacing 'wrong' (i.e. not lexicon-compatible) spelling by 'good' spelling. Phenomena covered are British/American alterations in English, and old-new orthography (old '*schuß*' -> new '*schuss*'), as well as ascii->'real' words ('*groesser*' -> '*größer*'). The lists are between 2.000 (English) and 15.500 (German) entries in size.

Modus operandi: The tool first splits a text into character classes. Only characters of the same class are linked into one token: alphanumerics and digits are concatenated, while for others each character is a single token.

The tool then normalises the single tokens, by collapsing multiple tokens into a single one if required (e.g. for URLs, digit+punctuation, letter+hyphen etc.), and normalises the resulting tokens for orthography, case information etc.

Relevant phenomena addressed are:

- letters and digits: *111s* or *111's*, ordinals like *12th*, *2nd*, hours like *5pm* or *3.40am*, units like *237km/h*
- punctuations inside of tokens, like in the case of URLs
- digits and punctuations (*2,5:2,5* or *3:1* or *12.12.2112*)

In these cases, tokens are formed from several character classes, and have to be re-merged into one token.

WSDL	http://80.190.143.163:8080/panaceaV2/services/LTTokenizer?wsdl

Table 19 WS Details for LT Tokenizer

4.4.4 Lemmatiser – Lexical Analysis

The first component of this tool is a dictionary lookup. The tool tries to find a token in the dictionary, and extract from it the lemma (i.e. the canonical form of the token), and linguistic annotations, i.e. elements of a tagset.

Tagset: Lexical Analysis is based on a tagset. LTMorpho provides three Linguatec developed tagsets building upon each other:

- the **Basic Tagset (BTag)** consists of the main parts of speech; it has 12 elements. It is used for deep parsing as well.
- the **Standard tagset (STag)**, which defines grammatical categories on top of the basic tagset, and based on the syntactic distribution of the described elements (like: common noun, full finite verb, etc.); it has 88 elements.
- the **Extended Tagset (XTag)**, which gives additional morphological information (like gender, number, tense etc.) on top of the Standard Tagset.

An example for a member of the extended tagset would be: '*PnPo-GmNpCaP2*'. Basic tag is pronoun (Pn), standard tag is possessive pronoun (PnPo), extended tag uses the features: Gender=masculine, Number=plural, Case=accusative, Person=2.

Language Resources: The main challenge for the lexical analyser in a shallow analysis environment is the size and organisation of the dictionary, as a single point of maintenance is a basic requirement for each dictionary setup. In the LTLemmatiser, a full word dictionary is used, compiled from a Basic Lemma Dicitionary. Depending on the tagset used, the size of the dictionary differs significantly: The German Full Word Dictionary has 5.470.000 entries with the Basic Tagset, 5.790.000 entries with the Standard Tagset, and 17.750.000 entries with the Extended Tagset, all for 3.700.000 lemmata.

Modus Operandi: The LTLexLookup component does a search for a dictionary entry, and returns the linguistic annotation found there. It analyses 31.700 tokens per second, on a standard PC.

WSDL	http://80.190.143.163:8080/panaceaV2/services/LTLemmatizer?wsdl

Table 20 WS Details for LT Lemmatizer

4.4.5 Decomposer

The entries that are not in the dictionary need to be further analysed, in order to reduce the amount on unknown types. As composition is one of the new word formation processes in German, a decomposer component is used to analyse unknown words, and copy all relevant linguistic information from the head of the compound.

The decomposer is part of the lemmatizer; however, it can also be used as a stand-alone tool, the input being a list of words.

Tagset: It turned out that the decomposer needs a specific tagset, reflecting the distributional properties of words participating in decomposition. For example, most function words behave the same way in compounds, while some verbal elements need very detailed description. Therefore, the decomposer uses a tagset that reflects such properties. The tagset consists of 61 elements and is described in the LT Documentation.

The decomposer uses the following language resources:

- Decomposer **Dictionary**: The dictionary of the decomposer contains all morphemes that can participate in a decomposition. Each entry consists of the following information elements: a text form; a lemma; a DTag (one of the decomposer tags); additional information. The decomposer dictionary contains about 460.000 entries.
- Decomposer **Irregular Dictionary**: This is a dictionary of irregular forms and exceptional decompositions. It consists of about 11.000 entries. Each entry is identified by <textform, lemma, POS> and gives the elements of which the decomposition consists.
- Decomposer **Transition Table**: This is a matrix that controls which decomposer tag can follow a given other tag. It is used to decide if a candidate decomposition element can follow an existing element in the chart. The matrix is 61 x 61 in size, and has binary values, i.e. it either allows or forbids a given transition.
- Decomposer **Disambiguation Rules**: There are many cases where several decompositions are possible for a given input word. In this case, the system must try to find the best (correct) decomposition. To do this, filter rules are applied. There are about 20 such rules. They are encoded into a numeric schema that is applied during decomposition.

Modus Operandi: The first step is a chart-based breadth-first analysis whereby from a given point in the input string all lexically possible continuations are checked. Each continuation candidate undergoes a check in the transition table to find if such a continuation is possible; if so then the candidate is inserted into the chart.

Next, the different decomposition hypotheses are built, and scored according to the local and global scores given by the rule scoring.

Finally, the hypotheses are filtered, compound parts of irregular entries are replaced by the decompositions in the irregular dictionary, and the hypotheses are ranked according to their scores.

Different output formats are produced for different purposes, among others a prettyprint format, and a format that is compatible to the input requirements of the MOSES MT system.

The decomposer analyses about 11.000 words per second. It runs on a file of unknowns extracted from the lemmatiser output by a small webservice 'LTUnkExtractor'.

WSDL will be	http://80.190.143.163:8080/panaceaV2/services/LTDecomposer?wsdl http://80.190.143.163:8080/panaceaV2/services/LTUnkExtractor?wsdl

Table 21 WS Details for LT Decomposer

4.4.6 Local Defaulter

In case the decomposer returns a string as 'unknown' this token needs to be annotated with linguistic information somehow; a tagger would not like a tag 'unknown' occurring in all kinds of possible contexts. It is the task of the defaulter to provide such annotations. The component is called '*local* defaulting' as only the unknown string itself is considered, and no context information is used. Corpus-based extraction of information would be called '*contextual* defaulting'.

