1.360.1

Engineering the Policy-making Life Cycle

Seventh Framework Programme – Grant Agreement 288147

# Opinion Mining Prototype Evaluation, version 1

| | |
|---|---|
| Document type: | Report |
| Dissemination Level: | PU |
| Editor: | Luis Torgo |
| Contributing Partners: | INESC PORTO |
| Contributing WPs: | WP6 |
| Estimated P/M (if applicable): | n.a. |
| Date of Completion: | 26 November 2013 |
| Date of Delivery to EC | 26 November 2013 |
| Number of pages: | 29 |

**ABSTRACT**

This document describes the first version of the evaluation of the opinion mining software component.

## 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 opinion mining (OM) component provides information on the sentiment of the population concerning a set of topics that are deemed relevant to both the global level optimizer and to the individual level simulation. However, the outcomes of the OM component can also be useful for user modelling. Namely, policy makers may find it useful to investigate the popularity trends of the topics most relevant to 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.

Deliverable 6.2 described the architecture (c.f. Figure 1) as well as the technical details of the OM prototype. This prototype already implements most the target features for this component of the e-Policy decision support system.

The overall task of the OM prototype is to be able to *infer the current sentiment of the population* concerning a *pre-defined set of energy-related topics*. To achieve this goal the OM prototype first crawls a set of *pre-defined e-participation sites* searching for new posts of the population that may discuss the selected topics. After this crawling stage the system will make *two main decisions concerning each new post*: i) which if any of the topics are discussed in the post; and ii) if it a post is relevant, what is the sentiment expressed in the post. To make these decisions the OM prototype needs to develop (learn) models that are able to classify correctly new posts in terms of these two issues. As shown in Figure 1, the two crucial components of the OM prototype are the **Classifier** and the **Learner**, where the former uses the models obtained by the latter.

This deliverable provides a first evaluation of the OM prototype and namely how the two key components of the prototype carry out the crucial task of inferring the sentiment of the population. More precisely, we report on the development and comparative evaluation of a set of alternatives for learner and classifier.

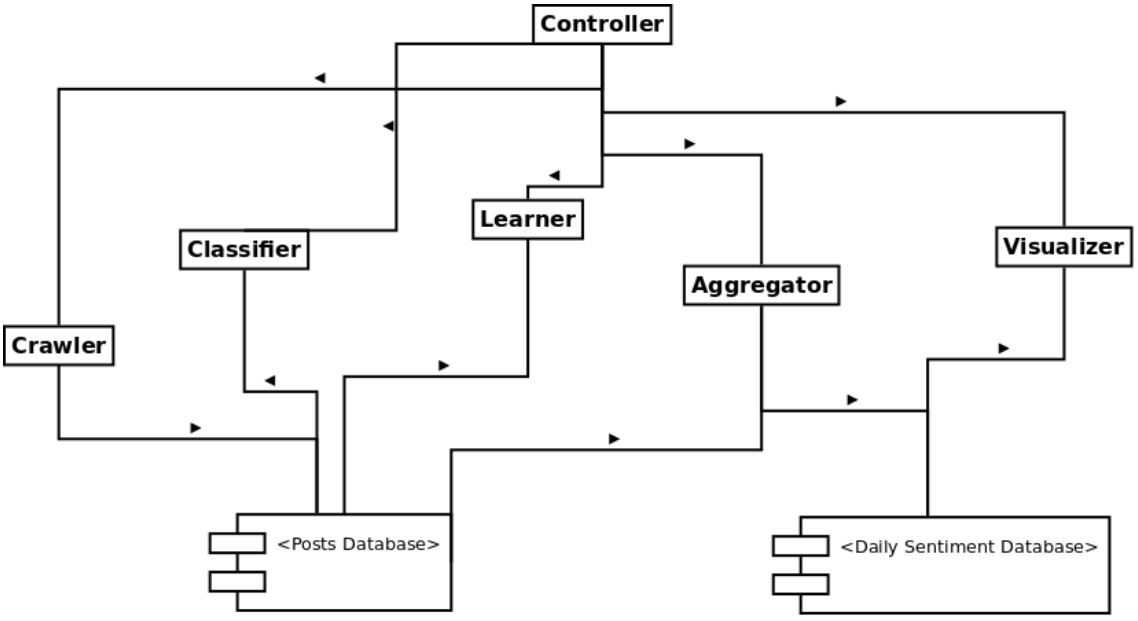


Figure 1: The proposed OM prototype architecture.

The structure of this deliverable is as follows. Section 2 formalizes the prediction tasks being addressed by the sentiment classifiers used by the OM prototype to infer the sentiment expressed in a text document. Section 3 describes in detail the solutions we have considered to solve these tasks. In Section 4 we describe the evaluation methodology that was selected to check the quality of the proposed solutions, while in Section 5 we present and discuss the results of this evaluation.

# 2 Problem Formalization

Text mining addresses the problem of analysing and extracting information from text documents. In our case the goal is to infer the sentiment expressed in each document concerning a pre-defined set of topics related to energy policies. Nowadays there is a massive amount of data available on the internet, providing an invaluable source of information on the opinion of people concerning almost every possible topic. Different e-participation tools facilitate the task of expressing our opinion. Having a system capable of classifying documents automatically will allow us to exploit massive amounts of data and extract useful information on the sentiments expressed by the public.

As we have mentioned before, given a new post, our text mining models need to be able to predict whether the post expresses sentiment concerning a set of pre-defined energy-related topics. Any of these topics may, or may not be, mentioned in the post. If a topic is mentioned then the OM should infer the sentiment expressed in the post. We therefore decomposed the problem into two separate text mining tasks:

- **Topic identification** - decide for the select set of $q$ topics which ones are mentioned in the document, i.e. make a set of $q$ binary decisions.
- **Sentiment scoring** - for each topic that is mentioned in a document decide what is the sentiment concerning that topic on a pre-defined scale.

Concerning the first task, we have followed two different approaches. The first approach takes the $q$ binary decisions (is or is not mentioned in the document) as independent and thus we essentially develop $q$ binary classifiers. The second approach tries to take advantage of eventual correlations among the $q$ topics and uses models that forecast the $q$ binary values at the same time.

With respect to the second task of sentiment scoring we have followed approaches that forecast the sentiment score for each topic individually.

All the tasks mentioned above can be seen as instances of predictive tasks. Predictive modelling has the general goal of inferring the value of some set of unknown variable(s) $\mathbf{y}$ from the values of known variables $\mathbf{x}$ that are supposed to influence the values of the $\mathbf{y}$. Specifically, we are assuming that there is some form of functional dependency of the values of the variable(s) in $\mathbf{y}$ on the values of the $\mathbf{x}$ (frequently referred to as the predictors). In our concrete application the information available for the models to make their decisions is in the form of a text document (a post at some web site). This means that if we assume the information in a text document is represented as a feature vector, we can look at our tasks as instances of standard predictive tasks.

Predictive tasks can be described as data analysis problems where one assumes that there is a functional dependency between a target variable $Y$ and a set of descriptor variables (or predictors) $X_1, X_2, \cdots, X_p$. The goal of predictive modelling is to infer this function from a sample of mappings between values of the predictors and the target variable, i.e. a (*training*) data set $\{\langle \mathbf{x}_i, Y_i \rangle\}_{i=1}^{N}$, where $\mathbf{x}$ is a *feature vector* formed by values of the $p$ predictor variables $X_1, X_2, \cdots, X_p$.

In data mining the two most common instances of predictive tasks are known as regression and classification. In regression we use the provided training data set to induce a model of the unknown function,

$$Y = f(\mathbf{x}) \tag{1}$$

where $Y$ is the **numeric** target variable and $\mathbf{x}$ is the vector of predictor variables $X_1, X_2, \cdots, X_p$.

