

---

# **Eurosentiment Documentation**

***Release 0.1***

**J. Fernando Sánchez**

October 11, 2014



<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	Getting Access to the Platform . . . . .	3
1.2	Managing Subscriptions . . . . .	3
1.3	The Eurosentiment Format . . . . .	9
1.4	Tools . . . . .	16
<b>2</b>	<b>Services and Resources</b>	<b>29</b>
2.1	Services . . . . .	29
2.2	Resources . . . . .	32
<b>3</b>	<b>Language Resource Pool Management</b>	<b>39</b>
3.1	The LRP Management Application API . . . . .	39
<b>4</b>	<b>FAQ</b>	<b>41</b>
4.1	Frequently Asked Questions . . . . .	41
<b>5</b>	<b>Indices and tables</b>	<b>43</b>
<b>6</b>	<b>To-Do</b>	<b>45</b>
	<b>HTTP Routing Table</b>	<b>47</b>



The aim of the [EUROSENTIMENT](#) project is to create a pool for multilingual language resources and services for Sentiment Analysis.

With this pool, both language resource owners and sentiment analysis service developers can share their knowledge and experience with the Sentiment Analysis community. But it also enables content providers to benefit from sentiment analysis with little effort.

You might want to check the [EUROSENTIMENT portal](#) to start enjoying the language resource pool (LRP). It contains a list of all the existing language resources and services provided by the project partners. Or you can register and contribute with your own language resources and services.

If you're still not sure about joining, please check our [demonstrator](#) to see what EUROSENTIMENT can offer you, and get an example of our language resources and services.

This is the reference documentation for the Language Resource Pool and its services, as well as the API to access other information



---

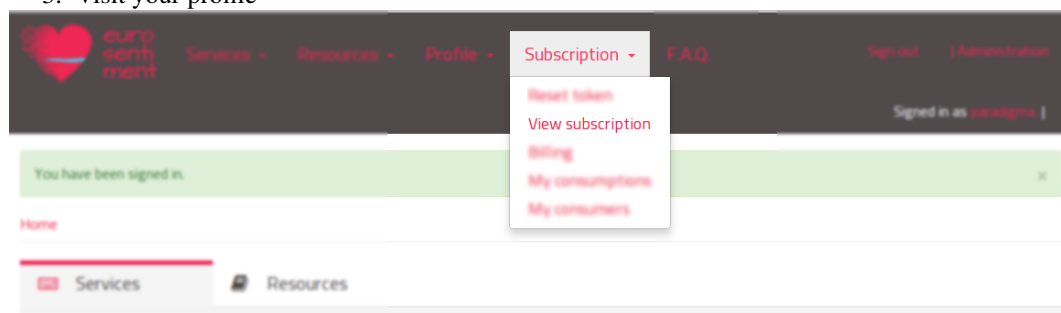
## Getting started

---

### 1.1 Getting Access to the Platform

To consume the services provided by EUROSENTIMENT programmatically, you need to provide an access token in each request. It can be easily obtained following these simple steps:

1. Go to the [EUROSENTIMENT LRP](#)
2. Log in (or sign up if you don't have an account already)
3. Visit your profile



4. Copy your Token

### 1.2 Managing Subscriptions


In this menu, the users can check their account subscription, their activities in consuming/producing services and resources, and their bills/reports.

#### 1.2.1 My Subscription

By clicking on **Subscription -> My Subscription**, users are presented with this page:

In this page, users can see:

- Their type of account (Language Resource Owner or Service Developer or Content Provider)

[Services](#) [Resources](#) [Profile](#) [Subscription](#) [F.A.Q.](#) [Sign out](#) [Administration](#)

Signed in as **paradigma** |

[Home](#) » [View subscription](#)

### Modality

Content Provider

### Token

78dab260-d17d-42fa-b1ac-d3fe84e942ca

### Start date

Dec. 22, 2012, 4:17 p.m.

### End date

Dec. 23, 2015, 4:17 p.m.

### Bill cycle day

18

### State

APPROVED

[Services](#) [Resources](#) [Profile](#) [Subscription](#) [Demos](#) [Help](#) [Sign out](#)

Signed in as **TestConsumer** |

[Home](#) » [View subscription](#)

## Welcome to "My Subscription"

Eurosentiment allows you to access resources and APIs via an access token, which you can get here. This way Eurosentiment allows precise access control and accounting resource usage. To learn more about this and see how to use this code please refer to the [documentation](#).

### User Type

Content Provider [Change modality](#)

### Token

f6a9204d-fccd-40b5-af15-37812eb3e1f0 [Reset Token](#)

### Start date

Aug. 4, 2014, 1:19 a.m.

### End date

Oct. 31, 2015, midnight

### Bill cycle day


1

### State

APPROVED

EUROSENTIMENT [License](#)



[Services](#) ▾[Resources](#) ▾[Profile](#) ▾[Subscription](#) ▾[Demos](#) ▾[Help](#) ▾[Sign out](#)

Signed in as **TestConsumer** |

[Home](#) » [View subscription](#)

## Welcome to "My Subscription"

Eurosentiment allows you to access resources and APIs via an access token, which you can get here. This way Eurosentiment allows precise access control and accounting resource usage. To learn more about this and see how to use this code please refer to the [documentation](#).

User Type

Content Provider [Change modality](#)

Token

f6a9204d-fccd-40b5-af15-37812eb3e1f0 [Reset Token](#)

Start date

Aug. 4, 2014, 1:19 a.m.

End date


Oct. 31, 2015, midnight

Bill cycle day

1

State

APPROVED

EUROSENTIMENT[License](#)

- Their personal access token;

Home » [View subscription](#)

### Welcome to "My Subscription"

Eurosentiment allows you to access resources and APIs via an access token, which you can get here. This way Eurosentiment allows precise access control and accounting resource usage. To learn more about this and see how to use this code please refer to the [documentation](#).

User Type

Content Provider [Change modality](#)

Token

f6a9204d-fccd-40b5-af15-37812eb3e1f0 [Reset Token](#)

Start date

Aug. 4, 2014, 1:19 a.m.

End date

Oct. 31, 2015, midnight

Bill cycle day

1

State

APPROVED

EUROSENTIMENT [License](#)

- The account registration start/end date, the day of their monthly bill calculation and the status of their account.

Users can change the type of their account or reset the token whenever, through the **Change modality** and **Reset token** buttons.

### 1.2.2 My Consumptions

In the section **Subscription -> My consumptions** consumers can see their own statistics:

Every time that users access a language resource or an analysis service, the system tracks this access by exploiting the personal access token; once a day, the stats module reads the logs and calculates the statistics.

The section **Subscription -> My consumers** shows the consumptions by other users of the resources/services belonging to the current user.


[Services ▾](#)
[Resources ▾](#)
[Profile ▾](#)
[Subscription ▾](#)
[Demos ▾](#)
[Help ▾](#)
[Sign out](#)

Signed in as **TestConsumer** |

[Home](#) » [View subscription](#)

## Welcome to "My Subscription"

Eurosentiment allows you to access resources and APIs via an access token, which you can get here. This way Eurosentiment allows precise access control and accounting resource usage. To learn more about this and see how to use this code please refer to the [documentation](#).

### User Type

Content Provider [Change modality](#)

### Token

f6a9204d-fccd-40b5-af15-37812eb3e1f0 [Reset Token](#)

### Start date

Aug. 4, 2014, 1:19 a.m.

### End date

Oct. 31, 2015, midnight

### Bill cycle day

4

### State

APPROVED


[EUROSENTIMENT](#)

[License](#)


[Services ▾](#)
[Resources ▾](#)
[Profile ▾](#)
[Subscription ▾](#)
[Demos ▾](#)
[Help ▾](#)
[Sign out](#)


Signed in as **TestConsumer** |

[Home](#) » [My Consumptions](#)

## Global Statistics

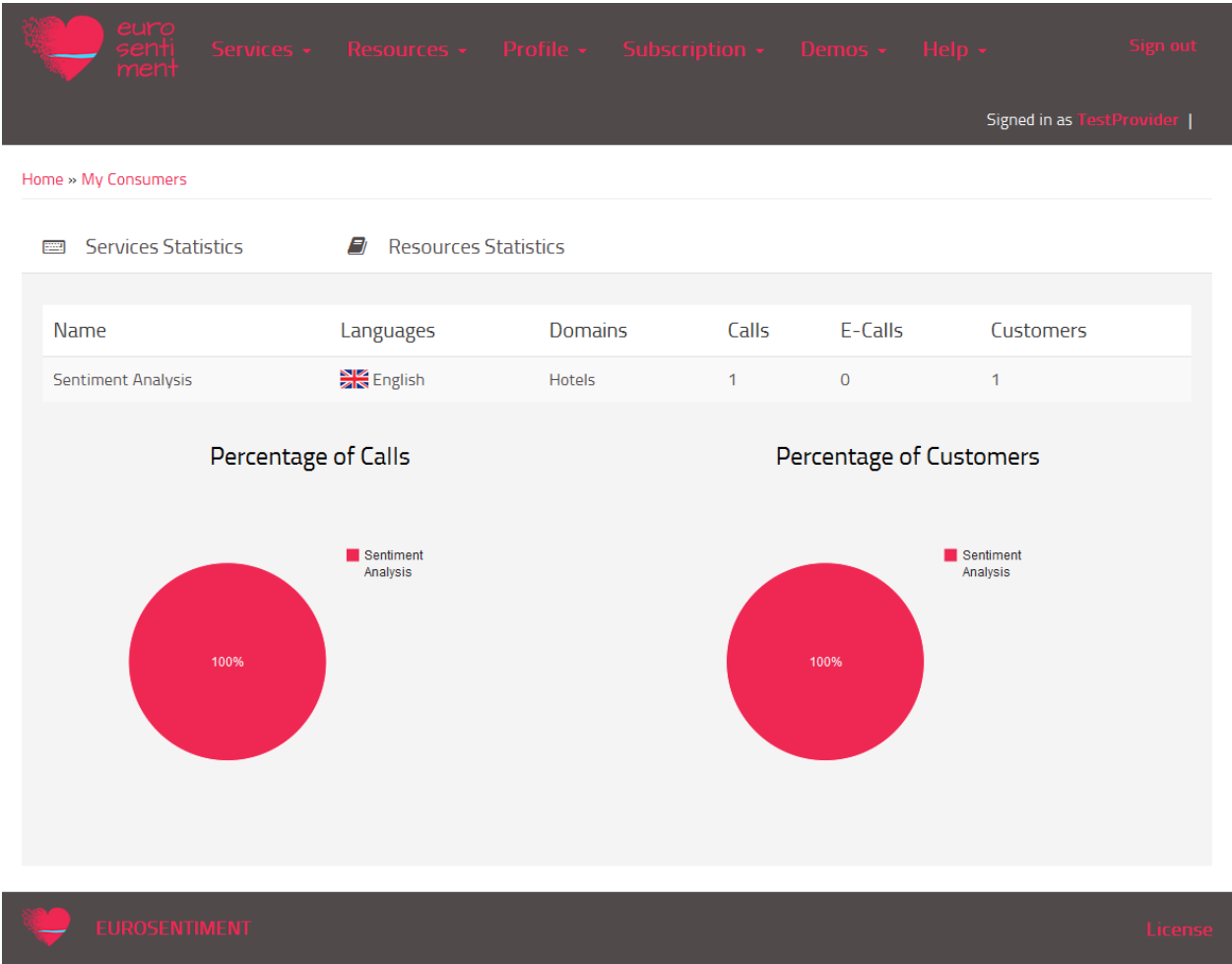
[Calls](#) [E-Calls](#) [Queries](#) [E-Queries](#) [Downloads](#) [E-Downloads](#)

