ePolicy

Engineering the Policy-making Life Cycle

Seventh Framework Programme – Grant Agreement 288147

Prototype of the Opinion Mining System version 2

Document type:	Deliverable
Dissemination Level:	Public
Editor:	Luis Torgo
Document Version:	1.0
Contributing Partners:	INESC PORTO
Contributing WPs:	WP6
Estimated P/M (if applicable):	n.a.
Date of Completion:	31 March 2014
Date of Delivery to EC	31 March 2014
Number of pages:	23

ABSTRACT

This document describes the second and final version of the prototype of the opinion mining system. It provides a brief description of : (i) the goals of this system; (ii) its main functionality requirements; (iii) the architecture of the proposed solution; (iv) some example use cases; (v) a more detailed description of the main components of the system; and (vi) how to install and use the prototype.

The ePolicy Consortium consists of the following partners: University of Bologna; University College Cork, National University of Ireland, Cork; University of Surrey; INESC Porto, Instituto de Engenharia de Sistemas e Computadores do Porto, Fraunhofer – Gesellschaft zur Foerderung der Angewandten Forschung E.V.; Regione Emila-Romagna; ASTER – Società Consortile per Azioni; Università degli Studi di Ferrara.

Authors of this document:

Luis Torgo[1], Pedro Coelho[2]

[1]: Luis Torgo
INESC Porto
email: ltorgo@dcc.fc.up.pt
[2]: Pedro Coelho
INESC Porto
email: pedro.s.coelho@inesc.pt

Contents

This page has been intentionally left blank.

1 Executive Summary

This deliverable describes the final version of the opinion mining (OM) prototype. This software component is part of the ePolicy decision support system and has as main goals to be able to infer the sentiment of the population concerning a set of pre-defined energy-related topics. The outcomes of this system can be explored individually by policy makers, and also be used as inputs to other components of the ePolicy decision support system. On top of this integration within the ePolicy decision support system, we have also targeted our work at being able to deliver a software prototype that could be used outside of the ePolicy domain. With this purpose we have developed the OM prototype to be able be executed as a stand alone software system that is able to address a specific opinion mining task :- infer the sentiment of the population regards a set of topics as expressed by posts on a set of web sites.

In this deliverable we describe the main features and functionalities of the OM prototype and present the architecture of the developed software. The outcomes of this tool can be explored in two different ways: (i) through the ePolicy decision support system web application interface; and (ii) through the graphical user interface we have developed for our stand alone system. This stand alone tool targets two different types of users: (i) standard users that want to explore the outcomes of the system concerning the inference of the sentiment regards a set of topics; and (ii) admin users that want to either tune / change the behaviour of the system or create new tasks / problems.

We describe the software requirements for the usage of the developed prototype. Moreover, we provide detailed information on how to install this software on a standard desktop computer and also on how to use it.

This deliverable is strongly related with Deliverable D 6.2 where we have described the first version of this software component. The content of the current document is also strongly related with Deliverable D 6.3 where we formalize the tasks being solved by the OM prototype and describe the different alternative solutions that were considered for these tasks taking into account the state of the art on opinion mining that was described in Deliverable D 6.1.

2 Introduction

The main goal of the opinion mining (OM) component is to provide information on the sentiment of the population concerning a set of relevant topics to both the global level optimizer and the individual level simulation. However, the outcome of the OM component can also be useful for user exploration. Namely, policy makers may find use in exploring the tendencies in terms of opinions of the population concerning a series of topics of interest for their job. This user-driven exploration of the sentiment of the population concerning a pre-defined set of topics is the other major goal of the OM component.

The opinion mining component uses sentiment analysis techniques from the field of text mining to infer the sentiment and opinion of the population on a set of pre-defined topics. This inference is carried out over textual sources (e.g. blog posts or other messages available in e-participation tools). For each textual source a sentiment score will be inferred by the OM component and these scores will then be aggregated (e.g. on a daily basis) to form an overall score of the population concerning a topic at a certain time. The exploration of these aggregated scores over time will allow to provide insights on the tendencies of these scores across a certain time horizon.

The OM software component has two main working modes: the *training mode* and the *standard usage mode*. In the standard usage mode this software component is carrying out the following sequence of operations:

- Crawl a *pre-defined set of e-participation sites* in search for new posts by the population;
- If new posts are found, fetch them into a local data base;
- For each new post use the previously learned models to infer the expressed sentiment concerning a set of *pre-defined topics*.

The end result of this working mode is thus continuously feeding a data base with new text documents and the corresponding sentiment scores concerning a set of pre-defined topics. At the same time, the standard usage mode will also determine on a regular basis (e.g. daily) what is the *aggregated score value for each of the target topics*. This aggregated score is a function of the individual scores that were inferred for each new text document that was fetched from the sites.

The main goal of the training or admin mode is to use text mining tools to learn models that are able to classify new text documents concerning the sentiment they express on a set of topics. These models are learned by example, i.e. using a set of textual documents that were pre-classified by a human expert on the domain, in a process usually known as supervised learning. Using this training data set sentiment classification models are learned, which can later be used in the standard usage mode. The training mode will be seldom used as it performs very specialized tasks that change the way the standard usage mode behaves. Typically it will be executed at the time the opinion mining problem is being created, or if new documents classified by a human expert become available.

We can identify several types of users interacting with the OM software component. Users with “administration” roles will be able to:

1. Create a new opinion mining problem / domain / task;
2. Change individual components of an existing problem

The first of these tasks means that the OM prototype can actually be used in other domains outside of the scope of the ePolicy project. This extended functionality was developed in response to a challenge by project reviewers. In this context, the OM prototype can be used to mine the sentiment expressed on e-participation tools concerning any set of selected topics. The second task involves changing or tuning the issues that characterize the opinion mining tasks we address with this software tool. These are:

- Change or tune the sentiment classification algorithms;
- Change the document web sources (the crawled e-participation tools);
- Change the set of topics for which the sentiment is to be inferred;
- Tag crawled documents for sentiment concerning the active set of topics

These tasks available for users with administration roles are very specific and require deep knowledge on the OM prototype.

Normal standard users, like policy makers, want to take advantage of the outcome of the OM component to better grasp the sentiment of the population concerning a set of topics related to energy policies. These users "consume" the outcome of the standard usage mode of the OM component. The main goal of these users is to have an answer to the general question: "What is the sentiment of the population concerning topic X?". This type of questions can take slightly different shapes like: i) "What is the *current* sentiment?" or ii) "What is the tendency in the last Y months (or other time frame) of this sentiment score?". This sort of functionality will be provided by an user interface to the OM component taking advantage of the autonomous sentiment inference that the system carries out for the new posts that appear in the pre-defined set of e-participation tools when working in standard usage mode.

Finally, the outcome of the OM component may also be used by other components of the ePolicy decision support system. In effect, other components of this system will use the sentiment of the population concerning different topics as inputs of their own functionalities.

3 Main Functional Requirements of the OM Component

The main outcome of the opinion mining (OM) component are values of sentiment scores for a pre-defined set of topics. These scores are obtained through time with a certain pre-defined regularity, i.e. they are in effect a time series. The scores are derived from the sentiment scores assigned to each individual post that appears in the e-participation sites. The OM component keeps feeding a data base with these aggregated sentiment scores. This information can then be used by other software components of the ePolicy decision support system, like the global optimizer or the individual level simulator. At the same time these scores can also be provided to the user by means of carefully selected data visualization techniques with the goal of helping policy makers to understand the sentiment of the population regarding a set of relevant topics.