In classification we have a similar inference problem but the domain of the target variable is a finite set of labels, i.e. $Y$ is a **nominal** variable.

Our first task of topic identification is an instance of a classification task as our target is to decide if the topics are or not mentioned in a post, i.e. a decision with two possible outcomes that are nominal (usually known as *binary* classification problems). As we have mentioned we will follow two approaches to obtain these binary decisions for our pre-defined $q$ topics. The first addresses this as $q$ independent binary classification problems. The second approach considers this as an instance of a *multivariate* classification problem with $q$ target variables. In multivariate classification the target is a vector of variables and not a single variable, i.e.

$$\mathbf{y} = f(\mathbf{x}) \tag{2}$$

where $\mathbf{y}$ is a vector of target variables $Y_1, Y_2, \cdots, Y_q$; and $\mathbf{x}$ is the vector of predictor variables $X_1, X_2, \cdots, X_p$.

We address the task of sentiment scoring as $q$ separate score prediction tasks. The sentiment on a certain topic can be expressed in many ways. Usual formats include positive vs negative sentiment, or some rating scale. We follow the latter approach by trying to infer the sentiment in a document in terms of a $-2, -1, 0, 1, 2$ scale, where negative numbers represent negative sentiment, while positive numbers the opposite. Coarser or finer granularities would be possible, but the approaches we will describe are generalizable to these other solutions as long as they can be regarded as values of an ordinal variable.

Given that our target variable is the value of the sentiment on an ordered fixed scale, i.e. an ordinal scale, we have a particular type of prediction task that differs from the more standard regression and classification tasks. Still, given the limited amount of available methods that address ordinal target variables, we have solved this sentiment scoring task using regression and classification approaches. In order to do so, we have designed pre-processing steps that will be described in the next section.

# 3 Implemented Solutions

In the previous section we have identified and formalized two main prediction tasks as driving the key components of the OM prototype: i) topic identification; and ii) sentiment scoring.

Both tasks share the same input - a set of documents. In more formal terms this means that all tasks share the same predictor variables, the only difference being on what they predict (i.e., the target variables of the prediction models).

In this section we describe how we address the two tasks. As a first step, we present the data representation that translates the text document into a set of variable values, required by both tasks.

## 3.1 Document Representation

The way we represent a document can have an impact on the obtained models and on the respective predictive performance. The literature describes several ways of representing a text document as a feature vector, with the most popular alternatives being the Bag of Words (BOW) and the N-gram representations.

The N-gram representation involves the creation of sequences of N-words. For example, on a 2-gram representation, the sentence 'I went to the garden today' could generate 3 groups of 2-grams, 'I went', 'to the', 'garden today'. Then, after discovering all the groups in our corpus, we count how many times they appear in the document and assign this value to the group. This type of representation tries to keep some information about the sequence of the words or the context in which each word appears.

The Bag of Words (BOW) representation, the one we have adopted, is the most frequent approach. We represent the document by separating the sentences into single words. For example, on the previous referred sentence, we can identify the words 'I', 'went', 'to', 'the', 'garden' and 'today'. This strategy usually proceeds by identifying all words in a given corpus (eventually after some pre-processing steps like stop word removal, or word stemming) and then by counting the occurrences of each identified word on each document. This means that the features or predictor variables used to represent the texts in a data set will be this (often large) set of identified words. As values of these predictors an usual choice is to assign the frequency (the number of times the word appears on the document, or term frequency (tf)). Another option is the tf-idf (term-frequency inverse-document-frequency) score which modifies the term frequency with a factor related to the importance of each word (term) of a document within a collection of documents. If the word appears more frequently in the collection of documents then its tf-idf value will be high. This tells us which words separate documents better (if they only appear in few documents then they distinguish these from the others).

On both representations, we need to decide what to do with all the words found in a corpora. Do all of them interest us? Should we, for example, keep numbers and punctuation? Although some of these decisions may be domain-dependent, frequent pre-processing stages include: (i) removal of stop words; (ii) removal of punctuation and numbers; and (iii) word stemming.

In summary, although many alternatives exist for representing the information in a text document we have selected the frequently used bag of words representation using term frequency as values. We have also opted to remove stop words, punctuation and numbers and apply word stemming. In order to reduce the number of words, we have removed sparse terms with a factor of less than $0.95$. This resulted in using a total of $172$ words whose frequency will be the predictor values describing each text document.

## 3.2 Topic Identification

The first step of the OM prototype for inferring the sentiment expressed in a post consists in identifying the topics that are addressed in this text document. In Section 2 we have mentioned that we have tried two approaches to this identification task.

The first approach consists of handling this as $q$ separate binary classification problems, where $q$ is the number of pre-selected topics. For each of these $q$ prediction problems a binary classification model was developed using the available training data with the goal of approximating the function $Y = f(\mathbf{x})$, where $Y$ takes two possible values, e.g. $yes$ and $no$, meaning that the respective topic is (or is not) mentioned in the document being analysed that is represented by the feature vector $\mathbf{x}$ (as we have seen the frequency of 172 pre-selected words). To address these $q$ binary classification tasks we have learned several alternative models using different machine learning algorithms that will be detailed in Section 4.2.

The second approach tackles the $q$ topics identification problem using a single multivariate classification model. The idea/motivation is to try to explore eventual correlations among the $q$ topics. With this purpose we tried different variants of the Clus [1] system.

## 3.3 Sentiment Scoring

The second step of the OM prototype is to infer the sentiment expressed in each document that was identified as mentioning a certain topic.

The selected scoring scale can be regarded as the domain of an ordinal variable. As mentioned before, few algorithms are available to address predictive tasks with ordinal target variables. In this context, we have implemented a different approach not to limit the range of solutions to this task. Namely, we have followed two different paths to the $q$ sentiment score prediction tasks: i) predict the score using classification models; and ii) using regression models.

Classification algorithms do not assume any ordering of the values of the target variable, which we have seen is not true in our sentiment scale. An order among the values means that it is worse to misclassify a document with sentiment $-2$ as having sentiment $2$, than classifying it as having sentiment $-1$. Classification algorithms consider all errors equally serious and thus can not cope with the above distinction. To achieve this distinction we can resort to cost matrices. A cost matrix is a $c \times c$ matrix where $c$ is the number of possible labels of the target variable. The rows and columns of this matrix represent the possible values for the predictions and true values of any test case. The entries in the matrix specify a value (a cost) for each possible combination of predicted and true target variable value. Using

these matrices we can specify the costs such that it is more costly for the model to predict a value of $2$ for a document with true sentiment of $-2$, than the cost of predicting $-1$. This means that through cost matrices we can convey the order information to the classification models by means of different costs of the errors, and thus use the large variety of classification algorithms in our $q$ sentiment scoring tasks.

Regression tasks assume that the target variable is numeric, which means that there is an implicit ordering among its values. This allows us to handle the different types of sentiment scoring errors naturally without having to resort to cost matrices as in classification. Still, regression methods allow interpolation among values, which means that some model could come up with a predicted sentiment score of $1.234$. In order to force the predictions into our selected sentiment scale, when using regression tools, we will re-scale the predicted values back to the original scale by applying the following rounding function to the predictions:

$$f(x) = \begin{cases} 2 & \text{if } x \geq 1.5 \\ 1 & x \in ]0.5, 1.5] \\ 0 & x \in ]-0.5, 0.5] \\ -1 & x \in ]-1.5, -0.5] \\ -2 & \text{if } x < -1.5 \end{cases} \tag{3}$$

In summary, we have tried, evaluated and compared different variants of several classification and regression algorithms (c.f. Section 4.2) with the goal of solving sentiment scoring for each of the $q$ topics. When one of the $q$ topics is identified as being present in one new post using the models of the previous section, the respective sentiment scoring model is used to forecast the sentiment score expressed in that document regards that topic.