## Specific Statistics

Domain	Language	Calls	E-Calls	Queries	E-Queries	Downloads	E-Downloads
Hotels	 English	1	0	0	0	0	0


[EUROSENTIMENT](#)

[License](#)



### 1.2.3 Billing

In the section **Subscription -> Billing** the consumer can see his own bills:

Home » Billing

€ Bills Reports

Date	N° Service calls	N° Resource calls	N° Resource downloads	Total	Download
2014-08-04 00:00:00.0	1	0	0	200.5 €	<a href="#">Download</a>

« 1 »

EUROSENTIMENT [License](#)

The picture above shows the bills of a user; the tabs allow the user to see revenue reports too. In this case, the user hasn't registered any services/resources, so the reports contain **zero** in every field. The total amount of fees is the result of the sum between the subscription fees (200) and the consuming fees (0,5). The billing module exploits the logs for calculating the bills. The calculation is performed once a month, when the current day is the same of the day of the account registration date: for each account, the billing module calculates a bill and a revenue report.

Users can download their bills/reports:

## 1.3 The Eurosentiment Format

The Eurosentiment format is an extension of already established formats for Linguistic Linked Data ([lemon](#), [NIF](#), [Onyx](#), [Marl](#), etc.) for their use in Sentiment Analysis. It covers the description of lexica and corpora on the language resource side, and the results from services.

In the following sections we will cover both cases individually.

### 1.3.1 The EUROSENTIMENT format for services and corpora

The Eurosentiment format is an extension of the [NIF](#) format data model for use in Sentiment Analysis. However, [NIF](#) and the Eurosentiment differ in one respect: Eurosentiment sets [JSON-LD](#) as its primary serialisation format, whereas [NIF](#) defaults to XML+RDF or turtle. It includes properties from [Marl](#), [Onyx](#) and other ontologies that complement those in [NIF](#) for sentiment and emotion tagging. However, [NIF](#) and the Eurosentiment differ in one respect: Eurosentiment sets [JSON-LD](#) as its primary serialisation format, whereas [NIF](#) defaults to XML+RDF or turtle.

JSON-LD is a subset of JSON that makes it possible to embed semantic information in plain JSON objects. It retains full compatibility with JSON while adding useful information.

Home » Billing

€ Bills Reports

Date	N° Service calls	N° Resource calls	N° Resource downloads	Total	Download
2014-08-04 00:00:00.0	1	0	0	200.5 €	<a href="#">Download</a>

« 1 »

Apertura di TestConsumer\_135.pdf

È stato scelto di aprire:

**TestConsumer\_135.pdf**  
 tipo: Adobe Acrobat Document (1,3 kB)  
 da: http://217.26.90.243:8080

Che cosa deve fare Firefox con questo file?

☐ Apriro con Adobe Reader 9.5 (predefinita)

☒ Salva file

☐ Da ora in avanti esegui questa azione per tutti i file di questo tipo.

OK Annulla

EUROSENTIMENT License

By using this serialisation format, Eurosentiment targets both semantic web developers and traditional developers alike.

## Overview

```
{
  "@context": [
    "http://demos.gsi.dit.upm.es/eurosentiment/static/context.jsonld",
  ],
  "@id": {{ processID }},
  "analysis": [
    {
      "@id": {{ analysisID }},
      "@type": [
        {{ analysisType }}
      ],
      "prov:wasAssociatedWith": {{ agent }},
      "dc:language": {{ language}},
      "marl:maxPolarityValue": {{ minValue }},
      "marl:minPolarityValue": {{ maxValue }}
    }
  ],
  "...",
  "domain": {{ domain }},
  "entries": [
    {
      "@id": {{ entry_id }},
      "dc:subject": {{ topic }},
      "emotions": [
        {
```

```

    "prov:generatedBy": {{ analysisID }},
    "onyx:hasEmotion": [
        {
            "onyx:hasEmotionCategory": {{ emotions[i].category }},
            "onyx:intensity": {{ emotions[i].emotion_intensity }}
        },
        [...]
    ]
}
[...]
```

```

],
"opinions": [
    {
        "prov:generatedBy": {{ analysisID }},
        "marl:polarityValue": {{ opinions[i].polarityValue }},
        "marl:hasPolarity": {{ opinions[i].polarity }},
        "marl:describesObject": {{ opinions[i].described_object }},
    },
    [...]
],
"nif:isString": {{ string_representation }},
"strings": [
    {
        "nif:anchorOf": {{ strings[i].value }},
        "itsrdf:taIdentRef": {{ strings[i].entity }},
        "nif:posTag": {{ strings[i].posTag }},
        "nif:lemma": {{ strings[i].lemma }}
    },
    [...]
]
},
[...]
```

**processID** Is the ID of the process that gathered the results.

**domain** Domain detected in the entries, or used by the analysis

**analysis** A set of results can be produced by combining the results from several analysis processes. Each of them needs to be described here.

**analysisID** Each of the analysis needs an unique URI so that the generated opinions/emotions can be linked to it. A set of results may aggregate the results from independent analysis (e.g. a sentiment analysis and an emotion analysis)

**analysisType** Example: *marl:SentimentAnalysis* or *onyx:EmotionAnalysis*

**algorithm** [In marl] Algorithm that was used to generate the results

**agent** Responsible for or creator of the analysis

**language** Language that the analysis uses. e.g. “*es*”

**minValue** [In marl opinions] Minimum value of the opinion value

**maxValue** [In marl opinions] Maximum value of the opinion value

**domain** Domain where the analysis was run. e.g. *wnd:electronics*

**entry\_id** Each entry must have a unique URI

**topic** The subject or subjects of the entry. e.g. *wnd:electronics*

**emotions** The emotions found in the context. Depending on the theory of emotions used, emotions can be categorised and/or be defined by different dimensions. This example represents the usual case which is a model using categories.

**category** Category of the emotion. e.g. *wna:Hatred*

**emotion\_intensity** Intensity of the emotion as defined by the algorithm

**opinions** The opinions found in the context.

**polarity** Polarity of the opinion. e.g. *marl:Positive*

**polarityValue** Numerical value of the polarity, as a floating point

**described\_object** Object that the opinion is about

**string\_representation** Plain text representation

**strings** A NIF context can be subdivided in substrings, which have their own properties. This is usually done to associate a particular string with an entity in Named Entity Recognition

**strings[i].value** Text representation

**strings[i].entity** Entity the string represents

**strings[i].posTag** Part-of-speech tag

**strings[i].lemma** Lemma of the word

## Context

The JSON-LD context contains semantic information about the properties in the JSON document, including convenient prefixes or namespaces. The Eurosentiment context would look like this:

```
{
  "@context": {
    "dc": "http://purl.org/dc/terms/",
    "dc:subject": {
      "@type": "@id"
    },
    "emotions": {
      "@container": "@list",
      "@id": "onyx:hasEmotionSet",
      "@type": "onyx:EmotionSet"
    },
    "marl": "http://www.gsi.dit.upm.es/ontologies/marl#",
    "nif": "http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#",
    "onyx": "http://www.gsi.dit.upm.es/ontologies/onyx#",
    "opinions": {
      "@container": "@list",
      "@id": "marl:hasOpinion",
      "@type": "marl:Opinion"
    },
    "prov": "http://www.w3.org/ns/prov#",
    "rdfs": "http://www.w3.org/2000/01/rdf-schema#",
    "analysis": {
      "@id": "prov:wasInformedBy"
    },
    "entries": {
      "@id": "prov:generated"
    }
  }
}
```



```

    },
    "strings": {
      "@reverse": "nif:hasContext",
      "@type": "nif:String"
    },
    "wnaffect": "http://www.gsi.dit.upm.es/ontologies/wnaffect#",
    "xsd": "http://www.w3.org/2001/XMLSchema#"
  }
}

```

## Examples

- Annotating one entry using a fictitious service (<http://example.com/analyse>) provided by <http://example.com>.  
Input: “My ipad is an awesome device”.

```

{
  "@context": [
    "http://demos.gsi.dit.upm.es/eurosentiment/static/context.jsonld"
  ],
  "results": {
    "analysis": [
      {
        "@id": "http://example.com/analyse",
        "@type": [
          "marl:SentimentAnalysis"
        ],
        "dc:language": "en",
        "marl:maxPolarityValue": 10.0,
        "marl:minPolarityValue": 0.0,
        "prov:wasAssociatedWith": "http://example.com"
      }
    ],
    "entries": [
      {
        "@id": "http://example.com/analyse?input=My%20ipad%20is%20an%20awesome%20device",
        "opinions": [
          {
            "marl:polarityValue": 9,
            "marl:hasPolarity": "marl:Positive",
            "marl:describesObject": "http://dbpedia.org/page/IPad",
            "prov:generatedBy": "http://example.com/analyse",
          }
        ],
        "nif:isString": "My ipad is an awesome device",
        "strings": [
          {
            "@id": "http://example.com/analyse?input=My%20ipad%20is%20an%20awesome%20device#char=3,6",
            "nif:anchorOf": "ipad",
            "itsrdf:taIdentRef": "http://dbpedia.org/page/IPad"
          }
        ]
      }
    ]
  }
}

