SEVENTH FRAMEWORK PROGRAMME

PANACEA

SEVENTH FRAMEWORK PROGRAMME
THEME 3
Information and communication Technologies

PANACEA Project

Grant Agreement no.: 248064

Platform for **A**utomatic, **N**ormalized **A**nnotation and
Cost-**E**ffective **A**cquisition
of Language Resources for Human Language Technologies

WP-4.4: Report on the revised Corpus Acquisition & Annotation subsystem and its components

Dissemination Level:	Public
Delivery Date:	November 02, 2011
Status – Version:	V1.0
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)
D7.2	First Evaluation Report (T14)
D7.3	Second evaluation report (T23)

Table of Contents

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 provided documentation on the initial functional prototype of this subsystem, while this deliverable presents updates in the revised subsystem during the second development cycle of the project.

The deliverable is structured as follows. A revised version of the Focused Monolingual Crawler (FMC), that has been implemented according to the results of the first evaluation cycle and the reviewers' comments in the first annual review report, is described in section 2. New and revised versions of tools for corpus normalization (cleaning and deduplication) are detailed in section 3. Section 4 provides documentation on the NLP tools introduced for the first time in the subsystem. These tools focus mainly on sentence splitting/tokenization and POS tagging/lemmatization for English (EN), French (FR), Spanish (ES), German (DE), Italian (IT) and Greek (EL).

2 Focused Monolingual Crawler

This section describes a revised version of the Focused Monolingual Crawler available as a web service (*ilsp_mono_crawl*) in the PANACEA factory. The FMC is used for building monolingual domain-specific LRs by crawling web documents with rich textual content. The initial FMC integrated modules for fetching and parsing HTML web pages, text classification, boilerplate removal and exporting of acquired documents in the cesDOC format described in *D3.1 Architecture and Design of the Platform, Sec. 6.1.2*. The revised version of the FMC exploits new methods and/or modifications of existing modules with the purpose of providing larger and qualitatively better, in-domain corpora.

2.1 The revised FMC

The first version of the FMC employed an adaptation of the Combine[1] open-source crawler and it was used for the construction of the first version of the in-domain monolingual corpora (MCv1) delivered as *D4.3 Monolingual corpus acquired in five languages and two domains*. Even though the default crawling strategy (the Breadth-First algorithm proposed by Pinkerton, 1994) was changed to a more efficient approach for focused crawling (the Best-First algorithm proposed by Cho, 1998), the architecture of that version was not scalable for larger crawls. Since the issue of crawler's scalability was raised in the first review report, we implemented a revised version of the FMC that adopts a distributed computing architecture based on Bixo[2], an open source web mining toolkit that runs on top of Hadoop[3] (a well-known framework for distributed data processing). In addition, Bixo also depends on the Heritrix[4] web crawler and makes use of ideas developed in the Nutch[5] web-search software project. These two open source frameworks for mining data from the web were mentioned as alternatives at the first review meeting by the reviewers.

The MCv1 dataset consisted of monolingual corpora for English, Spanish, Italian, French and Greek in the Environment (ENV) and Labour Legislation (LAB) domains. The relatively small size of MCv1 (500-800 web documents for each language/domain combination, resulting to just more that 1M tokens for each combination) was mentioned in the review report as an indication of problems in the architecture of the initial FMC. Following this comment, we used the revised FMC to construct the second version of the monolingual corpora (MCv2) which targeted the same language/domain combinations. The size of the produced MCv2 corpora[6] 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. More details about the quantity of the acquired data will be provided in the forthcoming *D7.3 Second evaluation report*, where the acquisition performance of the FMC will be discussed.

[1] http://combine.it.lth.se/documentation/
[2] http://openbixo.org/
[3] http://hadoop.apache.org/
[4] http://crawler.archive.org
[5] http://nutch.apache.org/about.html
[6] As scheduled in D4.3, MCv2 has been delivered internally to project partners in T20 to be augmented with automatic morphosyntactic annotations.