# 4 Evaluation Methodology

This section describes the key aspects of the methodology that was followed to evaluate the proposed solutions. We start by describing the data that was available for this evaluation. We then provide more details concerning the models that were used in implementing the solutions outlined in Section 3. Finally, we discuss the metrics that were used to evaluate the different alternatives as well as the experimental methodology that was followed to obtain reliable performance results.

## 4.1 The e-Policy data set

The e-Policy project is concerned with energy policies for the region of Emilia-Romagna in Italy. In this context, all activities concerning the involvement of the population with e-participation tools will naturally use the Italian language. Most of the existing research on text mining is carried out with the English language but work on other languages is growing [2]. Especially in huge global events such as the Olympics or Soccer World Championships, it is very important for the media to be able to extract and process large amounts of data as fast as possible which makes the study and development of this field very important

in all languages. On the e-Policy project, having efficient models and tools tailored for the Italian language is essential.

In terms of the goals of opinion mining within the project the consortium has decided to focus on 14 main topics and 3 subcategories (economic, environmental and technological aspects) for each, totalling 42 topics. The goal of the tools developed within the project is to infer the sentiment of the population concerning these 42 topics and also to provide information on tendencies of this sentiment along time, so that the eventual impact of decisions taken by policy makers can be measured. The list of 14 selected main topics is the following:

- Photovoltaic
- Thermal
- Wind power
- Hydroelectric
- Biomass
- Geothermal
- Biogas
- Fusion
- Biofuels
- Eco-Mobility
- Combustion
- Free energy
- Energy saving
- Waste to energy

As mentioned above, for each of these 14 topics, 3 different aspects were considered.

In terms of sources for mining the opinion of the public, the consortium has decided to start by exploring documents from two Italian websites [7, 8] - Energetic Ambient (Figures 2 and 3) and the Newclear blog (Figure 4). On both websites the different posts are structured as a hierarchy starting with a top post and then sub-sequent posts discussing this main post.


Figure 2: Energetic Ambient front page [7].


Figure 3: Energetic Ambient forum.

In the context of the OM prototype implementation we have developed crawlers for these two websites that have collected a data set with posts and some information associated with each post. Table 1 presents the information that is collected for each post by our crawlers, like the date, title and post counter of each post (if it is a main post or a reply to the main post), etc. In spite of the availability of all this information, the approaches described in this deliverable will only make use of the text of each post.

Predictive modelling requires a training set where the values of the target variables are known. In the context, we need a data set with posts that are tagged regards the sentiment expressed for each of the topics selected for this study. Tagging a large amount of posts for the 42 topics is a task that requires huge human resources with expertise in the energy field. Carrying out this task for the roughly 600 000 posts[1] that currently form our data base is not possible with the human resources available to the project. In this context, the amount of data (posts) that were read, analysed and tagged concerning the sentiment is limited, and amounts currently to around 800 text documents (c.f. Table 1). Moreover, we only have sentiment scores for 3 out of the 42 topics : 'Photovoltaic economic aspects', 'Photovoltaic environmental aspects' and 'Photovoltaic technology aspects'. Table 2 summarizes the number of available posts for each of the currently selected topics.

In summary, for the initial task of topic identification we will have a data set of formed by 857 posts tagged for the 3 topics. For the subsequent sentiment scoring tasks we will have data sets with different sizes depending on the topic, according to the numbers on Table 2. These numbers are obviously small, but we are working on trying to increase the number of tagged documents so that the statistical significance of the experimental evaluation of our models can be increased.

[1]This number is growing constantly as our crawlers are running in real time.


Figure 4: Newclear blog [8].

**Table 1** e-Policy Data set composition.

| Number of Documents | Number of Tagged Documents | Features |
|---|---|---|
| 582382 | 857 | ID,Author ID, Title, Text, Date, Postcounter, URL, Blogname, Topic, Score |

**Table 2** e-Policy data set composition by topic.

| Topic | Number of Documents |
|---|---|
| Photovoltaic Economic Aspect | 501 |
| Photovoltaic Environmental Aspect | 63 |
| Photovoltaic Technology Aspect | 410 |

## 4.2 The Modelling Techniques

The predictive tasks we have described in Section 3 use three types of models: i) multivariate classification; ii) classification; and iii) regression.

Concerning multivariate classification we have considered in our experiments the Clus [3] system. This learning algorithm is a decision tree and rule induction system that implements the predictive clustering framework described in [3]. While most decision tree learners induce classification or regression trees, Clus generalizes this approach by learning trees that are interpreted as cluster hierarchies. Such trees are called predictive clustering trees or PCTs. This system can handle multivariate classification tasks. We have considered several variants of this tool in our evaluation. Namely, we tested different ensemble parameters with the options Bagging, Random Forests, Random Subspaces and Bag of Subspaces. As for the other parameters of Clus, they were left with the default settings.

Concerning standard classification and regression tasks we have considered some of the most popular techniques: Random Forests, Support Vector Machines and Neural Networks. These

approaches not only are recognised as some of the best modelling tools as are able to address both classification and regression tasks.

Random Forests [4] are an ensemble learning method for classification and regression tasks composed of many decision trees created with the training data. Each tree is trained on a bootstrapped sample of the original dataset and each time a split node is created, only a randomly chosen subset of the predictors are considered for splitting. In terms of using random forests for prediction, their forecasts are the mode of the classes output by each tree in the ensemble in the case of classification tasks, or the average of the predicted values if it is a regression problem. In our experiments we have used the implementation available in the R package 'randomForest', ported from the original Fortran code by Andy Liaw and Matthew Wiener [9]. In terms of variants of these models we have considered different values for the parameter *ntree* which controls the number of trees to grow, and the parameter *mtry* that controls the number of variables randomly sampled as candidates for each split.

Support Vector Machines [5, 6], or SVMs, are a relatively recent modelling approach with a large success in many application domains. This approach is applicable to both classification and regression tasks. Nevertheless, the approach was originally developed for binary classification problems and it is easier to explain it within this setup. SVMs try to find a hyperplane that separates the cases belonging to each class (as for instance linear discriminants also do). With the goal of finding the hyperplane that maximizes the margin between the cases of the two classes, SVMs use quadratic optimization algorithms. Unfortunately, most real world problems are not linearly separable. The solution provided by SVMs to this problem consists in mapping the original data into a higher dimension input space where the cases belonging to the two classes can already be linearly separable. Although this solves the problem of linear separability, this creates another problem - applying the quadratic optimization algorithms on these high dimension spaces is computationally very demanding. To solve this extra problem SVMs use what is known as the kernel trick, which consists in using certain kernel functions that are cheap to compute and that are proven to lead to the same result as the expensive dot products that are used in the quadratic optimization algorithms when applying them in the high dimension space. These kernel functions are cheap to compute because they are calculated in the original, low dimension space. Still, their result is equal to the mentioned dot products which allows SVMs to obtain the hyper-planes in the high dimension space without having to carry out heavy computation steps on this space. This general approach has been generalized to both multi-class problems and regression tasks, and thus we can use this methodology in our tasks. We have used the SVM implementation available in the R package 'e1071' created by David Meyer [11]. In terms of different variants of SVMs we have we have varied the parameters *cost*, *epsilon* and *gamma*. The parameter *cost* sets the value associated with the cost of constraints violation, it is the 'C'-constant of the regularization term in the Lagrange formulation. The parameter *epsilon* controls the epsilon in the insensitive-loss function and *gamma* is a parameter used in the kernel.