```

- Annotating complex emotions in Spanish. Input: “Mi ipad me tiene harto”.

```
{
  "@context": [
    "http://demos.gsi.dit.upm.es/eurosentiment/static/context.jsonld"
  ],
  "results": {
    "analysis": [
      {
        "@id": "http://example.com/analyse",
        "@type": [
          "onyx:EmotionAnalysis"
        ],
        "dc:language": "es",
        "onyx:maxEmotionIntensity": 1.0,
        "onyx:minEmotionIntensity": 0.0,
        "prov:wasAssociatedWith": "http://example.com/"
      }
    ],
    "entries": [
      {
        "@id": "http://example.com/analyse?input=Mis%20ipad%20me%20tiene%20harto",
        "dc:language": "es",
        "opinions": [
        ],
        "emotions": [
          {
            "onyx:aboutObject": "http://dbpedia.org/page/IPad",
            "prov:generatedBy": "http://example.com/analyse",
            "onyx:hasEmotion": [
              {
                "onyx:hasEmotionCategory": "wna:dislike",
                "onyx:hasEmotionIntensity": 0.7
              },
              {
                "onyx:hasEmotionCategory": "wna:despair",
                "onyx:hasEmotionIntensity": 0.1
              }
            ]
          }
        ],
        "nif:isString": "My ipad is an awesome device",
        "prov:generatedBy": "http://example.com/analyse",
        "strings": [
          {
            "@id": "http://example.com/analyse?input=Mis%20ipad%20me%20tiene%20harto#char=3,6",
            "nif:anchorOf": "ipad",
            "itsrdf:taIdentRef": "http://dbpedia.org/page/IPad"
          }
        ]
      }
    ]
  }
}
```

## Other serialisation formats

The Eurosentiment format is semantic, as is the NIF Format. Although the preferred and mainly used serialisation format is JSON-LD, there are other serialisation formats that could be used as well.

For instance, it is particularly interesting to convert corpora to N-Triples for storage in a semantic server such as Virtuoso.

## Useful links

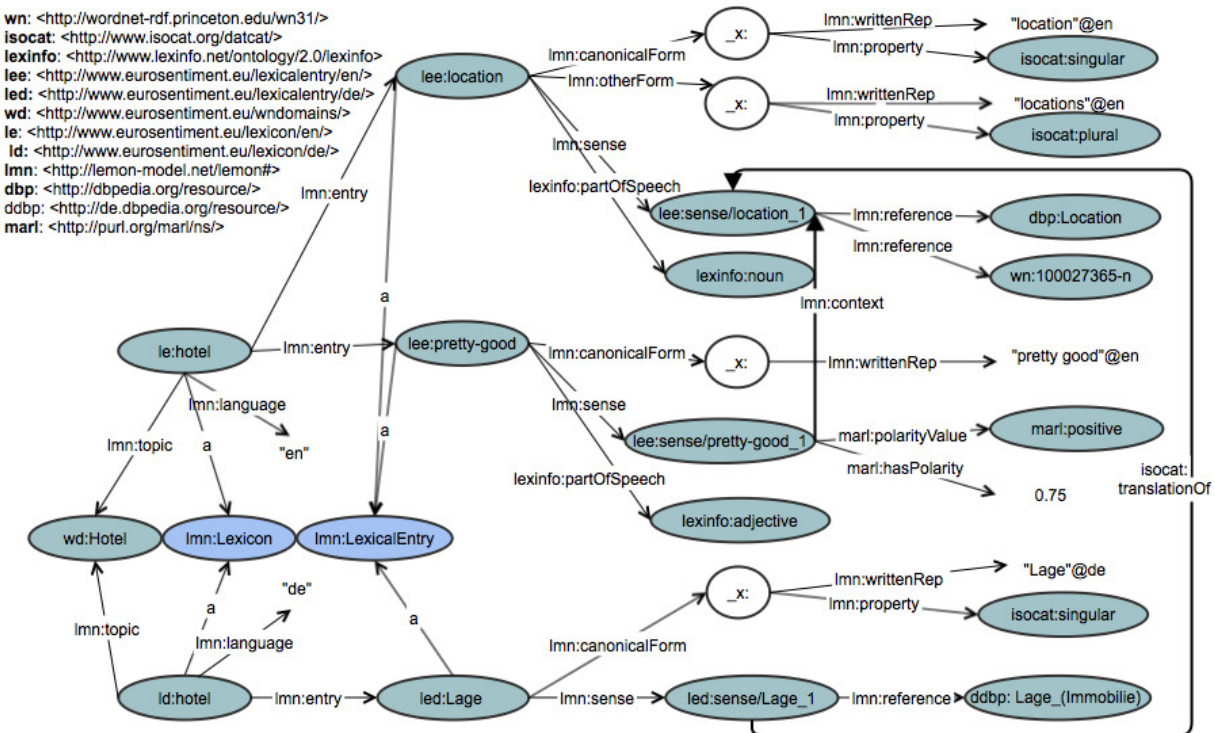
**NIF** <http://persistence.uni-leipzig.org/nlp2rdf/>

**JSON-LD** <http://json-ld.org>

### 1.3.2 The Eurosentiment Format for Lexica

The EUROSENTIMENT sentiment lexicons are represented in RDF using the **lemon** format extended with some properties from the **Marl** vocabulary for representing the sentiment information such as polarity and polarity value. Deliverable **D4.3** (p54-56) describes in details the model.

The following figure shows an example of RDF domain-specific lexicon.



In this figure we see snippets from two lexicons: a lexicon for the hotel domain in english (i.e. *le:hotel*) and a lexicon for the hotel domain in german (i.e. *ld:hotel*). The sentiment lexicons are composed of lexical entries: in our example *lee:location* and *lee:pretty-good* for the English lexicon and *led:Lage* for the german lexicon. Some of the lexical entries are domain aspects like *location* and *Lage* and some are sentiment words like *pretty good*.

Each lexical entry is defined by a `canonicalForm`, several `otherForm` properties representing the morphological variations, several senses and part of speech information. The connection of the EUROSENITMENT lexicons to other linguistic linked data datasets (i.e. DBpedia and Wordnet) happen at the sense level. For instance in our case, the sense of the *lee:location* lexical entry is linked to the <http://dbpedia.org/page/Location> entity and to the Wordnet synset: <http://wordnet-rdf.princeton.edu/wn31/100027365-n>.

In the case of sentiment lexical entries we use two more properties (i.e. *marl:PolarityValue* and *marl:hasPolarity*) to represent the sentiment information. In our example, *lee:pretty-good* has a positive polarity of 0.75 where the most positive value is 1 and the most negative value is -1.

The domain-specific aspect of the EUROSENTIMENT lexicons is given by the *lemon* property context which connects a sentiment word to a domain aspect. In the example, the *lmn:context* property signifies that the sentiment word *pretty good* has a positive value of 0.75 in the context of the *location* domain aspect.

We recall from [D4.3](#) that we generate domain-specific lexicons in other languages based on the initial english lexicon. For example, in our case, the german lexicon for hotel was automatically generated from the english lexicon for hotel. This relation between the two lexicons is represented via the *isocat:translationOf* relation between the senses of the translated lexical entries.

## 1.4 Tools

### 1.4.1 Eurosentiment Corpus Converter

#### Introduction

The purpose of the EUROSENTIMENT Corpus Converter is to translate information from legacy or non-semantic formats to the semantic formats used in EUROSENTIMENT.

The Corpus Converter has been tested with the corpora for sentiment and emotion analysis from the members of the consortium (mainly from Expert System and Paradigma Tecnológico). These corpora have been transformed to JSON-LD, using the Marl and Onyx vocabularies. Nevertheless, the Corpus Converter was designed as a generic tool to translate from and to a wide range of formats and vocabularies or ontologies.

