Engineering the Policy-making Life Cycle

Seventh Framework Programme – Grant Agreement 288147

Prototype of the Opinion Mining System version 1

Document type:	Internal Report
Dissemination Level:	Private
Editor:	Luis Torgo
Document Version:	1.0
Contributing Partners:	INESC PORTO
Contributing WPs:	WP6
Estimated P/M (if applicable):	n.a.
Date of Completion:	26 July 2013
Date of Delivery to EC	26 July 2013
Number of pages:	20

ABSTRACT

This document describes the first 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.

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 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: i) crawl a pre-defined set of e-participation sites in search for new posts by the population; and ii) if new posts are found, fetch them and infer their sentiment, storing the resulting sentiment scores in a local data base. 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.

We can identify several types of users interacting with the OM software component. Users with "administration" roles will be typically involved in the training mode of the system. These users will be able to tune, improve or create new sentiment classification models. Moreover, these specialized users will be allowed to include new textual sources into the set that is crawled by the OM component to check if new posts are available. Although such task (adding a new web source) could be in principle carried out by a "normal", non-technical user, the fact that this may involve several complex issues caused by different formats of the sites, precludes us from providing this as a general normal-user functionality. Another func-

tionality available to users with "administration" privileges is to augment the set of topics of interest for opinion mining. Once again this apparently simple task involves complex operations because adding a new topic involves learning new models, which in turn requires pre-labeled posts that will serve as a model training set. All these operations are not feasible in the time frame of a standard user, like a policy maker, that wants instant feedback from the system and will not be willing to wait for these complex operations to finish (for instance collecting a new training set for a new topic will involve the work of a human expert to label a large set of past posts - a time consuming task). In summary, administrators will interact with the OM component in a completely different manner than standard users and are required to be fully aware of the inner workings of this software component.

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.

2 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 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. This type of interaction will typically be carried out through configuration files that can change the behavior of the OM component. The main use cases for this particular users include:

1. tuning and changing the current sentiment classification models;
2. increasing the set of topics of interest;
3. increase the set of e-participation sites to be monitored.

2.1 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.

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.

3 The Architecture of the Proposed OM Software Component

The functional requirements presented in Section 2 involve tasks that have different time granularities 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 software agents that communicate with each other when necessary. For instance, when the crawling agent finds and downloads a new post from some site, it will send a message to the agent responsible for sentiment classification informing this agent that a new text document is available for being classified.

Figure 1 shows a UML class diagram with an overall perspective of the Opinion Mining (OM) software component. This component is formed by 5 main modules/agents: 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 classifier module that uses the learned models to assign a sentiment score to each newly found post (*Classifier* in the figure); iv) the aggregation module that has the task of turning the individual sentiment scores of each post into an aggregate sentiment score with a certain time regularity (*Aggregator* in the figure); and v) 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 based 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.

