Website Categorization: a Formal Approach and Robustness Analysis in the case of E-commerce Detection

Renato Bruni[a,*], Gianpiero Bianchi[b]

[a] *Dep. of Computer Control and Management Engineering,*
"Sapienza" University of Rome, Rome, Italy, e-mail: bruni@dis.uniroma1.it
[b] *Directorate for Methodology and Statistical Process Design ,*
Italian National Institute of Statistics "Istat", Rome, Italy, e-mail: gianbia@istat.it

Abstract

Website categorization has recently emerged as a very important task in several contexts. A huge amount of information is freely available through websites, and it could be used for example to accomplish statistical surveys, saving in costs. However, the information of interest for the specific categorization has to be mined among that huge amount. This turns out to be a difficult task in practice. In this work we propose a practically viable procedure to perform website categorization, based on the automatic generation of data records summarizing the content of each entire website. This is obtained by using web scraping and optical character recognition, followed by a number of nontrivial text mining and feature engineering steps. When such records have been produced, we use classification algorithms to categorize the websites according to the aspect of interest. We compare in this task Convolutional Neural Networks, Support Vector Machines, Random Forest and Logistic classifiers. Since in many practical cases the training set labels are physiologically noisy, we analyze the robustness of each technique with respect to the presence of misclassified training records. We present results on real-world data for the problem of the detection of websites providing e-commerce facilities, however our approach is not structurally limited to this case.

[*] Corresponding author

Keywords: Classification, Machine Learning, E-commerce, Feature Engineering, Text Mining, Surveys

1. Introduction

Text Mining, or Text Analytics, is the branch of Data Mining concerning the process of extracting high-quality information from texts, see for details Feldman & Sanger (2006); Aggarwal (2018) and references therein. When the text is extracted from the web, we also speak of Web Mining. This area underwent considerable improvements in recent years, with a number of concurrent factors contributing to its progress, first of all the continuous evolution of the Internet and the demand for effective and intelligent search and manipulation strategies. Modern web mining techniques require the integration of natural language processing with machine learning techniques on one side (to provide intelligence), and with image manipulation procedures on the other side (to be able to "read" graphical elements).

A very relevant problem in this field is the classification of text documents. This consists in using a set of text documents, each having a class label, to learn a *classifier*. The classifier is then used to automatically assign the class label to new unlabeled text documents. This task is required in a large variety of practical applications, in particular when the considered text documents are websites. In this case, the task is also called *Website categorization* or *Website classification*, see, e.g., Qi & Davison (2009). The number of web pages available on the Internet has constantly been increasing in the last decades, and nowadays a huge amount of data is freely available through this channel. The sheer size of the problem makes implausible a non-automatic intervention. On the other hand, the automatic extraction of statistical information from this source is extremely appealing, because it would produce large datasets with considerable savings, or it would provide a way to check or integrate datasets that are already available, increasing their quality.

However, website categorization turns out to be a very difficult task in practice, for the following reasons.

- The content of the generic website, consisting of thousands of pages in some cases, should be automatically processed. This is a complex process that requires the use of different Web Scraping and Text Mining phases. Web sites are of course not standardized, part of the information of a website is provided to the human users by means of the graphics rather than the text, etc.

- Effective and intelligent feature engineering techniques are required. Indeed, data obtained via automatic scraping inevitably contain a very large amount of information that simply represents noise for the categorization under analysis. This preponderant noise content should be reduced as much as possible to obtain a satisfactory performance in the classification phase. To this aim, quite articulated procedures had to be developed.

- Even after the noise reduction operations mentioned above, the classification problem itself has a very large dimension: the data records representing the websites generally need to have thousands of fields, and there are thousands of records. It is well known that such dimensionality issues can raise considerably the difficulty of the classification task.

- In typical applications of this task, the class labels of the training set are often inevitably noisy, for several reasons discussed below. Just to make a first example, any stored class label may become outdated whenever a website is changed or renewed, and this can happen at any time. We define *robustness* of a website categorization technique as the property of providing mainly correct output even in presence of a limited amount of errors in the input. Website categorization techniques need to possess some degree of robustness with respect to the presence of noise in the class label, in the sense that results should be influenced scarcely, and not catastrophically, by the presence of errors in the class labels.

In this work, we propose an overall approach to websites categorization capable to cope with the above difficulties. This approach is based on the use of automatic text scraping, image acquisition and processing, optical character recognition, natural language processing and text mining to create data records representing the websites, and on the use of classification algorithms (Support Vector Machines, Convolutional Neural Networks, Random Forest, Logistic classifiers) to perform the categorization. We apply this approach to the specific problem of the detection of websites providing e-commerce facilities, however the approach works at the formal level and thus it is not intrinsically limited to that case. In particular, we want to automatically determine whether the generic website allows to buy, or at least to order, goods or services, or not. This application arise from the Italian version of the European Community Survey on ICT usage and e-commerce in enterprises, an annual survey collecting data about enterprises. Though this determination may appear easily solvable by human inspection, it becomes in fact overwhelming demanding when the number of websites under analysis is large. We evaluate the performance and the robustness of our automatic approach using real-world data.

We obtain very encouraging results, which have been examined in Big Data Committee Report (2018) (Sect. 5.2), an official publication of the Italian National Institute of Statistics (Istat), which shows that the outcome of the proposed methodology is close enough to the outcome of the mentioned Survey on ICT to consider the proposed methodology a valid option. Moreover, the proposed methodology was able to produce the experimental statistics on usage of e-commerce in enterprises published by the Italian National Institute of Statistics (Istat), whose accuracy resulted not lower than those produced by the ICT survey, and which provide at the same time several advantages: reduction of the response burden; reduction of the measurement errors; possibility of more frequent detection (see also Barcaroli et al. (2018)). This is a very important result, because the proposed methodology will allow to make use of the Internet data instead of the traditional survey data in many future detections.

Therefore, the main contributions of this work are: the presentation of a practically viable formal approach to perform automatic categorization of websites; a comparison in this task of four classification algorithms based on different paradigms; an analysis of the robustness of this approach with respect to the presence of errors in the class label and a discussion on possible motivations for the observed robustness; a main step in the exploration of the possibility of using the web source to produce official statistics or other certified data.

Due to the wide span of the techniques used in our work, several works in the literature contain connections on some aspects. An extensive survey on text mining techniques is in Sebastiani (2002), while the specific case of web pages is surveyed in Qi & Davison (2009). The use of information obtained from the web is discussed, among others, in Gök, Waterworth & Shapira (2015). The authors show that web-based indicators offer additional insights when compared with more traditional indicators and, moreover, surveying a subject using web scraping does not allow the alteration in the behavior of the subject in response to being studied.

The case of web page categorization has been studied in a number of works which usually involve the use of feature extraction/selection techniques and classification algorithms (often Support Vector Machines or Deep Convolutional Neural Networks). In more detail, Li & Tian (2008) describe a classification approach based on Support Vector Machines using a two steps feature selection phase. Bhalla & Kumar (2016) describe an efficient categorization of web page contents by using a feature extraction tool based on the HTML document object model of the web page and a support vector machine kernel as the classification tool. Onan (2016) gives a comparative analysis of four different feature selections techniques, four different ensemble learning methods based on four different base learners. Kehagias at al. (2018) propose an automatic categorization of web service elements by means of Logistic Model Trees-based classifier, in conjunction with a data pre-processing technique that reduces the original feature-space dimension without affecting data integrity. López-Sánchez, Arrieta & Corchado (2019) propose a framework for the categorization of web pages on the basis of their visual content by means of a Deep Convolutional Neural Network for feature extraction. The detection of specific features on webpages or websites has been studied in the following works. The Website Key Object Extraction problem has been defined in Velásquez, Dujovne & L'Huillier (2011), proposing a solution based on a Semantic Web mining approach. Cuzzola et al. (2015) describe an approach to the detection and classification of daily deals from the web based on a semi-supervised method using sentence-level features.