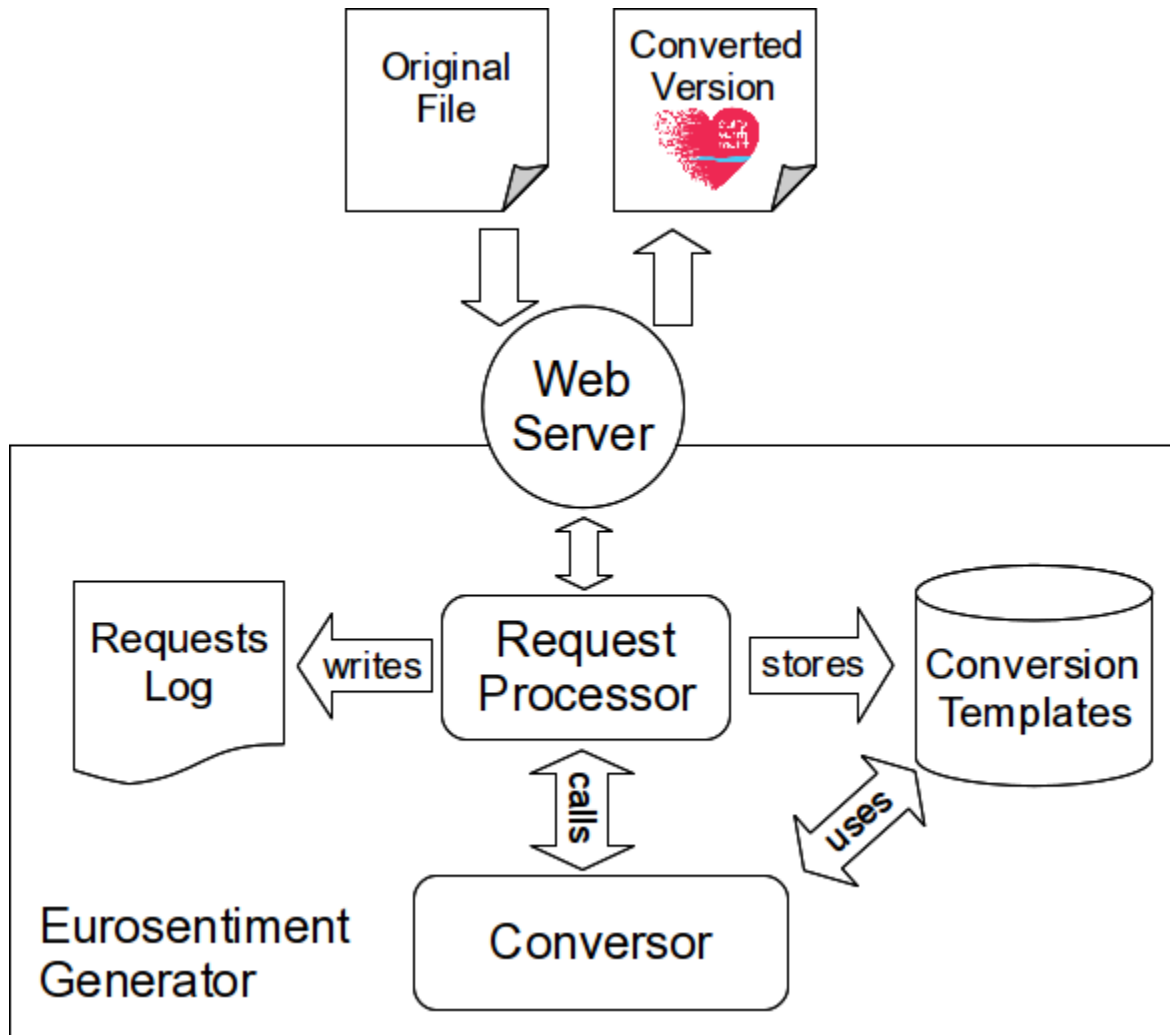
Translating a document does not require any technical qualification. The translation of documents can be done through a web portal. This is especially useful for demonstration and testing purposes or quick translations. However, in a real life scenario, there will be a big amount of information and files to be translated. Also, it should be possible to integrate our Corpus Converter in a pipeline or automated process, so that new content can be automatically converted. For these reasons, the Corpus converter also exposes a REST API. It only takes a POST request to translate a document.

The Corpus Converter has been designed to be extensible and to separate the technical aspects from the content and formats being translated. Our Corpus Converter itself is a convenient platform, but the actual translation is performed following a set of “Translation Templates”. These templates have access to the data in the original file, and determine the result of the translation.

An administrative web interface has been developed to make it easy to add new formats, improve translation templates or access the usage statistics.

#### Architecture

As can be seen in the figure below, the Corpus Converter is made out of A Web Server proxies all the user requests to the Request Processor. This processor is a Django application that has two roles: providing a REST and an Administrative interface. On the administrative side it deals with authorisation and authentication of users. It also stores the translation templates provided by the staff in the database. On the REST side, it forwards all valid user translation requests to the Translator module. This module opens the original file in any of the supported formats, it applies the requested translation template with its content, and then returns the resulting document.



### Translation Templates

For convenience, we used an easy-to-use templating language and engine for the Corpus Converter called [Jinja2](#). Despite its user-friendliness, it is a really powerful language. It features advanced loops, conditional clauses, functions and filters. It is essentially a stripped-down or subset of Python.

The file contents can be accessed from within the template like a stream or iterator. The Translator reads the original document, feeding it line by line (for plain text) or row by row (in spreadsheet files). Then, using Jinja2, it is easy to iterate over it and extract all the relevant information.

In addition to all the basic filters in Jinja2, the EUROSENTIMENT templates can also use a set of specific filters that make it easier to tokenise the input.

The annex contains a complete template.

### Supported Formats

As of this writing, the EUROSENTIMENT Corpus Converter accepts corpora in the following formats:

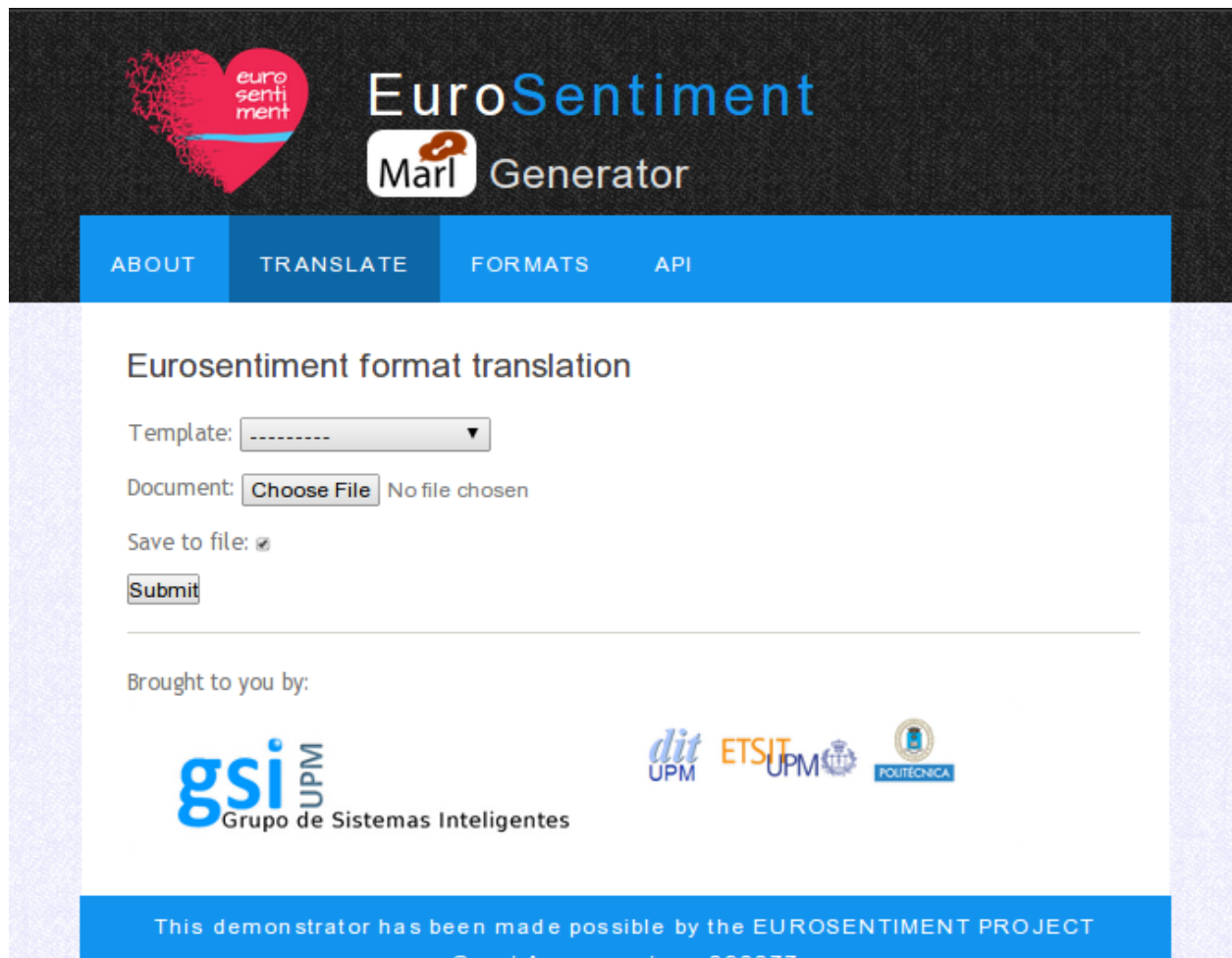
- Paradigma Tecnológico's Human Annotated Corpora (Tab-Separated-Values)

- Paradigma Tecnológico's Machine Annotated Corpora (Tab-Separated-Values)
- Paradigma Tecnológico's Synset-Aligned Corpora (Tab-Separated-Values)
- Expert System's Machine Annotated Corpora (Microsoft Excel)
- Expert System's Human Annotated Corpora (Microsoft Excel)
- Expert System's Corpora with Emotions (Microsoft Excel)
- TripAdvisor corpora (Raw text with custom tags)

## Usage

### Translating a document

Documents can be translated via the Web Interface or using the REST interface. Actually, the form in the Web is simply a convenient way of accessing the REST interface which shows all the available templates and a field to upload the desired file.



The screenshot displays the 'EuroSentiment Generator' web interface. At the top, there is a navigation bar with links for 'ABOUT', 'TRANSLATE', 'FORMATS', and 'API'. The 'TRANSLATE' tab is currently selected. Below the navigation bar, the main heading is 'Eurosentiment format translation'. The form includes a 'Template:' dropdown menu, a 'Document:' section with a 'Choose File' button and the text 'No file chosen', a 'Save to file:' checkbox which is checked, and a 'Submit' button. At the bottom of the form, it says 'Brought to you by:' followed by logos for 'gsi UPM Grupo de Sistemas Inteligentes', 'dit UPM', 'ETSIT UPM', and 'POLTÉCNICA'. A blue footer bar at the very bottom contains the text: 'This demonstrator has been made possible by the EUROSENTIMENT PROJECT Grant Agreement no: 286277'.

Figure 1.1: Translating a document through the web

The Corpus Converter endpoint takes the following parameters:

- input (i): The original file to be translated
- informat (f): The format of the original file
- intype (t) [Optional]:
  - direct (default)
  - url
  - file
- outformat (o):
  - json-ld
  - rdxml
  - turtle (default, to comply with NIF)
  - ntriples
  - trix
- base URI (u) [Optional]: base URI to use for the corpus
- prefix (p) [Optional]: prefix to replace the base URI
- template (t) [Optional]: ID of the template to use. If it is omitted, a template to convert from informat to outformat will be used, or a template from informat to another format (e.g. json-ld), with automatic conversion.
- toFile [Optional]: Whether the result should be sent in the response (default) or written to a file. For convenience, this value defaults to False when using the Web Form.