Artificial Neural Networks [10, 14] are models with a strong biological inspiration. They are composed by a set of units (neurons) that are connected. These connections have an associated weight and the learning process consists of updating these weights. Each unit has an activation level and means to update this level. Some of these units are connected to the out-

side, being called input and output neurons. Each unit has one simple task, receive the input impulses and calculate its output as a function of these impulses. This calculation is divided in two parts: a linear computation of the inputs and a non-linear computation (activation function). Different activation functions provide different behaviours. Some examples of common functions are the Step function, the Sign function and the Sigmoid function. The units can also have thresholds that represent the minimum value of the weighted sum of the inputs that activates the neuron. There are two main types of Artificial Neural Networks:- the feed-forward networks and the recurrent networks. The feed-forward networks have unidirectional connections (from input to output), without cycles, while the recurrent networks have arbitrary connections. Usually the networks are structured in layers. On a feed-forward network each unit is connected only to units on the following layers while on a recurrent network this does not happen and the network can have feedback effects, possibly exhibiting chaotic behaviour. They usually take longer to converge. The learning process of Artificial Neural Networks consists of updating the weights of the connections. The most popular way to do this is by using the Backpropagation algorithm. Each example is presented to the network. Then, if the output produced is correct, nothing is done. If it is not correct then we need to re-adjust the network weights. In networks with multiple layers the adjustment is not simple as we need to divide the adjustments across the nodes and layers of the network. A detailed description of the back-propagating algorithm is given by David E. Rumelhart [12]. In our experiments we have used the implementation of feed-forward Artificial Neural Networks available in the R package 'nnet' created by Brian Ripley [13]. In terms of different variants of ANNs we varied the parameter *size* that controls the number of units in the hidden layer, and the parameter *decay* which controls the weight decay.

## 4.3 Evaluation Metrics

Our evaluation has the goal of comparing different approaches to 2 tasks: i) topic identification; and ii) sentiment scoring. We have seen that these two tasks are inherently different. This is also reflected on the choices we have made concerning the metrics used to characterize and compare the performance of the different alternatives.

For the topic identification task the goal is to decide for the three selected topics if a document mentions them or not. This means that for each document we want to compare the prediction of our approach (a vector of three decisions concerning the three topics, e.g. $\langle yes, no, yes \rangle$) against the ground truth (another vector of three decisions). The evaluation of any prediction model typically involves this type of comparisons across a large set of cases (a test set). Assuming the existence of a set of $N_{test}$ cases to be used to evaluate a model, we can count for each topic how many predictions were correct. More specifically, we can obtain a confusion matrix with counts for each of the four possible situations[2] , and for each topic. With these confusion matrices we can calculate standard statistics like Accuracy, Recall, Precision and the F-measure. Finally, we can average the scores on these statistics across the 3 topics and have a final score for each of the metrics for each alternative we have considered in our evaluation. These are the results we will show in Section 5.

[2]Number of possible combinations of true and predicted values given that we have binary classification tasks.

Regarding the sentiment scoring task we have to take into account that our target is an ordinal variable. Moreover, as we have seen we will consider both regression and classification algorithms to solve this task. Still, independently of the algorithm their predictions can be cast into the selected scale of sentiment. For comparing the true and predicted sentiment score of a document we have used a cost matrix that can express the notion that not all errors are equivalent. We have used as evaluation metric the total cost of the predictions. This evaluation metric assumes the existence of a cost matrix indicating the cost of each misclassification. Models should try to minimize this score. We have used the following cost matrix in our experimental comparisons:

**Table 3** Cost matrix used in our experiments.

| | -2 | -1 | 0 | 1 | 2 |
|---|---|---|---|---|---|
| -2 | 0 | 1 | 2 | 3 | 4 |
| -1 | 1 | 0 | 1 | 2 | 3 |
| 0 | 2 | 1 | 0 | 1 | 2 |
| 1 | 3 | 2 | 1 | 0 | 1 |
| 2 | 4 | 3 | 2 | 1 | 0 |

Given this cost matrix the total cost of the predictions of a model for a single topic, given a test set with $N_{test}$ documents is given by,

$$TC = \sum_{i=1}^{N_{test}} M_{\hat{y}_i, y_i} \tag{4}$$

where $M_{\hat{y}_i, y_i}$ is the entry in the cost matrix $M$ corresponding to a prediction of $\hat{y}_i$ for the document whose true value is $y_i$.

## 4.4 Experimental Methodology

Any evaluation procedure based on data must be concerned with the statistical significance of the reported results. With this goal in mind we have designed an experimental methodology that can provide reliable estimates of the evaluation metrics that were described in the previous section, and that will also allow for comparisons of the observed performance differences in terms of statistical significance levels.

The data sets to be used in our experimental comparison are different depending on the task being addressed. Still, for each of the problems, all considered model variants will be evaluated using the same train and test partitions of the available data. More specifically, for each predictive task we will estimate the performance of all alternatives included in our study by means of 10 repetitions of a 10-fold Cross Validation process. This means that all scores we will report are averages of 100 train+test trials with the respective modelling solution. This experimental procedure ensures a good level of statistical significance of our reported results. Moreover, the use of the same train+test partitions for all variants allows to perform paired comparisons among the alternatives to check the statistical significance of the

observed differences. These paired comparisons were carried out using the Wilcoxon Signed Rank test which is a non-parametric statistical hypothesis test that compares two related repeated measurements to assess whether each set population mean ranks differ.

# 5 Results of the Evaluation

In this section we present and analyse the results that we have obtained on our experimental comparisons. This is done separately for each task at hand, predicting the topic and assigning the sentiment score for each topic.

## 5.1 Topic Identification Results

In Table 4 we have a summary of the results of the best variant of each learning algorithm for the task of identifying which topics are mentioned in a document. These results are the average of the scores obtained on each topic. A random forest variant (you may check in the Annexes the parameter settings corresponding to each variant) stands out as the best performer in every statistic. The results among the other alternatives are more balanced when looking at all evaluation metrics.

**Table 4** Best performing models in topic prediction.

| Model | Precision | Recall | F1 | Accuracy |
|---|---|---|---|---|
| Msvm.v10 | 0.71±0.07 | 0.61±0.07 | 0.66±0.06 | 0.78±0.03 |
| **MrandomF.v8** | **0.75±0.02** | **0.70±0.07** | **0.72±0.05** | **0.82±0.02** |
| Mnnet.v3 | 0.67±0.07 | 0.69±0.07 | 0.68±0.06 | 0.77±0.03 |
| Mclus.v1 | 0.67±0.08 | 0.57±0.07 | 0.61±0.06 | 0.76±0.03 |

The results in Table 4 are averages of 100 train+test iterations (we are using 10 repeations of 10-fold cross validation). Figures 5, 6, 7 and 8 show a series of box-plots of the performance achieved by the same models across the different iterations of the $10 \times 10$-fold CV process on different statistics. These graphs provide a more detailed perspective on the distribution of the scores across the 100 repetitions. They indicate that not only random forests achieve the best performance but they are also more stable across all iterations, given their more compact box-plots.