Another observation in the review report concerned missing information about the distribution of the sub-domains of ENV and LAB in MCv1. In order to address this issue, the FMC was modified so as to categorize in-domain pages into one or more of the sub-domains that can be defined in the topic definition provided by the FMC user. The identified sub-domain(s) is/are stored in the <subdomain> element of the cesDOC file exported from each relevant web page. Therefore, a user can easily calculate how many documents/tokens belong in 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. Information about the distribution of subdomains in MCv2 will also be included in the *D7.3* report.

Apart from comments by reviewers, the revision of the initial FMC took into account the results of the first evaluation cycle reported in *D7.2 First evaluation report.* One of the results of the manual evaluation of an MCv1 sample showed that about 5% of the acquired documents contained at least one paragraph not in the targeted language. Therefore, the revised FMC applies the embedded language identifier at paragraph level as well. If a paragraph is not in the targeted language, FMC adds to its corresponding XML element the attribute *crawlinfo* with value *ooi-lang* (meaning "out-of-interest because of a language different from the main document language"). For an example, see the `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>
```

The optional attribute *topic* (see the `p61` paragraph in the listing above) has a string value including all terms from the topic definition detected in this paragraph.

Another finding from the manual evaluation was that a very large proportion of the documents (approx. 80%) contained at least one paragraph of only limited or no use and that about 10% of the documents included at least one paragraph wrongly segmented into two or more paragraphs. These results motivated us to improve the embedded cleaning module as follows. First, we examined Boilepipe[7], the module used by the FMC for boilerplate removal. We informed the developer of Boilerpipe that the <*sub*> HTML tag was wrongly considered as a paragraph separator. Moreover, we introduced simple heuristics that classify short paragraphs as out of interest and added the value *ooi-length* to the *crawlinfo* attribute of such paragraphs. 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>
```

[7] http://code.google.com/p/boilerpipe/

```
<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>
```

We also modified Boilerpipe in order to extract structural information about the web page examined. This information is encoded in an optional *type* attribute at paragraph level, with *title*, *heading* or *listitem* as its possible values (see `p43` in the listing above). Finally, in the revised FMC, paragraphs that have been classified as boilerplate are optionally kept in the XML file. An attribute *crawlinfo* with value *boilerplate* is used to mark them as such:

```
<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;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>
```

Another enhancement of the FMC concerns its crawling strategy. Instead of using only the score of the source web page as an in-domain relevance estimation for each link, FMC now compares the anchor text of each link (i.e. the text surrounding the link) with the terms of the topic definition. According to this approach, each link is ranked with a potential score influenced by the source web page relevance score and the estimated relevance of the link's anchor text. The link relevance score *l* is calculated by the following formula:

$$l = p / N + \sum_{i=1}^{M} n_i * w_i$$

where p is the relevance score of the source page, N is the amount of links originated form the source page, M 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.

As concluded 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 and forces the crawler to visit relevant web pages earlier.

2.2 FMC as Web service

The web-service is available at http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_fmc_row. It uses three mandatory parameters:

1. The *language* parameter expects a string value denoting the targeted language. Currently supported languages are English, French, German, Greek, Italian and Spanish. Each downloaded web page is analyzed as a whole by the embedded language identifier and its language is identified. If the document is not in the targeted language, it is discarded. In addition, the language identifier is applied at paragraph level and, if needed, the attribute *crawlinfo* with value *ooi-lang* is added.