The extraction of information from entire websites, more demanding than the case of single web pages, has also been considered. For instance, Thorleuchter & Van Den Poel (2012) predict the e-commerce company success by mining the text of its publicly-accessible website. An important information to detect from an entire website is "phishing", described as the art of echoing a website of a creditable firm intending to grab user's private information. Hadi, Aburub & Alhawari (2016) detect phishing websites using association rule mining and classification, and produces more accurate results than other traditional data mining classification algorithms. Mohammad, Thabtah & McCluskey (2014) assess how good rule-based data mining classification techniques are in predicting phishing websites and which classification technique is more reliable. The specific case of automatic detection of e-commerce availability from web data is described in Barcaroli et al. (2016), which examines the possibility of using information extracted from the web to provide data for ICT related statistical surveys. Blazquez et al. (2016) combine web scraping techniques with learning methods from Big Data, and provide results for a data set of 426 corporate websites of manufacturing firms based in France and Spain. An initial version of this work, using only a subset of the dataset considered here, different techniques and without robustness analysis, is in Bruni, Bianchi & Scalfati (2018).

This work is organized as follows. Section 2 explains the framework of the overall approach and the design of the robustness analysis. Section 3 describes all the operations that have been used to convert the generic website into a data record of reasonable size summarizing that entire website. Section 4 describes the classification algorithms that have been used for our analysis. Section 5 reports the results of experiments by considering the real case of websites belonging to enterprises operating in Italy. Finally, Section 6 draws conclusions.

2. Overview of Our Approach

This section provides an overview of our website categorization approach and the design of our robustness analysis.

The overall website categorization approach is sketched in Fig.1. Given a list of websites, for each of them we extract the text of the pages by means of an automatic scraping procedure. Also, we download from each website images and screen-shots and we process them with optical character recognition (OCR) using the technique developed by Smith (2007) to extract the additional information provided by the graphics. We use this material to prepare very large text files. Subsequently, we extract the training set and we perform several steps in order to identify and select only the part of the above information that is relevant for our categorization. This is done by using natural language processing techniques, such as tokenization, lemmatization and part-of-speech recognition (see,

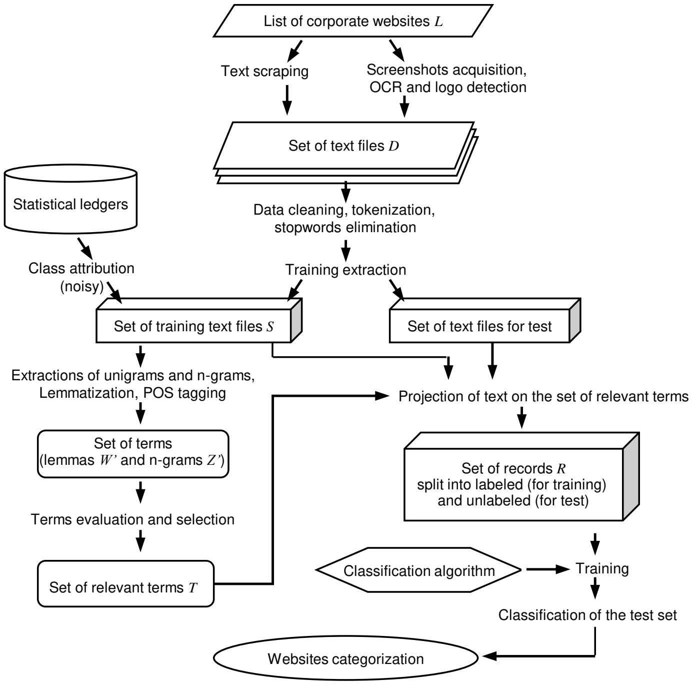
List of corporate websites L
Text scraping
Screenshots acquisition, OCR and logo detection
Set of text files D
Statistical ledgers
Data cleaning, tokenization, stopwords elimination
Class attribution (noisy)
Training extraction
Set of training text files S
Set of text files for test
Extractions of unigrams and n-grams, Lemmatization, POS tagging
Projection of text on the set of relevant terms
Set of terms (lemmas W' and n-grams Z')
Set of records R split into labeled (for training) and unlabeled (for test)
Terms evaluation and selection
Classification algorithm
Training
Set of relevant terms T
Classification of the test set
Websites categorization

Figure 1: The overall website categorization approach.

e.g., Bird, Klein & Loper 2009, Schmid 1995), After this, we apply term evaluation techniques to reduce the dimensions and obtain a set of standardized data records describing the above websites. Finally, we classify the obtained records by means of the following classification algorithms: Support Vector Machines (Chang & Lin 2001), Convolutional Neural Networks (Krizhevsky, Sutskever & Hinton 2012), Random Forest (Breiman 2001), Logistic classifiers Freedman (2009). Each of these classifiers requires to set algorithmic hyperparameters, which greatly affect the result of the classification phase. We formulate the problem of the parameters' choice as follows. We chose the parameters which maximize the harmonic mean of precision and sensitivity of the classification produced. This accuracy measure is called F1-score (Sokolova, Japkowicz & Szpakowicz 2006). To better estimate how the accuracy and the F1-score of the predicted classification will generalize, we introduce a n-fold cross validation scheme in the above described approach. We randomly partition the dataset into n parts of equal size. We extract the training set by taking $n-1$ parts out of the n parts in every possible way, and we predict the class label over the remaining part. Finally, we average the results of accuracy and F1-score over all these trials.

The classification paradigm is clearly based on the availability of a set of labeled records, called training set, that constitute the source of information to obtain a trained classifier. Therefore, to apply the described approach for website categorization, we need a set of websites for which we already have the class labels with respect to the considered categorization. However, the class labels obtainable in practical cases may easily contain errors, due to many reasons. For example, labels may have been assigned to websites by several human operators which follow slightly different labeling criteria, or which may be mistaken, especially on ambiguous cases; labels may have been assigned by automatic procedures which have difficulties on complex cases; labels may come from a stored source (e.g., statistical ledgers) and may simply become outdated whenever a website is renewed, which can happen at any time without notice; etc. Hence, in practical cases, the training set physiologically contains a certain

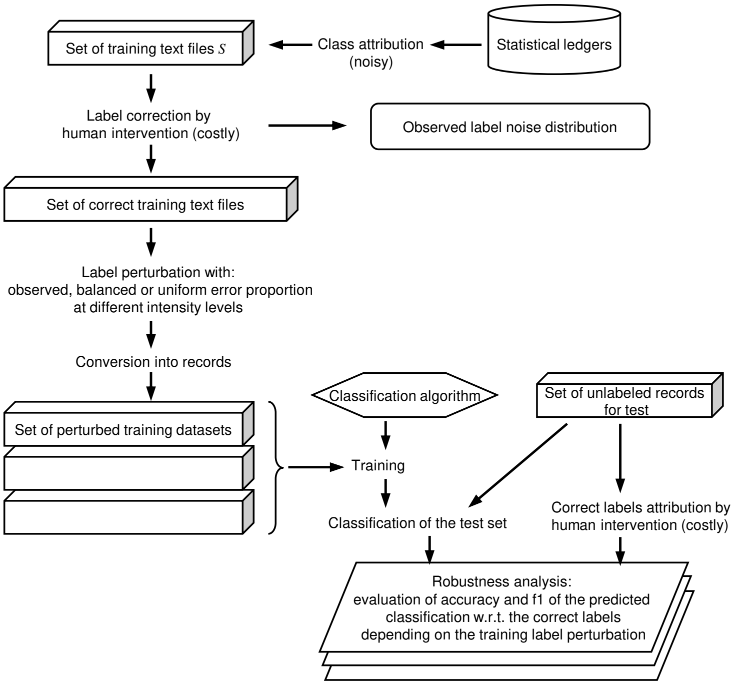
Set of training text files S
Class attribution (noisy)
Statistical ledgers
Label correction by human intervention (costly)
Observed label noise distribution
Set of correct training text files
Label perturbation with: observed, balanced or uniform error proportion at different intensity levels
Conversion into records
Classification algorithm
Set of unlabeled records for test
Set of perturbed training datasets
Training
Classification of the test set
Correct labels attribution by human intervention (costly)
Robustness analysis: evaluation of accuracy and f1 of the predicted classification w.r.t. the correct labels depending on the training label perturbation

Figure 2: Design of the robustness analysis for our website categorization approach.

portion of misclassified records. As a consequence, website categorization techniques should be influenced scarcely, and not catastrophically, by the presence of errors in the class labels. In other words, these techniques should possess some degree of robustness with respect to the presence of noise in the class label (recall that robustness for a software system can be defined as the degree to which a system can function correctly in the presence of invalid inputs or other errors IEEE Std 610.12 (1990)). To analyze the robustness of our categorization procedure with respect to the presence of noise in the class label, we initially had to remove all errors in the class labels by a very costly manual intervention. Then, we artificially perturbed at several intensity levels the class labels of the training set, in order to simulate the presence of noise. We finally applied the whole website categorization procedure at the perturbed datasets, to obtain an extensive analysis of the quality of the categorizations produced (see Fig.2).