The following language resources are used:

- Lists of foreign words: They are used to check if an unknown word comes from a foreign language. For this purpose, the word lists of the Language Identifier are re-used. Many unknown tokens in the test corpus are foreign language words.
- Default endings: These resources are created by a training component that correlates some linguistic information with string endings. Such information include: Tags (BTag, STag, XTag), lemma formation, gender defaulting, etc. It takes a list of example words, and linguistic annotations of them, and produces the longest common ending strings for this annotation.
  For the defaulting of the tag, the training component produces about 470 K correlations of endings and tags assignments; in the case of homographs, it also gives the relative weights of the different tags against each other, based on the training data.[21]

So far, only part-of-speech defaulting is done; other defaulting operations will concern lemma, gender, and others.

Modus operandi: At runtime, three defaulting steps are tried. First, the foreign word dictionary is looked up, to check if the unknown string is a foreign word. In case the word is found it is marked as (a special kind of) 'Common Noun'[22]. Next, a strategy to identify acronyms and other non-words, consisting of a mixture of digits, uppercase and lowercase letters is applied; it is supposed to cover strings like '*EU/2/08/091/004*' or '*CRF12*'. As for tag assignment, such strings can be common nouns ('*AKW'* = '*Atomkraftwerk'*) but also proper nouns ('CSU' = '*christlich soziale Union*'). Therefore, they are treated as homographs, leaving it to later components to tag them properly. Finally, the string undergoes local defaulting, looking up its ending in the defaulter resource. This will *always* produce an assignment. The STag (or a set thereof, in case of homographs) is returned.

[21] For the current setup, only the STag defaulter is used; following versions will default more features if the approach turns out to be viable.

[22] A tag like "FW" as in the STTS tagset does not really help, as its distribution would be completely unclear. Using the tags of the words in their respective language is not a good solution either; so classifying them as nouns is considered to do the least damage.

In a setup where a tagger is available, the defaulter is integrated into the tagger for cases where lexical analysis (incl. decomposition) returns an 'unknown' POS.

The defaulter analyses about 33K tokens per second. It runs on a file of unknowns extracted from the decomposer output by a small webservice 'LTUnkExtractor'.

WSDL will be:	http://80.190.143.163:8080/panaceaV2/services/LTDefaulter?wsdl http://80.190.143.163:8080/panaceaV2/services/LTUnkExtractor?wsdl

Table 22 WS Details for LT Defaulter

4.4.7 LTTagger

This component does shallow syntactic analysis, by assigning a single part of speech to each token in the sentence, using the Standard tagset of about 90 tags.

The tagger operates on a sentence lattice, containing a list of tokens, each token being a list of readings. The task of the tagger is to reduce the lattice to a list, i.e. remove readings which are incorrect in the given context, and keep just one reading per token. Removing readings is done by rules; the tagger is neither probabilistic (TreeTagger, TnT) nor transformational (Brill).

The main language resources used are tagging rules. They consist of condition–action patterns, conditions being configurations of tags, and actions performing disambiguation operations (mainly: KEEP this tag and remove all others, or REMOVE this tag and keep the others). The current version supports German only, and has about 400 tagging rules.

Modus Operandi: After initial preprocessing and pruning of 'impossible' readings (mainly for missing multiword parts), the tagger analyses the sentence lattice from left to right, firing all disambiguation rules starting at the current node; these rules simplify the sentence lattice to the right. Remaining ambiguities are resolved using the local probabilities of the parts-of-speech involved. A final cleanup step builds proper lemmata (e.g. for German split verbs).

Although tests are still ongoing it seems that the current version matches probabilistic taggers for German (TreeTagger, TnT) in quality (~96% accuracy).

WSDL will be:	http://80.190.143.163:8080/panaceaV2/services/LTTagger?wsdl

Table 23 WS Details for LT Tagger

4.5 Tools for Greek hosted by ILSP

This section discusses services and corresponding basic ILSP NLP tools for tokenization and sentence splitting, tagging and lemmatization. More information can be found in Prokopidis et al. (2011). All tools are implemented in the Apache UIMA Java framework. They are OS-independent applications that accept input and produce output in UTF-8 and ISO-8859-7. Among other formats, the tools can process and generate PANACEA TO1 and TO2 formats. The tools are made available in the PANACEA platform as free services for research purposes.

4.5.1 ILSP Sentence Splitter and Tokenizer

ILSP Sentence Splitter and Tokenizer (ILSP SST) identifies paragraph, sentence and token boundaries in Greek texts. Identifying token and sentence boundaries involves resolving ambiguity in punctuation use since structurally recognizable tokens may contain ambiguous punctuation; this may be the case for numbers, alphanumeric references, dates, acronyms and abbreviations. Following common practice, the tokenizer makes use of a regular expression definition of words, coupled with precompiled, semi-automatically collected gazetteers of abbreviations. At a final stage, the tool detects the type of tokens, classifying them in one of the categories of Table 24.

TOKEN TYPE	**Description**	**Example**
TOK	The default token type	
DATE	Date	21-10-2008, 09/12/10
ENUM	Enumerators	1., 1.α., i.
CPUNCT	Closing punctuation	), »,], ’
OPUNCT	Opening punct.	(, «, [
PTERM	Terminal punctutation	? ; (GREEK QUESTION MARK) !... ?...
PTERM_P	Potentially terminal punctutation	. ... : ! ; (SEMI COLON)
PUNCT	Other punctutation	* -
DIG	Digit	1.1000, 1.234,567.890, 1,234.567,890
INIT	Initial	Μιλτ.
NBABBR	Abbreviations that cannot appear at the end of a sentence	κ.(ύριος/α), κκ.(ύριοι), σ.(ελίδα)
ABBR	Abbreviation	κλπ, κοκ, ΟΗΕ, δολ., δρχ

Table 24 Token types recognized by ILSP SST

URL	http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_sst_row
WSDL	http://nlp.ilsp.gr/soaplab2-axis/typed/services/ilsp.ilsp_sst?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/131
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/20 (ILSP Basic NLP Tools)

Table 25 WS Details for ILSP SST

4.5.2 ILSP FBT Tagger

ILSP FBT POS Tagger is an adaptation of the Brill tagger trained on Greek texts annotated for POS and several morphosyntactic features. ILSP FBT uses a PAROLE compatible tagset of 584 different tags, which capture the morphosyntactic particularities of the Greek language. See Table 26 for the basic POS tags used by the tagger[23].