2. The *termList* parameter expects a list of triplets (<relevance weight, term, topic-class>) that define a targeted domain or its subdomains. More details about constructing such topic definitions are reported in section 4.1 of *D4.3*. An example from the topic definition for "Environment" in English 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
```

   This term list is used to perform text classification as described in section 3.1.1 of *D4.2*. In order to favor precision, we introduced an additional constraint, which disallows storing a web document as relevant unless three distinct terms from the topic definition are found in this document. This implies that topic definitions should contain three or more terms.

3. The *urlList* parameter expects a list of seed URLs with which the crawler is initialized. These URLs should be relevant to the domain (i.e. contain positively-weighted terms from the topic definition).

The web service uses three 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 1 minute, which implies that only one cycle will run.

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*.

The output of the crawler is a text file containing a list of URLs pointing to stored web documents exported as cesDOC XML files. Following the enhancements in FMC's functionalities described in the previous subsection, the cesDOC files are now enriched with attributes providing more information about the process outcome. Specifically, paragraph elements (<p>) in the XML files may contain the following attributes:

1. *crawlinfo* with possible values: *boilerplate*, meaning that the paragraph has been considered boilerplate; *ooi-length*, denoting that this paragraph is so short that either it is not useful, or it can confuse the language identifier; and *ooi-lang*, denoting that the paragraph is not in the targeted language.

2. *type* with possible values: *title*, *heading* and *listitem*.

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

As an example, the following paragraph has been detected as a *listitem* that contains two terms from the ENV_EN topic definition. The *crawlinfo* attribute is not needed in this example, since this is a relatively long paragraph in English, which was not classified as boilerplate.

