Implementing Web Classification for TLDs

Luca Deri [1], Maurizio Martinelli [1], Daniele Sartiano [2], Michela Serrecchia [1], Loredana Sideri [1], Sonia Prignoli [1]
IIT/CNR, Pisa, Italy [1]
Computer Science Department, University of Pisa, Italy [2]

{name.surname}@iit.cnr.it, sartiano@di.unipi.it

***Abstract*—On the market there are many commercial web classification services and a few publicly available web directory services. Unfortunately they mostly focus on English-speaking web sites, making them unsuitable for other languages in terms of classification reliability and coverage.**

This paper covers the design and implementation of a web-based classification tool for TLDs (Top Level Domain). Each domain is classified by analysing the main domain web site, and organised it in categories according to its content. The tool has been successfully validated by classifying all the registered .it Internet domains, whose results are presented in this paper.

Keywords—Internet Domain, Web-Content Classification, HTTP crawling, Web Mining, SVM.

I. Introduction and Motivation

Web classification is the method of classifying a website main content or topic according to a set of defined categories. Leading companies provide classification services to their customers either by means of a database that customers included on their products, or as a cloud service. In addition to commercial companies [1, 2], there are also publicly accessible web directories such as the popular Open Directory Project [3] that is the largest web directory fully maintained by a community of volunteers editors. In ODP, web sites are organised in categories that are further divided into subcategories. Even though ODP is a pretty large directory (it contains more than 4 million entries) in practice it has several limitations as it is not updated too often, many entries are outdated, and it is mostly focusing on web sites written in English language with limited coverage of other languages (e.g. there are only 162k classified Italian sites). Commercial web classification services cover many languages and countries but they have the same limitation of ODP: popular web sites accessed daily by million of users are classified properly, whereas not so popular web sites are often not classified or placed in the wrong category. This fact has been validated by the authors who purchased a classification service provided by two leading companies active on this market, classifying 1’000 .it web sites selected randomly, and comparing the results with a manual annotation. Company B has better accuracy than company A when classifying two popular categories, but over 50% of the domains in the test-set are unknown or unclassified. The outcome of this test has shown that these services are excellent for popular web sites but have various limitations when classifying non popular web sites. Instead when these services classify popular .it web sites, they are very reliable and accurate. The same behaviour can be observed analysing the results provided by commercial web analytics services such as alexa.com that misclassify non-popular .it web sites by placing them in a wrong category.

	Company A	Company B
Unknown Domain	20%	56%
Unrated Domain	27%	14%
Detection Accuracy Category Food	36%	60%
Accuracy Detection Category Hotel	0%	67%

1. Evaluation of two leading web content classification services over a test-set of 1,000 .it domain names

The authors of this paper work for the Italian .it Internet domain Registry (Registro.it) ccTLD, and thus focus mostly on the Italian-speaking community. Currently there are more than 2.8 millions of .it Internet domains that have been registered by Italian and non Italian subjects. If present, the main domain web site (i.e. www.<domain name>.it) is often written in Italian as well other official languages (German and French), even tough many sites provide also an English version, and a few are written in a different language. From our experiments with commercial web classification tools, we have realised that using them to classify the .it registered domains would not have been wise for various reasons:

- Table 1 shows that commercial web classification services for non-English languages are not optimal.
- Classification categories are not homogeneous and often they are either too specific or too broad.
- Publicly available directories such as ODP cover less than 10% of .it registered domain names.
- Even under the strong limitations of commercial tools classification, using them for periodically classifying the .it domains would have been very expensive (in terms of service cost to pay) and without any result guarantee as companies do not disclose how their classification service works, what is the classification accuracy and how often they scan a domain for content.

For the above reasons we have decided to create a web classification tool able to characterise .it registered domains by classifying their main domain web site. The idea is to create a directory for .it sites classified according to an identified set of categories. Goal of this work is not to develop yet another web classification tool and position it according to the state of the art. Instead what is novel in this paper, is to fully classify a ccTLD (country code Top Level Domain) using a home-grown tool that is royalty free, accurate in classification, small in space (i.e. we do not need to extract TBs of data to classify the whole .it), able to operate on non-English web sites, and able to periodically update the categorised sites. As this year Italy hosts the universal exposition Expo 2015, we have decided to focus on the classification of the agrifood industry as it appears from the registered .it domain names, leaving the classification of non-food web sites to the second part of the project.