In this context, the main functional requirements of the OM component are:

1. to be able to classify new documents that appear in a pre-defined set of forums of e-participation, concerning the expressed sentiment on another pre-defined set of topics;
2. to aggregate the sentiment predicted for the new documents into a single aggregated score with a certain regularity (e.g. daily);

3. to accept as input a set of topics and a set of e-participation sites, which are to be used to monitor the sentiment of the population.

These main functionalities are to be used in different ways. Namely, we distinguish three main classes of "agents" that may interact with the opinion mining component:

1. other software components of the ePolicy decision support system;
2. a "normal" user (e.g. policy maker);
3. a user with administration privileges.

The interaction with other software components is carried out through a data base management system. Namely, the OM component will store the different sentiment scores in a data base and this information can be accessed by other components according to their needs. In particular, both the global optimizer and the individual level simulator will use the predicted sentiment scores as inputs in their tasks.

The interaction with a normal user is to be provided by a graphical user-interface. The main use case we envisage is a policy maker wishing to know what is the sentiment of the population concerning a certain topic of interest to her/his activity. The OM component should be able to provide this information in a useful manner to the user.

Finally, the interaction with an user with administration privileges should be carried out outside of the ePolicy decision support system as this is a type of interaction that is very specific to the OM component and requires knowledge of the inner workings of this software. With this goal we have also developed a specific stand-alone user interface to the OM prototype that is independent of the ePolicy decision support system, and allows users with administration privileges to perform some specific tasks that can change the behaviour of the prototype.

3.1 Other Requirements of the Opinion Mining Component

The opinion mining (OM) software component should be implemented using free software to facilitate its deployment in different economic settings. The development of this software requires a programming environment for text mining that allows the inference of the sentiment on textual documents. Moreover, the OM component also requires a data base management system for storing the information on the documents and the inferred sentiments.

The development of the OM component should be carried out with the aim of making its adaptation to other domains and/or regional contexts, a relatively easy task. This means that it should be able to function independently of the ePolicy decision support system.

The key inputs of the OM component are: i) the set of topics of interest for which we want to infer the sentiment of the population, and ii) the set of e-participation sites/tools that are to be crawled by the system in search of new expressed opinions by the population. Changing these two input parameters should be made as easy as possible by the OM component. Still, one should be aware that there are technical requirements that must be present for this to be an easy task. Namely, to enable automatic crawling by the OM component, new sites should be able to provide information on new posts using RSS feeds. Without this, crawling these sites will require the adaptation of the OM software and thus will hardly be regarded as an easy task. Moreover, increasing the set of topics of interest also brings technical challenges.

Namely, new topics require new sentiment classification models. For these to be obtained we need to provide the OM component with a set of past documents/posts that are pre-labelled regarding their sentiment for these new topics by some human expert. This requires specialized human intervention and it is a time-consuming task as the larger the set of documents the better.

4 Design Decisions for the Implementation of the OM Prototype

In Deliverable D6.1 [2] we have presented a state of the art on opinion mining. As seen in that deliverable that are plenty of options available for trying to mine text documents for sentiment or opinions. The goals of the ePolicy project and the role of the OM prototype within the project decision support system (as specified by the functional requirements described in Section 3), imposed a series of constraints in terms of what should be implemented in the OM prototype. Namely, our system is heavily constrained by the requirement that we should be able to infer the sentiment for a set of *pre-defined topics*. This means that all existing text mining approaches that try to infer the topics present in text documents (frequently known as topic discovery) are out of the scope of this work - our set of target topics is established by the users of the system. This means that from a text mining perspective the goals of our prototype given a new text document and the pre-defined set of target topics, are:

1. Infer whether each of the topics is addressed or not in the document;
2. For each of the mentioned topics, infer the sentiment of the author within a given sentiment scale.

As mentioned in Section 2 of Deliverable D6.3 [1], these two goals translate in two types of tasks: (i) topic identification; and (ii) sentiment scoring. Sections 2 and 3 of Deliverable 6.3 formalize these tasks and describe how we have decided to implement them, given the set of options available in the current state of the art.

5 The Architecture of the Proposed OM Software Component

The functional requirements presented in Section 3 involve tasks that have different time granularity and specific needs. For instance, crawling a set of e-participation sites is a task that has a fast pace, while learning new sentiment classification models will be done on rare occasions.

These facts have lead us to develop a solution for the OM software component that is formed by a set of modules and a database that is used to communicate information between each module or other software components of the ePolicy decision support system.

Figure 1 shows a UML class diagram with an overall perspective of the Opinion Mining (OM) software component. This component is formed by 4 main modules: i) the user interface (*GUI* in the figure); ii) the module responsible for crawling the pre-defined set of sites for new posts of the population (*Crawler* in the figure); iii) the predictor module that uses the learned models to assign a sentiment score to each newly found post (*Predictor* in the figure); and iv) the module responsible for learning the opinion mining models for each of the pre-defined topics (*Learner* in the figure). All these modules use a data base to store and obtain information that is necessary to their tasks. The outcomes produced by these modules are also stored in the data base, allowing other components of the ePolicy decision support system to use these results for their own tasks.

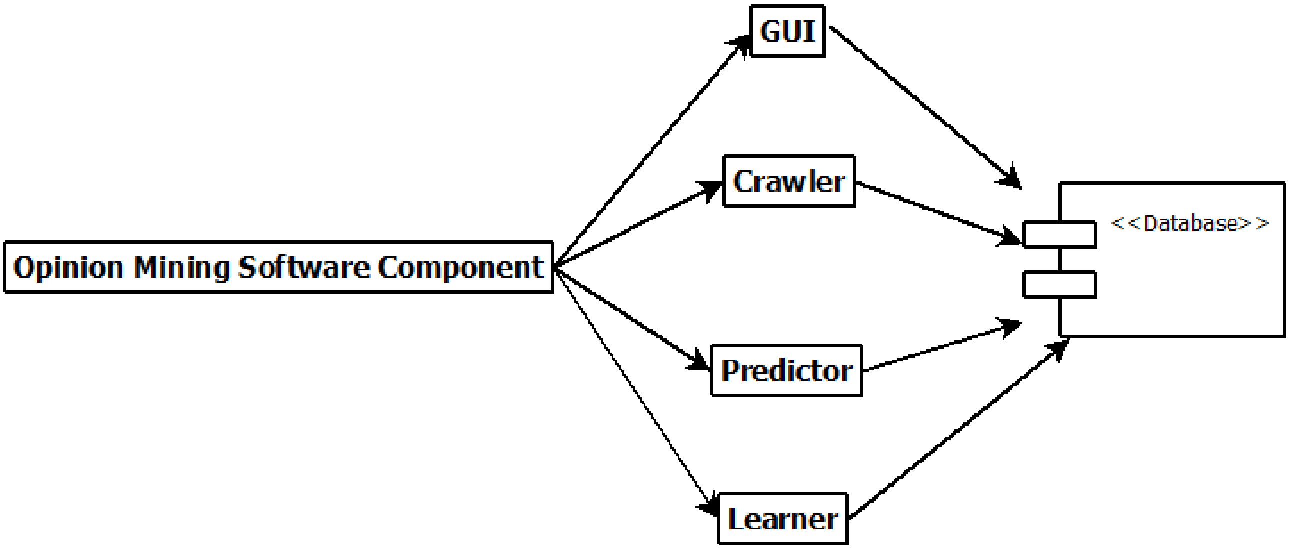


Figure 1: The overall picture of the Opinion Mining Software Component.