ILSP FBT assigns initial tags by looking up tokens in a lexicon created from a manually annotated corpus of approx. 455K tokens. The lexicon is augmented by ILSP manually compiled lexica. A suffix lexicon is used for initially tagging unknown words. 799 contextual rules are then applied to correct initial tags. When a token exists in the known words lexicon, rules can change its tag only if the resulting tag exists in the token's entry in this lexicon. The tool's accuracy has been tested against a 90K corpus with manually annotated POS tags. The tagger's accuracy reaches 97.48 when only basic POS is considered. When all features (including, for example, gender and case for nouns, and aspect and tense for verbs) are taken into account, the tagger's accuracy is 92.52. The tool can be accessed and integrated via the information from Table 27.

POS	Description
Ad	Adverb
Aj	Adjective
AsPpPa	Preposition + Article combination
AsPpSp	Simple preposition
AtDf	Definite article
AtId	Indefinite article
CjCo	Coordinating conjunction
CjSb	Subordinating conjunction
COMP	A composite word form
CPUNCT	Closing punctuation

POS	Description
OPUNCT	Opening punctuation
PnDm	Demonstrative pronoun
PnId	Indefinite pronoun
PnIr	Interrogative pronoun
PnPe	Personal pronoun
PnPo	Possessive pronoun
PnRe	Relative pronoun
PnRi	Relative indefinite pronoun
PTERM	Terminal punctuation
PtFu	Future particle

[23] For a full description of the tagset, including, for example, features for noun case and verb tense, see http://sifnos.ilsp.gr/nlp/tagset_examples/tagset_en/

POS	Description	POS	Description
DATE	Date	PtNg	Negative particle
DIG	Digit	PtOt	Other article
ENUM	Enumeration element	PtSj	Subjunctive particle
INIT	Initial	PUNCT	Other punctuation
NmCd	Cardinal numeral	RgAbXx	Abbreviation
NmCt	Collective numeral	RgAnXx	Acronym
NmMl	Multiplicative numeral	RgFwOr	Foreign word in its original form
NmOd	Ordinal numeral	RgFwTr	Transliterated foreign word
NoCm	Common noun	VbIs	Impersonal verb
NoPr	Proper noun	VbMn	Main verb

Table 26 Basic POS tags together with their subcategorizations

URL	http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_fbt_row
WSDL	http://nlp.ilsp.gr/soaplab2-axis/typed/services/ilsp.ilsp_fbt?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/128
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/20 (ILSP Basic NLP Tools)

Table 27 WS Details for ILSP FBT

4.5.3 ILSP Lemmatizer

Following POS tagging, a lexicon-based lemmatizer retrieves lemmas from ILSP's Greek Morphological Lexicon[24]. This resource contains 66K lemmas, which in their expanded form extend the lexicon to approximately 2M different entries.

When a token under examination exists in the lexicon with a unique lemma, this lemma is returned. When two or more lemmas exist, the lemmatizer uses information from the POS tags assigned by ILSP FBT to disambiguate. For example, the token ενοχλήσεις will be assigned the lemma ενοχλώ "to annoy", if tagged as a 2nd person singular, present tense verb; on the other hand, it will be assigned the lemma ενόχληση "annoyance", if it is tagged as a common plural noun. The lemmatizer can be accessed and integrated via the information from Table 28.

URL	http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_lemmatizer_row
WSDL	http://nlp.ilsp.gr/soaplab2-

[24] http://www.ilsp.gr/en/services-products/langresources/item/32-ilektronikomorfologiko

	axis/typed/services/ilsp.ilsp_lemmatizer?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/129
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/20 (ILSP Basic NLP Tools)

Table 28 WS Details for ILSP Lemmatizer

4.5.4 ILSP Dependency Parser

The ILSP Dependency Parser is based on the MaltParser system for dependency parsing (Nivre et al, 2007). The parser has been trained on the Greek Dependency Treebank (Prokopidis et al, 2005), a resource that comprises data annotated at several linguistic levels. As of 2012, GDT[25] contained 118K tokens in 4963 sentences, while more annotated texts on different domains are being added. The scheme used during manual annotation includes 25 main relations (Table 29) and is based on an adaptation of the guidelines for the Prague Dependency Treebank[26]. The scheme allows for simple and intuitive descriptions of structures common in languages which, like Greek, exhibit a flexible word order. Since dependency relations are directly encoded, without the presupposition of any default constituent structure from which all others are derived, representation for the main relations in a sentence is straightforward. Non-projective structures are also allowed in the scheme.

Dep. Rel	Description	Dep. Rel.	Description
`Pred`	Main sentence predicate	`Coord`	A node governing coordination
`Subj`	Subject	`Apos`	A node governing apposition
`Obj`	Direct object	`*_Co`	A node governed by a Coord
`IObj`	Indirect object	`*_Ap`	A node governed by an Apos
`Adv`	Adverbial dependent	`AuxC`	Subord. conjunction node
`Atr`	Attribute	`AuxP`	Prepositional node
`ExD`	A node whose parent node is not present in the sentence (ellipsis)	`AuxV`	Particles or auxiliary verbs attached to a verb

Table 29 Common dependency relations in the Greek Dependency Treebank

In n-fold experiments using this dependency set and automatically assigned POS and lemmas, we have trained models on the GDT that showed an overall labeled attachment score (i.e. the proportion of tokens attached to the correct head and assigned the correct dependency relation) of 76.42% and an overall unlabeled attachment score of 84.52%. Precision and recall for the subject relation reached 83.49% and 89.46% respectively. The parser can be accessed and integrated via the web service detailed in Table 30.

URL	http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_depparser _row
WSDL	http://nlp.ilsp.gr/soaplab2-axis/typed/services/ilsp.ilsp_depparser?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/178
PANACEA MyExperiment	http://myexperiment.elda.org/workflows/88 (ILSP NLP Tools)

[25] http://gdt.ilsp.gr
[26] http://ufal.mff.cuni.cz/pdt2.0/

Workflow(s) using the WS	

Table 30 WS Details for ILSP Dependency Parser

4.5.5 ILSP Named Entity Recognizer

The ILSP Named Entity Recognizer WS for Greek uses MENER (Maximum Entropy Named Entity Recognizer). MENER is a Maximum Entropy approach to NE recognition that combines sentence based local evidence with document based global evidence (a set of features drawn from other occurrences of a word within the same document). The system is compatible with the ACE (Automatic Content Extraction) scheme, catering for the recognition and classification of the following types of NEs: person (PER), organization (ORG), and location (LOC) (Giouli et al., 2006). When processing news & politics data from various sources (NE classes: LOC, ORG, PER) MENER has shown a 93% F-measure overall.