II. Related Work

Web classification has been a hot research topic for a decade, as it enables focused crawling, improves the web search, and it is the cornerstone of contextual advertising, as well web analysis. It exploits many methods and techniques developed for text classification, even though it differs from it in a few aspects [5]:

- Unlike documents and books, web collections do not have a list of structured documents.
- Web pages are semi-structured documents that are linked through anchors.

In [6] the authors proposed a web page classifier that uses features [12] extracted through web page summarisation algorithms. In [7] the authors used a directed graph to represents the topological structure of the website, in which they extracted a strongly connected sub-graph and then applied a page rank algorithm to select topic-relevant resources. Other approaches extracts context features from neighbouring web pages, for example anchor of the link, and the surround headings [8]. Most methods used to classify web content rely on support vector machines (SVM). A SVM [9] is a supervised learning method that performs discriminative classification. The algorithm implements classifications by exploiting a training set of labeled data. Formally the SVM constructs the optimal hyperplane under the condition of linearly separable. SVMs are very popular in text and web classification [10] due to the good results that can be achieved using them.

III. Web Domain Classification: Design and Implementation

Web classification is an activity divided in two distinct steps: web page retrieval and page content classification. As previously stated, one of the goals of this project is to create a web classification tool able to scale to million of sites, and thus implement a classification process that requires just a few web pages to correctly classify a site. For this reason we have designed our system to require just a few pages from a site in order to classify the site. In order to validate classification results limiting human intervention, we have decided to develop two different classifiers that can both exploit the same retrieved web data (i.e. we do not want to crawl the same web site twice). The first classifier is based on SVMs that are a solid technology used in this field, and the other is based on a novel approach we developed. The idea is to select the best algorithm for our needs and use the other one as validation tool.

The rest of this section covers the tool used for downloading web pages, and the design principles as well implementation details of the two web classifiers.

A. Web Crawling

A web crawler (or spider) is an application that downloads web site content. Crawlers download web pages, parse its content in order to extract hyperlinks, and recursively visits them until a limit is reached (i.e. a maximum number of pages is downloaded). Even though there are many open-source crawlers available on the market, we have decided to develop our own able to satisfy our requirements that include (but are not limited to):

- Automatically discard non relevant pages such as “Contacts”, “Impressum”, “Legal”, “Links” that are not helping in categorisation and might confuse the method.
- Recognise parked and under-construction web sites so they can be discarded immediately without any further processing.
- Detect splash messages and landing pages, so that the crawler can follow the correct hyperlinks without analysing content not meaningful to analyse.
- Visit first hyperlinks internal to the site we’re crawling, then those that are external, starting first from sub-domains (e.g. www.subdomain.domain.it) and then all the others. In essence we prefer to go deep in the site being crawled rather than jumping on hyperlinks that point to external sites.
- Create an index of the downloaded pages, and parse them by generating an additional file that contains only the textual part of the web page. This choice allows applications that access the page, to avoid parsing the page one more time and access web page content without paying attention to the HTML markup.

We have developed the crawler in C using the cURL library for downloading web content, and libXML2 for parsing the retrieved page, extract textual content including meta-tags, and getting the list of hyperlinks to follow. While the page is downloaded, the crawler parses the page in memory and creates on disk a single text file per domain containing the text extracted from each individual page. Such file contains the textual part of the pages as well the text of selected meta-tags as earlier described on this section. Using a 100 Mbit Internet connection and a low-end server, it is possible to crawl all the main sites of the .it registered domains (limiting the download to 10 web pages per site), save their content on disk, and parse the HTML, in less than a day. Removing the limitation of one thread visiting one physical host at a time, could dramatically reduce the download time but like previously explained this limitation is compulsory and thus it cannot be removed.

B. Probabilistic Web Page Classification

The first method we developed is based on probabilistic web page classification [11]. The whole idea behind this method is the following: if site X belongs to category Y, then the site X must contain several words that are relevant for Y mixed with a few others that are not relevant and thus can be discarded. The creation of relevant/non-relevant word dictionaries has been done manually in order to fine tune the process, more than what an automated system (in theory) could do. Dictionaries for all the categories have been created as follows:

- First we have defined 13 categories (12 agrifood plus non-agrifood) that, as previously explained earlier in this paper, will initially focus only on agrifood, and as follow-up work on non-agrifood.
- We have extracted the list of all registered .it domains (~2.8 million) from the domains database.
- Eight people have manually classified about 4,000 .it web site domains including agrifood and non-agrifood web sites. Having classified the same domain by multiple people should prevent from misclassifying domains.
- Exploiting the text file generated by the crawler for each valid .it domain, we have coded a tool in python that reads all the words contained in such file, lemmatise them using some existing dictionaries (Italian, English, French and German as they are the official languages in Italy) of the Tanl pipeline [12], and computes the term frequency–inverse document frequency (TF-IDF). Stop-words are automatically discarded.
- Using the result of the previous step, we have manually created two dictionaries including the words we considered relevant and those that are not. Very relevant (e.g. salami)/ irrelevant (e.g. sex) words are marked with a sign to give them a higher/negative score in the categorisation process.