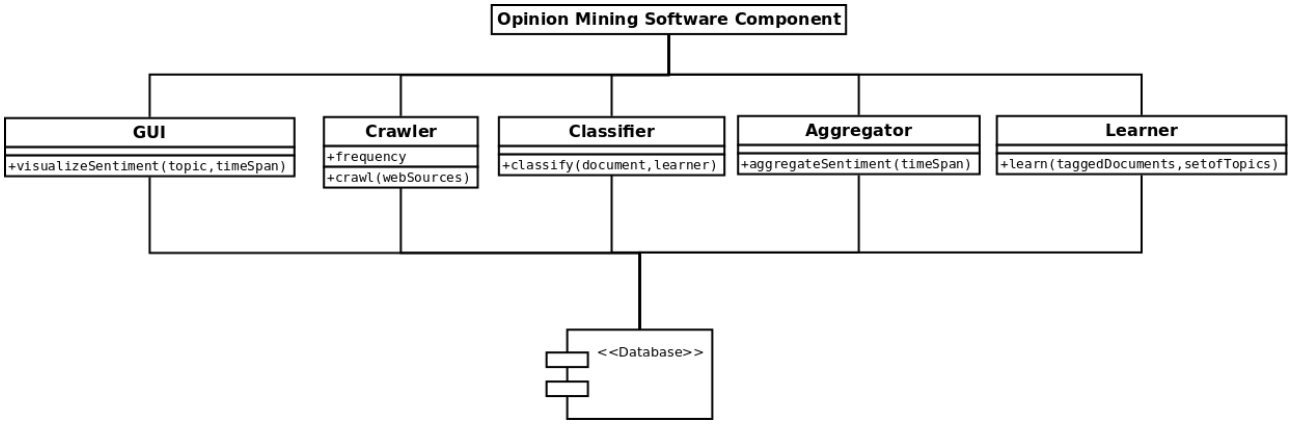

Figure 1: The overall picture of the Opinion Mining Software Component.

As mentioned in Section 2 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 behaviour, either by tuning the existing models or by extending them by including other topics of interest or by adding new sources of e-participation.

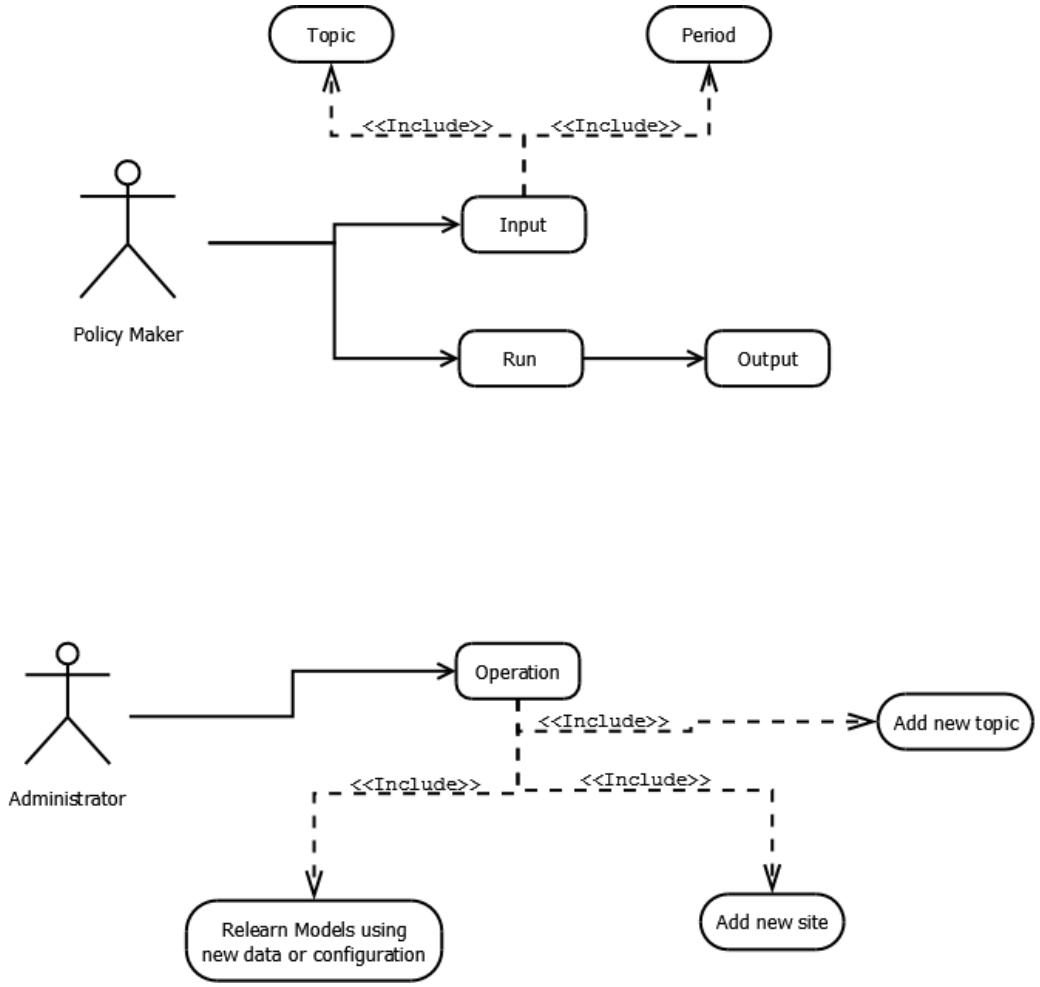

Figure 2: The different types of use cases of the OM component.