Using the command line tool *curl*, a request can be made like this:

```
curl http://demos.gsi.dit.upm.es/eurosentiment/marlgenerator/process -F"intype=file" -F"informat=Exar
```

## Adding a template

Editing a template is simple. First, visit the administration URL. If it is your first login or if your session expired, you will be greeted by a login screen:

Just enter your username and password, and the administration interface should appear.

It is also possible to add a format from this menu, clicking on the “Plus” icon:

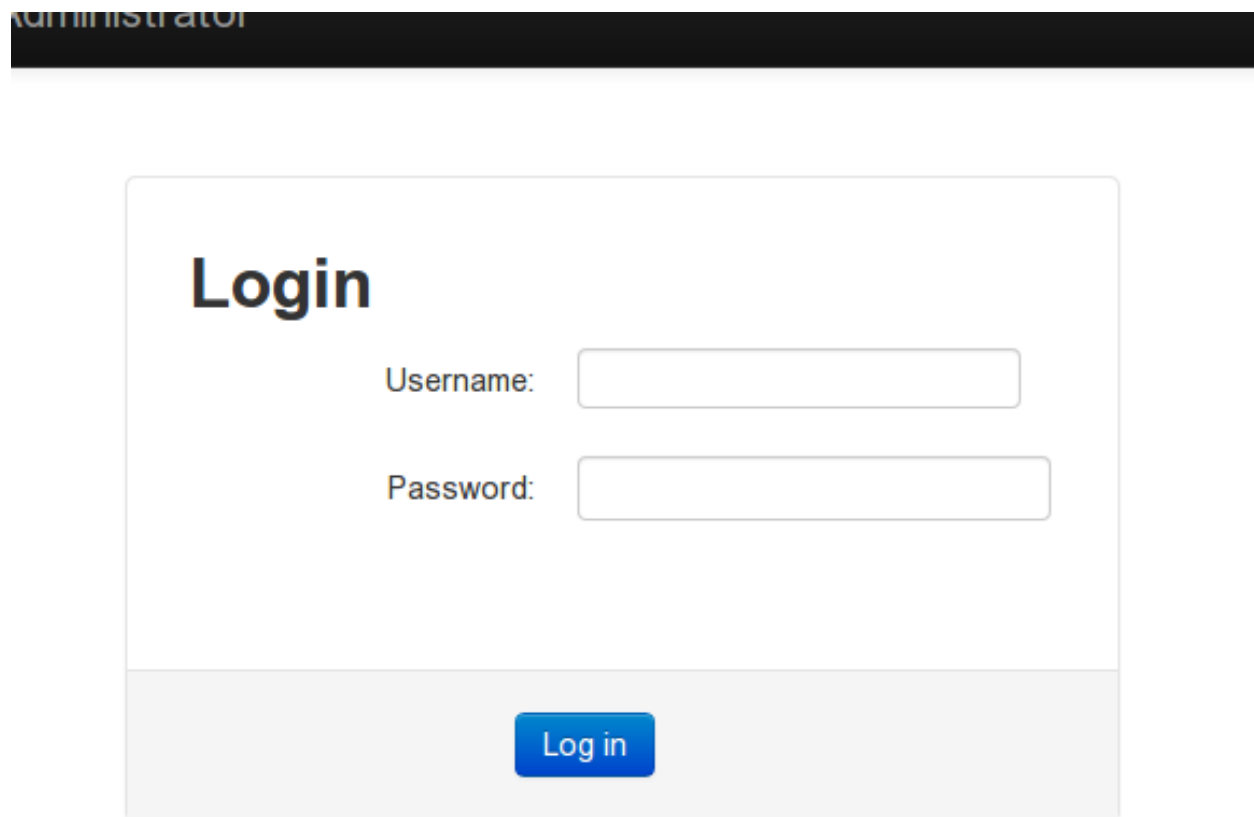
## Checking usage statistics

Once logged in as a superuser, you can also add new users and check the requests that have been made for each format.

To check the requests, click on “Translation Requests” in the administration panel.

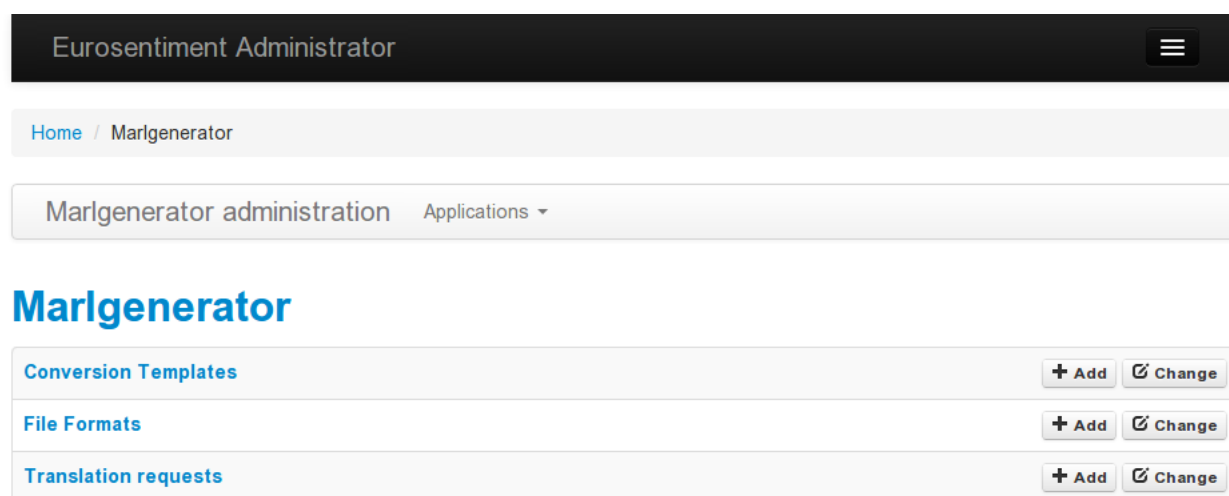
In addition to simply checking the requests, it is also possible to filter the requests using different parameters. This feature is especially useful if you want to study the popularity of a format, or to compare different templates for the same formats.

## Example Template



The image shows a login prompt form. At the top, there is a dark header bar with the word "administrator" in a light color. Below this, the form is centered on a light gray background. The form has a title "Login" in a large, bold, black font. Below the title, there are two input fields: "Username:" and "Password:". Each field is a simple rectangular box with a light gray border. Below the input fields, there is a blue button with the text "Log in" in white. The button has a slight shadow and is centered horizontally.

Figure 1.2: Login prompt




The image shows the administration interface of the Eurosentiment Administrator. At the top, there is a dark header bar with the text "Eurosentiment Administrator" on the left and a hamburger menu icon on the right. Below the header, there is a breadcrumb trail: "Home / Marlgenerator". Below this, there is a section titled "Marlgenerator administration" with a dropdown menu labeled "Applications". Below this, there is a section titled "Marlgenerator" in a large, bold, blue font. Below this, there is a table with three rows: "Conversion Templates", "File Formats", and "Translation requests". Each row has a "+ Add" button and a "Change" button with a pencil icon.

Conversion Templates	+ Add	Change
File Formats	+ Add	Change
Translation requests	+ Add	Change

Figure 1.3: Administration Interface



Eurosentiment Administrator 

[Home](#) / [Marlgenerator](#) / [Conversion Templates](#) / ExpSys\_MA\_to\_Marl

Change Conversion Template History


Fields in **bold** are required.


Template name:

Template: 

```
{% set sheet=f.sheet_by_index(0) %}
{
  "@context": {
    "marl": "http://gsi.dit.upm.es/ontologies/marl#",
    "expsys": "http://expertsystems.com/domains#",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "prov": "http://www.w3.org/ns/prov#",
    "marl:domain": {
      "@type": "@id"
    }
  },

```

Input format:  

Output format:  

Delete Save and add another Save and continue editing Save

Figure 1.4: Editing a template

Add File Format

Fields in **bold** are required.

**Format name:**

**File extension:**

csv - Comma Separated Val ▼

**MIME type:**

text/plain

**Description of the format:**

Example of the format:

Choose File

No file chosen

Save

Figure 1.5: Adding a format on the fly

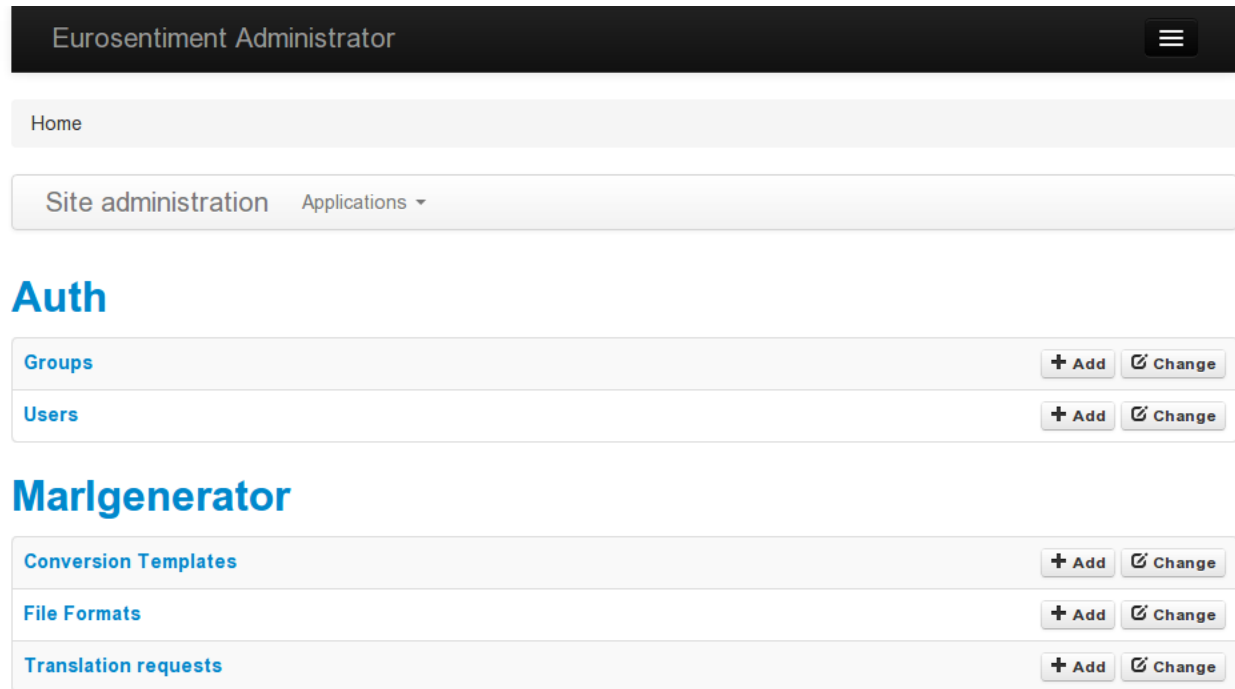







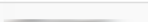
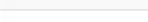
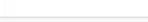
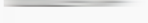