For each domain web site, the probabilistic classifier takes as input the text extracted from the crawler and complements it with the split domain name. For instance the domain name freshalohe.it is split into fresh and alohe. This information is added to the web page content downloaded by the crawler. As sometimes Internet domain names have nothing to do with their real content, we do not base the classifier just on the domain name but we merge the split words with the rest of the site content. The classification process is straightforward: all the domain words are stored on three different hash tables (one for very/relevant, another for not-very/relevant and another for other) where each key is the matching word and the value is the number of occurrences found. The classifier assigns a domain to a category by counting the number of matching words and matching word occurrences in each hash, and then decides based on the results found. In essence a domain is assigned to a category if a) there are enough positive words found, b) positive words (both in occurrence and number) are more than double of the negative words c) very negative words are less than a threshold and less than half of the very positive words. In other words a match between a domain and a category is found when there are enough matches found, and negative words are very few and much less than positive words both it terms of number and occurrence.

C.SVM-based Web Page Classification

The SVM-based classifier is based on the popular libSVM[1]. Instead of using the page text generated by the crawler, this classifies parses the HTML page, extracts the text according to the features described below on this section by selecting the relevant HTML tags, converts the text to lower-case and tokenise it using the NLTK[2] library. As in the former classifier, words are lemmatised, and stop-words discarded. The features used by the classifier take into account the structure of the web page by interpreting HTML tags accordingly. Extracted words are grouped into clusters of similar words using word2vec, a tool that a) implements the continuous bag-of-words and skip-gram architectures for computing vector representations of words, and b) applies the k-means algorithm for computing the word clusters. Using the Italian wikipedia, we have obtained 800 word clusters. In order to represent the context web page, we extracted the following features for each web page:

- HTML TITLE, IMG, and META tags. In the latter case we consider only attributes a) name, b) keywords, c) description and d) classification, as well e) property only restricted to title and description.
- HTML tag A: extract the tag text only if the HREF attribute is not pointing to an external site.
- The web page domain name is tokenised for computing all the possible n-grams of length 4 or longer that are contained in the OpenOffice dictionary. Internationalised domain names (IDN) are ignored.
- HTML BODY: we extract and tokenise all the text contained in the BODY tag.
- Positive and negative list of words according to the dictionaries used by the former classifier.
- Word cluster: for each word extracted in the HTML BODY tag, a word is used as feature only if such word is contained in one of the above word clusters.

In SVMs it is crucial to select the features used for classification. In order to find the optimal setup, we ran several tests on a set of 10,000 domains with all the possible features combination with a cutoff of 1 and 10, tested with/without stop-word removal. The best configuration we obtained has an accuracy of over 97%, and it uses word clusters, HTML meta and title tags, domain name split, positive and negative dictionary.

IV. Experiments

The experiments have been performed on a test-set of 5,600 domains randomly selected from the list of .it registered domains and manually classified. Each domain has been classified by at least two persons to reach an agreement on the domain category. During this project we have learnt that not all registered domains have an active web site: about 5% of the domains have a parking web page, and about 25% do not have a web site at all. The classification outcome is evaluated using the standard metrics precision, recall and F1 [13]. The precision is a metric that highlights how much the prediction is correct, whereas the recall indicates what portion of the classified data has been correctly identified. High precision gives and idea of the correctness of the results, whereas the recall highlights how much data has been

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[2] http://www.nltk.org

correctly classified. The F1 score measures the whole accuracy in terms of precision and recall, and thus it is the indicator of how good is a given classifier. The following table highlights the results of the two classifiers when classifying agrifood vs non-agrifood.

Classifier	Precision	Recall	F1
Probabilistic	91,4%	91,4%	91,4%
SVM	91,0%	84,0%	88,0%
Union	88,7%	93,9%	90,9%
Intersection	94,3%	77,5%	85%

2. Classification results evaluation for agrifood classification.

The probabilistic classifier outperforms the SVM classifier in both precision and recall, featuring a score well above 90% thus making it quite an excellent tool [14]. We have also evaluated how to combine the two approaches together in order to improve the results. With no surprise the union has a better recall but worse precision with respect to the probabilistic method, and the opposite for the intersection. However in terms of F1 the union and intersection of results do not improve the probabilistic classifier, that still outperforms both of them. The probabilistic classifier produces better results than the one based on SVM, probably because it is based on a fine-tuned manual word selection that is more accurate than an automatic system.