As mentioned in Section 3 the OM component interacts with different types of users. Figure 2 provides an illustration of the typical use cases of the two types of users: i) the policy maker and ii) the administrator of the component. Essentially, policy makers will use the system to visualize some form of synthesis of the sentiment of the population concerning a certain topic, whilst the administrator users will essentially tune the system to change its behavior, either by tuning the existing models or by extending them by including other topics of interest or by adding new sources of e-participation.

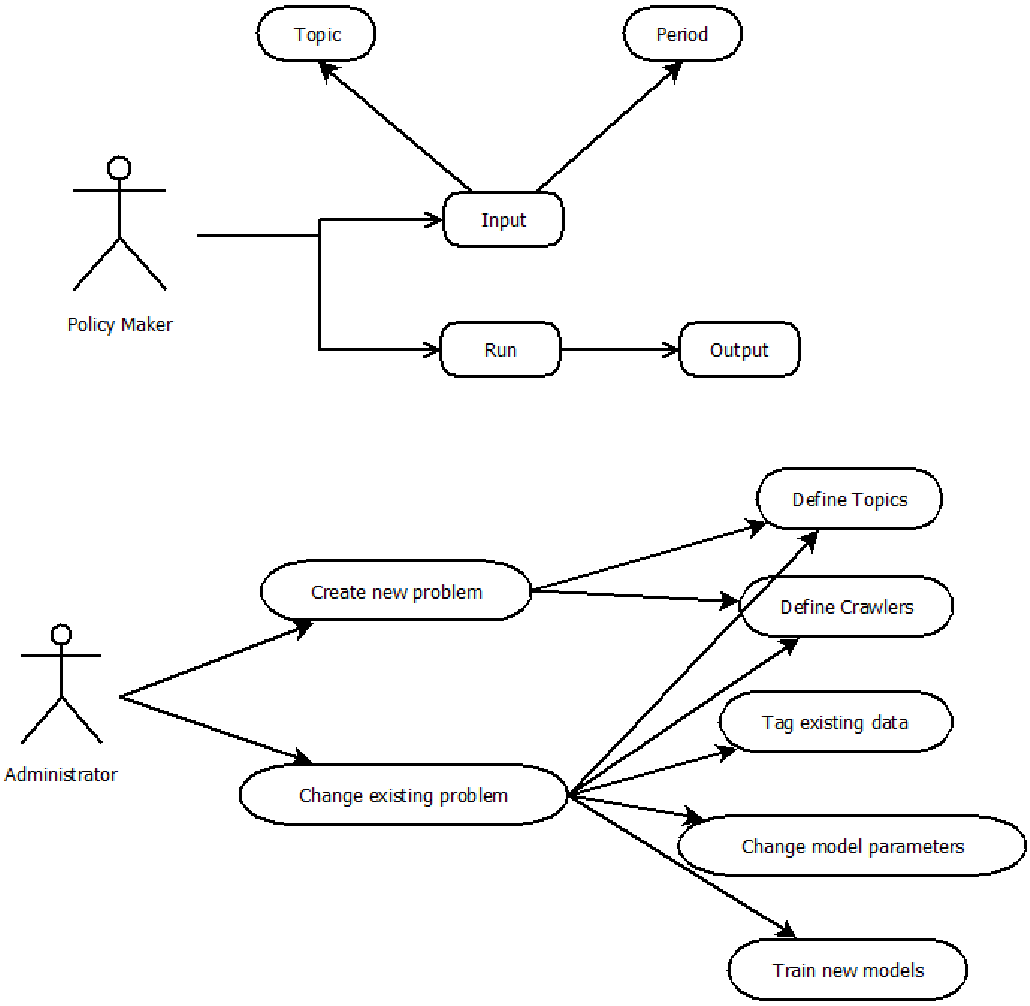


Figure 2: The different types of use cases of the OM component.

Finally, in Figure 3 we illustrate the typical interaction of the policy maker with the OM component using its graphical user interface. This consists in carrying out the operation of visualization of the sentiment on a certain topic (either the current sentiment or the tendency of this sentiment over a time span), and getting as result some form of visualization of this information derived by the OM component.

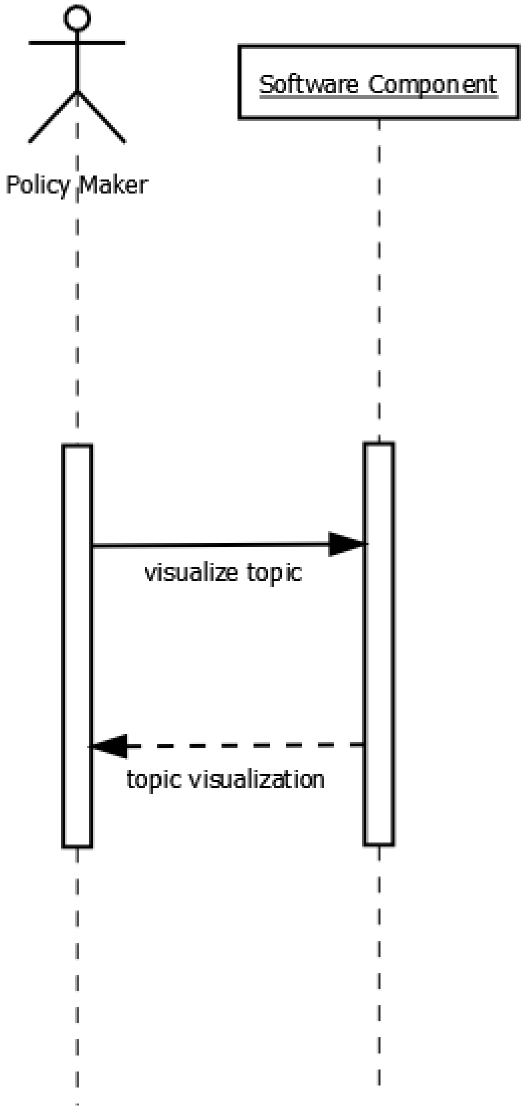


Figure 3: A sequence diagram showing the typical interaction of a policy maker with the OM component.

6 Technical Description of the Opinion Mining Prototype

The prototype of the OM software component is implemented using a set of modules that communicate with each other and pass information through a data base management system. As mentioned before this is justifiable by the rather different types of tasks we have in this prototype, each with different time frames and eventually involving different types of users and interactions. This type of architecture also brings several advantages in terms of maintaining the software and even allows for different modules to be executed on different computers, which may be advantageous given the different computational requirements that each module has.

6.1 Crawler

The crawler module is responsible for launching processes that crawl each selected web site. Since each website has its own structure, different individual crawlers have been developed. This also means that it is not easy to automate the addition of new web sites as it may require specific individual settings. In this context, changing the settings of the crawler (e.g. by adding new web sites) is a task for users with administration role and it may require the development of specific software components that are able to crawl the new sites that may have some particularities that preclude the use of some general crawling component.