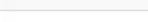

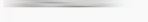
Figure 1.6: Superuser panel

```
{
  "@context": [
    "http://demos.gsi.dit.upm.es/eurosentiment/static/context.jsonld",
  ],
  "@id": "{{ linesplit(f.name, "/") [-1] }}",
  "analysis": [
    {
      "@id": "{{ linesplit(f.name, "/") [-1] }}#MachineAnnotated",
      "@type": [
        "marl:SentimentAnalysis"
      ],
      {% if language %}
        "dc:language": "{{ language }}",
      {% endif %}
      "marl:maxPolarityValue": 10.0,
      "marl:minPolarityValue": 0.0,
      "prov:wasAssociatedWith": "pt:agent"
    }
  ],
  "entries": [
{% for line in f %}
{% set i=linesplit(line, "\t") %}
{% set node="_:BlankNode%s" % loop.index %}
{% set text = i[0] %}
{% set syntax=linesplit(i[1][1:-1], ",") %}
{% set pol= i[2] | float %}
    {
      "@id": "{{ node }}",
      "opinions": [
```

Eurosentiment Administrator 

Home / Marlgenerator / Translation requests

Select translation request to change [+ Add translation request](#)  Filter 

<input type="checkbox"/>	Started 	Template used	Input format	Output format	Request IP
<input type="checkbox"/>	Oct. 22, 2013, 11:23 a.m.	PT_HA_to_Marl	PT Human Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 22, 2013, 8:49 a.m.	ExpSys_HA_to_Marl	Expert System's Human Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 22, 2013, 8:44 a.m.	ExpSys_MA_to_Marl	Expert System's Machine Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 22, 2013, 4:13 a.m.	ExpSys_HA_to_Marl	Expert System's Human Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 22, 2013, 4:09 a.m.	PT_HA_to_Marl	PT Human Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 22, 2013, 3:53 a.m.	PT_SA_to_Marl	PT Synsets	Marl JSON-LD	
<input type="checkbox"/>	Oct. 22, 2013, 3:42 a.m.	PT_HA_to_Marl	PT Human Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 22, 2013, 3:33 a.m.	PT_HA_to_Marl	PT Human Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 22, 2013, 2:39 a.m.	ExpSys_HA_to_Marl	Expert System's Human Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 21, 2013, 11:43 a.m.	PT_MA_to_Marl	PT Machine Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 21, 2013, 11:37 a.m.	PT_SA_to_Marl	PT Synsets	Marl JSON-LD	
<input type="checkbox"/>	Oct. 21, 2013, 11:02 a.m.	ExpSys_HA_to_Marl	Expert System's Human Annotated	Marl JSON-LD	
<input type="checkbox"/>	Oct. 21, 2013, 10:57 a.m.	ExpSys_MA_to_Marl	Expert System's Machine Annotated	Marl JSON-LD	


Action:    0 of 13 selected

Figure 1.7: Log of requests

Eurosentiment Administrator

[Home](#) / [Marlgenerator](#) / Translation requests

Select translation request to change

+ Add translation request

Search

Filter

<input type="checkbox"/>	Started	Template used
<input type="checkbox"/>	Oct. 22, 2013, 11:23 a.m.	PT_HA_to_Marl
<input type="checkbox"/>	Oct. 22, 2013, 8:49 a.m.	ExpSys_HA_to_Marl
<input type="checkbox"/>	Oct. 22, 2013, 8:44 a.m.	ExpSys_MA_to_Marl
<input type="checkbox"/>	Oct. 22, 2013, 4:13 a.m.	ExpSys_HA_to_Marl
<input type="checkbox"/>	Oct. 22, 2013, 4:09 a.m.	PT_HA_to_Marl
<input type="checkbox"/>	Oct. 22, 2013, 3:53 a.m.	PT_SA_to_Marl
<input type="checkbox"/>	Oct. 22, 2013, 3:42 a.m.	PT_HA_to_Marl
<input type="checkbox"/>	Oct. 22, 2013, 3:33 a.m.	PT_HA_to_Marl
<input type="checkbox"/>	Oct. 22, 2013, 2:39 a.m.	ExpSys_HA_to_Marl
<input type="checkbox"/>	Oct. 21, 2013, 11:43 a.m.	PT_MA_to_Marl
<input type="checkbox"/>	Oct. 21, 2013, 11:37 a.m.	PT_SA_to_Marl
<input type="checkbox"/>	Oct. 21, 2013, 11:02 a.m.	ExpSys_HA_to_Marl
<input type="checkbox"/>	Oct. 21, 2013, 10:57 a.m.	ExpSys_MA_to_Marl

Action:

Go

0 of 13 s

BY STARTED

Any date

Today

Past 7 days

This month

This year

BY TEMPLATE USED

All

PT\_HA\_to\_Marl

PT\_SA\_to\_Marl

PT\_MA\_to\_Marl

ExpSys\_HA\_to\_Marl

ExpSys\_MA\_to\_Marl

BY INPUT FORMAT

All

PT Human Annotated

Marl JSON-LD

PT Synsets

PT Machine Annotated

Expert System's Human Annotated

Expert System's Machine Annotated

BY OUTPUT FORMAT

All

Figure 1.8: Filtering requests

```
        {

{% if pol%}
                                "marl:polarityValue": {{ pol }},
{% if pol > 5 %}
                                "marl:hasPolarity": "marl:Positive"
{% elif pol < 5 %}
                                "marl:hasPolarity": "marl:Negative"
{% else %}
                                "marl:hasPolarity": "marl:Neutral"
{% endif %}
{% endif %}
        }
    ],
    "nif:isString": {{ text | escapejs }},
    "strings": [
        {% for s in syntax %}
        {
            {% set parts=linesplit(s, ";;") %}
            "nif:anchorOf": {{ parts[0] | escapejs }},
            "nif:posTag": "pt:{{ parts[1] }}",
            "nif:lemma": {{ parts[2] | escapejs }} }
            {% if not loop.last %}, {% endif %}{% endfor %}
        ]
    } {% if not loop.last %} , {% endif %}
{% endfor%}
    ]
}
```

### 1.4.2 Eurosentiment Playground

EUROSENTIMENT provides services and resources for Sentiment Analysis in several languages. There are several utilities, code snippets and instructions on how to make use of the platform publicly available. However, all of them require the installation of a third party tool or the use of a programming language to consume the API. The EUROSENTIMENT Playground solves this problem by providing an easy-to-use web interface to make API calls. Read our simple instructions and start using EUROSENTIMENT today!

The playground is available here: <http://demos.gsi.dit.upm.es/eurosentiment-playground/>

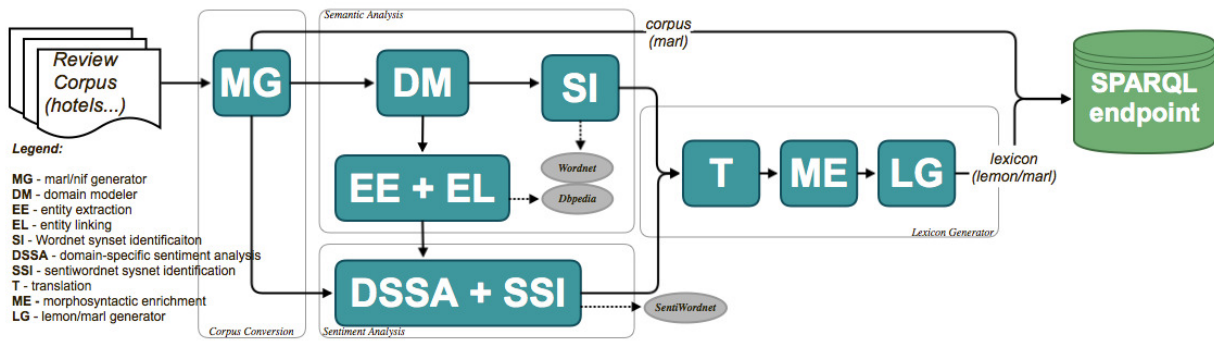
### 1.4.3 Language Resource Adaptation Pipeline

In this section we describe the Language Resource Adaptation Pipeline various components and provides links to the source code for these components.

The Language Resource Adaptation Pipeline (a.k.a. LRAP) implements a methodology for legacy language resource adaptation that generates domain-specific sentiment lexicons organized around domain entities described with lexical information and sentiment words described in the context of these entities.

The outcome of the Language Resource Adaptation Pipeline are annotated corpora represented the NIF/Marl format and domain-specific sentiment lexicons represented in RDF using the Lemon/Marl format. The legacy language resources are enriched with semantics and additional linguistic information from resources like DBpedia and BabelNet.