The ILSP NERC WS can be accessed and integrated via the information from Table 31.

URL	http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_nerc_row
WSDL	http://nlp.ilsp.gr/soaplab2-axis/typed/services/ilsp.ilsp_nerc?wsdl
PANACEA Catalogue Entry	http://registry.elda.org/services/179
PANACEA MyExperiment Workflow(s) using the WS	http://myexperiment.elda.org/workflows/88 (ILSP NLP Tools)

Table 31 WS Details for ILSP NERC

5 Publications list

Three conference papers have been produced in the context of WP4:

Mastropavlos, Nikos; Papavassiliou, Vassilis. (2011). Automatic Acquisition of Bilingual Language Resources. In *Proceedings of the 10th International Conference on Greek Linguistics*. Komotini, Greece: 1-4 September 2011.

Pecina, Pavel; Toral, Antonio; Way, Andy; Papavassiliou, Vassilis; Prokopidis, Prokopis; Giagkou, Maria. (2011). Towards using web-crawled data for domain adaptation in statistical machine translation. In Forcada, Mikel L.; Depraetere, Heidi and Vandeghinste (Eds.) In *Proceedings of the 15th conference of the European Association for Machine Translation* (EAMT 2011). Leuven, Belgium: 30-31 May 2011, pp.297-304. **(***In collaboration with PANACEA WP5, Parallel corpus & derivatives***)**

Prokopidis, Prokopis; Georgantopoulos, Byron; Papageorgiou, Haris. (2011). A suite of NLP tools for Greek. In *Proceedings of the 10th International Conference on Greek Linguistics*. Komotini, Greece: 1-4 September 2011.

6 References

Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., Padró, M. (2006). FreeLing 1.3*:* Syntactic and semantic services in an open-source NLP library. In *Proceedings of the Fifth conference on International Language Resources and Evaluation (LREC'06)*. Paris, France European Language Resources Association (ELRA).

Attardi, G. (2006). Experiments with a Multilanguage Non-Projective Dependency Parser. In *Proceedings of the Tenth Conference on Natural Language Learning*, New York, (NY).

Attardi, G., Chanev, M.& Dell'Orletta, F. (2007). Tree Revision Learning for Dependency Parsing, In *Proceedings of the Human Language Technology Conference*.

Baroni M, Bernardini S, Ferraresi A, Zanchetta E. (2009). The WaCky Wide Web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation* 43(3):209–226

Bergmark D., Lagoze C., Sbityakov A. (2002) Focused crawls, tunneling, and digital libraries. *In: Agosti M, Thanos C (eds) Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science*, vol 2458, Springer Berlin, Heidelberg, pp 49–70

Cho, J., Garcia-Molina, H., and Page, L. (1998) Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30, 1–7, 161–172.

Voula Giouli, Alexis Konstandinidis, Elina Desypri, Harris Papageorgiou (2006). Multi-domain Multi-lingual Named Entity Recognition: Revisiting & Grounding the resources issue. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*. Genoa, Italy. European Language Resources Association (ELRA).

Héctor Martínez, Jorge Vivaldi and Marta Villegas (2010). Text handling as a Web Service for the IULA processing pipeline" in Calzolari, Nicoletta et al. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Paris, France European Language Resources Association (ELRA).

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S. and Marsi, E. (2007). MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2), 95-135.

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

Lluís Padró and Miquel Collado and Samuel Reese and Marina Lloberes and Irene Castellón. (2010). FreeLing 2.1: Five Years of Open-Source Language Processing Tools. In *Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010)*. La Valletta, Malta. European Language Resources Association (ELRA).

Pecina, P., A. Toral, A.Way, V. Papavassiliou, P. Prokopidis, and M. Giagkou. (2011). Towards Using Web-Crawled Data for Domain Adaptation in Statistical Machine Translation. In *Proc. of the 15th Annual Conference of the European Associtation for Machine Translation*, pp 297–304, Leuven, Belgium.

Pavel Pecina, Antonio Toral, Vassilis Papavassiliou, Prokopis Prokopidis, and Josef van Genabith. (2012). Domain Adaptation of Statistical Machine Translation using Web-Crawled Resources: A Case Study. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, Trento, Italy, pp. 145-152, 2012.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL.*

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

Prokopis Prokopidis, Elina Desypri, Maria Koutsombogera, Haris Papageorgiou, and Stelios Piperidis (2005). Theoretical and Practical Issues in the Construction of a Greek Dependency Treebank. In Montserrat Civit, Sandra Kubler, and Ma. Antonia Marti, editors, *Proceedings of The Fourth Workshop on Treebanks and Linguistic Theories*, pages 149-160, Barcelona, Spain, 2005.

Prokopidis, P., Georgantopoulos, B., and Papageorgiou, H. (2011). A suite of NLP tools for Greek. *10th International Conference on Greek Linguistics*. Komotini, Greece.

Qin J, Chen H. (2005). Using Genetic Algorithm in Building Domain-Specific Collections: An Experiment in the Nanotechnology Domain. *In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, IEEE Computer Society, Washington, DC, USA, p 102

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing (NeMLaP-1)*, pp. 44–49.

Theobald, M., Siddharth, J., and Paepcke, A. (2008). SpotSigs: Robust and Efficient Near Duplicate Detection in Large Web Collections. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and development in information retrieval*.

Vivaldi P.J. (2009) Corpus and exploitation tool: IULACT and bwanaNet. In *I International Conference on Corpus Linguistics (CICL 2009)*, pp 224-239. Universidad de Murcia.