6.2 Predictor

This module handles the sentiment inference that is carried out over new unlabelled posts. It can be done on a pre-scheduled basis (for example, classify the daily posts at the end of the day). The predictor uses the current best opinion mining models for each of the topics being considered, applying them to new posts to get a sentiment score for each of the topics. These predicted sentiment scores are then stored in the data base for use by other components of the e-Policy decision support system.

6.3 Learner

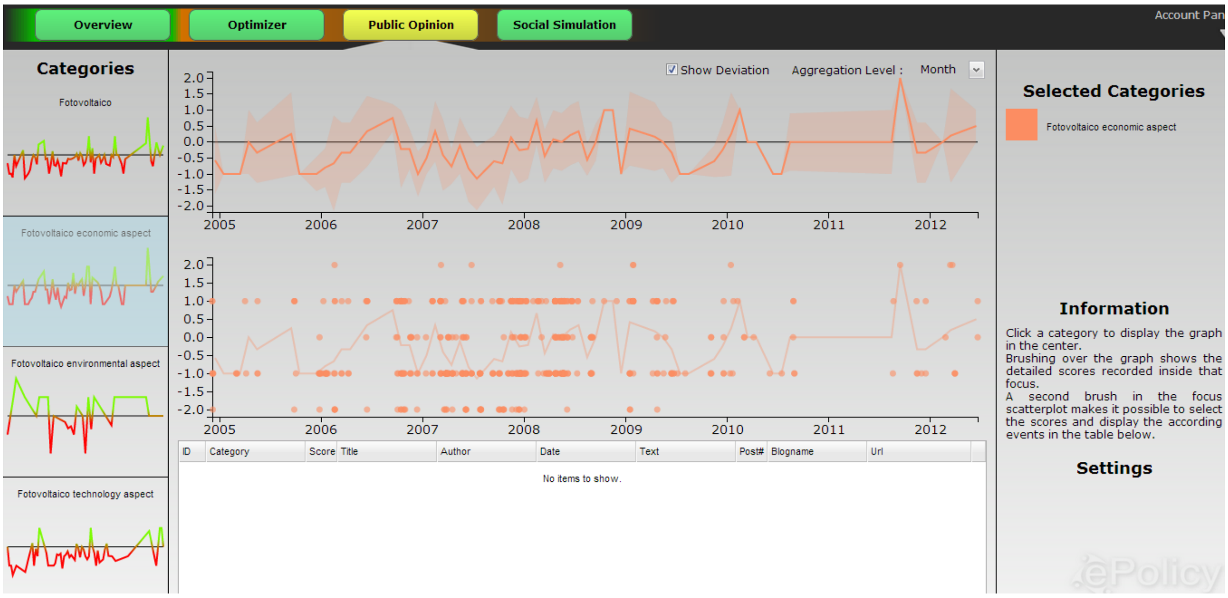
The learner module is responsible for obtaining new opinion mining models when new documents tagged by a human expert on the domain become available. The arrival of new tagged data is very rare due to the high costs in terms of human resources to manually label posts. In this context, the execution of this module is typically triggered manually by an administrator user.

6.4 Graphical User Interface (GUI)

With the goal of allowing the usage of the OM prototype in context outside of the ePolicy project, we have decided to provide two types of graphical user interfaces. The first, is a component of the user interface to the ePolicy decision support system that was developed by Fraunhofer. This GUI allows a policy maker to obtain information concerning the sentiment of the population towards the set of selected energy-related topics. The second is provided by a stand alone graphical user interface that we have develop that allows the interaction of two types of users (admins and normal users) for the available set of problems / domains. This second GUI is completely independent of the ePolicy project goals and can be seen as a general tool for inferring the sentiment concerning a series of topics as expressed on some selected web sites.

6.4.1 ePolicy Decision Support System GUI

Fraunhofer's GUI (Figure 4) allows the user to interact with the different components of ePolicy decision support system. This GUI includes a tab for exploring the public opinion concerning a series of energy-related topics. In this section of this GUI a user can select the topic to inspect as well as the aggregation level (e.g. monthly, weekly, etc). This interface also allows the user to drill down to individual posts (Figure 5) as the user may be interested in knowing more about some particular posts (e.g read the post).

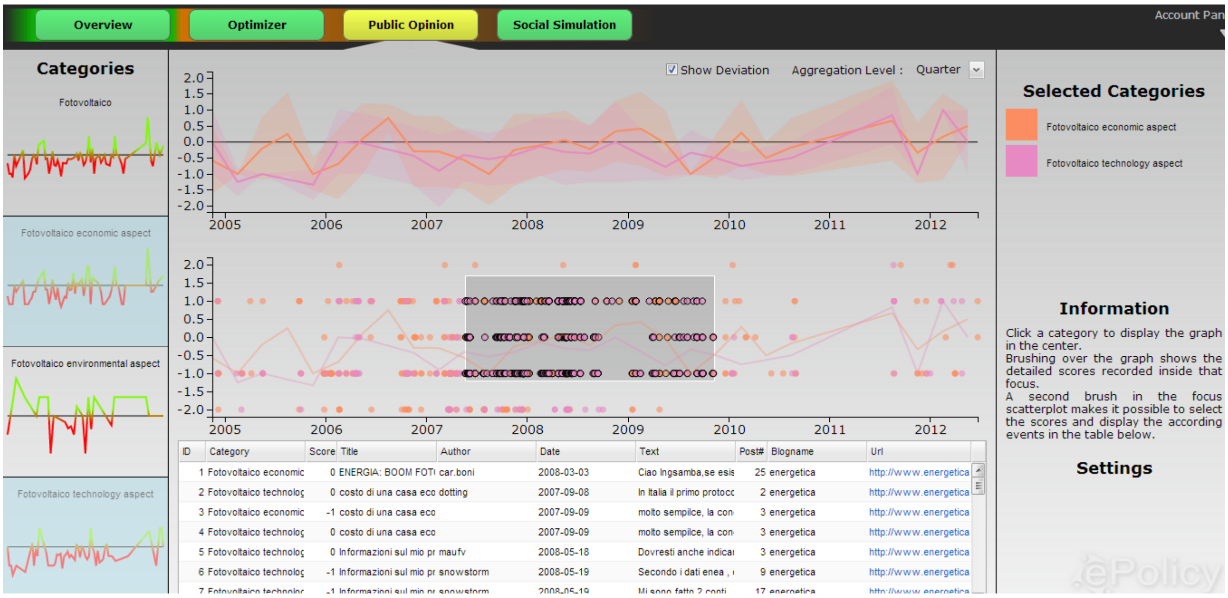


Click a category to display the graph in the center.
Brushing over the graph shows the detailed scores recorded inside that focus.
A second brush in the focus scatterplot makes it possible to select the scores and display the according events in the table below.

ID	Category	Score	Title	Author	Date	Text	Post#	Blogname	Url

No items to show.

Figure 4: ePolicy decision support system - exploring public opinion.



Click a category to display the graph in the center.
Brushing over the graph shows the detailed scores recorded inside that focus.
A second brush in the focus scatterplot makes it possible to select the scores and display the according events in the table below.