There are four main steps of the LRAP as shown in the Figure bellow:



1. **Corpus Conversion:** normalizes the different language resources to a common schema based on Marl and NIF; The corpus convertor tool was described earlier in a separate section.
2. **Semantic Analysis:** extracts the domain-specific entity classes and named entities and identifies links between these entities and concepts from the LLOD Cloud. The Semantic Analysis step consists of: **Domain Modeller (DM)**, **Entity Extraction (EE)**, **Entity Linking (EL)** and **Synset Identification (SI)** components.
3. **Sentiment Analysis:** extracts contextual sentiments and identifies SentiWordNet synsets corresponding to these contextual sentiment words. The Sentiment Analysis step consists of: **Domain-Specific Sentiment Polarity Analysis (DSSA)** and **Sentiment Synset Identification (SSI)** components.
4. **Lexicon Generator:** uses the results of the previous steps, enhances them with multilingual and morphosyntactic information and converts the results into a lexicon based on the lemon and Marl formats. The Lexicon Generator step consists of: **MorphoSyntactic Enrichment (ME)**, **Machine Translation(T)** and **lemon/Marl Generator(LG)** components.

Different language resources are processed with variations of the given adaptation pipeline. For more details on the LRAP and the domain-specific lexicons it generates please check our dissemination material:

- Presentation at 5th International Workshop on EMOTION, SOCIAL SIGNALS, SENTIMENT & LINKED OPEN DATA: “Generating Linked-Data based Domain-Specific Sentiment Lexicons from Legacy Language and Semantic Resources” - Gabriela Vulcu, Paul Buitelaar, Sapna Negi, Bianca Pereira, Mihael Arcan, Barry Coughlan, Fernando J. Sanchez and Carlos A. Iglesias
- Poster at the Data Challenge at the 3rd Workshop on Linked Data in Linguistics “Linked-Data based Domain-Specific Sentiment Lexicons” - Gabriela Vulcu, Raul Lario Monje, Mario Munoz, Paul Buitelaar and Carlos A. Iglesias





---

## Services and Resources

---

### 2.1 Services

The Eurosentiment Portal offers a series of services that are useful for Sentiment Analysis such as entity recognition or domain detection, as well as sentiment and emotion analysis themselves.

#### 2.1.1 How to add a new service

This section describes how a new service is added to the EUROSENTIMENT LRP.

**Service upload steps:**

0. Precondition: the user is logged in: <https://portal.eurosentiment.eu/> with his username and password
1. Click on the ‘Services’ tab -> ‘Add a service’
2. **Fill in the ‘Service creation form’**
  - Name -> give a name of your service
  - Description -> add a detailed description of your service
  - HTTP method -> select the HTTP method and fill in the service’s access URI
  - Credentials -> if the service endpoint is not public, please provide an access token
  - Request cost -> fill in the cost per request to your service
  - Request limit -> fill in the limit of requests a user of the service should not exceed
  - Language -> select from the language drop-down list
  - Domain -> select from the drop-down list
  - Fill in the Fact sheet with similar details and send an email to [eurosentimentpt@gmail.com](mailto:eurosentimentpt@gmail.com)
3. Click on the ‘Create’ button

#### 2.1.2 NIF API

**GET** `/services/server/access/` (*service\_id*)

Access the service at *service\_url*. The *service\_id* can be retrieved from the service page in the Portal. Since the requests to the server are likely to be long, **POST** `/services/server/access/` (*service\_id*) is recommended.

**Example request:**

```
GET /service/access/SentimentAnalysisExample?input=I%20love%20EUROSENTIMENT HTTP/1.1
Host: eurosentiment.eu
Accept: application/json, text/javascript
```

**Example response:**

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/javascript

{
  "@context": [
    "http://eurosentiment.eu/context.jsonld",
    {
      "@base": "http://eurosentiment.eu/service/access/SentimentAnalysisExample#"
    }
  ],
  "analysis": [
    {
      "@id": "SentimentAnalysisExample",
      "@type": "marl:SentimentAnalysis",
      "dc:language": "en",
      "marl:maxPolarityValue": 10.0,
      "marl:minPolarityValue": 0.0
    }
  ],
  "domain": "wndomains:electronics",
  "entries": [
    {
      "opinions": [
        {
          "prov:generatedBy": "SentimentAnalysisExample",
          "marl:polarityValue": 7.8,
          "marl:hasPolarity": "marl:Positive",
          "marl:describesObject": "http://eurosentiment.eu",
        }
      ],
      "nif:isString": "I love EUROSENTIMENT",
      "strings": [
        {
          "nif:anchorOf": "EUROSENTIMENT",
          "nif:taIdentRef": "http://eurosentiment.eu"
        }
      ]
    }
  ]
}
```

**Query Parameters**

- **input** (*i*) – No default. Depends on informat and intype
- **informat** (*f*) – one of *turtle* (default), *text*, *json-ld*
- **intype** (*t*) – one of *direct* (default), *url*
- **outformat** (*o*) – one of *turtle* (default), *text*, *json-ld*
- **prefix** (*p*) – prefix for the URIs

**Request Headers**

- **Accept** – the response content type depends on *Accept* header
- **X-Eurosentiment-Token** – optional OAuth token to authenticate

**Response Headers**

- **Content-Type** – this depends on *Accept* header of request

**Status Codes**

- **200 OK** – no error
- **404 Not Found** – service not found

**POST** `/services/server/access/` (*service\_id*)

The same as the previous method. This is the recommended method.

**Form Parameters**

- **i/input** – No default. Depends on *informat* and *intype*
- **f/informat** – one of *turtle* (default), *text*, *json-ld*
- **t/intype** – one of *direct* (default), *url*
- **o/outformat** – one of *turtle* (default), *text*, *json-ld*
- **p/prefix** – prefix for the URIs

**Request Headers**

- **Accept** – the response content type depends on *Accept* header
- **X-Eurosentiment-Token** – optional OAuth token to authenticate

**Response Headers**

- **Content-Type** – this depends on *Accept* header of request

**Status Codes**

- **200 OK** – no error
- **404 Not Found** – service not found

## 2.1.3 Create your own services

If, instead of using any of the provided services you want to roll your own, you can also contribute to the Eurosentiment pool by publishing your service. The [Eurosentiment github repository](#) contains a series of tutorials that will help you get started with the process, including some complete examples of sentiment analysis service in different programming languages.

As of this writing, there are examples in:

- Java: <https://github.com/eurosentiment/eurosentiment-tutorial-java>
- Python: <https://github.com/eurosentiment/eurosentiment-tutorial-python>
- Ruby: <https://github.com/eurosentiment/eurosentiment-tutorial-ruby>

## 2.2 Resources

### 2.2.1 How to add a new language resource

This section describes how a new language resource is added to the EUROSENTIMENT LRP. You can also watch a video (<https://www.dropbox.com/s/86dgo4u4k9gf0i6/EUROSENTIMENT-adding-a-new-LR.mp4?dl=0>) that describes step-by-step the process.

#### Resource upload steps:

0. Precondition: the user is logged in: <https://portal.eurosentiment.eu/> with his username and password
1. Click on the 'Resources' tab -> 'Add a resource'
2. **Fill in the 'Resource creation form'**
  - Name -> give a name of your LR
  - Graph URI prefix -> the platform will suggest you a unique graph URI (if you like it, leave it like it is)
  - Short Name -> give a short name of your LR. It will be used to dynamically generate the graph URI
  - Language -> select form the language drop-down list
  - Application Domain -> select form the drop-down list
  - Description -> add a detailed description of your language resource.
  - Resource type -> chose from the drop-down list the resource type you want to add.
  - Access control -> select the type of license under which you want to publish the LR
  - Upload the LR file
  - Fill in the Fact sheet with similar details and send an email to [eurosentimentpt@gmail.com](mailto:eurosentimentpt@gmail.com)
3. Click on the 'Create' button
4. **What will happen behind the scenes:**
  - an email is sent to the EUROSENTIMENT LRP
  - you are notified by email that your LR was submitted and will be reviewed before added to the LRP as soon as possible
  - an administrator will carefully read the resource submission and will decide if the language resource will be added to the language resource pool
  - if the LR is provided in an existing format then move to 4.6
  - if the LR is provided in a new format not known previously to the platform, the administrator will develop a specific language resource adaptation pipeline
  - the administrator runs the language resource adaptation pipeline and the provided LR is processed, converted to RDF and linked-data and the result is uploaded to the SPARQL endpoint
  - you are notified by email that your language resource was processed
5. Click on the 'Resources' -> 'List Own resources' and see your newly added LR that can be used by you or other users of the EUROSENTIMENT LRP

## 2.2.2 Resources API

### POST /sparql

The posts tagged with *tag* that the user (*user\_id*) wrote.

#### Example request:

```
POST /sparql HTTP/1.1
Host: eurosentiment.eu
Content-Length: 199
x-eurosentiment-token: 23aee871-d18d-4afa-c2e3-283f8ae9232ca
Accept-Encoding: gzip, deflate, compress
Accept: */*
content-type: application/json
```

```
{"query": "PREFIX lemon: <http://lemon-model.net/lemon#>\nSELECT * FROM <http://www.eurosentimen
```

#### Example response:

```
HTTP/1.1 200 OK
date: Mon, 07 Jul 2014 16:46:21 GMT
content-length: 596
content-type: application/json
server: Jetty(8.1.10.v20130312)

{
  "head": {
    "link": [],
    "vars": [
      "s",
      "sense"
    ]
  },
  "results": {
    "bindings": [
      {
        "s": {
          "type": "uri",
          "value": "http://www.eurosentiment.eu/dataset/hotel/es/gabvul/9/lexicalentry"
        },
        "sense": {
          "type": "uri",
          "value": "http://www.eurosentiment.eu/dataset/hotel/es/gabvul/9/lexicalentry"
        }
      },
      {
        "s": {
          "type": "uri",
          "value": "http://www.eurosentiment.eu/dataset/hotel/es/gabvul/9/lexicalentry"
        },
        "sense": {
          "type": "uri",
          "value": "http://www.eurosentiment.eu/dataset/hotel/es/gabvul/9/lexicalentry"
        }
      }
    ],
    "distinct": false,
    "ordered": true
  }
}
```

```
}
```

:data :reqheader Accept: the response content type depends on

*Accept* header

### Request Headers

- *Authorization* – optional OAuth token to authenticate

### Response Headers

- *Content-Type* – this depends on *Accept* header of request

### Status Codes

- *200 OK* – no error
- *404 Not Found* – there's no user

## 2.2.3 Language Resource Types

This section describes the type of language resources that the EUROSENTIMENT LRP supports. Based on the initial list of language resources (see [here](#)) from the project we identified the following supported formats. please not in brackets the number of the Lngua resource addressed by the listed format.

1. **domain-specific-lexicon-TSV** (5): Tab-separated-values file that describes sentiments in the context of domain aspects (e.g. myFile.tsv). The header of the TSV file should have the following columns:

```
entityWNid entityPOS entity sentiWNid sentiPOS sentiment sentiScore
```

where:

- *entityWNid*: WordNet30 synset ID of the domain aspect (e.g. 02671062).
- *entityPOS*: part of speech of the domain aspect (i.e. n, a, v, r).
- *entity*: domain aspect as string (e.g. “access”).
- *sentiWNid*: WordNet30 synset ID of the sentiment associated with the domain aspect (00979366).
- *sentiPOS*: part of speech of the sentiment word (i.e. n, a, v, r).
- *sentiment*: sentiment word as string (e.g. “quick”)
- *sentiScore*: the polarity value of the sentiment (a rational number between -1 and 1, where -1 is very negative, 0 is neutral and 1 is very positive).

e.g.