7 Appendix

7.1 FMC configuration file

```
<?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>3</value>
           <description>Minimum number of terms that must exist in clean
                  content of each web page in order to be stored.</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>
    <max_depth>
           <value>4</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>
   <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" />
           <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>32</value>
           <description>Max number of fetching threads for each host</description>
    </max_connections_per_host>
    <max_fetched_per_host>
           <value>50000</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>
```

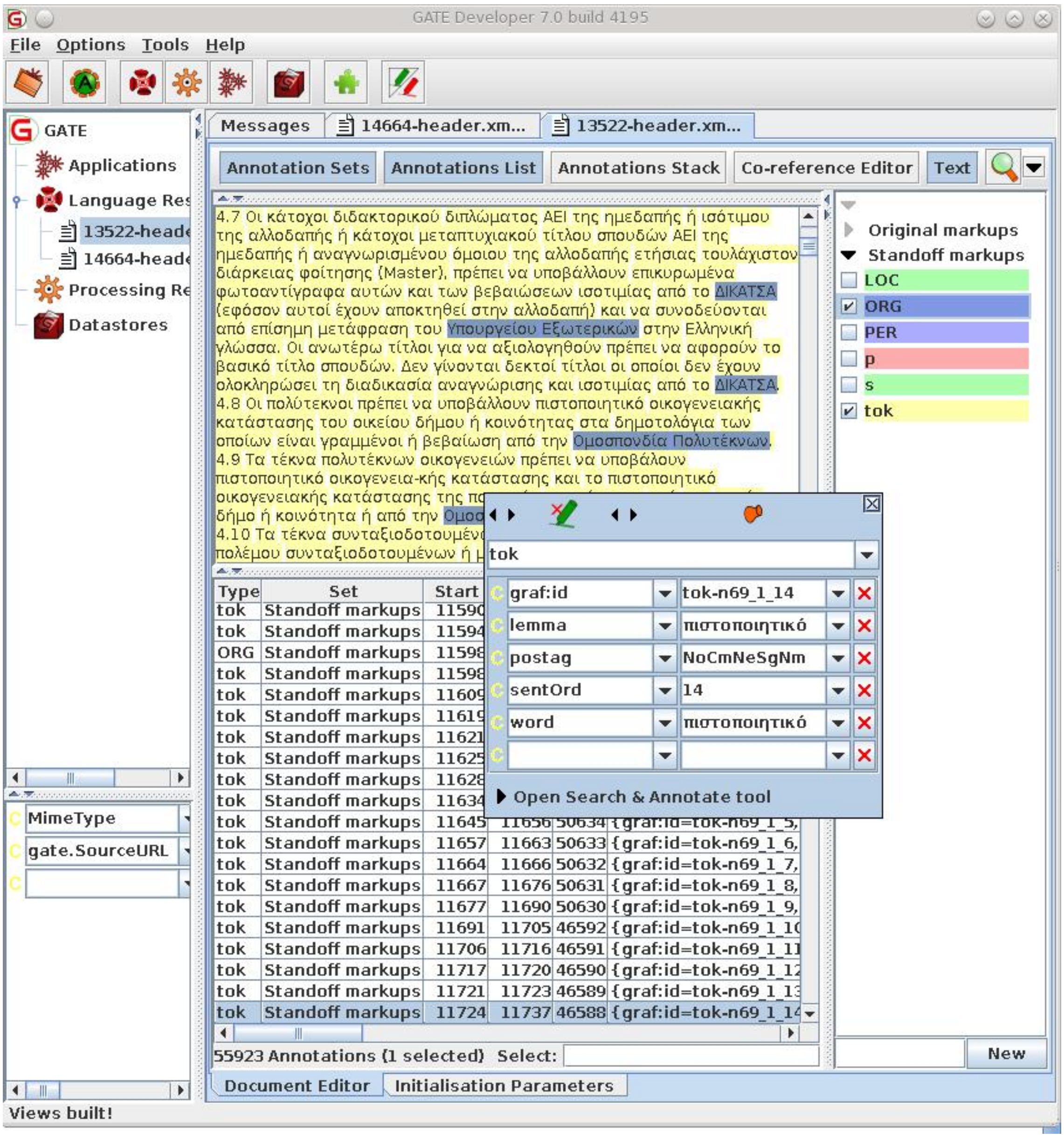
7.2 Data annotated with PANACEA NLP tools

The monolingual corpora collected in the framework of PANACEA and described in Section 2.1.11 of this deliverable were augmented with annotations provided by the NLP tools detailed in Section 4. The annotations generated included POS tags and lemmas for all monolingual partitions (i.e. EL, EN, ES, FR and IT) and were stored in PANACEA Travelling Object 1 format, which is based on the XCES standard. For partitions with dependency and named entity annotations, versions of the data stored in PANACEA Travelling Object 2 format (based on the GrAF standard) are also provided. PANACEA partners have made available these annotated data as detailed in Table 32.

Partner	Lang	Format	Annotations	URL
ILSP	EL	TO2	POS, Lemma, Dependency, NEs	http://nlp.ilsp.gr/panacea/D4.3/data/201209/graf/
DCU	EN	TO1	POS, Lemma	http://www.computing.dcu.ie/~atoral/panacea/to1/
UPF	ES	TO2	POS, Lemma, Dependency	http://gilmere.upf.edu/panacea-data/mcv2/dependency/
DCU	FR	TO1	POS, Lemma	http://www.computing.dcu.ie/~atoral/panacea/to1/
CNR	IT	TO2	POS, Lemma, Dependency	http://langtech3.ilc.cnr.it/html/panacea/storage/mcv2_graf_env/list_content_dir/ http://langtech3.ilc.cnr.it/html/panacea/storage/mcv2_graf_lab/list_content_dir/

Table 32 Monolingual corpora automatically annotated with WP4 NLP tools

Data in both format standards can be imported, explored and further processed in well known integrated development environments for NLP. For example, Figure 4 shows a GrAF document from the LAB_EL collection with several types of annotations (including POS, lemma, ORG, LOC and PER) opened in the GATE[27] Developer GUI.