Classifier	Precision	Recall	F1
Agriculture	85,8%	75,2%	80,1%
Wine	84,5%	90,2%	87,3%
(Olive) Oil	79,1%	88,8%	83,7%
Breeding	52,2%	95,1%	67,5%
Farmhouse	86,4%	95,8%	90,8%
Pasta and Bread	61,5%	85,2%	71,4%
Fishing and Aquaculture	77,7%	58,3%	66,6%
Meat Curing	80,0%	90,1%	84,8%
Dairy Foods	90,5%	82,6%	86,4%
Agriculture (Other)	72,2%	59,1%	65,0%
Beverages (no wine)	86,6%	95,4%	90,8%
Restaurant and Catering	56,2%	73,8%	63,8%
Overall	75,7%	85,2%	80,2%

3. Classification results evaluation for agrifood with probabilistic classifier.

In addition, for some categories we have very few classified domains that make the SVM prediction inaccurate whereas a human can still identify the keywords of such category. On the other hand the probabilistic classifier requires some manual tuning made by language and field experts, whereas for the SVM it is sufficient to manually assign a domain to a category letting the system automatically select the words to use in the classification based on the specified features. Seen the above results, we have selected the tool based on the probabilistic approach to split agrifood web sites according to the 12 categories we defined. A domain matches the expected categories when the tool places it into the same categories that human classifiers used. When a domain overlaps multiple categories, we have decided to place it into the most relevant one. The previous table highlights the classification results using the probabilistic approach. The overall F1 score is over 80% that is considered as a good result in literature. It is encouraging to note that in case of incorrect results, the classifier has not associated a totally unrelated category to the domain, but has rather failed to identity the most relevant category in the set of identified categories. Considered that web sites do not always report in their text content the same information they represent with pictures, we believe that the tool we developed has produced outstanding classification results.

V. Final Remarks

This paper has covered the design and implementation of a web classification system focusing on .it web sites. The whole idea has been to create a classification system able to permanently classify a large number of continuously changing web sites. The system has been used in the context of the Universal Exposition Expo 2015 to classify the agribusiness sites active in the .it ccTLD, and divide them into sub-categories. While the system is operational since some months, we plan to extend it to non agrifood, and thus classify the whole .it domain database.

References

1. BrightCloud Inc., BrightCloud Web Classification Service, http://www.brightcloud.com/pdf/BCSS-WCS-DS-us-021814-F.pdf
2. SimilarWeb Inc, Our Data & Methodology, http://www.similarweb.com/downloads/our-data-methodology.pdf, 2014.
3. AOL Inc, Open Directory Project (ODP), http://dmoz.org.
4. Y. Zhang Zhang, N. Zincir-Heywood, and E. Milios, Summarizing web sites automatically, Proc. of AI'03, 2003.
5. Q. Xiaoguang, and B. D. Davison, Web page classification: Features and algorithms, ACM Computing Surveys (CSUR) 41.2 (2009):12.
6. S. Dou, et al., Web-page classification through summarization, Proc. of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004.
7. Z. Ji-bin, et al., A Web Site Classification Approach Based On Its Topological Structure, Int. J. of Asian Lang. Proc. 20.2 (2010):75-86.
8. G. Attardi, A. Gulli, and F. Sebastiani, Automatic Web page categorization by link and context analysis, Proc. of THAI. Vol. 99. No. 99. 1999.
9. V. Vapnik, Probabilistic learning theory. Adaptive and learning systems for signal processing, communications, and control, John Wiley & Sons, 1998.
10. X. Weimin, et al., Web Page Classification Based on SVM, Proc. of WCICA 2006, IEEE, 2006.
11. V.F. Fernandez, et al., Naive Bayes Web Page Classification with HTML Mark-Up Enrichment, Proc. of ICCGI '06, Aug. 2006.
12. G. Attardi, S. Dei Rossi, and M. Simi, The tanl pipeline, Proc. of Workshop on Web Services and Processing Pipelines in HLT, co-located LREC. 2010.
13. D. M. Powers, Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation (Tech. Rep.)., Journal of Machine Learning Technologies 2 (1): 37–63, 2011.
14. C. Goutte and E. Gaussier, A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation, Proc. of ECIR '05, 2005.