Large Scale Web-Content Classification

Luca Deri [1], Maurizio Martinelli [1], Daniele Sartiano [2], Loredana Sideri [1]

IIT/CNR, Via Moruzzi 1, 56124 Pisa, Italy [1]
Computer Science Department, University of Pisa, Largo B. Pontecorvo 3, 56127 Pisa, Italy [2]
{luca.deri, maurizio.martinelli, loredana.sideri}@iit.cnr.it, {sartiano}@di.unipi.it

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

Abstract: Web classification is used in many security devices for preventing users to access selected web sites that are not allowed by the current security policy, as well for improving web search and for implementing contextual advertising. There are many commercial web classification services available on the market 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 classifying 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.

1 INTRODUCTION

Web classification is the method of classifying a website main content or topic according to a set of defined categories. Classification of web sites is an activity on which many security tools rely on. Anti-spam, parental controls, web security, anti-phishing and URL filters could not exist without classification services that can assist network software to make the right decisions. 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. The former has the advantage of guaranteeing low response time at the price of being slightly outdated, the latter is always updated with the drawback of the network latency to reach the service provider. In addition to commercial companies (BrightCloud, 2014; SimilarWeb, 2014; Blocksi, 2014; zvelo, 2014), there are also publicly accessible web directories such as the popular Open Directory Project (AOL, 2015) 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. ODP features some regional categories that include web sites written in a specific language (e.g. German) or coming from a geographical region (e.g. France). 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).

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

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

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 site are often not classified or placed in the wrong category. This fact has been validated by the authors who purchased a classification service offered by two leading companies active on this market, classifying 1’000 .it web sites selected randomly, and comparing the results with manual manually 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 on a wrong category.

Another aspect to consider when doing web classifications, is that the web site content is not stable over-time.

Table 2: Content changes on a random set of 2’000 .it web sites, over 60 a days time period.

Content Changes	Web Sites %
Exactly Same Content	56,1%
Almost (< 10% changes)	29,8%
Major Differences in Content	12,6%
Domain Registration Change	1,5%

While it is uncommon that a domain changes category over-time unless it is transfer to a new owner, domain content can change more often. This requires the development an automatic tool able to be run periodically as what is properly classified today might be outdated in a few weeks.

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 2.7 million .it Internet domains that have been registered by Italian and non Italian entities. If present, the main domain web site (i.e. www.<domain name>.it) is often written in Italian and other official languages (German and French), even tough many sites have 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 registered .it 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 registered .it 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 the main domain web site. The idea is to create a directory for .it sites classified according to an identified set of categories. While it is usually safe to assume that a domain contains homogeneous information (e.g. if www.xxx.it contains information about food, it is unlikely that yyy.xxx.it contains non-food information), we do not want to make this assumption. Instead, once we have classified the main domain web site, we associate this information with the registrant and thus categorise the registrants interests. As Registro.it has the list of all registered .it domain names, the result of this work is the creation of a comprehensive directory of .it sites as well a classification of registrants according to the categories we identified.

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 on 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. The result of our work could ignite the definition of follow-up research projects using the result of this work, as well the creation of a directory, far larger than ODP, based on web content for .it domains. It is worth to remark that even though this work has been triggered by our need to classify .it domain web sites, its scope it is broad and the tools and solutions identified throughout this work can be used in other contexts. In particular, as this year Italy will host the universal exposition[1], we have decided to focus on the classification of agrifood industry as it appears from the registered .it domain names.

The rest of the paper is structured as follow. Section 2 describes the related work and highlight the state of the art in web classification tools. Section 3 covers the design and implementation of the web classification system. Section 4 describes the validation process and experiments, section highlight

[1] Expo 2015, http://expo2015.org

some future work activities, and finally section 6 concludes the paper.

2 RELATED WORK

Web classification has been a hot research topic for a decade (Zhang Zhang 2003; Dumais, 2000), as it enables focused crawling (Soumen, 1999), improves the web search (Chandra, 1997), and it is the cornerstone of contextual advertising (Jung-Jin, 2009) as well web analysis. It exploits many methods and techniques developed for text classification, even though it differs from it in a few aspects (Xiaoguang, 2009):

- Unlike in 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 (Dou, 2004) the authors proposed a web page classifier that uses features extracted through web page summarisation algorithms. PEBL (Hwanjo, 2002) is a semi-supervised learning method that uses only positive examples for reducing the training set. In (Jin-bin, 2010) 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 (Soumen, 1998; Attardi, 1999).