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.

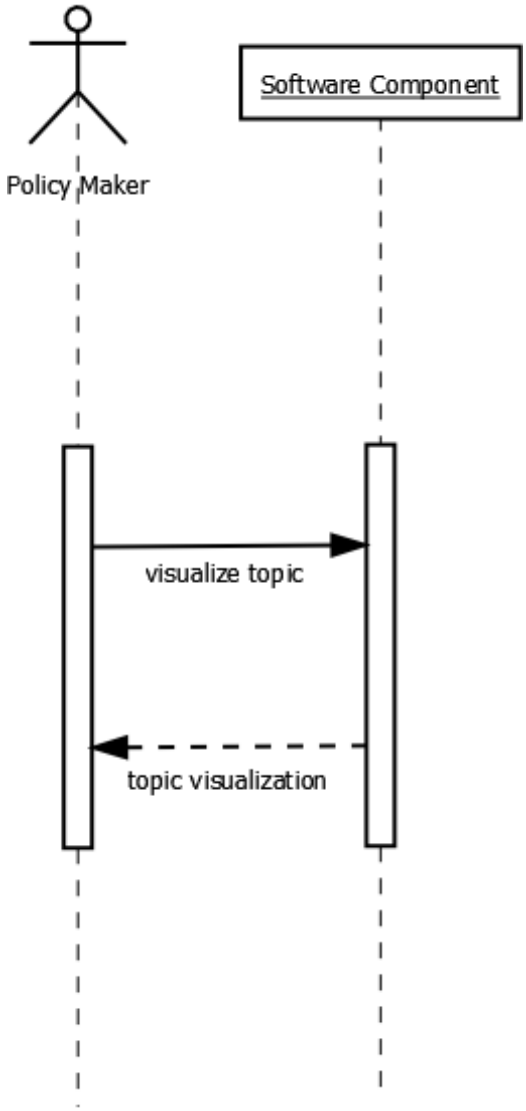

Figure 3: A sequence diagram showing the typical interaction of a policy maker with the OM component.

4 Technical Description of the Opinion Mining Prototype

The current prototype of the OM software component is implemented using software agents 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 module, 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 agents to be executed on different computers, which may be advantageous given the different computational requirements that each agents has.

Figure 4 provides an overview of the software agents that form the opinion mining software prototype. At the top level we have a controller agent, that can be regarded as the entry point to the prototype. One of the main tasks of this agent is to start GUI that will provide a kind

of control console for the users to interact with the OM prototype. Depending on the user interaction with this GUI different agents may be brought to life by the controller agent.