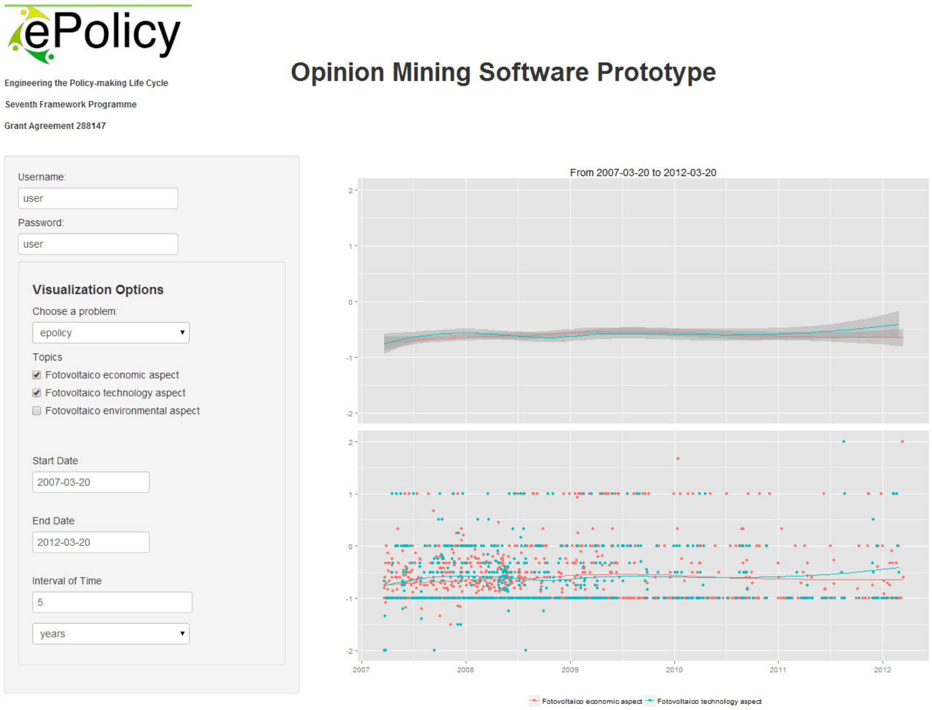
ID	Category	Score	Title	Author	Date	Text	Post#	Blogname	Url
1	Fotovoltaico economic	0	ENERGIA: BOOM FOTI	car.boni	2008-03-03	Ciao Ingsamba,se esis	25	energetica	http://www.energetica
2	Fotovoltaico technolog	0	costo di una casa eco	dotting	2007-09-08	In Italia il primo protocc	2	energetica	http://www.energetica
3	Fotovoltaico economic	-1	costo di una casa eco		2007-09-09	molto sempilce, la con	3	energetica	http://www.energetica
4	Fotovoltaico technolog	0	costo di una casa eco		2007-09-09	molto sempilce, la con	3	energetica	http://www.energetica
5	Fotovoltaico technolog	0	Informazioni sul mio pr	maufv	2008-05-18	Dovresti anche indicai	3	energetica	http://www.energetica
6	Fotovoltaico technolog	-1	Informazioni sul mio pr	snowstorm	2008-05-19	Secondo i dati enea ,	9	energetica	http://www.energetica
7	Fotovoltaico technolog	-1	Informazioni sul mio pr	snowstorm	2008-05-19	Mi sono fatto 2 conti	17	energetica	http://www.energetica

Figure 5: ePolicy decision support system - drilling down to specific posts.

6.4.2 Stand Alone OM Prototype GUI

The stand alone prototype includes a user interface that allows two types of users to login (admins and normal users), and provides a different set of functionalities for each of these types of users.

The interface for standard users allows them to select the problem / domain they wish to explore, and then for each domain it provides means for the user to select the topics she/he wants to visualize, as well as the time span to consider in this exploration. After this selection, the system draws a plot of the data in which a user can see the a line with the aggregated sentiment score along time together with a confidence band around the line reflecting the variability in this aggregated score. A second plot is also drawn with the individual sentiment scores assigned to each post, which lead to the aggregated sentiment expressed by the mentioned lines. Figure 6) shows an illustration of this part of the GUI.



ePolicy

Engineering the Policy-making Life Cycle

Seventh Framework Programme

Grant Agreement 288147

Opinion Mining Software Prototype

Username: user

Password: user

Visualization Options

Choose a problem: epolicy

Topics

☑ Fotovoltaico economic aspect

☑ Fotovoltaico technology aspect

☐ Fotovoltaico environmental aspect

Start Date: 2007-03-20

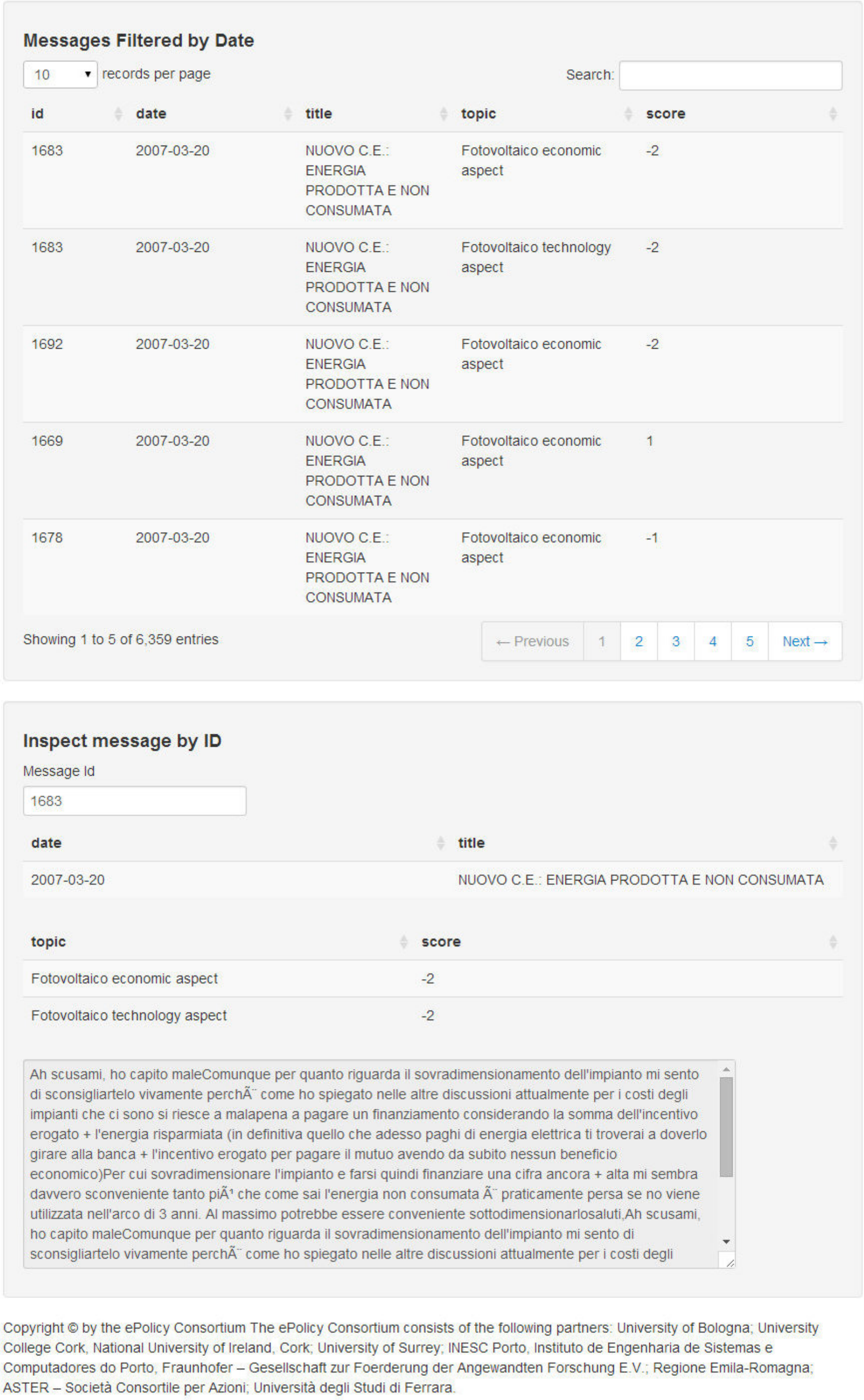
End Date: 2012-03-20

Interval of Time: 5 years

From 2007-03-20 to 2012-03-20

Figure 6: Standard user GUI - exploring the aggregated sentiment of topics.