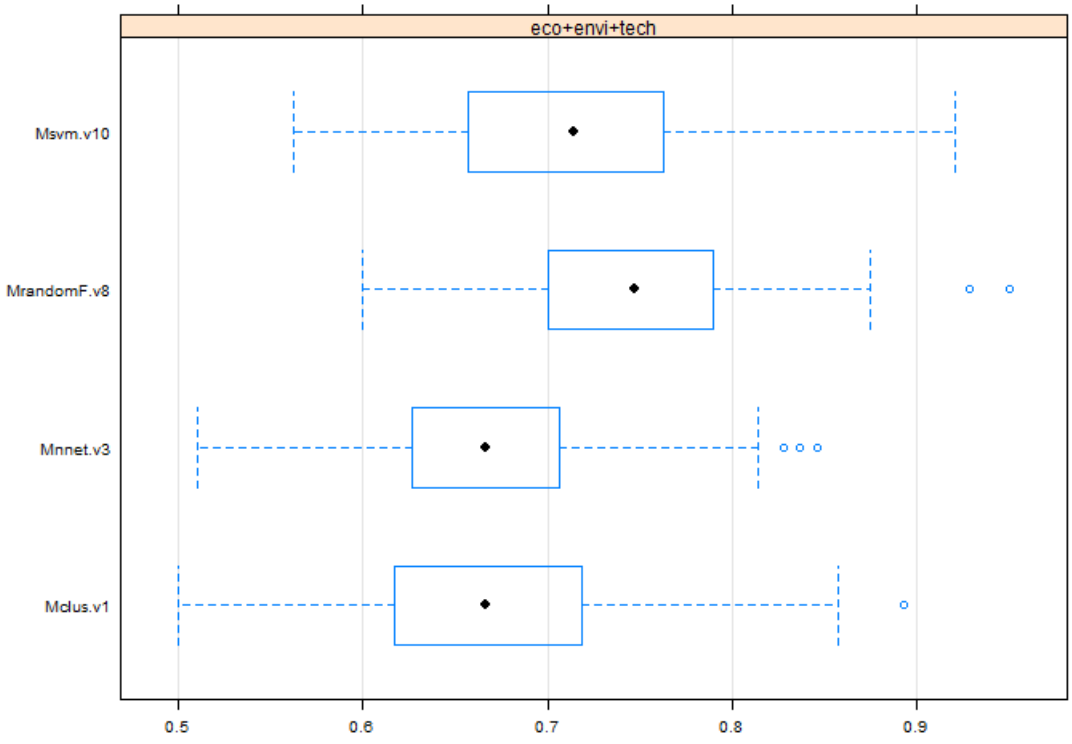


Figure 5: Precision of the models in the experiments.

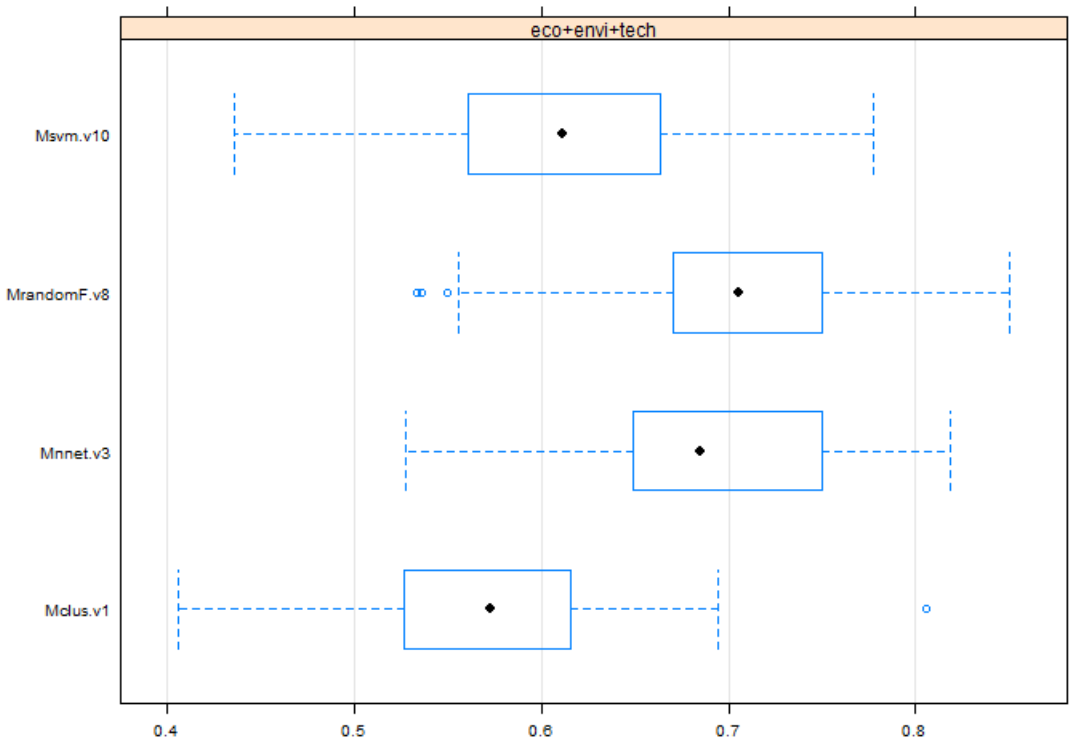


Figure 6: Recall of the models in the experiments.

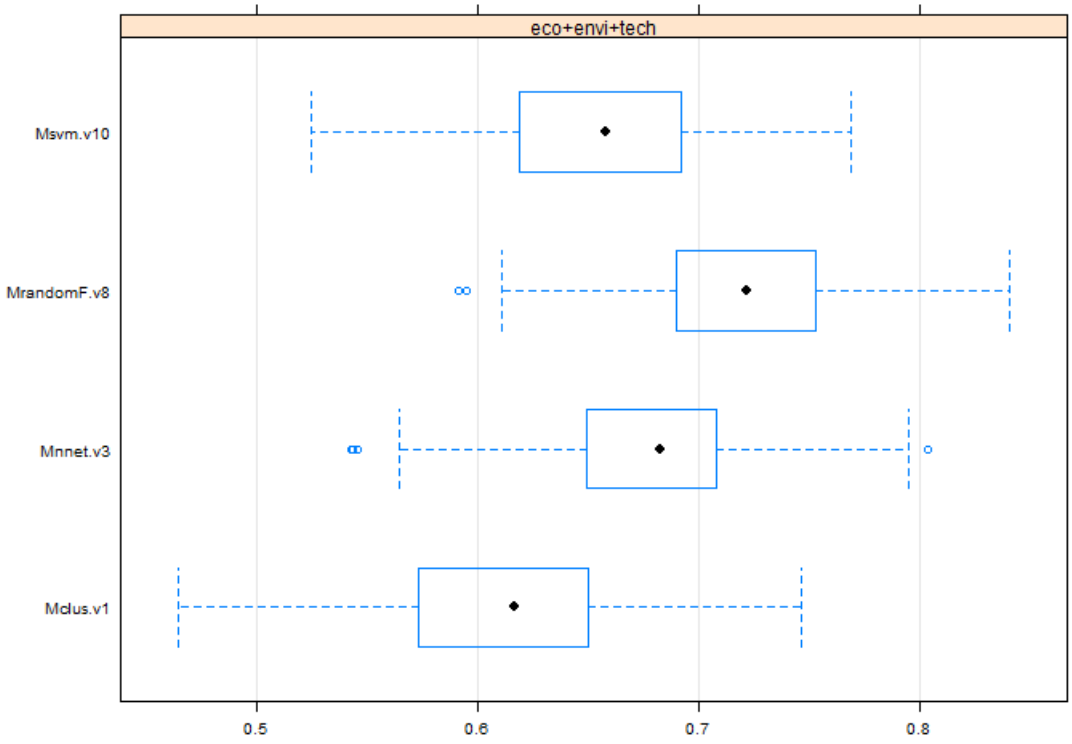


Figure 7: F1 measure of the models in the experiments.