Most methods used to classify web content rely on support vector machines (SVM). A SVM (Vapnik, 1998; Sun, 2002) is a supervised learning method that performs discriminative classification. The algorithm implements classifications by exploiting a training set of labelled data. Formally the SVM constructs the optimal hyperplane under the condition of linear separable. SVMs are very popular in text (Joachims, 1998) and web classification (Sun, 1002; Hwanjo, 2002; Weimin, 2006) due to the good results that can be achieved using them.

3 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 (up to 10 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). If both classifiers would be perfect, then the classified results would overlap. In practice as shown in literature, classification accuracy above 80% is considered a good result. This means that there are tenth of thousand of sites (when categorising almost 3 million sites) that would fall into two different categories when classified by the two methods. As shown in table 3, the intersection of results produced by both classifiers increases the accuracy thus we have a high confidence to have been classified correctly, while restricting human analysis to those sites that do not belong to the same category.

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.

3.1 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). There are several open-source crawlers available such as HTTrack and Apache Nutch (Marill, 2004), but none of them we tested was flexible enough for the project, as we required:

- Ability to crawl up to a certain number of pages starting from the main page, discarding non HTML pages. The download limit per site should be per page and not per URL depth (as most crawlers do) as this might require a larger number of pages to be downloaded.
- Automatically discard non relevant pages such as "Contacts", "Impressum" , "Legal" 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 wasting time with pages that do not have meaningful content 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. This practice is necessary to avoid following resources not local the site when there are local hyperlinks to visit. 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 including relevant tags such as the meta tags keywords, description and content. 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.
- Before downloading a URL, the crawler must resolve the symbolic IP address to a numerical IP, and make sure that the same physical host is not receiving too many simultaneous requests. This feature is necessary to avoid HTTP servers from banning the crawler when downloading pages of different domain names hosted on the same physical host.

The crawler we have developed satisfies all the above requirements. It is written in C and it uses 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. The application is logically divided in threads of execution, each downloading a URL. The redis key/value database is used to store the list of domains to crawl, as well the list of hyperlinks that have been extracted by the pages so far retrieved and not yet visited. In order to avoid sending too many request to the same physical host, when a hyperlink has to be visited, it is placed on a different redis queue whose queueId is computed as 'hash(numerical IP of the hyperlink) % number of concurrent threads'. This algorithm guarantees that only one thread at time visits pages served by the same IP address. Downloaded pages are saved on disk in text format on a name hierarchy; this is in order to avoid placing all the files on the same directory. 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. Domains without a web site registered, landing pages or parked sites, are detected by searching specific sentences in the HTML (e.g. "web site under construction") and do not trigger the generation of any textual domain file.

Using a 100 Mbit Internet connection and a low-end server, it is possible to crawl all the main sites of the 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. During this crawler development we have learnt that not all registered domains have an active web site: about 5% of the registered domains have a parking web page, and about 25% do not have a web site at all.

3.2 Probabilistic Web Page Classification

The first method we developed is based on probabilistic web page classification (Fernandez, 2006; Vinu, 2011). 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 the categories, that as previously explained earlier on this paper, will initially focus only on agrifood, and then classify them into the various agrifood categories.
- In order to select domains that are more likely to be in the agrifood business (and thus ease the creation of dictionaries), we have selected a set of words such as "pizza" and "drink" and extracted domain names containing those words. In addition to this we merged them with other randomly selected domain names from the complete .it domain list. This is because for each category we have to define a positive dictionary (words that belong to a given category), and a negative dictionary (words that should not belong to the category), and a "other" dictionary (words that can relevant but too generic such as "product" or "industry"). The need of a negative dictionary is justified by the need to discard information that is close to what we are looking for but not enough. Example in order to distinguish agritourism from hotel or real-estate, we need to make sure that the web site contains terms related to the agribusiness (e.g. wine-tasting, or oil production) but not terms like mortgage, valet parking, or congress centre.
- Exploiting the text file generated by the crawler for each valid .it domain, we have written a python tool 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 (Attardi, 2010), and computes the term frequency–inverse document frequency (TF-IDF) (Rajaraman, 2011). Stop-words are automatically discarded.