3. From the Websites to the Data Records

This section describes all the operations that have been used to convert the generic website into a data record of reasonable size summarizing the website. The description has been particularized to the case studied in the experiments: the detection of e-commerce, which arise from the Italian version of the European Community Survey on ICT usage and e-commerce in enterprises, an annual survey collecting data on the use of ICT, the internet, e-government, e-business and e-commerce in enterprises.

We initially receive a list L of websites that must be categorized. For each $l_i \in L$, we use a web scraping procedure that reads and saves the content of the website, that is the text appearing in its pages. We need to set limits on the amount of data that can be downloaded from a website. Thus, the scraping procedure starts from the homepage and continues with all the other pages reachable from it, up to a certain depth, that can be selected. The underlying idea is that the pages that are too nested are less relevant for the analysis, while they would mainly introduce noise. Also, we use a global maximum for the number of pages that can be downloaded from a single website.

On the other hand, besides the text, for the pages processed by the scraping procedure we also read additional information: the attributes of HTML elements, the name of the image files, the keywords of the pages. Moreover, we perform Optical Character Recognition (OCR) on all the types of images that appear in those pages, in order to read also the words provided as images. In fact, these words are often written with such a special emphasis because they are particularly relevant (consider for instance the case of logos, or 'buy' and 'pay' commands, etc.). Similarly, we take a screen-shot of the homepage and perform OCR on it, too. Indeed, many websites try to catch the eye on the key sentences of the homepage by adopting unusual writing techniques or symbols. OCR is achieved by using the Tesseract open source OCR Engine, initially developed by Hewlett Packard Research before 1995 and subsequently by Google Research since 2006 (Smith 2007). The regions of the images containing text are located by using the Marvin open source image processing framework (Archanjo, Andrijauskas & Munoz 2008).

The above operations produce, for each l_i, a text file d_i. The set of all such text files is D. These text files are very large, each of them may contain more than 10 million words. Unfortunately, the overwhelming majority of this information is simply irrelevant for the categorization. Therefore, we need to identify and select only the part of the above information that is relevant for the categorization required, and to exclude the remaining parts as much as possible.

To perform such a selection, we use the following steps. We initially clean the text by removing all non-alphabetic symbols (e.g. $\%, \&, =$, etc.), and by inserting white spaces to detach the words (tokenization). Then, we remove the stop-words (articles, prepositions, etc.), since their generic meaning has practically no relevance for the categorization task. Subsequently, we identify a training set S composed of 50% of the elements in D. The elements in S should have the class label for the categorization under analysis. As explained above, in practical cases, S may very easily contain misclassified elements. Hence, the class label is often noisy. Moreover, the dataset may be imbalanced, in the sense that the proportions of the records in the different classes are not comparable. For example, in the case of the detection of e-commerce, negative records, i.e., those not providing e-commerce facilities, constitute about 80% of D.

After this, we extract from S two dictionaries: the set W of the uni-grams, i.e., the single words, appearing in those files, and the set Z of the n-grams appearing in the same files. N-grams are sequences of n adjacent words that are typically used together. An example is “credit card”, which is a bi-gram, i.e., has $n = 2$.

For the set of uni-grams W, we perform lemmatization, that is, the inflectional ending of each word is removed in order to return the word to its basic lemma. This operation allows to group together the different inflected forms of a word (e.g., plurals of nouns, tenses of verbs, etc.) so they can be analyzed as a single item. Since we are working with websites of enterprises operating in Italy, this is done for Italian and English language. All words not belonging to these two languages are discarded. Clearly, the choice of the languages can be changed without affecting the fundamental aspects of our approach. Lemmatization is performed by using the software TreeTagger. This tool was initially described in Schmid (1995) and it was subsequently developed by several contributors.

For the set of n-grams Z, we do not perform lemmatization, since substituting words with their basic lemmas may result in losing the identity of many n-grams, which are generally built with specific inflectional forms. On the other hand, for Z we do perform part-of-speech recognition (POS tagging), that is, each word is identified as a particular part of speech (e.g., a noun, a verb, etc.). This is done again in Italian and English language, and again with the software TreeTagger. All words not belonging to these two languages are discarded. Moreover, we discard all n-grams that cannot represent meaningful concepts. For example, in the case of bi-grams, we keep only the pairs that are syntactically well-composed. This means they must be composed by: noun and verb; noun and adjective; noun and adverb; noun and noun; verb and adjective; verb and adverb.

Thus, we obtain at this stage the following two sets of terms:

1. Set W', whose component terms are single lemmas in Italian or English language.
2. Set Z', whose component terms are syntactically well-composed n-grams in Italian or English language.

Now, for each of the terms of W' and Z', we must compute a measure of its relevance for the categorization under analysis by means of a *Term Evaluation* (TE) function. There exist several possible TE functions, representing several metrics; for example Chi Square, Information Gain (also known as Mutual Information), Gain Ratio, etc. (Debole & Sebastiani 2004).

We have selected for our experiments the so-called Chi-square metric (χ^2), since it appears appropriate and it is one of the most frequently used in text categorization. Indeed, χ^2 statistics is employed in many fields to measure how the results of an observation differ from the results expected according to an initial hypothesis (Pearson 1900). In our case, we make the hypothesis of dependence between the generic term $w \in W'$ and the class (positive or negative) of the generic file d containing w, and thus we measure the dependence of w from the class, with lower values corresponding to lower dependence. Since we have two classes, for each $w \in W'$, we compute a score $s(w) = \chi^2_+(w) + \chi^2_-(w)$, where χ^2_+ is called positive score and χ^2_- negative score. The positive score is defined as follows:

$$\chi^2_+ = \frac{p(p_{11}p_{22} - p_{12}p_{21})^2}{(p_{11} + p_{12})(p_{21} + p_{22})(p_{11} + p_{21})(p_{12} + p_{22})}, \tag{1}$$

where p_{11} is the number of occurrences of w in positive files; p_{12} is the total number of occurrences of w; p_{21} is the number of all distinct words occurring in positive files; p_{22} is the total number of all distinct words; and $p = p_{11} + p_{12} + p_{21} + p_{22}$. The negative score is defined similarly, except that p_{11} becomes the number of occurrences of w in negative files and p_{21} the number of all distinct words occurring in negative files.

Similarly, for any n-gram $z \in Z'$, we compute a score $s(z) = \chi^2_+(z) + \chi^2_-(z)$, where χ^2_+ is the positive score and χ^2_- the negative score. These scores are again based on the described Chi-square metric, and the basic idea is now to measure the dependence between the presence of the words constituting the generic $z \in Z'$ and the class of the generic file d containing z. Assuming z is a bi-gram, the positive score is defined as follows:

$$\chi^2_+ = \frac{q(q_{11}q_{22} - q_{12}q_{21})^2}{(q_{11} + q_{12})(q_{21} + q_{22})(q_{11} + q_{21})(q_{12} + q_{22})}, \tag{2}$$

where q_{11} is the number of positive files containing all the words constituting z; q_{12} is the number of positive files containing only the first word of z; q_{21} is the number of positive files containing only the second word of z; q_{22} is the number of positive files not containing any of the words constituting z; and $q = q_{11} + q_{12} + q_{21} + q_{22}$. The negative score is defined similarly, except that all the above values are computed for negative files. In case z has $n \geq 3$, the above formula is consequently expanded. We extract n-grams using all the values of n from 2 to 5. In case an n-gram with n words is fully contained in a larger n-gram with $(n+1), \ldots, 5$ words, we remove the larger n-gram. This because we assume the presence of the shorter n-gram more significant than that of the larger one. We observe that, in our experiment, this practically leads to using mainly bi-grams. However, this reveals not to be a limitation, and it also provides computational advantages.