Standard users can also drill down to individual posts. A table is presented with all posts within the selected time span (Figure 7), with a search box that allows easy filtering of these posts. The user may also select and individual post (through its ID) to obtain the specific text of this post together with the assigned sentiment scores for each topic.



Messages Filtered by Date

id	date	title	topic	score
1683	2007-03-20	NUOVO C.E.: ENERGIA PRODOTTA E NON CONSUMATA	Fotovoltaico economic aspect	-2
1683	2007-03-20	NUOVO C.E.: ENERGIA PRODOTTA E NON CONSUMATA	Fotovoltaico technology aspect	-2
1692	2007-03-20	NUOVO C.E.: ENERGIA PRODOTTA E NON CONSUMATA	Fotovoltaico economic aspect	-2
1669	2007-03-20	NUOVO C.E.: ENERGIA PRODOTTA E NON CONSUMATA	Fotovoltaico economic aspect	1
1678	2007-03-20	NUOVO C.E.: ENERGIA PRODOTTA E NON CONSUMATA	Fotovoltaico economic aspect	-1

Showing 1 to 5 of 6,359 entries

Inspect message by ID

Message Id: 1683

date	title
2007-03-20	NUOVO C.E.: ENERGIA PRODOTTA E NON CONSUMATA

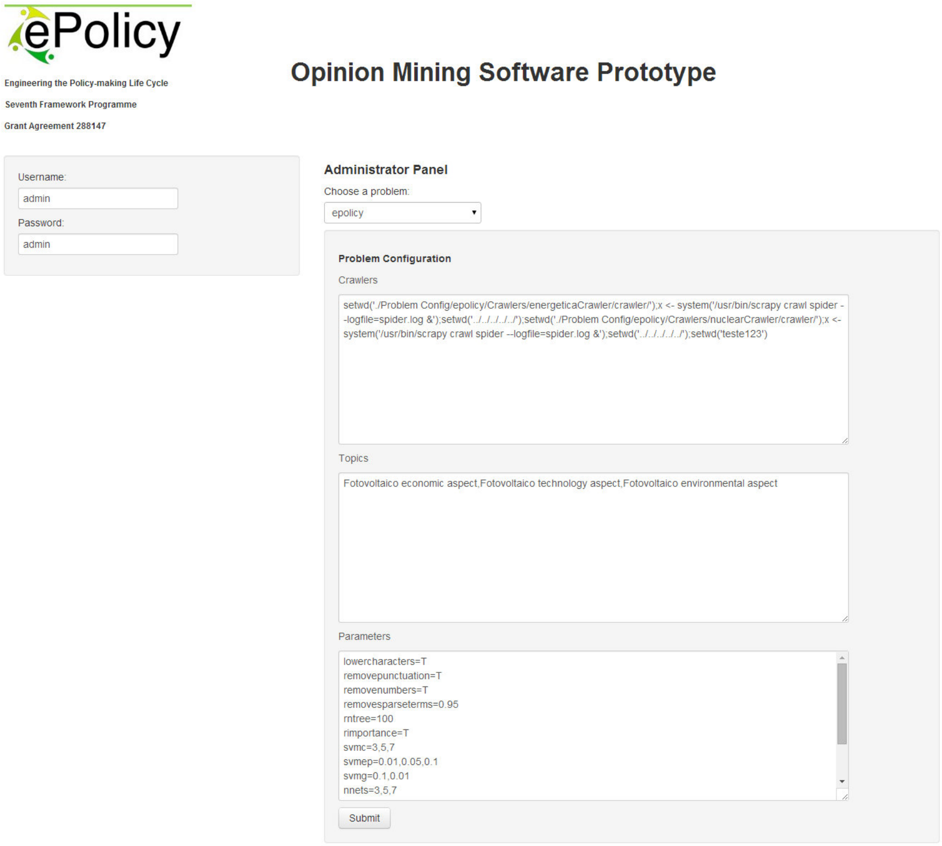
topic	score
Fotovoltaico economic aspect	-2
Fotovoltaico technology aspect	-2

Ah scusami, ho capito maleComunque per quanto riguarda il sovradimensionamento dell'impianto mi sento di sconsigliartelo vivamente perchÃ¨ come ho spiegato nelle altre discussioni attualmente per i costi degli impianti che ci sono si riesce a malapena a pagare un finanziamento considerando la somma dell'incentivo erogato + l'energia risparmiata (in definitiva quello che adesso paghi di energia elettrica ti troverai a doverlo girare alla banca + l'incentivo erogato per pagare il mutuo avendo da subito nessun beneficio economico)Per cui sovradimensionare l'impianto e farsi quindi finanziare una cifra ancora + alta mi sembra davvero sconveniente tanto piÃ¹ che come sai l'energia non consumata Ã¨ praticamente persa se no viene utilizzata nell'arco di 3 anni. Al massimo potrebbe essere conveniente sottodimensionarlosaluti,Ah scusami, ho capito maleComunque per quanto riguarda il sovradimensionamento dell'impianto mi sento di sconsigliartelo vivamente perchÃ¨ come ho spiegato nelle altre discussioni attualmente per i costi degli

Copyright © by the ePolicy Consortium The ePolicy Consortium consists of the following partners: University of Bologna; University College Cork, National University of Ireland, Cork; University of Surrey; INESC Porto, Instituto de Engenharia de Sistemas e Computadores do Porto, Fraunhofer – Gesellschaft zur Foerderung der Angewandten Forschung E.V.; Regione Emila-Romagna; ASTER – Società Consortile per Azioni; Università degli Studi di Ferrara.

Figure 7: Standard user GUI - exploring the individual posts.

The stand alone OM prototype also provides and administrator graphical user interface (Figure 8).



ePolicy

Engineering the Policy-making Life Cycle

Seventh Framework Programme

Grant Agreement 288147

Opinion Mining Software Prototype

Username: admin

Password: admin

Administrator Panel

Choose a problem: epolicy

Problem Configuration

Crawlers

```
setwd('./Problem Config/epolicy/Crawlers/energeticaCrawler/crawler/');x <- system('/usr/bin/scrapy crawl spider -
-logfile=spider.log &');setwd('../../../../../');setwd('./Problem Config/epolicy/Crawlers/nuclearCrawler/crawler/');x <-
system('/usr/bin/scrapy crawl spider --logfile=spider.log &');setwd('../../../../../');setwd('teste123')
```

Topics

Fotovoltaico economic aspect,Fotovoltaico technology aspect,Fotovoltaico environmental aspect

Parameters

```
lowercharacters=T
removepunctuation=T
removenumbers=T
removesparseterms=0.95
rntree=100
rimportance=T
svmc=3,5,7
svmep=0.01,0.05,0.1
svmg=0.1,0.01
nnets=3,5,7
```

Figure 8: Administrator GUI.