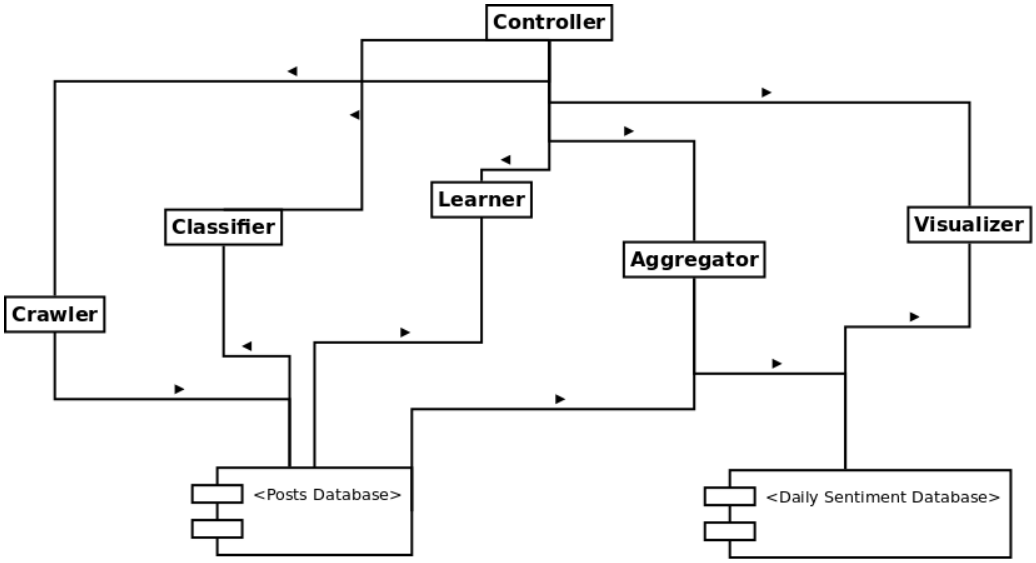

Figure 4: A class diagram showing the Opinion Mining prototype module.

With the exception of the individual website crawlers, all system components were developed using the R programming language [1]. The crawlers of each website were implemented using the Python Scrapy technology. This means we are using free and open source software that includes no acquisition costs, which was one of our requirements for this prototype.

Each agent runs as an individual process and is able to communicate with the other agents if necessary. In the following sub-sections we will provide some further details on each individual agent.

4.1 Controller

The controller is the main agent of the system and it is the entry point for the opinion mining prototype. This software agent is responsible for coordinating all other agents of the prototype and also for handling the user interface. When starting this agent, depending on the configuration files, a GUI is launched with a simple control console that allows different users to start different types of interaction with the OM component. This GUI is most probably going to be changed for the final version of the OM prototype, mainly due to the work on WP7. This control console allows users to start or stop the system, summarize[1] and visualize the sentiment regards some topic and if the user is an administrator it also allows her/him to add a new topic, add a new site or instruct the system to relearn new models. Figure 5 shows the main window of the control console provided by the GUI in the current version of the OM prototype.

[1]In this first version this functionality is still not implemented, though it will be essentially a different form of presenting the information already provided by the visualization agent.

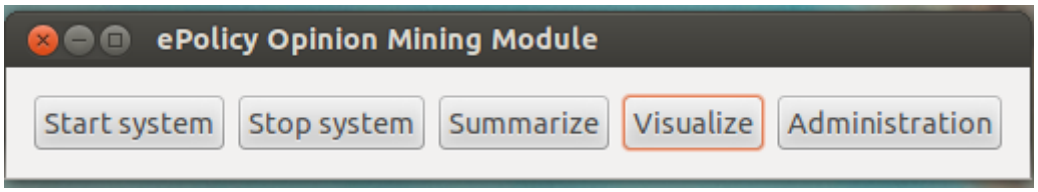

Figure 5: The control console of the OM software prototype.

4.2 Crawler

The crawler agent 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 that are typically restricted to users with administration access and knowledge of the system.

Once the crawler agent finds a new post in some of the sites, a message informing the controller of this event is sent.

4.3 Classifier

This software agent handles the classification of unlabelled posts. It can be done both in real time (each time a new post arrives the classifier is executed to classify it) or on a pre-scheduled basis (for example, classify the daily posts at the end of the day). The classifier uses the current best classification models for each of the topics being considered, applying it 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 agents or components of the e-Policy decision support system.

The execution of this software agent is controlled by the controller agent based either on the arrival of new posts (if running in real time), or on a regular schedule (e.g. at the end of each day).

4.4 Learner

The learner software agent is responsible for obtaining new sentiment classification models when new pre-labelled data is available. Currently, this agent is executed manually as the arrival of new labelled 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 agent is typically triggered manually by an administrator user.

4.5 Aggregator

The aggregator is a software agent that takes care of the task of aggregating the classification of each individual post into an aggregated sentiment score with some time granularity (tipically on a daily basis). With a certain frequency the agent is executed by the controller and it gathers all the posts of the past period (e.g. past day) together with the respective sentiment

scores obtained by the classifier agent, generating an aggregated sentiment score for each topic from these individual scores. These aggregated scores are then stored on the data base for use by other software components.