After this, all terms in W' and in Z' are sorted by decreasing score values, and we finally select among them the terms constituting the relevant information. We take from W' all the terms with a TE score larger than a threshold τ_w and up to a maximum of α_w terms. Similarly, we take from Z' all the terms with a TE score larger than τ_z and up to a maximum of α_z terms. Generally, we set these parameters in order to obtain a set T of about 1000 terms, where the uni-gram terms are about 800 and the n-gram terms are the remaining part. These values were selected as good compromises between accuracy and speed in our experiments. If we select more than 1000 terms, the procedure becomes slower with very modest improvements in accuracy, while, if we select less than 1000 terms, the procedure loses an accuracy not sufficiently compensated by the improvements in speed.

Now, for each $d_i \in D$, we project it on the set T, that is, we reduce d_i to a binary vector r_i of size $|T|$. The h-th element of r_i will be denoted by r_i^h and it is computed as follows:

$$r_i^h = \begin{cases} 1 & \text{if the } h\text{−th term in } T \text{ is present in } d_i \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Vector r_i constitutes the data record summarizing d_i, as anticipated above. The set of all records is R. The class c_i of $r_i \in R$, when available, is

$$c_i = \begin{cases} 1 & \text{if the } i\text{−th website in } L \text{ offers e−commerce} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

4. The Classification Phase

Many different classification approaches have been proposed in the literature, based on different data models and mathematical techniques. It is well known that there is not a single approach capable to outperform all the others on every instance. However, given a specific category of problems, it is empirically possible to identify which approaches generally provide the best performances for that category. For our type of website categorization problems, we have performed preliminary tests with several classifiers and the most promising preliminary results have been obtained with:

- Support Vector Machines (SVM);
- Convolutional Neural Networks (CNN);
- Random Forests (RF);
- Logistic Classifiers (LC).

Hence, we have selected these four classifiers for our full analyses. For each record r_i, the fields r_i^h corresponding to the terms constitute the input or independent variables; the class c_i constitutes the target or dependent variable.

Support Vector Machines. SVMs are supervised learning models that build a deterministic linear classifier. They are based on finding a separating hyperplane that maximizes the margin between the extreme training data of opposite classes. New examples are then mapped into that same space and predicted to belong to a class, on the basis of which side of the hyperplane they fall on. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, by implicitly mapping their inputs to a higher dimensional space, see also Chang & Lin (2001); Vapnik (1995). This classifier requires to set some algorithmic hyperparameters, in particular when using as kernel a Gaussian radial basis function. The main parameters in this case are the penalty parameter c of the error term and the kernel coefficient γ. They greatly affect the result of the classification, In particular, we chose the combination of values $(\bar{c}, \bar{\gamma})$ which maximizes the harmonic mean of precision and sensitivity of the classification produced, called F1-score (Sokolova, Japkowicz & Szpakowicz 2006). By denoting with TP the number of true positive records produced, by TN that of true negatives, by FP that of false positives and by FN that of false negatives, this means:

$$(\bar{c}, \bar{\gamma}) = \arg\max_{(c,\gamma)} \frac{200\ TP}{2TP + FP + FN}. \tag{5}$$

We solve this minimization problem by using a grid search approach, see also Chang & Lin (2001) for details. The tolerance for the stopping criterion in SVM was set at the default value of 0.001; the maximum number of iterations at 1000.

For SVM, RF and LC we use the Python implementations that are available through the functions: SVC(); RandomForestClassifier(); LogisticRegression() in scikit-learn (Pedregosa et al. 2011), which is a machine learning package currently included into scientific Phyton distributions.

Convolutional Neural Networks . An atificial neural network is a collection of connected nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection can transmit a signal from one neuron to another. Neurons are typically organized in layers, and such a network can perform classification tasks by training the neuron input weights to minimize misclassification errors. Convolutional Neural Networks (CNN) are composed of an input and an output layer, and of multiple hidden layers of different types, in particular convolutional layers. They apply a convolution operation to the input, passing the result to the next layer. CNN are designed to process data that come in the form of multiple arrays. This architecture should better detect distinctive local motifs regardless to their position in the multiple array, and for this reason CNN are particularly effective in image recognition. However, the computational requirements are particularly heavy, because the network must be deep, even if the convolution operation reduces the number of free parameters, allowing the network to be deeper with fewer parameters.

We use the Python implementation provided by Bhavesh & Oswal (2016), which requires to set the following algorithmic hyperparameters. The largest number of terms for single record, called “sequence_length”, and the overall total number of terms in our records, called “vocab_size”, are respectively 1,000 and

149,628, as a result of the text mining steps described in the previous Section. The number of classes in the output layer is 2 (positive and negative). The dimensionality of our embeddings is called "embedding_size". For this parameter we use a grid search over the values 100, 200, 300 with the objective of maximizing the above mentioned F1-score. This defines the size of our embedding layer, which has shape [vocabulary_size, embedding_size]. The parameters for the convolutional filters are called "num_filters" and "filter_sizes". We use 512 for the first one, while we grid search over 3, 4, 5 for the second one. This means that we have filters that slide over 3, 4 and 5 words respectively. The fraction of neurons disabled for the droput phase, called "dropout_keep_prob", is set to 0.5 during training, and to 1 (disable dropout) during evaluation. Finally, the number of epochs used is 2, 5 and the batch size is 30.

Note that Bhavesh & Oswal (2016) was explicitly designed to work directly on texts, so in theory it could directly take the websites as input, without the use of our text selection procedures. However, similar attempts failed in practice because the memory demand in this case was overwhelming.

Random forests. Decision trees are a supervised learning model that maps observations about the input variables to conclusions about the target variable. The goal is to create a decision tree that predicts the value of the target variable based on combinations of the values of the input variables. Each internal node is associated with a decision concerning the value of an input variable that best splits the training set. Different algorithms can be used to determine the input variables associated with the internal nodes, see also Loh (2014). This methodology is generally quite effective and computationally light, however it often exhibits a tendency to overfit the training set. This means that the model produced by the classifier may become unnecessarily complex in the attempt to excessively fit the peculiarities of the available data. To overcome similar problems, *Ensemble* techniques have been developed. Those techniques generate many weak learners and combine their outputs in order to obtain a classification that is generally both accurate and robust.

In particular, Random Forest (RF) is an ensemble learning method that operates by generating a multitude of decision trees. The global output is obtained by computing the mode of the outputs of the individual trees. Additional details can be found in Ho (1998); Breiman (2001). Random forests are generally more robust and can achieve better performances than the single decision trees, and can be extended to process very large datasets (see, e.g., Genuer et al. (2017)). For this reason, we use such a version of the decision tree methodology in our experiments. The number of trees used in our experiments has been set at 500.

Logistic classifiers. Logistic regression is a regression model where the target variable is categorical; hence, it can be used to perform a classification. This approach is called Logistic Classifier (LC). It measures the relationship between the target variable and one or more independent variables by estimating the probabilities using a logistic function, which is the cumulative logistic distribution, see also Freedman (2009). Logistic regression can be seen as a special case of the generalized linear model and thus analogous to linear regression. This approach is often used in practice because it is computationally light and it possesses a fair power of generalization.

5. Experimental Results

We apply the described procedure to a sample of the Italian version of the European Community Survey on ICT usage and e-commerce in enterprises, an annual survey collecting data on the use of ICT, the internet, e-government, e-business and e-commerce in enterprises. The survey covers the universe of enterprises with 10 or more employees, for a total number of about 184,000 enterprises in 2017. The list L contains 4,755 websites chosen randomly in order to be a representative sample for the full survey. The number of words downloaded for each website goes up to 3,000,000. Each item in L originally had a class value, although, as observed, this information may be noisy. By applying the techniques described in Section 3, we produce a set R of 4,755 records. Each $r_i \in R$ has 1000 fields, 800 obtained from uni-grams and 200 from n-grams, and the class label c_i. A record of this type is positive if the corresponding website offers e-commerce facilities, and it is negative otherwise.

Since only about 20% of the entries are positive, the dataset is very imbalanced. This increases the difficulty of the classification phase. Indeed, it is very easy to reach an 80% of classification accuracy by simply predicting all records as negative. However, this result would be completely useless from the practical point of view. In fact, obtaining the correct identification of the positive records constitutes the main goal in this type of problems, and this identification is particularly challenging.