```
<p id="p45" type="listitem" topic="dumping of waste;natural
resources">
"If the administration gets its way, thousands of streams, wetlands
and other waters would no longer be protected by the law, allowing
industry to dredge, fill or dump waste into them without a permit
and without notifying the public." — July 11, 2003 [ Natural
Resources Defense Council, 7/11/2003 ]
</p>
```

3 Corpus normalization tools and services

This section describes two new PANACEA web services dedicated to removal of boilerplate and detection of duplicate documents.

3.1 Cleaner

The Cleaner aims to detect and remove boilerplate text that typically is not related to the main content (e.g. navigation links, advertisements, disclaimers, etc.) from a web document. It is an extension of the Boiperplate remover service described in subsection 3.3 of *D4.2 Initial functional prototype*. The main extensions concern the extraction of structural information about the web documents, the employment of the new version of Boilerpipe tool (Kohlschütter et al, 2010), and an enhancement that allows exporting in a cesDOC format as described in *D3.1*.

The web-service is available at http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_cleaner_row. It uses one mandatory parameter:

1. The *input* parameter expects 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) a cesDOC file containing metadata of the web document and the clean text only, and iii) a cesDOC 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 deduplication 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 m*ethodsList* parameter sets the method for removing boilerplate. Boilerpipe provides six methods (ArticleExtractor, ArticleSentencesExtractor, DefaultExtractor, KeepEverythingExtractor, LargestContentExtractor, 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 boileplate. The 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*> (in case the output is a cesDOC file).

5. The t*ermList* 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 2.2 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. The value of this attribute will be the found terms.

3.2 DeduplicatorMD5

The *DeduplicatorMD5* web service aims to detect and discard (near) duplicate documents appearing in a corpus. We employed the deduplication strategy included in the Nutch framework[8], which involves the construction of a text profile based on the quantized word frequencies, and an MD5 hash for each document. The web-service of the *DeduplicatorMD5* is available in http://nlp.ilsp.gr/soaplab2-axis/#ilsp.ilsp_deduplicatormd5_row. It uses two mandatory parameters:

1. The *input* denotes a file containing a list with URLs pointing to the files to be deduplicated.

2. The *inputType* denotes the type of the files to be deduplicated. These files could be text or cesDOC files similar to the ones provided by the FMC.

The *DeduplicatorMD5* also contains two optional parameters:

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

2. The *quantValue* is a parameter of the exploited algorithm. 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 deduplication.

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

4 NLP tools and services

This section catalogues and describes the NLP tools introduced to the revised CAA subsystem as web services. The current CAA subsystem includes services that provide **Sentence Splitting**, **Tokenization**, **POS Tagging** and **Lemmatization** functionalities for English, French, German, Greek, Italian and Spanish, i.e. for all languages targeted by PANACEA.

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[9] 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 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 have investigated two approaches. UPF, DCU and CNR have built specific web services[10] to perform I/O conversions from and to their tools. ILSP has adapted its NLP tools by integrating importers and exporters from and to the common encoding format.

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.

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

[10] See, among others, the UPF, DCU, and CNR converters at Appendix E, 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[11] 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

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

4.1.2 Berkeley tagger

Berkeley tagger is a webservice 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

4.2 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 5 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 Workflow(s) using the WS	http://myexperiment.elda.org/workflows/5, http://myexperiment.elda.org/workflows/22, http://myexperiment.elda.org/workflows/23

Table 6 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[12] texts with PAROLE[13] 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	(wsdl can be found on the registry)
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 7 WS Details for Freeling

The Freeling services accept a set of optional parameters, which we briefly describe in Table 8. 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
nodict	Whether to perform dictionary search

[12] The Freeling service hosted by UPF can be used for POS tagging and lemmatisation of, among others, English and Catalan.
[13] http://nlp.lsi.upc.edu/freeling/doc/userman/parole-es.html,

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 8 Optional parameters for Freeling services

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 9 WS Details for freeling_it

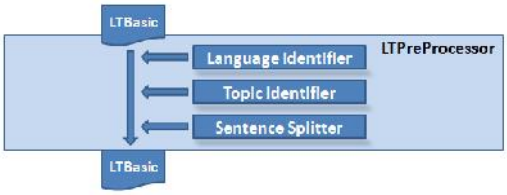
The freeling_it service accepts a set of optional parameters regarding multiword detection, named-entity and output-format. These parameters are briefly described in Table 10.

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*)

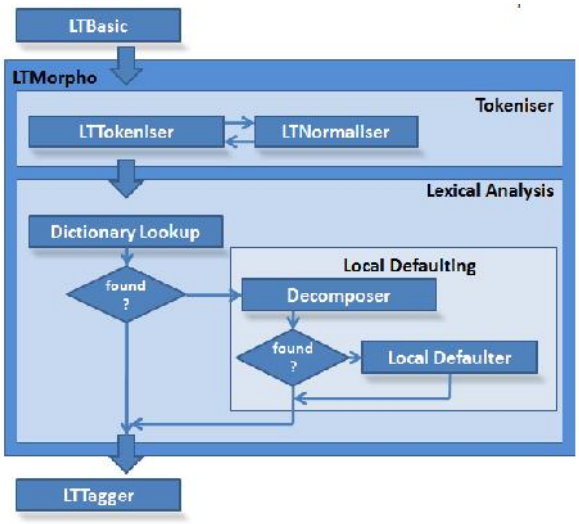
Table 10 Optional parameters for freeling_it

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 is still under development and will be added to the PANACEA toolbox in a later stage.

4.4.1 Topic Identification

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

Language resources

There are two main resources: a taxonomy of topics, and the features for each topic.