```
02671062 n access 00979366 a quick 0.5
```

2. **entiment-lexicon-CSV** (45 (needs to be converted), 54): Comma-separated-values file that describes the sentiment words and their polarities from a domain. (e.g. myFile.csv). The header of the CSV file should have the following columns:

```
sentiment,sentimentPOS,sentimentScore,morphosyntacticVariations
```

where:

- *sentiment*: the sentiment word in the domain.
- *sentimentPOS*: the part-of-speech of the sentiment word.

- *sentimentScore*: the polarity value of the sentiment (a rational number between -1 and 1, where -1 is very negative, 0 is neutral and 1 is very positive).
- *morphosyntacticVariations*: the sentiment morphosyntactic variations.

e.g.

```
Besserung|NN 0.40 Besserungen,Besserungnen
```

3. **review-corpus-no-polarity** (36): A file containing one review per line.

e.g.

```
The location was great and the staff friendly. I like it!
```

```
The room was a bit too small.
```

...

4. **review-corpus-overall-polarity-TSV** (3, 31, 34): A tab-separated-values file with reviews and overall polarity. The header of the TSV file should have the following columns:

```
reviewText overallPolarity
```

where:

- *reviewText*: A string that contains the review (no tabs in the string).
- *overallPolarity*: The overall polarity of the given review text (rational number between -1 and 1, where -1 is very negative, 0 is neutral and 1 is very positive).

e.g.

```
Rien à dire. Très bon produit de qualité. 1.0
```

5. **review-corpus-pos-lemma-wn-overall-polarity-TSV** (2, 4): A tab-separated-values file with columns of the following form:

```
reviewText annotation overallPolarity
```

where:

- *reviewText*: A string that contains the review (no tabs in the string).
- *annotation*: A list with comma-separated values for each word in the text review containing: word, part-of-speech, lemma and a list of possible WordNet30 synsetIDs.
- *overallPolarity*: The overall polarity of the given review text (rational number between -1 and 1, where -1 is very negative, 0 is neutral and 1 is very positive).

e.g.

```
Excellent location. [Excellent;;JJ;;excellent;;[], location;;NN;;location;;[n#01051331]]
0.8
```

6. **review-corpus-pos-lemma-wn-overall-polarity-Excel** (7,8,9,10,11,12,13,14,17,18,20,21,22,50,51,52,53): A .gz compressed folder containing Excel files with several reviews per file. Each review in the Excel file is spread over several lines. The header of the Excel file is: TEXT LEMMA WN\_POS WN\_SYNSEST DOMAIN SENTIMENT EMOTION.

where:

- *TEXT*: the full review text in the first line; subsequent lines have one word of the review per line.
- *LEMMA*: nothing in the first line; in subsequent lines it describes the lemma value of an individual word from the review.

- *WN\_POS*: nothing in the first line; in subsequent lines it describes the part-of-speech value of an individual word from the review.
  - *WN\_SYNSET*: nothing in the first line; in subsequent lines it describes the WordNet30 synset ID value of an individual word from the review.
  - *DOMAIN*: the domain of the review (only in the first line)
  - *SENTIMENT*: (only in the first line)
  - *EMOTION*: (only in the first line)
7. **dataset-rdf** (25, 26, 27, 29, 37(needs conversion to RDF),42,55): RDF dump (\*.nt.gz) of linked data dataset like WordNet, DBpedia, BabelNet.
  8. **aspects-review-corpus-TripAdvisor** (49): A file with annotated reviews and aspect ratings. Each review in the file is spread over several lines where each line starts with a dedicated tag as in the example below.

e.g.

```
<Author>IndieLady
<Content>Lovely hotel, unique decor, friendly front desk staff [...]
<Date>Nov 13, 2008
<No. Reader>-1
<No. Helpful>-1
<Overall>4
<Value>5
<Rooms>4
<Location>5
<Cleanliness>4
<Check in / front desk>5
<Service>5
<Business service>-1
```

9. **aspects-review-corpus-Amazon** (44): A file that consists of plain text reviews for products with custom ratings annotations that spread over several lines. The marker for a new review is [t] whereas the numbers in brackets stand for the rating of a certain aspect in the review. See below an example:

e.g.

```
[ t ] the best 4mp compact digital available camera[+2]## this camera is
perfect for an enthusiastic amateur photographer . picture [+3] ,
macro[+3]## the pictures are razor sharp , even in macro . . .
```

10. **Opener lexicon** : Semicolon-separated-values file with the following columns:

wordnetSynsetID; POStag; polarity; confidence; lemmas; manualReviewFlag  
where:

- *wordnetSynsetID* : wordnet 30 synsetID.
- *POStag* : part-of-speech tag
- *polarity* : sentiment polarity which can be -1, 0 or 1 for negative, neutral and positive respectively.



- *confidence* : confidence assigned by the propagation algorithm
- *lemmas* : lemmas of this synset in wordnet separated by comma
- *manualReviewFlag* : -1 if no manual review has been done and + 1 if they have been reviewed.

e.g.

```
eng-30-09366317-n;n;positive;0.3125;natural_elevation,elevation;-1
```

```
eng-30-07961016-n;n;neutral;0.3125;clod,glob,ball,chunk,clump,lump;-1
```



---

## Language Resource Pool Management

---

### 3.1 The LRP Management Application API

It is possible to retrieve management information from the LRP such as listing a user's own services or the subscription details.

The basic endpoint is: <http://217.26.90.243:8080/EuroSentimentServices/services/server>

#### **GET /listAll**

Get a list of all the services in the platform.

##### **Example request:**

```
GET /listAll
Host: portal.eurosentiment.eu
Accept: application/json, text/javascript
```

##### **Example response:**

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/javascript

[
  {
    "credentials": "",
    "lastModification": "2014-06-30 10:24:47.0",
    "request_limit": 10000,
    "serviceMethod": "POST",
    "serviceUrl": "http://54.187.254.3:8000/language_detector",
    "sid": "sptdl0407",
    "state": "enabled",
    "url": "http://217.26.90.243:8080/EuroSentimentServices/services/server/access/sptdl0407"
  }
]
```

##### **Status Codes**

- 200 OK – no error
- 404 Not Found – there's no user

---

**Todo**

Describe the LRPMA API in detail

---

## 4.1 Frequently Asked Questions

1. **What is EUROSENTIMENT?** EUROSENTIMENT is a platform for the distribution of multilingual sentiment analysis resources available online for the first time.
2. **How does it work?** The EUROSENTIMENT platform works as a hub where different players can sell and buy sentiment analysis resources and services.
3. **What are the benefits in using EUROSENTIMENT?** EUROSENTIMENT is a comprehensive market to implement a fully featured service.
4. **Which kind of roles the platform support?** The platform supports Service Providers (anybody selling a service), Resource Providers (anybody selling a linguistic resource), Content Providers (anybody selling contents) and Consumers (anybody buying services, resources and contents).
5. **How EUROSENTIMENT can help me developing my business?** EUROSENTIMENT is the common place where demand and supply meet to provide the critical mass for boosting your activity.
6. **How shall I earn from EUROSENTIMENT?** As sentiment analysis solution implementer you can sell your services and also the resources they are based on.
7. **What about if I'm a researcher?** EUROSENTIMENT foreseen license and access facilitations for academic usage.
8. **Shall I benefit in using the EUROSENTIMENT resources for my research activity?** EUROSENTIMENT aims to distribute the best resources available on the market, providing to your research the state of the art tools to validate and extend your results.
9. **Which type of billing services the EUROSENTIMENT platform does support?** The accessed resources will be billed per download volume, the services for processing volume.
10. **Does the platform support an API?** The EUROSENTIMENT platform provides a fully featured REST based API.
11. **Which kind of domains are supported?** Currently we support the Electronics and Hotels domains.
12. **Which languages are supported?** Currently we support English, Spanish, Italian, French, Catalan and Portuguese.
13. **Shall I test the EUROSENTIMENT services?** The EUROSENTIMENT platform provides a demo area which allows to test all available resources.
14. **Which kind of license restrictions are supported/applied?** The licenses are defined by the resource owners and will follow their specific policies.

15. **Which kind of security standards does it offer?** The platform provides standard HTTPS certificate-based security verifications. Registered users are restricted to configurable group operational grants.
16. **What can I do as Service Provider?** As a Service Provider I can register my already implemented service to the platform, which will provide user accounting, billing and tracking features.
17. **What can I do as Language Resource Owner?** As Language Resource Owner I can upload my resources for public download previous payment and licence acceptance.
18. **What can I do as Content Provider?** As Content Provider I can upload my resources for public download previous payment and licence acceptance.

---

## Indices and tables

---

- *genindex*
- *modindex*
- *search*





---

## To-Do

---

---

### Todo

Describe the LRPMA API in detail

---

(The *original entry* is located in `/var/build/user_builds/eurosentiment/checkouts/master/lrpma/api.rst`, line 43.)



### **/listAll**

GET /listAll, [39](#)

### **/services**

GET /services/server/access/(service\_id),  
[29](#)

POST /services/server/access/(service\_id),  
[31](#)

### **/sparql**

POST /sparql, [33](#)