- Using the result of the previous step, we have manually created the dictionaries by including the words we considered more relevant. Very relevant (e.g. salami)/irrelevant (e.g. sex) words are marked with a plus sign to give them a higher 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. The algorithm used is pretty simple: using a dictionary whose words are sorted by length, we find the longest word included in the domain name. When a match is found the matching word is removed from the domain name, and then we find the next match until the string has zero length or no match is found. In order to support overlapping domain words (e.g. areaperta.it to be split in area and aperta), when the matching word is removed a one char padding, before/after the matching word, is left on the domain word. The classification process is straightforward: all the domain words are stored on three different hash tables (one for relevant, another for not-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.

3.3 SVM-based Web Page Classification

The SVM-based classifier is based on the popular libSVM[2]. Instead of using the page text generated by the crawler, this classifier 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[3] 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 (Mikolov, 2013), 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 and 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 using a method similar to (Min-Yes, 2005) for computing all the possible n-grams of length 4 or longer that are contained in the OpenOffice dictionary[4]. 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.
- TFIDF: the term frequency–inverse document frequency of the body's words.

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

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

[4] http://extensions.openoffice.org/en/search?f[0]=field_project_tags%3A157

4 EXPERIMENTS

4.1 Dataset

We have randomly selected 6’000 domains from the list of all registered .it domains (~2.8 millions). Each domain has been classified by at least two persons using a web-interface we have developed. This is to reach an agreement on the domain category. Out of the domain set, we obtained a dataset of about 5.600 domain with valid annotations; the following table highlights the number of valid annotations for every agrifood categories and non-agrifood category.

Table 3: Valid Annotations per Category.

Category	Annotations	Percentage
Agriculture	301	5,3%
Wine	366	6,5%
Cooking Oil	103	1,8%
Breeding	190	3,4%
Farmhouse	679	12,0%
Pasta and Bread	237	4,2%
Fishing and Aquaculture	41	0,7%
Meat Curing	86	1,5%
Dairy Foods	250	4,4%
Agriculture (Other)	370	6,5%
Beverages (no wine)	117	2,1%
Restaurant and Catering	277	4,9%
Non-agrifood	2.654	46,8%
Total	5.671	

The valid annotations have been split in two groups: 80% of the corpus has been used as development set, the remaining 20% as test set. The development set has been further split in two groups: 80% of the has been used as training set, and the remaining 20% as dev-set for the feature selection.

4.2 Training and Feature Selection

In SVMs it is crucial to select the features used for classification. In order to find the optimal setup for the SVM configuration we have used a simple wrapper algorithm: we ran several test using all possible subset of features, with a cutoff of 1 and 10 and with/without stop word removal. Each subset was used to train a new model which was tested on a dev-set. For all the prediction results, we have counted the number of errors made on the dev-set and we have chosen the features subset which had the minimum error rate.

The feature selection has been divided in two distinct steps. The first step was to find the best features subset for the agrifood and non-agrifood domains classification. In this case the best configuration that we obtained uses positive and negative list of words, HTML meta, title, body, img and a tag. The second step was to find the best features subset to classify the agrifood domains into 12 agrifood categories: the best configuration that we obtained uses word clusters, HTML meta, title and body tag, domain name split and TFIDF. As the two configurations do not overlap, one for the first, and one for the second step.

4.3 Classification Results

The classification outcome is evaluated using the standard metrics precision, recall and F1 (Powers, 2011). 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 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.

Table 4: Classification results evaluation for agrifood classification.

Classifier	Precision	Recall	F1
Probabilistic	91,4%	91,4%	91,4%
SVM	91,0%	84,0%	88,0%
Union	88,7%	93,92%	90,95%
Intersection	94,3%	77,48%	85,05%

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 (Goutte, 2005). We have also evaluated how to combine the two approaches together in order to improve the results. In the above table we have depicted the union and intersection of classification results reported by both approaches. 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. 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 classification based on the specified features. This said we have decided not to discard the SVM classifier, but rather to use both of them to further tune the classification system. In fact with million of domains to classify, it is helpful to limit manual analysis/debugging to a subset of the results rather than to the whole set. For this reason using the two classifiers we can restrict our search mostly in the set of domains that are present in the union but not on the intersection of both methods. Both classifiers report a prediction confidence for each classified site. The following figure depicts the number of correct predictions when compared to the returned prediction confidence.