- The taxonomy of the topic identifier consists of about 40 topics, organised 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 textforms, and they can contain multiwords; using multiwords

improves results significantly, esp.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 11 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.

Language Resources

The sentence splitter uses the following 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 12 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.

Language resources

Normaliser lists: 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 UK-US 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 13 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 14 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.

Language Resources

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 15 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'.

Language Resources

The following 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.[14]

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'[15]

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

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

[15] 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.

('*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.

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 16 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, this way enabling the system to perform preprocessing for machine translation.

Language Resources

The main resources are tagging rules: Rules consist of condition–action patterns, conditions being configurations of linguistic categories and actions performing some disambiguation operations.

Modus Operandi

After initial preprocessing and pruning of 'impossible' readings, the tool performs from left to right in a cyclic way, applying disambiguation rules to the input until all ambiguous constructions are disambiguated, or no rules can fire any more. A final cleanup step takes care of remaining constructions.

The tagger is still under construction.

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

Table 17 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 XML files. 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 18.

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 18 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 19 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 20 for the basic POS tags used by the tagger[16].

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 21.

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
DATE	Date

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
PtNg	Negative particle

[16] 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
DIG	Digit
ENUM	Enumeration element
INIT	Initial
NmCd	Cardinal numeral
NmCt	Collective numeral
NmMl	Multiplicative numeral
NmOd	Ordinal numeral
NoCm	Common noun
NoPr	Proper noun

POS	Description
PtOt	Other article
PtSj	Subjunctive particle
PUNCT	Other punctuation
RgAbXx	Abbreviation
RgAnXx	Acronym
RgFwOr	Foreign word in its original form
RgFwTr	Transliterated foreign word
VbIs	Impersonal verb
VbMn	Main verb

Table 20 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 21 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[17]. 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 22.

URL	http://nlp.ilsp.gr soaplab2-axis/#ilsp.ilsp_lemmatizer_row
WSDL	http://nlp.ilsp.gr/soaplab2-axis/typed/services/ilsp.ilsp_lemmatizer?wsdl

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

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 22 WS Details for ILSP Lemmatizer

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. 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.) 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. Proceedings of the 10th International Conference on Greek Linguistics. Komotini, Greece: 1-4 September 2011.

6 Conclusions and Workplan

In this deliverable, we presented the revised version of the CAA subsystem of the PANACEA platform. Compared to the initial version described in D4.2, this version reflects improvements during PANACEA's second development cycle and includes updated and new tools focusing on corpus acquisition (revised Focused Monolingual Crawler), normalization (dedicated cleaning and deduplication tools) and NLP (tools for sentence splitting/tokenization and POS tagging/lemmatization in EN, DE, EL, ES, IT, FR). The integration into the PANACEA platform of web services corresponding to these tools complies with the *Description of Work* document and the solution path detailed in D4.1 *Technologies and tools for corpus creation, normalization and annotation*.

The final version of the PANACEA platform is planned to include NLP modules focusing on parsing and/or chunking for all languages targeted by the project. In more detail, the workplan for the CAA development in the context of WP4 will include the tasks sketched below:

- T24: **Internal deliverable.** Partners adapt NLP tools focusing on parsing and/or chunking for DE, EN, EL, ES, IT, FR. The I/O of all tools will be conformant with the common encoding format documented in *D3.1 Architecture and Design of the Platform.* These tools will be part of the final version of the CAA subsystem.

- T29: **D4.5.** Final version of the prototype and documentation, as required for the D7.4 Third evaluation report (T30).

7 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, Proc. 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, Proc. of the Human Language Technology Conference.

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

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 conference on International Language Resources and Evaluation (LREC'10)*. Paris, France European Language Resources Association (ELRA).

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. 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).

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*.

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

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.

Appendix

A. Extract from a Greek document with POS and lemma annotations