4.7 Οι κάτοχοι διδακτορικού διπλώματος ΑΕΙ της ημεδαπής ή ισότιμου της αλλοδαπής ή κάτοχοι μεταπτυχιακού τίτλου σπουδών ΑΕΙ της ημεδαπής ή αναγνωρισμένου όμοιου της αλλοδαπής ετήσιας τουλάχιστον διάρκειας φοίτησης (Master), πρέπει να υποβάλλουν επικυρωμένα φωτοαντίγραφα αυτών και των βεβαιώσεων ισοτιμίας από το ΔΙΚΑΤΣΑ (εφόσον αυτοί έχουν αποκτηθεί στην αλλοδαπή) και να συνοδεύονται από επίσημη μετάφραση του Υπουργείου Εξωτερικών στην Ελληνική γλώσσα. Οι ανωτέρω τίτλοι για να αξιολογηθούν πρέπει να αφορούν το βασικό τίτλο σπουδών. Δεν γίνονται δεκτοί τίτλοι οι οποίοι δεν έχουν ολοκληρώσει τη διαδικασία αναγνώρισης και ισοτιμίας από το ΔΙΚΑΤΣΑ.
4.8 Οι πολύτεκνοι πρέπει να υποβάλλουν πιστοποιητικό οικογενειακής κατάστασης του οικείου δήμου ή κοινότητας στα δημοτολόγια των οποίων είναι γραμμένοι ή βεβαίωση από την Ομοσπονδία Πολυτέκνων.
4.9 Τα τέκνα πολυτέκνων οικογενειών πρέπει να υποβάλουν πιστοποιητικό οικογενεια-κής κατάστασης και το πιστοποιητικό οικογενειακής κατάστασης της πο
δήμο ή κοινότητα ή από την Ομοσ
4.10 Τα τέκνα συνταξιοδοτουμένω
πολέμου συνταξιοδοτουμένων ή μ

Type	Set	Start			
tok	Standoff markups	11590			
tok	Standoff markups	11594			
ORG	Standoff markups	11598			
tok	Standoff markups	11598			
tok	Standoff markups	11609			
tok	Standoff markups	11619			
tok	Standoff markups	11621			
tok	Standoff markups	11625			
tok	Standoff markups	11628			
tok	Standoff markups	11634			
tok	Standoff markups	11645	11656	50634	{graf:id=tok-n69_1_5,
tok	Standoff markups	11657	11663	50633	{graf:id=tok-n69_1_6,
tok	Standoff markups	11664	11666	50632	{graf:id=tok-n69_1_7,
tok	Standoff markups	11667	11676	50631	{graf:id=tok-n69_1_8,
tok	Standoff markups	11677	11690	50630	{graf:id=tok-n69_1_9,
tok	Standoff markups	11691	11705	46592	{graf:id=tok-n69_1_10
tok	Standoff markups	11706	11716	46591	{graf:id=tok-n69_1_11
tok	Standoff markups	11717	11720	46590	{graf:id=tok-n69_1_12
tok	Standoff markups	11721	11723	46589	{graf:id=tok-n69_1_13
tok	Standoff markups	11724	11737	46588	{graf:id=tok-n69_1_14

55923 Annotations (1 selected) Select:

tok

graf:id	tok-n69_1_14
lemma	πιστοποιητικό
postag	NoCmNeSgNm
sentOrd	14
word	πιστοποιητικό

Figure 4 A GrAF document from the LAB_EL collection examined in GATE

7.3 N-grams generated from the monolingual collections

Based on the annotated ENV & LAB monolingual collections for EL, EN, ES, IT and FR, we have generated word and word/tag/lemma n-grams. N-grams are accompanied by their observed frequency counts. The length of the n-grams ranges from unigrams (single words) to five-grams.

We attempted to extract only non-boilerplate portions of the crawled web pages of each collection. Sentence and token boundaries were automatically detected in the remaining data.

[27] http://gate.ac.uk

Tokens were automatically annotated with POS, morphosyntactic descriptions, and lemmas, with natural language processing tools described in this deliverable.

During the generation of the n-grams, all tokens that were guessed to be URLs, were mapped to the special word `<unk>` (for "unknown word"). The beginning of each sentence was marked with `<s>`, the end of a sentence was marked with `</s>`. The inserted tokens `<s>` and `</s>` were counted like other words and appear in the n-gram table. Table 33 provides details on the number of tokens and sentences of the non-boilerplate corpora from which the n-grams were produced, together with sizes for 1-5 word and word/pos/lemma datasets. N-grams for all languages will be delivered by PANACEA under Creative Commons licenses.

	EL		EN		ES		IT		FR	
	ENV	LAB	ENV	LAB	ENV	LAB	ENV	LAB	ENV	LAB
Tokens	31,713,750	24,069,160	46,553,583	45,134,225	49,860,040	58,067,619	35,998,740	70,438,164	42,780,009	46,992,912
Sentences	1,185,312	948,768	1,700,436	1,407,448	1,882,063	1,969,934	525,544	1,175,164	1,235,107	1,232,707
1-grams	435,189	363,977	910,884	606,569	652,771	582,537	459,854	547,482	824,972	664,984
2-grams	3,860,716	3,104,722	17,548,238	12,561,445	4,829,362	4,856,859	4,109,124	5,247,091	13,738,611	11,776,185
3-grams	9,767,383	7,725,039	75,302,270	5,7544,641	13,777,133	13,993,266	11,773,233	16,698,423	60,196,054	54,793,250
4-grams	13,683,940	10,650,455	148,085,024	120,877,352	21,883,389	22,815,785	17,243,938	26,014,257	127,183,968	121,160,297
5-grams	14,954,020	11,513,191	207,570,812	177,588,852	25,791,987	27,262,680	19,268,554	29,778,353	187,590,869	185,045,596
1_wpl-gms	502,131	418,272	2,067,718	1,389,737	681,386	610,630	492,735	580,555	1,904,750	1,551,122
2_wpl-gms	4,226,159	3,391,983	43,474,713	31,331,648	4,973,657	3,391,983	4,291,429	5,497,232	34,026,699	29,506,040
3_wpl-gms	10,137,987	8,011,442	190,553,115	146,303,726	13,956,369	14,226,180	11,962,668	17,009,683	153,763,338	140,824,253
4_wpl-gms	13,863,606	10,783,050	382,596,116	313,620,971	21,990,758	10,783,050	17,338,887	26,182,756	333,484,505	318,910,719
5_wpl-gms	15,019,194	11,558,504	545,236,920	468,455,667	25,843,317	22,966,815	19,305,246	29,843,367	501,199,941	495,887,230