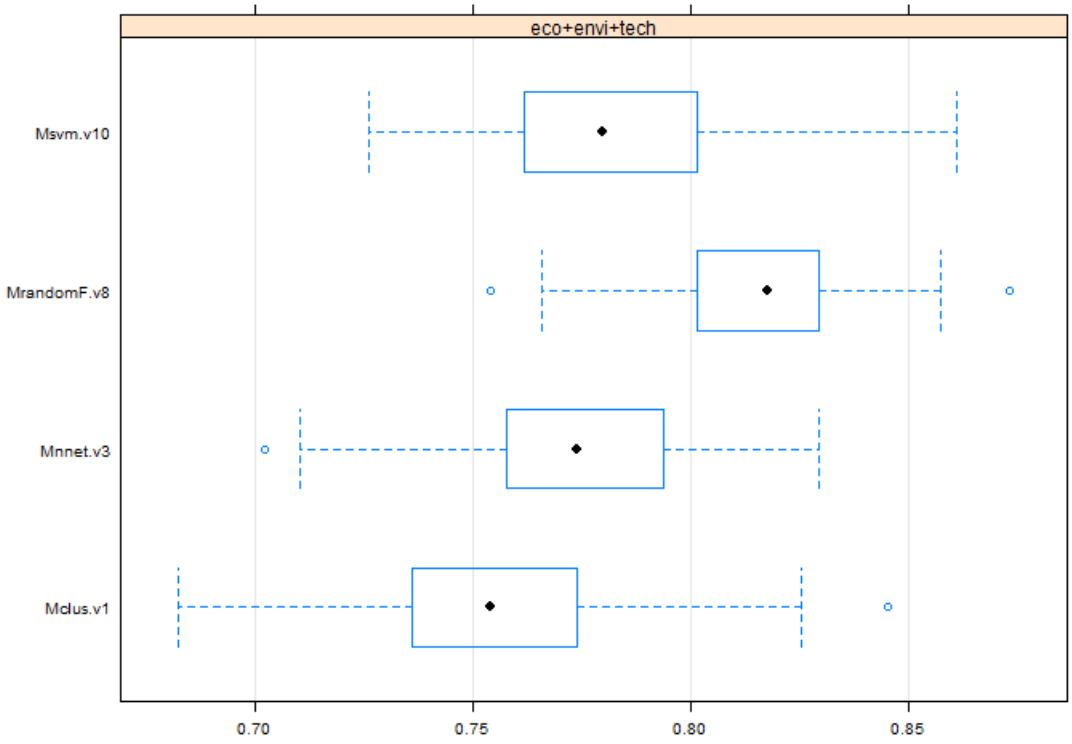


Figure 8: Accuracy of the models in the experiments.

To check that the differences in average performance reported on Table 4 are statistically

significant, we performed a Wilcoxon Signed Rank test. This is a non-parametric statistical hypothesis test that compares two related repeated measurements to assess whether each set population mean ranks differ. Table 5 shows the results of this test when comparing the average score of the best random forest against the other best variants. For each of the competitors, in the column "Statistical Significance" we may have: i) no signal, if the difference between the performance of the competitor and the random forest is not significant (more specifically confidence below 90%); ii) one signal of the difference is significant at the 95% level; or iii) two signals for 99% confidence. Plus signs (+) indicate that the average score of the competitor is significantly higher than that of the random forest, whilst minus signals (−) indicate a significantly lower score of the competitor. Please note that having higher or lower scores may have different meanings depending on the quantity being estimated. In effect, for certain quantities we want to minimize them (e.g. a prediction error) so having a lower value is better, while for others (e.g. precision) we want to maximize them, so having a lower value is worse.

**Table 5** Statistical significance of the observed differences.

| Measure | Learner | Average | Standard Deviation | Statistical Significance |
|---|---|---|---|---|
| Precision | Msvm.v10 | 0.71 | 0.07 | −− |
| | **MrandomF.v8** | 0.75 | 0.07 | N/A |
| | Mnnet.v3 | 0.67 | 0.07 | −− |
| | Mclus.v1 | 0.67 | 0.08 | −− |
| Recall | Msvm.v10 | 0.61 | 0.07 | −− |
| | **MrandomF.v8** | 0.70 | 0.07 | N/A |
| | Mnnet.v3 | 0.69 | 0.07 | |
| | Mclus.v1 | 0.57 | 0.07 | −− |
| F1 | Msvm.v10 | 0.66 | 0.06 | −− |
| | **MrandomF.v8** | 0.72 | 0.05 | N/A |
| | Mnnet.v3 | 0.68 | 0.05 | −− |
| | Mclus.v1 | 0.61 | 0.06 | −− |
| Accuracy | Msvm.v10 | 0.78 | 0.03 | −− |
| | **MrandomF.v8** | 0.82 | 0.02 | N/A |
| | Mnnet.v3 | 0.77 | 0.03 | −− |
| | Mclus.v1 | 0.76 | 0.03 | −− |

These tests allows us to conclude that the random forests are the best options to predict the topics of the documents and by a statistically significant margin at least for the variants that were considered on this comparative experiment. We should also remark that the values of both precision and recall are interestingly high. In effect, when the best model says a document mentions a certain topic, on average it is correct roughly on 75% of the cases (average precision of 0.75). Moreover, when a document mentions a topic our random forest is able to capture this event on roughly 70% of the cases (recall of 0.70). For a model learned with such few amount of labelled documents this is an interesting performance.

## 5.2 Sentiment Scoring Results

The evaluation for this task is done topic by topic as each topic may have rather different sentiment from the population. Each subsection includes a table with a summary of the results, box-plots of the performance of the models in the experiment and the statistical significance tests.

In terms of performance metric for evaluating the alternative models we have already mentioned that we would use the Total Cost (c.f. Equation 4, page 16) of the model predictions. However, to have a better idea of the value of the obtained scores we will also report the total cost achieved by a naive model (**MmodePred.v1** on the tables) that will predict for all documents the Mode of the sentiment score observed on the training set (i.e. the most frequent score). Using this baseline Total Cost score we also calculate the Relative Cost for each model, which is the ratio of its score over the score of this baseline.

For this predictive task we will have two alternatives (regression and classification) for each of the modelling approaches we have considered. The model names ending with the letter "r" (e.g. **Msvmr.v10**) denote regression approaches, while the others represent the classification techniques.

Regards the results in terms of the statistical significance of the observed differences we should remark that in the case of the used metrics (Total Cost and Relative Cost), having lower values is better. This means that when we have a competitor with the results of the paired comparisons between denoted with plus signals it means that it is worse than the best alternative.

### 5.2.1 Photovoltaic Economic Aspect

The results show in Table 6 indicate that a random forest variant achieved the best score (i.e. lower Total and Relative Costs) for the topic "photovoltaic economic aspect". Figure 9 reinforces this idea by showing the distribution of the results of the different alternatives across the 100 repetitions. Moreover, the outcome of the paired comparisons in Table 7, shows that the difference to the other models is statistically significant with 99% confidence.

Concerning the issue on whether it is better to use a classification or regression approach, the results are not conclusive as they depend on the base models. For random forests classification it is clearly the way to go, but for the other techniques this is not so clear.

It should also be mentioned that when compared to the baseline of always predicting the most frequent sentiment score, most models behave rather badly, with the exception of the random forest using a classification approach.