```
<p id="p11">
<s id="s11">
<t id="t227" word="Για" tag="AsPpSp" lemma="για"/>
<t id="t228" word="πρώτη" tag="NmOdFeSgAcAj" lemma="πρώτος"/>
<t id="t229" word="φορά" tag="NoCmFeSgAc" lemma="φορά"/>
<t id="t230" word="οι" tag="AtDfMaPlNm" lemma="ο"/>
<t id="t231" word="επιστήμονες" tag="NoCmMaPlNm" lemma="επιστήμονας"/>
<t id="t232" word="μελετούν" tag="VbMnIdPr03PlXxIpAvXx" lemma="μελετώ"/>
<t id="t233" word="τρόπους" tag="NoCmMaPlAc" lemma="τρόπος"/>
<t id="t234" word="για" tag="AsPpSp" lemma="για"/>
<t id="t235" word="την" tag="AtDfFeSgAc" lemma="ο"/>
<t id="t236" word="προσαρμογή" tag="NoCmFeSgAc" lemma="προσαρμογή"/>
<t id="t237" word="των" tag="AtDfNePlGe" lemma="ο"/>
<t id="t238" word="σπάνιων" tag="AjBaNePlGe" lemma="σπάνιος"/>
<t id="t239" word="ειδών" tag="NoCmNePlGe" lemma="είδος"/>
<t id="t240" word="στις" tag="AsPpPaFePlAc" lemma="στου"/>
<t id="t241" word="περιβαλλοντικές" tag="AjBaFePlAc" lemma="περιβαλλοντικός"/>
<t id="t242" word="μεταβολές" tag="NoCmFePlAc" lemma="μεταβολή"/>
<t id="t243" word="που" tag="PnReFe03PlNmXx" lemma="που"/>
<t id="t244" word="προκαλούνται" tag="VbMnIdPr03PlXxIpPvXx" lemma="προκαλώ"/>
<t id="t245" word="από" tag="AsPpSp" lemma="από"/>
<t id="t246" word="την" tag="AtDfFeSgAc" lemma="ο"/>
<t id="t247" word="αλλαγή" tag="NoCmFeSgAc" lemma="αλλαγή"/>
<t id="t248" word="του" tag="AtDfNeSgGe" lemma="ο"/>
<t id="t249" word="κλίματος" tag="NoCmNeSgGe" lemma="κλίμα"/>
<t id="t250" word="." tag="PTERM P" lemma="."/>
</s>
</p>
```

Sentence: Για πρώτη φορά οι επιστήμονες μελετούν τρόπους για την προσαρμογή των σπάνιων ειδών στις περιβαλλοντικές μεταβολές που προκαλούνται από την αλλαγή του κλίματος.

Translation: For the first time, scientists study ways for the adaption of rare species to environmental changes due to climate change.

This sentence was processed with the ILSP NLP services described in Section 4.5.

B. Extract from an EN document with POS annotations