To study the robustness, it was necessary to first remove the noise in the class label. Therefore, the class labels have been interactively checked by human intervention. This operation led to reverse about 8% of positive records into negative ones, and about 3.33% of negative records into positive ones. The fact that the class noise is concentrated more on positive records has been frequently observed on our instances of the problem. After this, we obtain a set of records R' with correct class labels.

To perform the classification task, we select in R' a training set S of 2,377 records, that is 50% of the total dataset. The extraction have been randomly performed 3 times, and all performance results are averaged on the 3 trials. To tackle the issue of imbalance in the training set (see also He & Ma (2013)), we operate as follows. First, we perform a partial resampling by randomly undersampling the majority class and oversampling the minority class (by replication), until obtaining a dataset of the same size but containing roughly 40% positive entries. Then, we adjust the misclassification costs (computed during the training phase) by using weights inversely proportional to the class frequencies in the resampled training set.

Given the above training set S, we apply to it a random perturbation in the class label, in order to reintroduce class noise but at a controlled level. This is done at eight different levels of intensity and with the three following proportions in the distribution of the class errors:

1. *Observed error proportion.* We introduce errors in the class using an error proportion 2.4 times larger for positive records than for negative ones, as it was the error observed in our dataset.
2. *Balanced error proportion.* We introduce exactly the same number of class errors in reach class, independently of the size of each class.
3. *Uniform error proportion.* We introduce errors in the class using an error proportion that reflects the size of each class, so that the frequency of errors is the same all over the dataset. In our case, roughly 20% of the entries are positive, and the remaining 80% of the entries are negative. Therefore, the uniform error proportion has a number of errors on negative records that is about 4 times larger.

We obtain 8 x 3 = 24 versions of the training set, denoted by S_h^k. Index $h = 1, \ldots, 8$ represents the perturbation level, from 0% (not perturbed at all) to 21% (a strong perturbation); index $k = 1, \ldots, 3$ represents the perturbation model: *observed, balanced, uniform.* The actual number of training records whose class have been changed to obtain each S_h^k is reported in Table 1 below. Note that the total number of positive records in S is only 432; hence some of the highest perturbation levels reverse the class of more than half of the positive training records. This means that the information provided to the classifier to be able to predict the positive class has been strongly altered.

After this, we perform the training phase of the four classifiers (RF, SVM, LC, CNN) described in Section 3 for each of the above 24 training sets S_h^k. The objective in the training phase is the maximization of F1, as shown in equation (5). This is obtained for RF and LC in less than 5 minutes of computation for each training set S_h^k. SVM and CNN classifiers, on the other hand, requires a much more elaborate training phase. In their case, we perform a grid search to find the best parameters, requiring respectively about 20 and 16 minutes in average for each training set S_h^k and using 3-fold cross-validation. Note that the use of a grid search approach is standard in practical applications, and even though it cannot theoretically guarantee to determine exactly the optimal parameters, it is generally regarded as a very reasonable compromise between time and performance.

Table 1: Number of perturbed training records in each perturbation scheme.

	Observed		Balanced		Uniform	
Total Perturbation	pos.	neg.	pos.	neg.	pos.	neg.
3%	50	21	36	36	13	58
6%	101	42	71	71	26	117
9%	151	63	107	107	39	175
12%	201	84	143	143	51	234
15%	252	105	178	178	62	292
18%	302	126	214	214	77	351
21%	352	147	250	250	90	409

When the training phase is completed, we use the learned classifiers to predict the class for all the records in the test sets $T = R' - S$. Subsequently, by knowing the real class of each $r_i \in T$, we compare it to the predicted class so we can compute the confusion matrix corresponding to each classifier and each training set S_h^k. The elements of each confusion matrix (true positives TP, false negatives FN, true negatives TN, false positives FP) are then used to evaluate the following performance measures:

- Accuracy a, defined as the percentage of correct predictions over all predictions:
$$a = \frac{100(TP + TN)}{TP + FN + TN + FP}.$$
- Precision p, also called the positive predictive value, defined as the percentage of true positive records in all positive predictions: $p = \dfrac{100\;TP}{TP + FP}$.
- Sensitivity s, also called the true positive rate, defined as the percentage of correct positive predictions in all real positive records: $s = \dfrac{100\;TP}{TP + FN}$.

- F1-score, which is the harmonic mean of the above described measures of precision and sensitivity:

$$F_1 = \frac{200\,TP}{2TP + FP + FN}.$$

Note that, for the detection of e-commerce, the latter one appears the most relevant performance measure, since it fully evaluates the correct identification of the positive records, that is the most important and difficult task. Therefore, in our experiments, besides the basic measure of accuracy, we consider the F1-score.

Table 2 compares the results of the three described classifiers using the correct training set S_1^1 and all the training sets perturbed with the observed error proportion $S_2^1 \dots S_8^1$. Table 3 compares the same information but using the training sets perturbed with the balanced error proportion $S_2^2 \dots S_8^2$. Finally, Table 4 compares the same information but using the training sets perturbed with the uniform error proportion $S_2^3 \dots S_8^3$.

Table 2: Results obtained when perturbing with the observed proportion.

	RF		SVM		LC		CNN	
Perturb. level	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
0%	90.45 %	73.51 %	88.81 %	70.31 %	87.76 %	65.86 %	86.20 %	67.05 %
3%	90.66 %	73.25 %	87.94 %	68.80 %	87.35 %	63.90 %	84.52 %	60.25 %
6%	90.15 %	70.97 %	86.63 %	64.68 %	87.30 %	62.37 %	84.06 %	58.75 %
9%	90.2 %	70.39 %	84.63 %	59.68 %	86.96 %	60.46 %	84.11 %	56.80 %
12%	89.91 %	68.50 %	82.93 %	56.21 %	87.59 %	60.32 %	84.00 %	56.05 %
15%	86.92 %	57.92 %	76.79 %	53.77 %	86.50 %	56.44 %	83.36 %	50.88 %
18%	84.57 %	48.96 %	71.86 %	49.15 %	85.87 %	52.94 %	82.90 %	49.50 %
21%	84.74 %	47.47 %	70.67 %	47.31 %	85.79 %	51.01 %	81.44 %	46.86 %

To better study the decrease of the classification performance at the increase of the noise in the class label, we also report Table 5, containing the performance

Table 3: Results obtained using balanced perturbation.

	RF		SVM		LC		CNN	
Perturb. level	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
0%	90.45 %	73.51 %	88.81 %	70.31 %	87.76 %	65.86 %	87.60 %	67.80 %
3%	90.19 %	72.33 %	88.18 %	68.95 %	87.34 %	64.71 %	84.75 %	60.45 %
6%	89.86 %	71.75 %	85.75 %	65.83 %	86.63 %	63.70 %	84.10 %	59.33 %
9%	89.49 %	70.79 %	86.54 %	64.04 %	86.03 %	62.10 %	83.70 %	57.15 %
12%	88.77 %	68.77 %	85.67 %	57.64 %	86.03 %	60.43 %	83.33 %	56.65 %
15%	85.79 %	60.42 %	80.12 %	55.04 %	85.37 %	60.27 %	83.15 %	56.03 %
18%	84.82 %	57.88 %	78.20 %	48.89 %	86.96 %	56.70 %	83.00 %	54.95 %
21%	80.91 %	47.96 %	75.15 %	47.08 %	84.18 %	55.97 %	82.35 %	53.56 %

Table 4: Results obtained using uniform perturbation.

	RF		SVM		LC		CNN	
Perturb. level	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
0%	90.45 %	73.11 %	88.81 %	70.31 %	87.76 %	65.86 %	86.35 %	67.00 %
3%	89.66 %	72.67 %	86.24 %	68.20 %	85.70 %	63.98 %	83.68 %	59.77 %
6%	89.02 %	72.38 %	85.83 %	65.44 %	85.37 %	63.04 %	82.50 %	59.20 %
9%	87.13 %	69.09 %	76.32 %	62.57 %	84.61 %	63.83 %	81.56 %	58.95 %
12%	86.99 %	68.71 %	73.43 %	58.69 %	83.05 %	62.07 %	81.50 %	58.82 %
15%	83.60 %	64.09 %	73.30 %	57.07 %	82.42 %	60.87 %	79.05 %	57.50 %
18%	80.07 %	57.99 %	69.28 %	49.69 %	79.48 %	57.97 %	78.93 %	57.34 %
21%	79.39 %	56.33 %	68.61 %	49.69 %	78.85 %	57.50 %	77.25 %	56.78 %

degradation corresponding to each perturbation level averaged over the 3 perturbation models. This performance degradation is computed as the reduction in accuracy and F1-score with respect to the values obtained when the class label contain no errors.