**Table 6** Best performing models for the photovoltaic economic aspect topic.

| Model | Total Cost | Relative Cost |
|---|---|---|
| Msvm.v10 | 45.81±6.91 | 0.98 |
| Msvmr.v18 | 45.12±3.95 | 0.97 |
| **MrandomF.v2** | **40.59±7.13** | **0.87** |
| MrandomFr.v1 | 47.54±3.31 | 1.02 |
| Mnnet.v3 | 48.42±6.59 | 1.04 |
| Mnnetr.v1 | 47.60±4.61 | 1.02 |
| MmodePred.v1 | 46.52±5.92 | 1.00 |

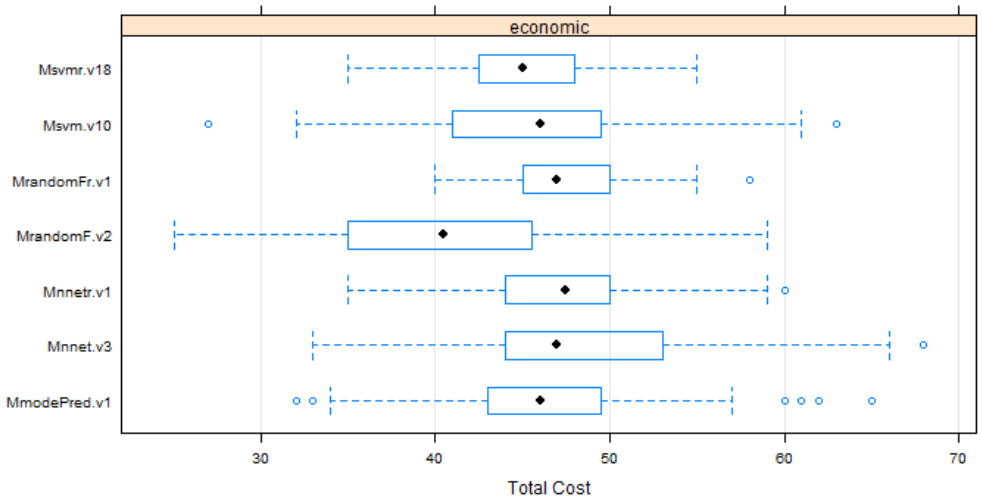


Figure 9: Distribution of the performance of the models in the experiments.

**Table 7** Statistical significance of the observed differences.

| Learner | Average | Standard Deviation | Statistical Significance |
|---|---|---|---|
| Msvm.v10 | 45.81 | 6.91 | ++ |
| Msvmr.v18 | 45.12 | 3.95 | ++ |
| **MrandomF.v2** | 40.59 | 7.13 | N/A |
| MrandomFr.v1 | 47.54 | 3.31 | ++ |
| Mnnet.v3 | 48.42 | 6.59 | ++ |
| Mnnetr.v1 | 47.60 | 4.62 | ++ |
| MmodePred.v1 | 46.52 | 5.92 | ++ |

### 5.2.2 Photovoltaic Environmental Aspect

The total costs obtained in this topic (Table 8) show that the classification approaches of SVMs and random forests are the most competitive, with average scores that are clearly ahead of the other alternatives, and moreover, which clearly overcome the baseline model. The results on Figure 10 confirm this idea and also show that the baseline has a very wide range of scores

compared to the other alternatives. The statistical significance tests on Table 9 confirm the superiority of the above mentioned two models, as well as their similar performance as there is no statistical significance on the difference between their scores.

**Table 8** Best performing models for the photovoltaic environmental aspect topic.

| Model | Total Cost | Relative Cost |
|---|---|---|
| Msvm.v10 | 4.40±1.89 | 0.67 |
| Msvmr.v10 | 5.30±1.45 | 0.81 |
| **MrandomF.v2** | **4.30±1.70** | **0.66** |
| MrandomFr.v1 | 5.36±1.30 | 0.82 |
| Mnnet.v5 | 5.99±1.86 | 0.91 |
| Mnnetr.v3 | 5.83±1.41 | 0.89 |
| MmodePred.v1 | 6.56±4.00 | 1.00 |

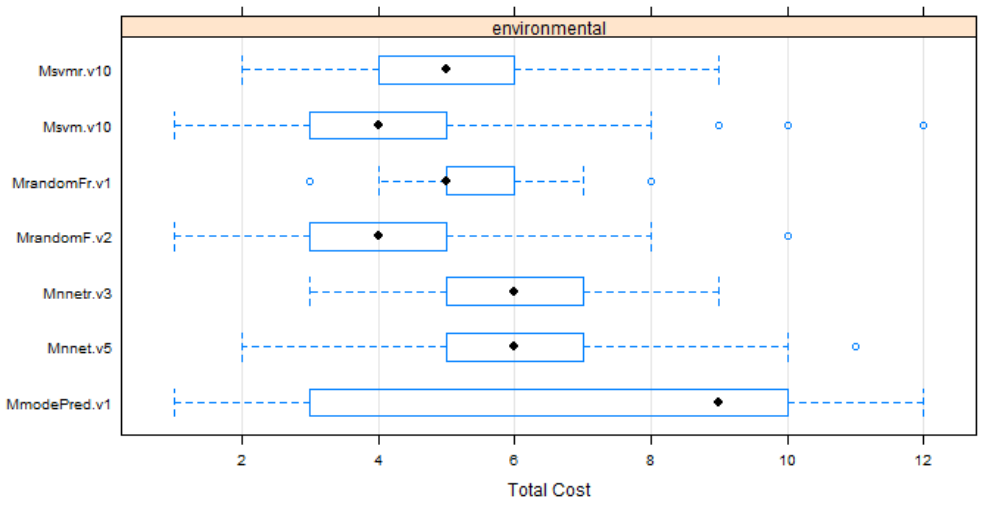


Figure 10: Performance of the models in the experiments.

**Table 9** Statistical significance of the observed differences.

| Learner | Average | Standard Deviation | Statistical Significance |
|---|---|---|---|
| Msvm.v10 | 4.40 | 1.89 | |
| Msvmr.v10 | 5.30 | 1.45 | ++ |
| **MrandomF.v2** | 4.30 | 1.70 | N/A |
| MrandomFr.v1 | 5.36 | 1.30 | ++ |
| Mnnet.v5 | 5.99 | 1.86 | ++ |
| Mnnetr.v3 | 5.83 | 1.41 | ++ |
| MmodePred.v1 | 6.56 | 4.00 | ++ |

### 5.2.3 Photovoltaic Technology Aspect

For the last topic we once again observe a similar performance of both SVMs and random forests when using classification. Still, all models have a rather disappointing performance when compared to the baseline, even the best alternatives. Still, the differences in spite of small, are statistically significant according to the results of Table 11.

**Table 10** Best performing models for the photovoltaic technology aspect topic.

| Model | Total Cost | Relative Cost |
|---|---|---|
| Msvm.v1 | 28.62±4.43 | 0.97 |
| Msvmr.v10 | 30.61±4.57 | 1.04 |
| **MrandomF.v9** | **27.98±4.70** | **0.95** |
| MrandomFr.v9 | 31.73±5.10 | 1.07 |
| Mnnet.v2 | 31.64±5.89 | 1.07 |
| Mnnetr.v3 | 32.49±5.85 | 1.10 |
| MmodePred.v1 | 29.56±4.36 | 1.00 |

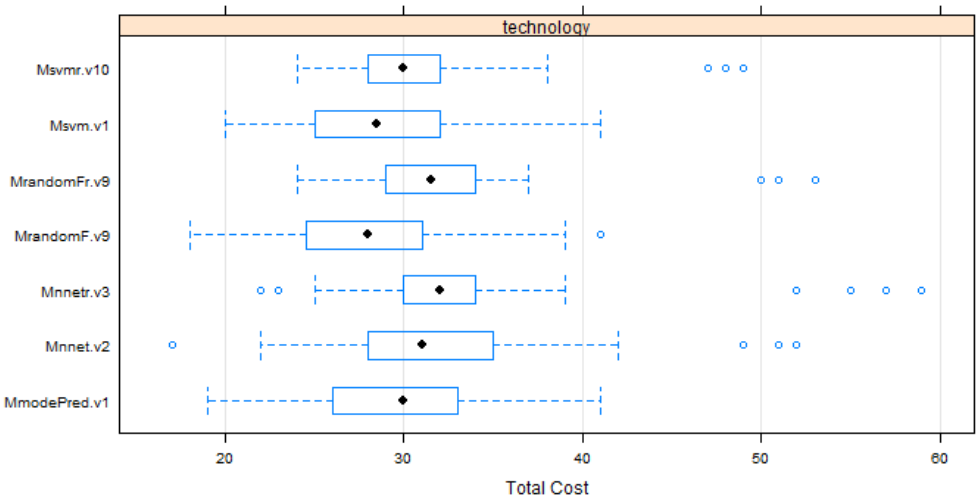


Figure 11: Performance of the models in the experiments.