```
<p id="p1">
       <s id="s1">
         <t tid="t_1" tag="CC" word="And"/>
         <t tid="t_2" tag="DT" word="the"/>
         <t tid="t 3" tag="NN" word="motor"/>
         <t tid="t 4" tag="IN" word="of"/>
         <t tid="t 5" tag="DT" word="the"/>
         <t tid="t_6" tag="NN" word="economy"/>
         <t tid="t_7" tag="," word=","/>
         <t tid="t 8" tag="DT" word="the"/>
         <t tid="t 9" tag="NN" word="construction"/>
         <t tid="t 10" tag="NN" word="industry"/>
         <t tid="t 11" tag="," word=","/>
         <t tid="t_12" tag="VBZ" word="is"/>
         <t tid="t_13" tag="IN" word="in"/>
         <t tid="t 14" tag="DT" word="a"/>
         <t tid="t 15" tag="JJ" word="deep"/>
         <t tid="t 16" tag="NN" word="crisis"/>
         <t tid="t_17" tag="." word="."/>
       </s>
     </p>
```

Sentence: And the motor of the economy, the construction industry, is in a deep crisis.

This sentence was processed with the Berkeley Tagger service described in Section 4.1.2.

C. Extract from an ES document annotated for POS and lemma

```
<p id="p1">
<s id="s1">
    <t id="t1" tag="DI0MS0" lemma="uno" word="Un"/>
    <t id="t2" tag="NCMS000" lemma="estudio" word="estudio"/>
    <t id="t3" tag="VMP00SM" lemma="realizar" word="realizado"/>
    <t id="t4" tag="SPS00" lemma="en" word="en"/>
    <t id="t5" tag="Z" lemma="14" word="14"/>
    <t id="t6" tag="NCFP000" lemma="ciudad" word="ciudades"/>
    <t id="t7" tag="AQ0FP0" lemma="español" word="españolas"/>
    <t id="t8" tag="VMIP3S0" lemma="observar" word="observa"/>
    <t id="t9" tag="DI0MS0" lemma="uno" word="un"/>
    <t id="t10" tag="NCMS000" lemma="aumento" word="aumento"/>
    <t id="t11" tag="SPS00" lemma="de" word="de"/>
    <t id="t12" tag="DA0MP0" lemma="el" word="los"/>
    <t id="t13" tag="NCMP000" lemma="ingreso" word="ingresos"/>
    <t id="t14" tag="AQ0CP0" lemma="cardiovascular" word="cardiovasculares"/>
</s>
```

Sentence: “Un studio realizado en 14 ciudades españolas observa un aumento de los ingresos cardiovasculares”.

Translation: A study carried out in 14 spanish cities shows an increase in cardiovascular examinations.

This sentence was processed with the Freeling tagging service described in Section 4.2.3.

D. Extract from an IT document annotated for POS and lemma