Table 5: Decrease (negative values) of the classification performance at the increase of the noise in the class label.

	RF		SVM		LC		CNN	
Perturb. level	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
3%	-0.28 %	-0.63 %	-1.36 %	-1.66 %	-0.96 %	-1.66 %	-2.40 %	-7.13 %
6%	-0.77 %	-1.68 %	-2.74 %	-4.99 %	-1.33 %	-2.82 %	-3.16 %	-8.19 %
9%	-1.51 %	-3.29 %	-6.31 %	-8.21 %	-1.89 %	-3.73 %	-3.59 %	-9.65 %
12%	-1.89 %	-4.72 %	-8.13 %	-12.80 %	-2.20 %	-4.92 %	-3.77 %	-10.11 %
15%	-5.01 %	-12.57 %	-12.07 %	-15.02 %	-3.00 %	-6.67 %	-4.86 %	-12.48 %
18%	-7.30 %	-18.43 %	-15.70 %	-21.07 %	-3.66 %	-9.99 %	-5.11 %	-13.35 %
21%	-8.77 %	-22.79 %	-17.33 %	-22.28 %	-4.82 %	-11.03 %	-6.37 %	-14.88 %

By analyzing the above results, we observe what follows.

1. The classification performance obviously degrades by increasing the perturbation level. However, this degradation is not so marked as it could be expected. Indeed, up to a perturbation level of 15%, the degradation is generally less than proportional to the perturbation introduced. Also, for some cases of Tables 2, 3, 4, one step of increase in the perturbation level does not worsen the classification performance. In other words, the whole procedure appears to possess a certain degree of robustness with respect to the presence of class errors in the training set. We hypothesize that this robustness is due to the use of the automatic classification algorithms. Indeed, it is well known that the capability of a classifier to generalize what is learned from the training set is strictly correlated with its ability to search for a “simple” model of the data, according to the so-called Occam’s razor principle, see also Domingos (1999). In our ex-

periments, by reversing the class of some training records, we are actually providing a limited amount of wrong information, carried by the records that have been perturbed, mixed with the correct information carried by the unperturbed records. The simplification ability of the automatic classifiers allows to override this amount of wrong information, at least until it remains a minoritary part of the whole information provided.

2. The robustness described above appears more marked for the "deep" classifier CNN. The degradation in the performance is indeed particularly slow for this technique. This is due to its recognized generalization ability even in very difficult cases of the classification problem, and the most perturbed datasets are indeed very difficult cases of the classification problem. This holds because, in such a binary classification, the noise they contain actually corresponds to information completely contradicting the correct information carried by unperturbed records. The performance degradation appears not marked also for LC, but this is due to the fact that its performance at zero perturbation is often the lower.
3. Data perturbed with the uniform model produce the best F1 performance, while those perturbed with the observed model produce the worst F1 performance. This holds because, for each given perturbation level, the observed model is the one that corrupts the largest amount of positive records, so their correct detection becomes more difficult. On the contrary, the uniform model is the one that corrupts the smallest amount of positive records, so the effect is the opposite of the former case.
4. Despite the intrinsic robustness observed in the classification approach, we have to note that there exists a kind of threshold effect, in particular for the "non-deep" classifiers. Indeed, when the perturbation level goes beyond 12 %, the performance degrades more sharply, since the amount of wrong information becomes consistent and it starts to cause a sort of "avalanche effect". This especially holds for the dataset perturbed with the observed model, for the reasons specified in the previous observation.
5. Random Forest classifier (RF) generally provides the best performances

in our experiments. However, the results of the other classifiers are not considerably worse, and in any case the trend is very similar. Hence, the overall classification results can be considered quite aligned. As known, in practical applications, any generic dataset has its inherent “level of difficulty”, in the sense that in many cases the classification accuracy cannot be pushed up to 100%, no matter which classifier and parameters’ combination is used. In other words, if we operate a bad choice for classifier/parameters, we may be able to worsen the results at will, but, on the other extreme, there exist a sort of upper limit in the classification performance that is obtainable on a dataset. Coming back to our case, the fact that the classification results are quite aligned allows us to deem that, after the careful parameters’ optimization and the learning phase, the performance of the classification produced by our classifiers is not far from the upper limit in the performance obtainable by a generic automatic classification strategy on the considered dataset.

To provide further insight on the robustness of the procedure, we also report the graphs of the decrease of the performance obtained when increasing the perturbation level. In particular, Fig. 3 reports the analysis of the accuracy of RF classifier; Fig. 4 reports the analysis of the F1 score of the same classifier; Fig. 5 reports the analysis of the accuracy of SVM classifier; Fig. 6 reports the analysis of the F1 score of the same classifier; Fig. 7 reports the analysis of the accuracy of LC; Fig. 8 reports the analysis of the F1 score of the same classifier; Fig. 9 reports the analysis of the accuracy of CNN; Fig. 10 reports the analysis of the F1 score of the same classifier. These figures allow to fully observe the evolution of the degradation in the classification performance, confirming the observations reported above.

The computational times and the memory usage of the whole procedures are quite reasonable, considering also the large size of the problem. In particular, using a PC with i7 processor and 16GB RAM, we experienced what follows. Given the set of text files D, the text mining operations which produce the set of records R and the whole classification phase, running all the three classifiers in sequence, require about 50 minutes in total. On the other hand, the generation of the set D from the initial list of website L by means of web scraping and OCR procedures is much more time consuming, requiring several hours; however this part is intended to be performed offline, and it can also be completely parallelized on several machines.

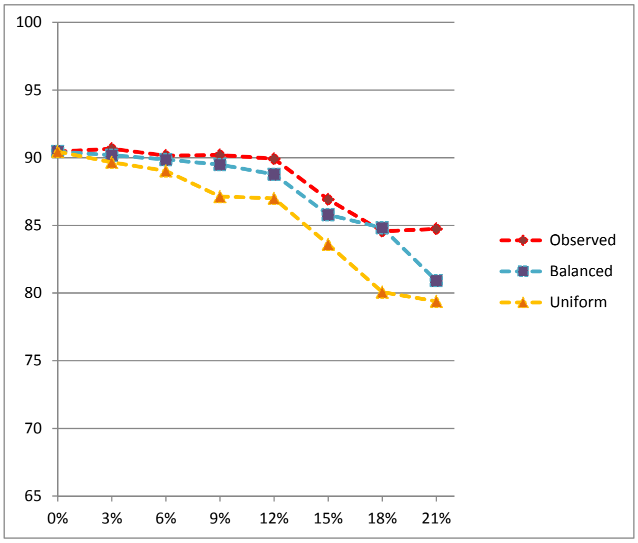

Figure 3: Accuracy of Random Forest classifier for different perturbations of the training set.

6. Conclusions

Website categorization has recently emerged as a very important task in several contexts, because it allows the automatic individuation of some feature of interest by solving a classification problem, for which very effective algorithms are nowadays available. Determining whether an enterprise website offers e-commerce facilities or not is a particularly interesting practical case of this problem, which is useful for example to obtain from the web the information that is currently been collected by an annual survey of the European Community on ICT related aspects in the enterprises. However, the importance of website

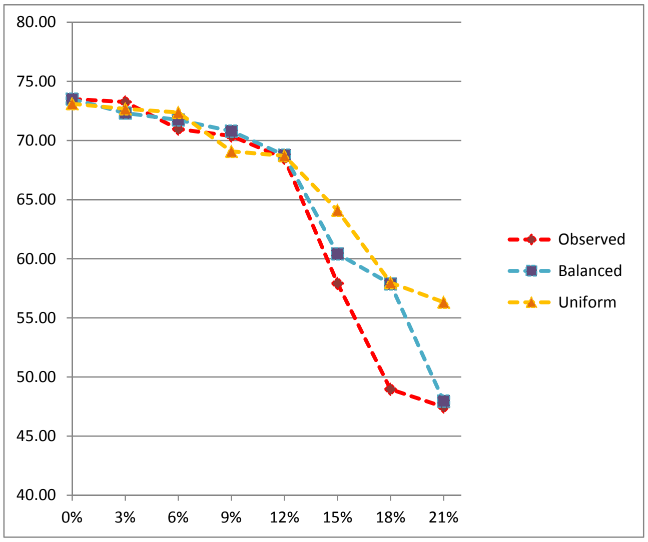

Figure 4: F1 score of Random Forest classifier for different perturbations of the training set.