In the first section of this interface admins can select the crawlers (providing a command which will be run by the system), select the topics and tune the parameters of the opinion mining models that are used by our prototype. Currently, it is possible to tune the following parameters of these models:

- lowercharacters: Possible values are T or F. This defines if in the pre-processing of the text documents all characters should be lowered.
- removepunctuation: Possible values are T or F. This defines if in the pre-processing of the text documents all punctuation should be removed.
- removenumbers: Possible values are T or F. This defines if in the pre-processing of the text documents all numbers should be removed.
- removesparseterms: Value between 0.01 and 1.00. This defines the value that should be used when removing sparse terms on the pre-processing of the text documents.
- rntree: Controls the number of trees to grow in the approaches that use random forests. The value should be a positive integer.
- svmc: Sets the value associated with the cost of constraints violation in support vector machines, it is the ’C’-constant of the regularization term in the Lagrange formulation. The value should be a positive integer.
- svmep: Controls the epsilon in the insensitive-loss function of support vector machines. The value should be a positive integer.
- svmg: Parameter used in the kernel of support vector machines. The value should be a positive integer.

- nnets: Controls the number of units in the hidden layer of neural networks. The value should be a positive integer. The value should be a positive integer.
- nnetd: Controls the weight decay in neural networks. The value should be a positive integer.

A second section of the admin GUI allows these users to train new opinion mining models by clicking on a button and also to inspect and tag new posts (Figure 9). For this latter task we provide a table where the available posts can be filtered by ID, date and title. Using these filtering facilities the user may drill down to a specific post and eventually tag it for sentiment concerning the available topics.

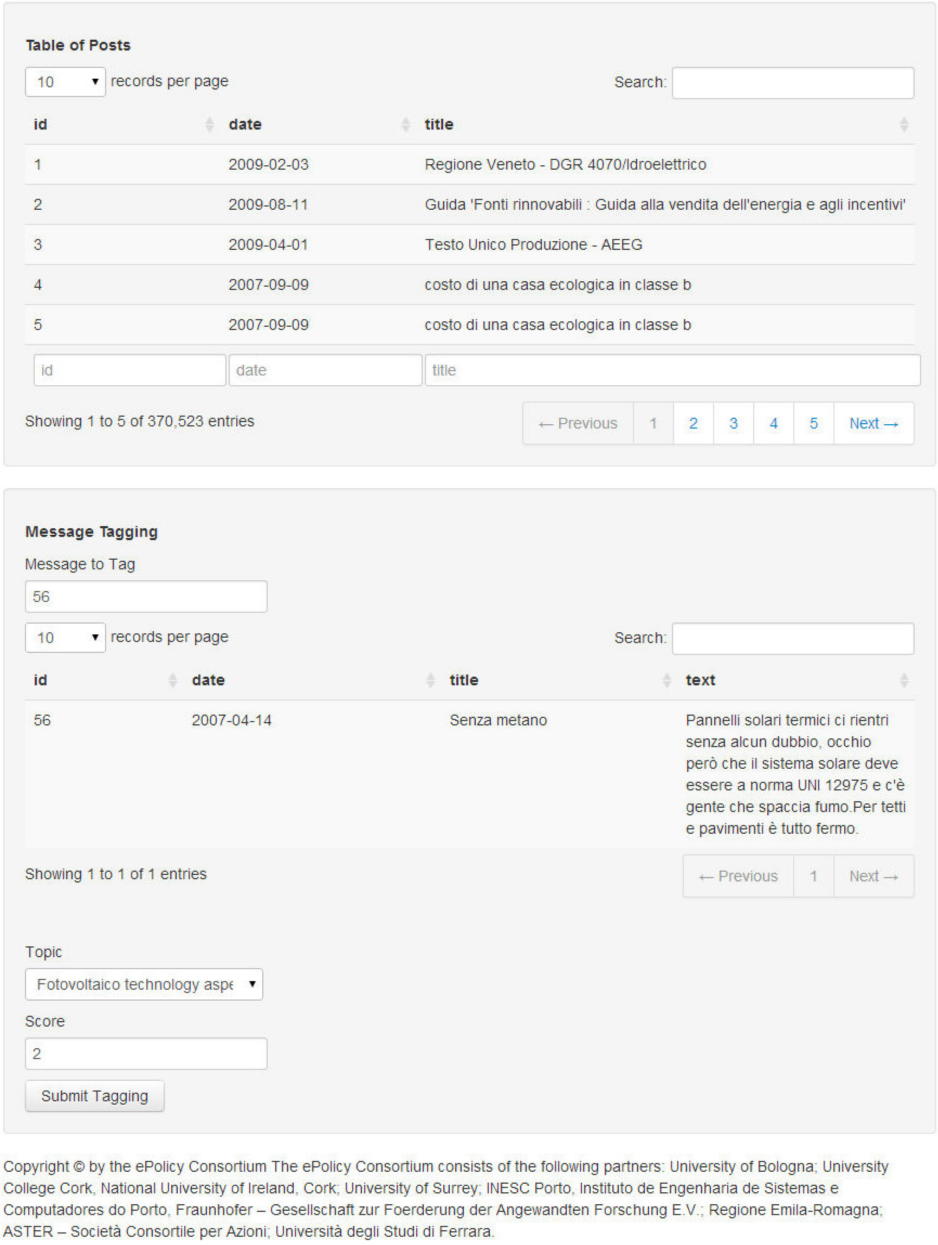


Table of Posts

10 records per page

Search:

id	date	title
1	2009-02-03	Regione Veneto - DGR 4070/Idroelettrico
2	2009-08-11	Guida 'Fonti rinnovabili : Guida alla vendita dell'energia e agli incentivi'
3	2009-04-01	Testo Unico Produzione - AEEG
4	2007-09-09	costo di una casa ecologica in classe b
5	2007-09-09	costo di una casa ecologica in classe b

Showing 1 to 5 of 370,523 entries

Message Tagging

Message to Tag: 56

10 records per page

Search:

id	date	title	text
56	2007-04-14	Senza metano	Pannelli solari termici ci rientri senza alcun dubbio, occhio però che il sistema solare deve essere a norma UNI 12975 e c'è gente che spaccia fumo.Per tetti e pavimenti è tutto fermo.