```
        <s id="s279">
          <t tid="t_8172" tag="RG" lemma="così" word="Così"/>
          <t tid="t_8173" tag="DA0MP0" lemma="il" word="i"/>
          <t tid="t_8174" tag="NCMP000" lemma="pesce" word="pesci"/>
          <t tid="t_8175" tag="Fc" lemma="," word=","/>
          <t tid="t_8176" tag="RG" lemma="non" word="non"/>
          <t tid="t_8177" tag="VMIP3P0" lemma="respirare" word="respirano"/>
          <t tid="t_8178" tag="RG" lemma="più" word="più"/>
          <t tid="t_8179" tag="Fc" lemma="," word=","/>
          <t tid="t_8180" tag="PI0MS000" lemma="tutto" word="tutto"/>
          <t tid="t_8181" tag="PP3CSD00" lemma="si" word="si"/>
          <t tid="t_8182" tag="VMIP3S0" lemma="degradare" word="degrada"/>
          <t tid="t_8183" tag="Fp" lemma="." word="."/>
        </s>
```

Sentence: Così i pesci, non respirano più, tutto si degrada.

Translation: So is it for the fish, they cannot breathe any more, everything is degraded.

This sentence was processed with the Freeling Italian service described in Section 4.3.1.

E. 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	DeduplicatorMD5	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
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 Dependency parsing	Freeling	UPF	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
POS tagging and Lemmatization of IT	Freeling_it	CNR	http://wiki2.ilc.cnr.it:8080/soaplab2-axis/#panacea.freeling_it_row
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
Converter from and to the crawlers' output;	PANACEA Conversor	UPF	http://ws04.iula.upf.edu/soaplab2-axis/#format_conversion.panacea_conv

from and to results of NLP tools to the common encoding format defined in D3.1			ersor_row
Converter from the Berkeley tagger output to the common encoding format	Berkeley_tagger2to	DCU	http://www.cngl.ie/panacea-soaplab2-axis/#panacea.berkeley_tagger2to_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

F. 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 1, 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.

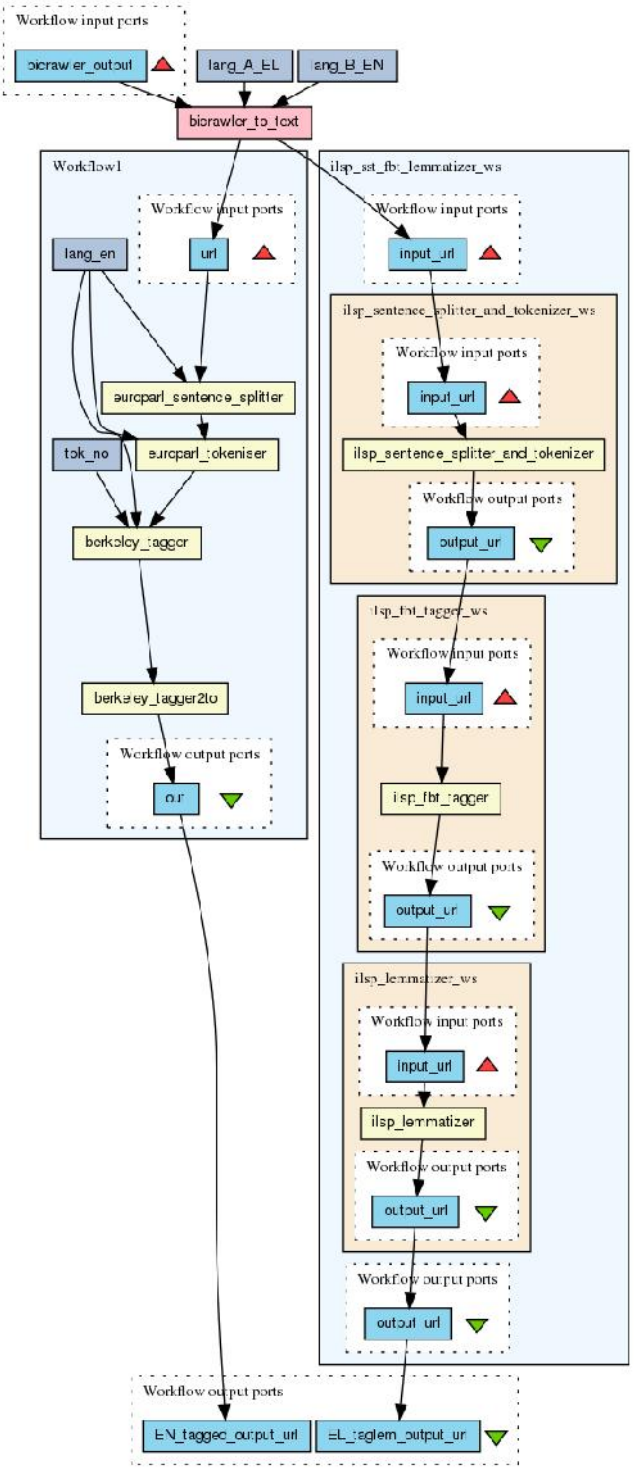


Figure 1 Processing an EN-EL pair of documents

In Figure 2, 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.

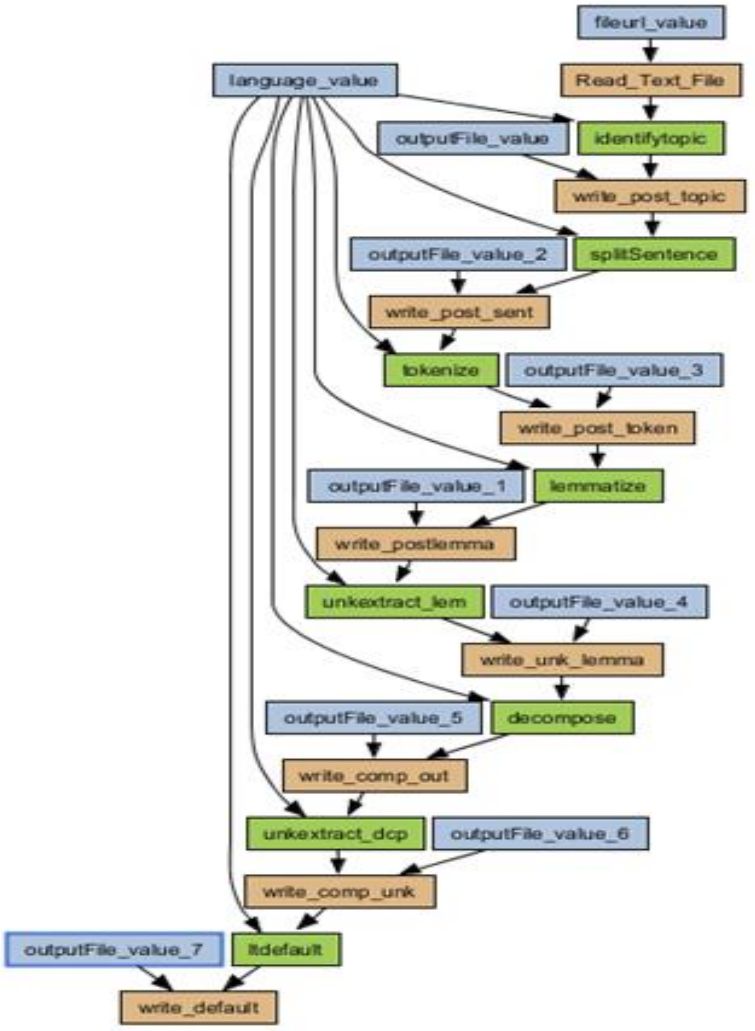


Figure 2 Lexical Analysis of crawled documents using Linguatec's services