4.6 Visualizer

The Visualizer software agent allows the user to check the evolution through time of the sentiment scores. In this first version of the OM software prototype this facility is still very limited and will be one of the main sources of changes for the final version of the prototype, particularly with the contribution of the expertise on visualization brought by the Fraunhofer project partner in WP7. The current version was essentially developed for demonstration purposes.

This component is called whenever the user wants to visualize the data. There is no communication involved besides the launching of the component.

Figure 6 shows the current interface where the user can select the information to visualize, while Figure 7 shows an example of the type of graphs we are currently producing.

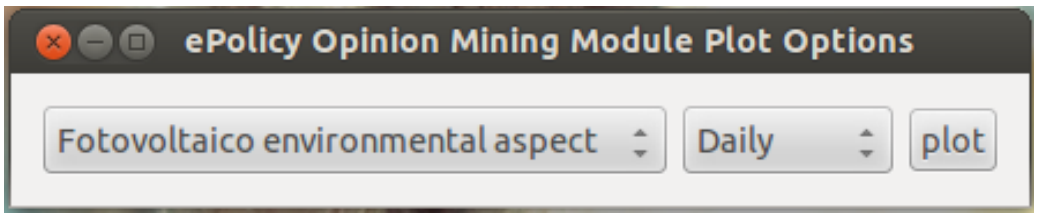

Figure 6: Visualization options.

Figure 7: Visualization of the sentiment of one topic.

These graphs show the user the individual scores of the posts that appeared during a certain time window as dots, and also some aggregation of these individual scores. In the example

of the figure the aggregation is show as a smoothing of the individual scores (a line) and also as a bandwidth capturing the variability of these individual scores. In Figure 7 we are showing this type of information for one specific topic.

4.7 The Database

The OM prototype uses several tables of a MySQL data base to store relevant information. Figure 8 describes these tables.