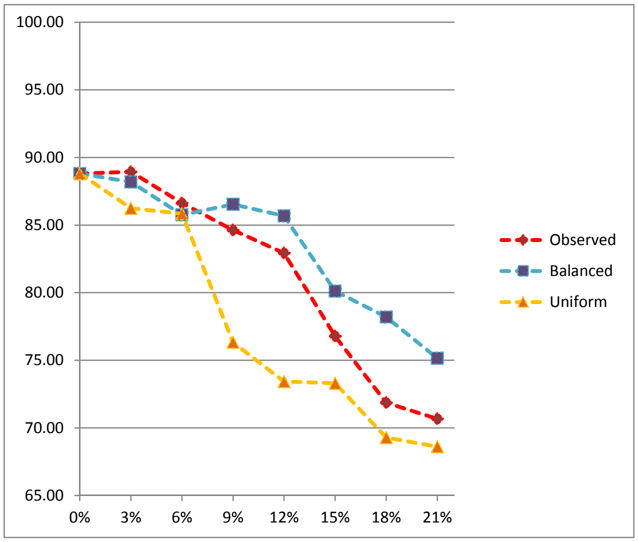

Figure 5: Accuracy of Support Vector Machines classifier for different perturbations of the training set.

645 categorization is not limited to this specific case. On the contrary, besides a number of other statistical surverys that could profit from website categorization, this operation would be useful in many different contexts for the study of

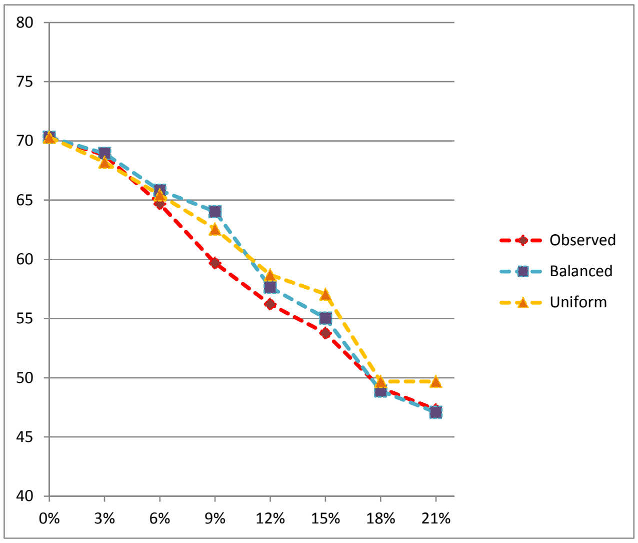

Figure 6: F1 score of Support Vector Machines classifier for different perturbations of the training set.

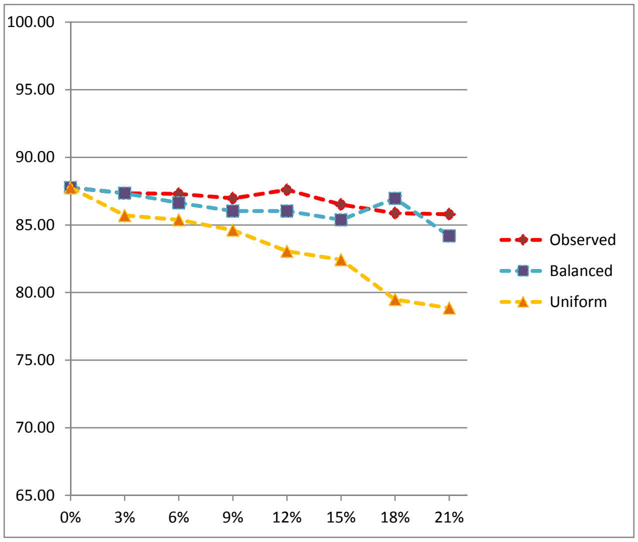

Figure 7: Accuracy of Logistic classifier for different perturbations of the training set.

organizations having websites (e.g., Universities, Public administrations, etc.).

Website categorization is however a difficult task. To use a classification algorithm, it is necessary to convert each website into a record describing the

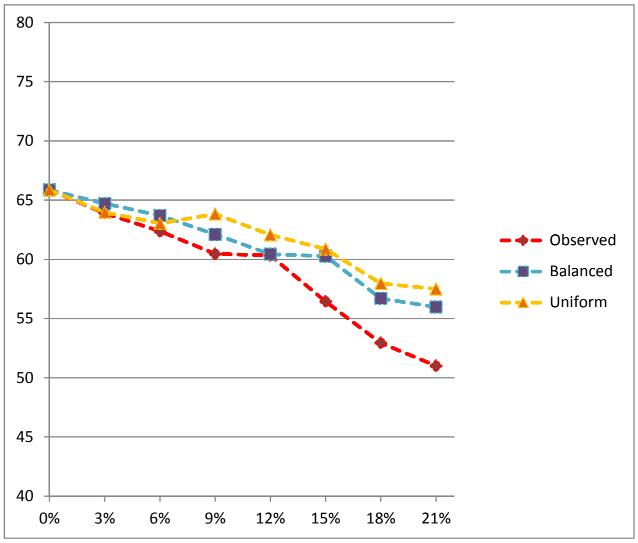

Figure 8: F1 score of Logistic classifier for different perturbations of the training set.

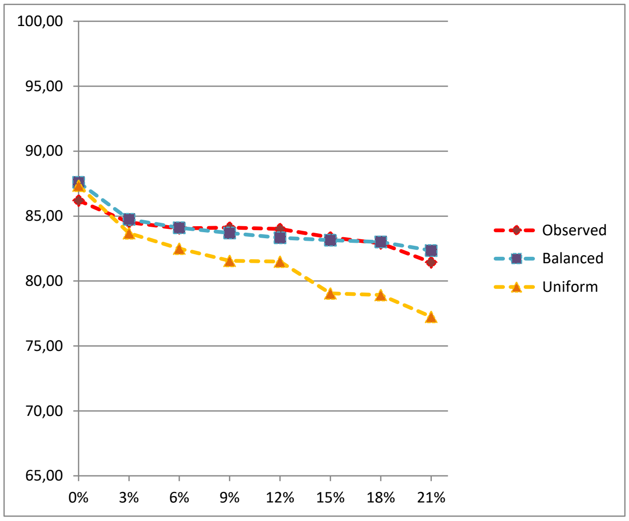

Figure 9: Accuracy of CNN classifier for different perturbations of the training set.

website in a compact and tractable manner. These records should contain only the relevant portion of the information contained in the websites, and this selection requires several text mining and feature engineering operations. On the other hand, we found that the use of classification algorithms provides also a

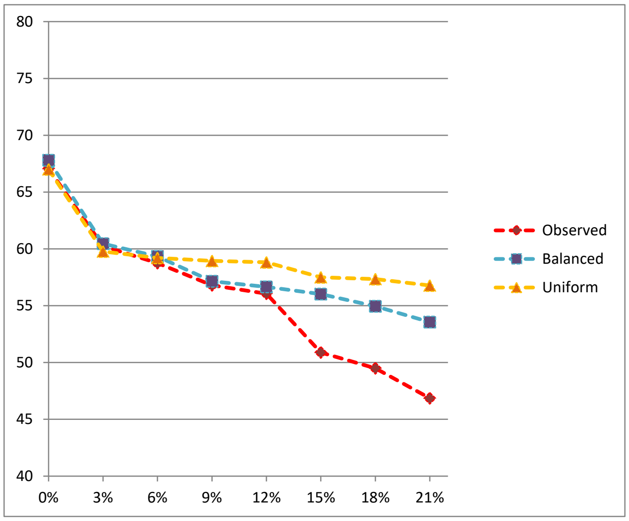

Figure 10: F1 score of CNN classifier for different perturbations of the training set.

certain degree of robustness with respect to the presence of errors in the class labels of the training set. Indeed, searching for a parsimonious model of the data, according to the so-called Occam's razor principle (Domingos 1999), provides on the positive side a generalization capability that results in the above mentioned robustness. This feature is very useful in practice, because in similar cases the class label physiologically contain some errors. Our experiments show that the overall procedure presented in this work constitute a practically viable technique that is able to perform automatic categorization of websites with a satisfactory degree of accuracy and robustness.