Table 33 Data sizes for the n-grams generated from PANACEA monolingual data

7.4 Web services for the CAA subsystem in the PANACEA platform

Functionality	**Web Service**	**Host of the WS**	**URL**
Monolingual Crawling	Focused Monolingual Crawler	ILSP	http://nlp.ilsp.gr/soaplab2-axis/# ilsp_mono_crawl
Bilingual Crawling	Focused Bilingual Crawler	ILSP	http://nlp.ilsp.gr/soaplab2-axis/# ilsp_bilingual_crawl
Boilerplate removal	Cleaner	ILSP	http://nlp.ilsp.gr/soaplab2-axis/# ilsp_cleaner
Duplicate removal	De-duplicator	ILSP	http://nlp.ilsp.gr/soaplab2-axis/# ilsp_deduplicatormd5
Sentence Splitting of EN and FR	Europarl sentence-splitter	DCU	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.europarl_sentence_splitter_row
Tokenization of EN and FR	Europarl tokeniser	DCU	http://www.cngl.ie/panacea-soaplab2-axis//#panacea.europarl_tokeniser_row
Lowercasing of EN and FR	Europarl lowercaser	DCU	http://www.cngl.ie/panacea-soaplab2-axis//#panacea.europarl_lowercase_row
Tagging of EN and FR	Berkeley_tagger	DCU	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.berkeley_tagger_row
Tagging of EN and FR	TreeTagger	DCU	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.treetagger_row
Preprocessing functionalities of ES	IULA Preprocess	UPF	http://kurwenal.upf.edu/soaplab2-axis/#chunking_segmentation.iula_preprocess_row
Tokenization of ES	IULA Tokenizer	UPF	http://kurwenal.upf.edu/soaplab2-axis/#tokenization.iula_tokenizer_row
PoS tagging and Lemmatization of ES	IULA Tagger	UPF	http://kurwenal.upf.edu/soaplab2-axis/#morphosintactic_tagging.iula_tagger_row
Tokenization, PoS tagging and	Freeling	UPF	Tokenizer: http://ws04.iula.upf.edu/soaplab2-axis/#tokenization.freeling_tokenizer_r

Dependency parsing			ow PoS tagging: http://ws04.iula.upf.edu/soaplab2-axis/#morphosintactic_tagging.freeling_tagging_row Dependency parsing: http://ws04.iula.upf.edu/soaplab2-axis/#syntactic_tagging.freeling_dependency_row
Tokenization, sentence splitting, NER, PoS tagging and Dependency parsing	Freeling 3	UPF	Tokenizer: http://ws04.iula.upf.edu/soaplab2-axis/#tokenization.freeling3_tokenizer_row Sentence Splitter: http://ws04.iula.upf.edu/soaplab2-axis/#segmentation.freeling3_sentence_splitter_row PoS tagging: http://ws04.iula.upf.edu/soaplab2-axis/#morphosintactic_tagging.freeling3_tagging_row Dependency parsing: http://ws04.iula.upf.edu/soaplab2-axis/#syntactic_tagging.freeling3_dependency_row
Dependency parsing	MALT parser	UPF	http://ws04.iula.upf.edu/soaplab2-axis/#syntactic_tagging.malt_parser_row
PoS tagging	Twitter NLP	UPF	http://ws04.iula.upf.edu/soaplab2-axis/#morphosintactic_tagging.twitter_nlp_row
NER	Anonymizer	UPF	http://ws04.iula.upf.edu/soaplab2-axis/#named_entity_recognition.anonymizer_row
POS tagging and Lemmatization of IT	Freeling_it	CNR	http://wiki2.ilc.cnr.it:8080/soaplab2-axis/#panacea.freeling_it_row
Dependency parsing of Italian	DESR	CNR	http://langtech3.ilc.cnr.it:8080/soaplab2-axis/services/panacea.desr?wsdl
Topic Identification	LT Topic Identifier	LT	http://80.190.143.163:8080/panaceaV2

			/services/LTTopicIdentifier?wsdl
Sentence Splitting	LTSentenceSplitter	LT	http://80.190.143.163:8080/panaceaV2 /services/SentenceSplitter?wsdl
Tokenization and Normalization	LT Tokenizer	LT	http://80.190.143.163:8080/panaceaV2 /services/LTTokenizer?wsdl
Lemmatization– Lexical Analysis	LT Lemmatizer	LT	http://80.190.143.163:8080/panaceaV2 /services/LTLemmatizer?wsdl
Decomposition	LT Decomposer	LT	http://80.190.143.163:8080/panaceaV2 /services/LTDecomposer?wsdl
Defaulter (Service assigning default tags to unknown words)	LT Defaulter	LT	http://80.190.143.163:8080/panaceaV2 /services/LTDefaulter?wsdl
Unknowns (Helper Service to collect unknowns from analysis outputs)	LTUnkExtractor	LT	http://80.190.143.163:8080/panaceaV2 /services/LTUnkExtractor?wsdl
Tagging	LTTagger	LT	http://80.190.143.163:8080/panaceaV2 /services/LTTagger?wsdl
Sentence Splitting and Tokenization of EL	ILSP Sentence Splitter and Tokenizer	ILSP	http://nlp.ilsp.gr soaplab2-axis/#ilsp.ilsp_sst_row
POS Tagging of EL	ILSP FBT Tagger	ILSP	http://nlp.ilsp.gr soaplab2-axis/#ilsp.ilsp_fbt_row
Lemmatization of EL	ILSP Lemmatizer	ILSP	http://nlp.ilsp.gr soaplab2-axis/#ilsp.ilsp_lemmatizer_row
Dependency parsing of EL	ILSP Dependency Parser	ILSP	http://nlp.ilsp.gr soaplab2-axis/#ilsp.ilsp_depparser _row
Recognition of Named Entities for EL	ILSP NERC	ILSP	http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_nerc_row
Converter from and to the crawlers' output; from and to results of NLP tools to the common encoding format defined in D3.1	PANACEA Conversor	UPF	http://ws04.iula.upf.edu/soaplab2-axis/#format_conversion.panacea_conversor_row
Converter from the Berkeley tagger output to the common	Berkeley_tagger2to	DCU	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.berkeley_tagger2to_row

encoding format			
Converter from the TreeTagger output to the common enconding format	treetagger2to	DCU	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.treetagger2to_row
Converter from the Freeling from Italian to the common encoding format	converter_freeling_to	CNR	http://wiki2.ilc.cnr.it:8080/soaplab2-axis/ panacea.converter_freeling_to
Converter from UIMA Common Analysis Structure XMI to GrAF	xmicas2graf	ILSP	http://nlp.ilsp.gr/soaplab2-axis/#ilsp.xmicas2graf_row

7.5 Taverna workflows built with PANACEA web services

In this appendix, we provide two example workflows for processing texts with CAA web services for the PANACEA platform. In Figure 5, an EN-EL pair of document is first tunneled to two different pipelines, one for each language. The Greek text is processed by NLP tools hosted at ILSP, while the Europarl tools and the Berkeley tagger hosted at DCU takes care of the English counterpart.