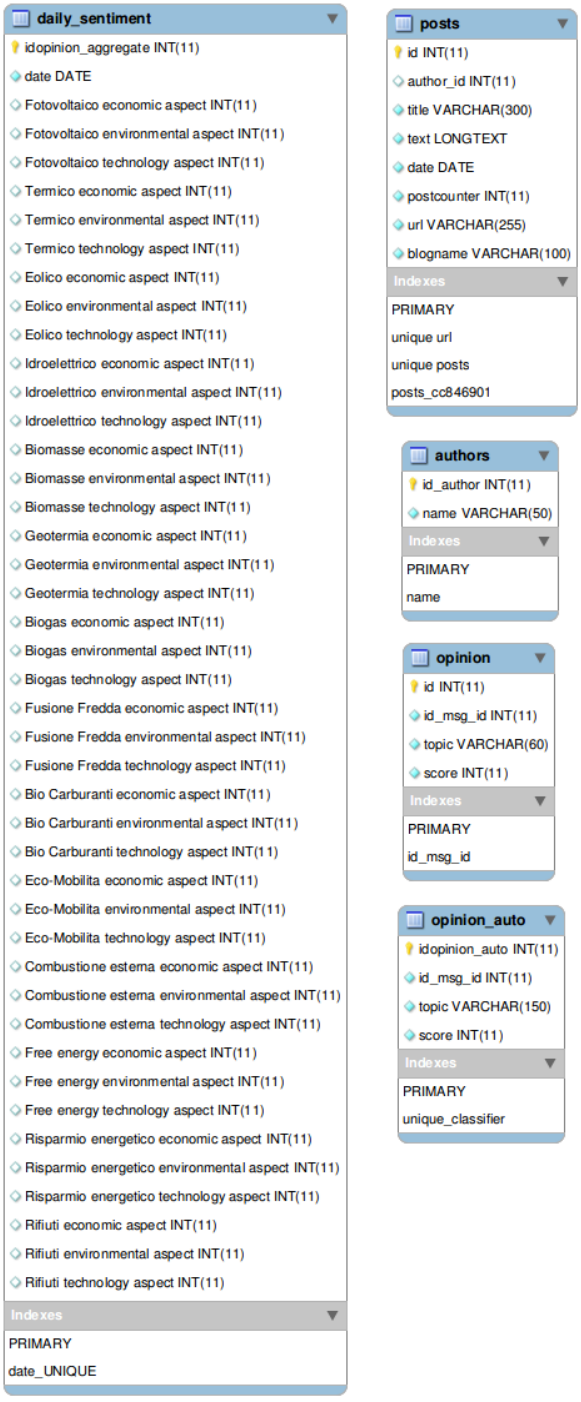

Figure 8: The used data base tables.

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 *authors* tables contains information on the authors of each post. In its current version the OM prototype is not taking advantage of this information. This information is also filled in by the crawlers.

The *opinion* table is the crucial table for the Learner agent. This table contains the sentiment scores assigned by a human expert to a sub-set of the available posts. This table is filled in using a web interface that was developed and made available to the human experts, to facilitate their labelling task. The information in this table is used by the Learner agent to obtain the sentiment classification models for each topic.

The *opinion_auto* table is identical in structure to the *opinion* table. However, the sentiment scores are the result of the application of the learned sentiment score models, and not the opinion of a human expert. This means that they are predictions of the OM prototype and thus have an associated degree of uncertainty. The information in this table is filled in by the Classifier agent and it is consumed by the visualization and summarization agents as well as any other software components of the e-Policy decision support system.

The *daily_sentiment* table contains the daily aggregated sentiment of all selected topics. They are the result of the Aggregator agent and may be used by any software components of the e-Policy decision support system.

5 Using the OM Software Prototype

The Opinion Mining software prototype can be operated locally by installing the software on some desktop computer, or can be run as a web service provided at INESC servers. In this first version of the software prototype some of the functionalities are limited, namely in terms of user interface and also in terms of summarization and visualization of the results of the sentiment classification models. These parts of the prototype will be the main goals to address for the final version of the prototype due in month M30. In this first version the main development efforts were centred on the system architecture and also on the key components of the prototype, so that a fully functional prototype could be implemented. In this context, we developed the crawlers for automatically fetching information from the selected e-participation sites, and also the components related with the sentiment classification models that will automate the assignment of sentiment scores to new posts on these sites. As a result the current first version of the prototype is able to crawl in real time for new posts on a set of pre-defined web sites and use sentiment classification models to tag these new posts in terms of the sentiment score regarding a set of topics. Future work on this software prototype will be coordinated with the work on WP7 with the goal of providing these results in a useful form to both normal users (e.g. policy makers) and other components of the e-Policy decision support system.

The following sub-sections provide some information for being able to use the current version of the prototype. Two alternatives exist: (i) install and run the software locally; or (ii) use the web service that provides access to the system that is being executed in real time at

INESC servers.

5.1 Local Installation

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.

Some parts of the software prototype currently have limitations in terms of the operating systems they can be used, mainly because they are still under development. Namely, the GUI (user interface) currently only runs under Linux and was in particular tested under the Ubuntu Linux distribution. These dependencies should be removed in the final version of the prototype due in month M30.

5.1.1 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[2]:
  - plyr
  - zoo
  - tm
  - Snowball
  - RWeka
  - rJava
  - RWekajars
  - DBI
  - RMySQL
  - gWidgets
  - ggplot2
  - Amelia
- Python 2.
  Downloadable at `http://www.python.org/download/releases/2.7.5/`
- Python Scrapy.
  Downloadable at `http://scrapy.org/download/`

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

5.1.2 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_om* 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.

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

- Run R in the folder where the prototype code is.
- In R, run the command : `source("go.R")`

5.2 Using the OM Prototype through a Web Service

As a Web Service, we provide the facilities that a normal user may be interested to explore, i.e. checking the sentiment scores of the population on some topics. All other operations, more related with the administration of the software prototype do not make much sense over the Web as they are meant to be carried out by expert users that are able to play with the configuration files of the system. Moreover, the prototype is being executed in real time at INESC servers so it also does not make sense to allow a web user to start and stop the system.