**Table 11** Statistical significance of the observed differences.

| Learner | Average | Standard Deviation | Statistical Significance |
|---|---|---|---|
| Msvm.v1 | 28.62 | 4.43 | + |
| Msvmr.v10 | 30.61 | 4.57 | ++ |
| **MrandomF.v9** | 27.98 | 4.70 | N/A |
| MrandomFr.v9 | 31.73 | 5.10 | ++ |
| Mnnet.v2 | 31.64 | 5.89 | ++ |
| Mnnetr.v3 | 32.49 | 5.85 | ++ |
| MmodePred.v1 | 29.56 | 4.36 | ++ |

# 6 Conclusions

This deliverable has described the first version of the evaluation of the opinion mining software prototype. This current evaluation will be complemented by Deliverable 6.5 (month 33) where the final evaluation of the full prototype will be presented. In the current deliverable we have focussed on the evaluation of the key components of this prototype - the components that are responsible for identifying which topics are mentioned in the posts and also what is the sentiment of the opinions expressed in those posts.

We have formalized the predictive tasks that are involved in the general task of inferring the sentiment of the population concerning a set of energy-related topics. We have described a series of approaches to these tasks and their implementation in our prototype. We have proposed a series of performance metrics and the associated experimental methodology for the evaluation these components of the OM prototype.

The results of our evaluation indicate that in general our models are able to perform their tasks with success. In effect, the scores in terms of topic identification are interesting even-though space for improvements still exist. In terms of sentiment scoring the results are also interesting but here we see the need for further adjustments / improvements in our models. The obvious way to try to improve our results is to obtain more labelled data, which is a time consuming task requiring large resources of human specialists. Still, we expect to be able to increase the size of the available training set in the near future.

# A Model Variants

In this appendix we describe the variants of the models detailing the parameter values that were used in each variant. The models whose name ends in *r* are the variants in which regression was used to obtain the results.

**Table 12** Random Forests parameters.

| Name | Number of trees | Mtry |
|---|---|---|
| randomF.v1 | 100 | 3 |
| randomF.v2 | 500 | 3 |
| randomF.v3 | 1000 | 3 |
| randomF.v4 | 100 | 5 |
| randomF.v5 | 500 | 5 |
| randomF.v6 | 1000 | 5 |
| randomF.v7 | 100 | 7 |
| randomF.v8 | 500 | 7 |
| randomF.v9 | 1000 | 7 |
| randomFr.v1 | 100 | 3 |
| randomFr.v2 | 500 | 3 |
| randomFr.v3 | 1000 | 3 |
| randomFr.v4 | 100 | 5 |
| randomFr.v5 | 500 | 5 |
| randomFr.v6 | 1000 | 5 |
| randomFr.v7 | 100 | 7 |
| randomFr.v8 | 500 | 7 |
| randomFr.v9 | 1000 | 7 |

**Table 13** Support Vector Machines parameters.

| Name | Cost | Epsilon | Gamma |
|---|---|---|---|
| svm.v1 | 3 | 0.01 | 0.1 |
| svm.v2 | 5 | 0.01 | 0.1 |
| svm.v3 | 7 | 0.01 | 0.1 |
| svm.v4 | 3 | 0.05 | 0.1 |
| svm.v5 | 5 | 0.05 | 0.1 |
| svm.v6 | 7 | 0.05 | 0.1 |
| svm.v7 | 3 | 0.1 | 0.1 |
| svm.v8 | 5 | 0.1 | 0.1 |
| svm.v9 | 7 | 0.1 | 0.1 |
| svm.v10 | 3 | 0.01 | 0.01 |
| svm.v11 | 5 | 0.01 | 0.01 |
| svm.v12 | 7 | 0.01 | 0.01 |
| svm.v13 | 3 | 0.05 | 0.01 |
| svm.v14 | 5 | 0.05 | 0.01 |
| svm.v15 | 7 | 0.05 | 0.01 |
| svm.v16 | 3 | 0.1 | 0.01 |
| svm.v17 | 5 | 0.1 | 0.01 |
| svm.v18 | 7 | 0.1 | 0.01 |
| svmr.v1 | 3 | 0.01 | 0.1 |
| svmr.v2 | 5 | 0.01 | 0.1 |
| svmr.v3 | 7 | 0.01 | 0.1 |
| svmr.v4 | 3 | 0.05 | 0.1 |
| svmr.v5 | 5 | 0.05 | 0.1 |
| svmr.v6 | 7 | 0.05 | 0.1 |
| svmr.v7 | 3 | 0.1 | 0.1 |
| svmr.v8 | 5 | 0.1 | 0.1 |
| svmr.v9 | 7 | 0.1 | 0.1 |
| svmr.v10 | 3 | 0.01 | 0.01 |
| svmr.v11 | 5 | 0.01 | 0.01 |
| svmr.v12 | 7 | 0.01 | 0.01 |
| svmr.v13 | 3 | 0.05 | 0.01 |
| svmr.v14 | 5 | 0.05 | 0.01 |
| svmr.v15 | 7 | 0.05 | 0.01 |
| svmr.v16 | 3 | 0.1 | 0.01 |
| svmr.v17 | 5 | 0.1 | 0.01 |
| svmr.v18 | 7 | 0.1 | 0.01 |

**Table 14** Neural Networks parameters.

| Name | Size | Decay |
|---|---|---|
| nnet.v1 | 3 | 0.1 |
| nnet.v2 | 5 | 0.1 |
| nnet.v3 | 7 | 0.1 |
| nnet.v4 | 3 | 0.01 |
| nnet.v5 | 5 | 0.01 |
| nnet.v6 | 7 | 0.01 |
| nnet.v7 | 3 | 0.05 |
| nnet.v8 | 5 | 0.05 |
| nnet.v9 | 7 | 0.05 |
| nnetr.v1 | 3 | 0.1 |
| nnetr.v2 | 5 | 0.1 |
| nnetr.v3 | 7 | 0.1 |
| nnetr.v4 | 3 | 0.01 |
| nnetr.v5 | 5 | 0.01 |
| nnetr.v6 | 7 | 0.01 |
| nnetr.v7 | 3 | 0.05 |
| nnetr.v8 | 5 | 0.05 |
| nnetr.v9 | 7 | 0.05 |

**Table 15** Clus parameters.

| Name | Ensemble Type |
|---|---|
| clus.v1 | Bagging |
| clus.v2 | Random Forest |
| clus.v3 | Random Subspaces |
| clus.v4 | Bagging of Subspaces |

# References


[1] Clus framework. URL: http://dtai.cs.kuleuven.be/clus/.

[2] C. Banea, R. Mihalcea, and J. Wiebe. Multilingual sentiment and subjectivity analysis. *Multilingual Natural Language Processing*, 2011.

[3] Hendrik Blockeel, Luc De Raedt, and Jan Ramon. Top-down induction of clustering trees. pages 55–63, 1998. URL: http://www.cs.kuleuven.ac.be/cgi-bin-dtai/publ_info.pl?id=20419.

[4] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[6] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[7] Energetica Ambiente Forum. `http://www.energeticambiente.it/index.php`.

[8] Newclear blog. `http://blog.forumnucleare.it/`.

[9] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[10] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[11] David Meyer. Package e1071. *R News*, 2012.

[12] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1:213, 2002.

[13] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

[14] B Yegnanarayana. *Artificial neural networks*. PHI Learning Pvt. Ltd., 2004.