Showing 1 to 1 of 1 entries

Topic: Fotovoltaico technology aspe

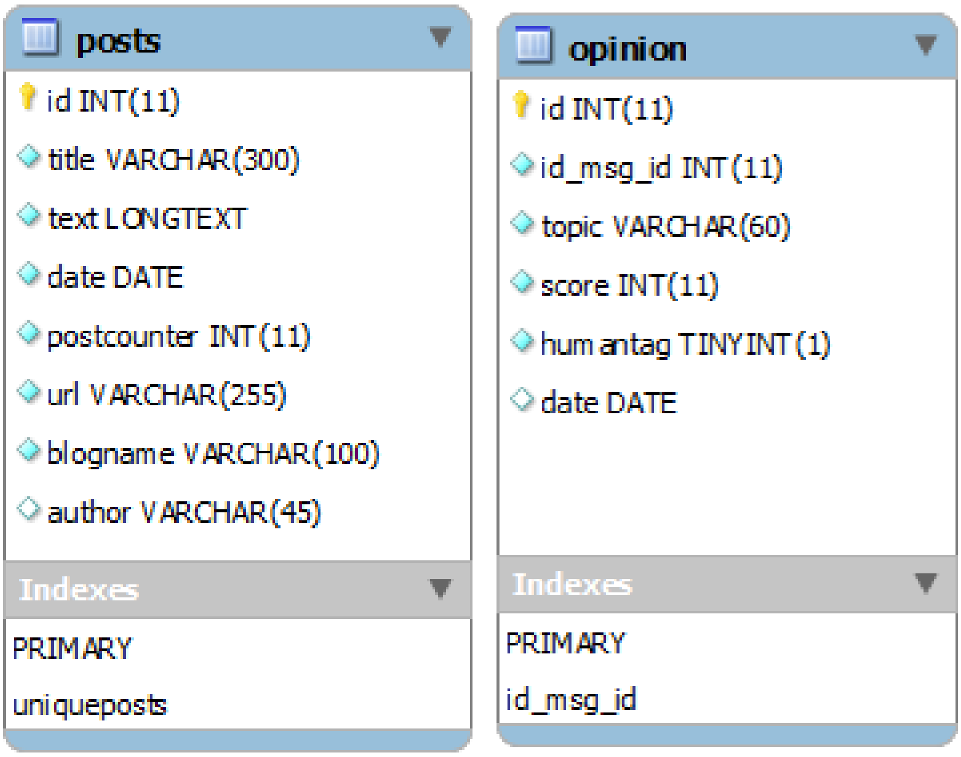
Score: 2

Copyright © by the ePolicy Consortium The ePolicy Consortium consists of the following partners: University of Bologna; University College Cork, National University of Ireland, Cork; University of Surrey; INESC Porto, Instituto de Engenharia de Sistemas e Computadores do Porto, Fraunhofer – Gesellschaft zur Foerderung der Angewandten Forschung E.V.; Regione Emila-Romagna; ASTER – Società Consortile per Azioni; Università degli Studi di Ferrara.

Figure 9: Administrator GUI - tagging posts.

6.5 The Database

The OM prototype uses one database with two tables for each problem / task. These tables store all relevant information obtained by the prototype and that may be used by other software components or by the GUI's to present it to the users. Figure 10 describes these tables.



posts
id INT(11)
title VARCHAR(300)
text LONGTEXT
date DATE
postcounter INT(11)
url VARCHAR(255)
blogname VARCHAR(100)
author VARCHAR(45)
Indexes
PRIMARY
uniqueposts

opinion
id INT(11)
id_msg_id INT(11)
topic VARCHAR(60)
score INT(11)
humantag TINYINT(1)
date DATE
Indexes
PRIMARY
id_msg_id

Figure 10: The used data base tables.

The *posts* table contains the original posts found at the crawled web sites. This table is filled in by the different crawlers launched by the Crawler agent.

The *opinion* table contains the sentiment scores concerning each topic for each post. This table contains two types of scores (determined by the value of the *humantag* field): (i) scores assigned by a human expert; and (ii) scores assigned by our opinion mining models. The former are crucial for the Learner agent as they will form the training data used to learn the opinion mining models. The latter are the result of the application of these models to newly crawled posts.

7 Using the OM Software Prototype

As we have mentioned before there are essentially two ways of using the OM prototype: (i) through the interface of the ePolicy decision support systems; or (ii) as a stand alone system. The former usage is the preferable setting for policy makers and includes not only access to the outcomes of our opinion mining component, but also all other ePolicy components. The use of this interface is through a web service and thus does not require the installation of any software at the users' computer. The access to the ePolicy decision support system web interface is through the following URL: `http://epolicy.igd.fraunhofer.de/epo/`

Accessing our prototype as a stand alone system requires the installation of the software at the users' computer. The following sub-sections provide information on this installation process.

7.1 Installing the Stand Alone OM Prototype

The OM software prototype is based on 4 main software tools, all of which are freely available for the most common desktop platforms: (i) the MySQL data base management system; (ii) the R programming language; (iii) the Python programming language; and (iv) Python Scrapy (this last part is required for the crawlers developed for the e-policy's specific problem).

7.1.1 Software Requirements

To run all parts of the current version of the OM software prototype you should make sure you have installed and running properly, the following software:

- MySQL Server.
  Downloadable at `http://dev.mysql.com/downloads/`
- R Platform.
  Downloadable at `http://www.r-project.org/`
- The following R packages must be installed[1]:
  - plyr
  - zoo
  - tm
  - Snowball
  - RWeka
  - rJava
  - RWekajars
  - DBI
  - RMySQL
  - gWidgets
  - ggplot2
  - shiny
  - DMwR
  - performanceEstimation
  - randomForest
  - e1071
  - nnet
  - xts
  - gridExtra

[1] To install an R package you should run R and execute the following command: `install.packages("packageName")`

- Python 2.
  Downloadable at `http://www.python.org/download/releases/2.7.5/`
- Python Scrapy.
  Downloadable at `http://scrapy.org/download/`

7.1.2 Necessary Data and Code

The code of the software prototype is available as a ZIP file at the following URL:
`http://www.dcc.fc.up.pt/~ltorgo/ePolicy/OMprototype/`

The ZIP file should be downloaded and un-zipped on some local folder.

At the same URL you will also find an SQL file that will create and populate the MySQL data base necessary to run the prototype. This SQL file should be executed in your own MySQL installation. It will create a data base named *epolicy* and the necessary tables, that will also be populated with the current data we have. Once the data base and respective tables are created you should create a user on your MySQL server with access to this data base. The credentials of this user (username and password) should be updated in the file *db.cfg* that contains the data base access configuration information and that should be in the folder where you un-zipped the code of the prototype.

7.2 Using the Stand Alone OM Prototype

Once all the installation steps are successfully carried out, to run the stand alone prototype it is enough to:

- Run R in the folder where the prototype code is.
- In R, run the command : `library(shiny);runApp('.')`

7.2.1 Exploring the Documents and Sentiment of a Problem / Domain

In order to visualize a problem, you must first write the username and password in the input boxes on the side panel. After this you can select the problem that you want to analyse from the available drop box, and check the topics you are interested in visualizing. The interface also allows you to specify a time span for your analysis.

7.2.2 Creating and Editing a Problem

In order to create or edit a problem, you must first write the username and password in the input boxes on the side panel. After this, a main panel with various options will be displayed as described before. Here if you are creating a new problem, you can define the problem's name, the possible topics and the crawlers command (the administrator must provide the crawlers to retrieve posts from web sources). Otherwise, if you are interested in changing a problem you will be able to change the topics and crawlers, tune the model parameters and inspect the text documents as well as tag them in terms of sentiment.

This page has been intentionally left blank.

References

[1] L. Torgo and P. Coelho. Opinion mining prototype evaluation, version 1. Technical Report D6.3, ePolicy research project consortium, 2013.

[2] L. Torgo, P. Coelho, V. Costa, D. Sangiorgi, and S. Franceschini. State of the art on opinion mining for policy evaluation and tools of e-participation. Technical Report D6.1, ePolicy research project consortium, 2013.