The web service providing access to the current version of the prototype is available at the following URL: `http://iris.dcc.fc.up.pt:8000/custom/OMprototype`

Figure 9 shows an example of the current version of the web service provided at this site. It contains two combo-boxes that allow the user to select the topic of interest and the frequency of the aggregated scores. In the current version of this prototype these options are limited to 3 alternative topics (or all of them together), and to daily aggregation of sentiment scores. The button "Visualize" will generate the graph corresponding to the selection of the user.

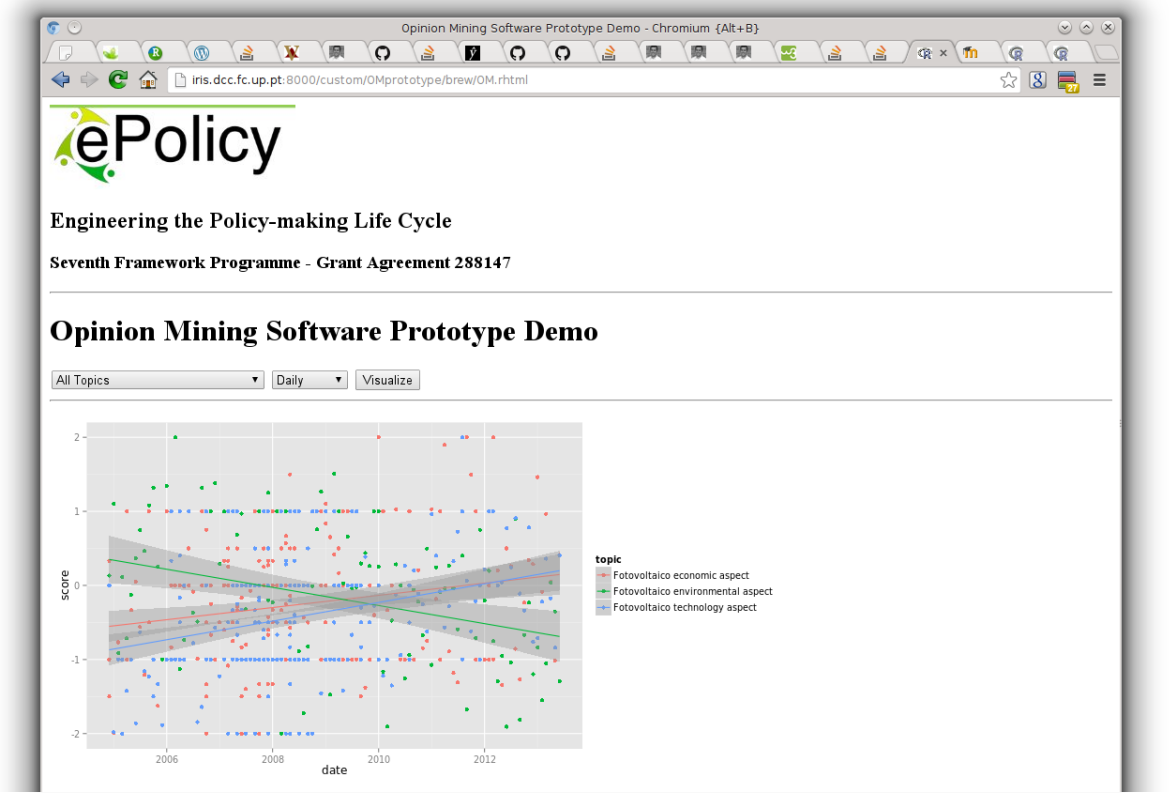

Figure 9: The Web Service of the Opinion Mining Prototype.

We should reinforce that most of the facilities concerning visualization are the topic of work package WP7 and thus will be subject to further changes/work in the near future..

This page has been intentionally left blank.

References

[1] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.