One additional important quality of the proposed procedure is that it works at the formal level, hence it could be adapted to solve different website categorizations problems arising in other contexts. The relevant terms are automatically extracted on the basis of their distribution in the text files, not by manually specifying them for the specific context. The knowledge of the language is provided by means of dictionary files, so they can be easily changed to work in different languages. Future work includes the application of this procedure to integrate National statistical ledgers on several other aspects of the enterprises. Another work currently undergoing by using the same procedure is the analysis of Universities by using their websites (see Bianchi et al. 2019).

References

Aggarwal C. C. (2018) Machine Learning for Text. Springer.

Archanjo G.A., Andrijauskas F., Munoz D. (2008) Marvin–A Tool for Image Processing Algorithm Development. Technical Posters Proceedings of XXI Brazilian Symposium of Computer Graphics and Image Processing.

Barcaroli G., Bianchi G., Bruni R., Nurra A., Salamone S., Scarnò M. (2016) Machine learning and statistical inference: the case of Istat survey on ICT. Proceeding of 48th scientific meeting of the Italian Statistical Society SIS 2016, Salerno, Italy. Editors: Pratesi M. and Pena C. ISBN: 9788861970618

Barcaroli G., Golini N., Righi P. (2018) Quality evaluation of experimental statistics produced by making use of Big Data. Proceeding of Q2018, European Conference on Quality in Official Statistics, Krakow, Poland.

Bhalla, V.K., Kumar, N. (2016) An efficient scheme for automatic web pages categorization using the support vector machine. New Review of Hypermedia and Multimedia 22, 223-242.

Bhavesh V. Oswal (2016) CNN-text-classification-keras, GitHub repository, `https://github.com/bhaveshoswal/CNN-text-classification-keras`

Bianchi G., Bruni R., Laureti Palma A., Perani G., Scalfati F. (2019) The corporate identity of Italian Universities on the Web: a webometrics approach. 17th International Conference on Scientometrics & Informetrics, 2019.

Big Data Committee (2018) Annual Report. Italian National Institute of Statistics (Istat), Rome, Italy. ISBN 978-88-458-1962-9

Bird S., Klein E., Loper E. (2009) Natural Language Processing with Python. OReilly Media.

Blazquez D., Domenech J., Gil J., Pont A. (2016) Automatic detection of e-commerce availability from web data. Proceedings of First International Conference on Advanced Research Methods and Analytics (CARMA2016), València, Spain.

Breiman L. (2001) Random Forests. Machine Learning. 45(1), 5–32.

Bruni R., Bianchi G. (2015) Effective Classification using Binarization and Statistical Analysis. IEEE Transactions on Knowledge and Data Engineering 27(9), 2349-2361.

Bruni R., Bianchi G., Scalfati F. (2018) Identifying e-Commerce in Enterprises by means of Text Mining and Classification algorithms. Mathematical Problems in Engineering Vol 2018, n. 7231920, 2018.

Chang C.-C., Lin C.-J. (2001) Training ν-support vector classifiers: Theory and algorithms. Neural Computation, 13(9), 2119-2147.

Cuzzola J., Jovanović J., Bagheri E., Gasević, D. (2015) Automated classification and localization of daily deal content from the Web. Applied Soft Computing Journal 31, 241-256.

Debole F., Sebastiani F. (2004) Supervised Term Weighting for Automated Text Categorization. In: Sirmakessis S. (ed) Text Mining and its Applications. Studies in Fuzziness and Soft Computing, vol 138. Springer, Berlin.

Domingos P. (1999) The role of Occam's razor in knowledge discovery. Data Mining and Knowledge Discovery 3, 409-425.

Feldman R., Sanger J. (2006) The Text Mining Handbook. Cambridge University Press.

Freedman D.A. (2009) Statistical Models: Theory and Practice. Cambridge University Press.

Genuer R., Poggi J.-M., Tuleau-Malot C., Villa-Vialaneix N. (2017) Random Forests for Big Data. Big Data Research 9, 28-46.

Gök, A., Waterworth, A., Shapira, P. (2015). Use of web mining in studying innovation. Scientometrics, 102(1), 653–671.

Hadi, W., Aburub, F., Alhawari, S. (2016) A new fast associative classification algorithm for detecting phishing websites(Article) Applied Soft Computing Journal 48(1), 729-734.

Hastie T., Tibshirani R., Friedman J. (2009) The Elements of Statistical Learning. 2nd ed., Springer.

He H., Ma Y. (eds) (2013) Imbalanced Learning: Foundations, Algorithms, and Applications IEEE Press.

Ho T.K. (1998) The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence. 20(8), 832–844.

IEEE Standard Glossary of Software Engineering Terminology, in IEEE Std 610.12-1990, pp. 1-84. doi: 10.1109/IEEESTD.1990.101064

Kehagias, D., Mavridou, E., Giannoutakis, K.M., Tzovaras, D., Hassapis, G. (2018) Automatic categorization of Web service elements. International Journal of Web Information Systems 14(2) 233-258.

Krizhevsky A., Sutskever I., Hinton G.E. (2012) Imagenet classification with deep convolutional neural networks. In Proc. Advances in Neural Information Processing Systems 25, 1090–1098.

Lam W., Ruiz M., Srinivasan P. (1999) Automatic text categorization and its application to text retrieval. IEEE Transactions on Knowledge and Data Engineering 11(6), 865-879.

Li, X., Tian, X. (2008) Two steps features selection and support vector machines for web page text categorization. Journal of Computational Information Systems 4(1), 133-138.

Loh W.-Y. (2014) Fifty years of classification and regression trees, International Statistical Review, 82(3), 329-348.

López-Sánchez, D., Arrieta, A.G., Corchado, J.M. (2019) Visual content-based web page categorization with deep transfer learning and metric learning. Neurocomputing 338, 418-431.

Mohammad, R.M., Thabtah, F., McCluskey, L. (2014) Intelligent rule-based phishing websites classification. IET Information Security 8(3), 153-160.

Onan A. (2016) Classifier and feature set ensembles for web page classification. Journal of Information Science 42(2), 150-165.

Pearson K. (1900) On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Philosophical Magazine 50(302) 157-175. DOI: 10.1080/14786440009463897

Pedregosa F. et al. (2011) Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830.

Qi X., Davison B.D. (2009) Web page classification: Features and algorithms. ACM Computing Surveys Vol 41(2), article 12.

Rehurek R., Sojka P. (2010) Software Framework for Topic Modelling with Large Corpora. Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Malta.

Schmid H. (1995) Improvements in Part-of-Speech Tagging with an Application to German. Proceedings of the ACL SIGDAT-Workshop. Dublin, Ireland.

Sebastiani F. (2002) Machine Learning in Automated Text Categorization. ACM Computing Surveys Vol 34(1), 1-47.

Smith R. (2007) An Overview of the Tesseract OCR Engine. in Proc. of the Ninth International Conference on Document Analysis and Recognition, 629-633, 2007 ISBN:0-7695-2822-8 IEEE Computer Society, Washington, USA.

Sokolova M., Japkowicz N., Szpakowicz S. (2006) Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation. In: Sattar A., Kang B. (eds) AI 2006: Advances in Artificial Intelligence. Lecture Notes in Computer Science, vol 4304. Springer, Berlin, Heidelberg

Thorleuchter D., Van Den Poel D. (2012) Predicting e-commerce company success by mining the text of its publicly-accessible website. Expert Systems with Applications 39(17), 13026-13034.

Vapnik V. (1995) *The Nature of Statistical Learning Theory*, second edition. Springer.

Velásquez J.D., Dujovne L.E., L'Huillier G. (2011) Extracting significant Website Key Objects: A Semantic Web mining approach. Engineering Applications of Artificial Intelligence, 24(8), 1532-1541.