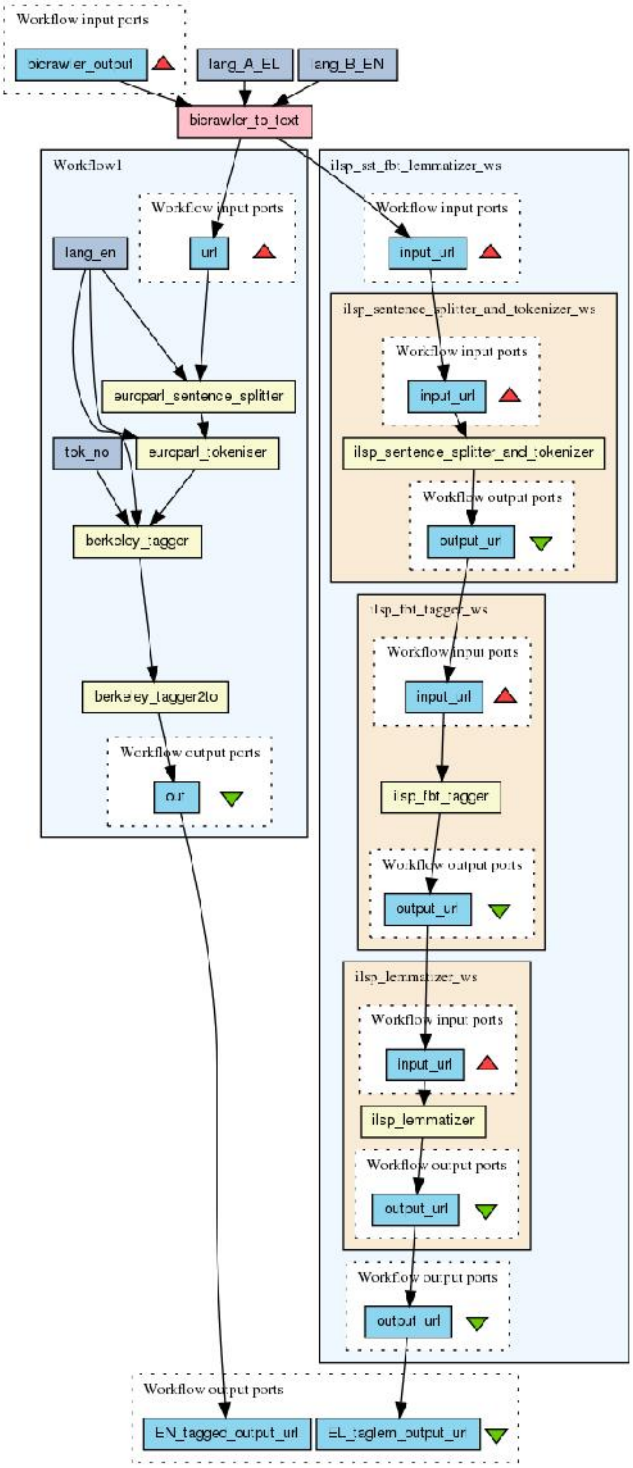


Figure 5 Processing an EN-EL pair of documents

In Figure 6, a German document crawled from the web is processed by LT's services. The document is first processed for topic identification, sentence and token boundary identification, and lemmatization. Following these processing stages, "unknown tokens" are decomposed and checked again. Finally, in case the decomposer classifies a token as 'unknown', this token is assigns them a default tag by the *ltdefault* service.

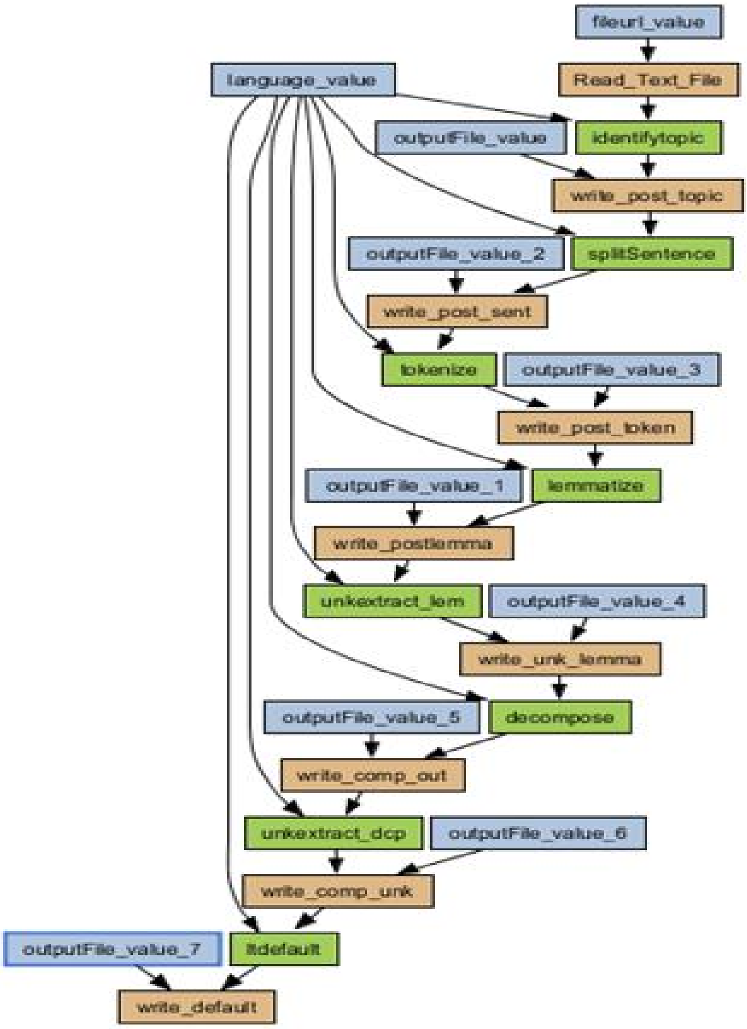


Figure 6 Lexical Analysis of crawled documents using Linguatec's services