Table 5: Prediction Percentage Distribution for the Probabilistic Classifier

Category	Precision	Recall	F1
Agriculture	85,8%	75,2%	80,1%
Wine	84,5%	90,2%	87,3%
Cooking 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%**

Table 6: Prediction Percentage Distribution for the SVM Classifier

Category	Precision	Recall	F1
Agriculture	90,8%	72,0%	80,3%
Wine	84,2%	74,4%	79,0%
Cooking Oil	94,4%	47,2%	63,0%
Breeding	76,9%	62,5%	69,0%
Farmhouse	90,5%	84,9%	87,6%
Pasta and Bread	81,8%	64,3%	72,0%
Fishing and Aquaculture	50,0%	18,2%	26,7%
Meat Curing	81,3%	56,5%	66,7%
Dairy Foods	97,8%	81,8%	89,1%
Agriculture (Other)	48,8%	69,3%	57,3%
Beverages (no wine)	96,2%	80,7%	87,7%
Restaurant and Catering	85,2%	73,0%	78,6%
Overall	**80,3%**	**72,0%**	**75,9%**

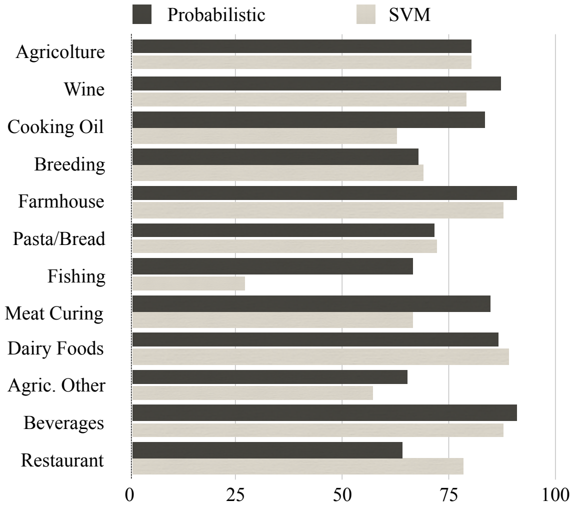

Figure 1: F1 Score Comparison: Probabilistic vs. SVM Classifier

Figure 1 shows that SVM F1 score decreases for those categories that have too few annotated domains as depicted in table 2. Instead both classifiers produce very similar F1 scores for most categories, where more annotated domains were used. As expected the manual tuning in the probabilist classifier has some benefits as in most categories it outperforms the one based on SVM, and it can produce good results even for those categories for which few domains have been annotated. This is because the dictionary of positive/negative words used by the probabilistic classifier includes using both the words extracted from the annotated domains and additional words added manually that are also relevant but not present in the annotate domains.

As shown in the figure below, the SVM classifier assigns to each domain a category and the probability of belonging to this category (axis x). The SVM classifier gives better results when this probability is high but the correct prediction percentage does not fall when the probability decreases.

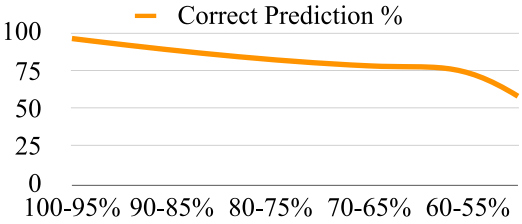

Figure 2: Correct Prediction Percentage Distribution for SVM-classifier.

While analysing Figure 1. we have manually looked at the results to see where the system could be improved. Beside the classification errors, we have noticed that many domains were not classified properly due to lack or little text on which to run the classifier. In addition to flash-only web sites where we have very little text (e.g. the <meta> tag), web sites sometimes have a complex page structure that makes difficult for the crawler to guess what are the pages that contain the most relevant information. A possible improvement could include the analysis of neighbouring web pages to guess the category of pages not classified due to lack of text. We have added in the crawler the ability to skip pages such as “Contact Us” or “Legal” but as future work we need to add further heuristic to skip pages that contain not too relevant text, and that can negatively influence the prediction, while adding the ability to follow HTML anchors often hidden in Javascript code. In fact, unless we try to discard content that is not relevant such as the above web pages, our system can be influenced by words that are present in the web pages but that are not relevant for our classification, and thus produce poor results.

The following figure depicts the agrifood distribution for .it registered domains. The most popular category is farmhouse, that is 50% bigger than the second category that is wine production. The first three categories account for 50% of the all the classified agrifood domains.

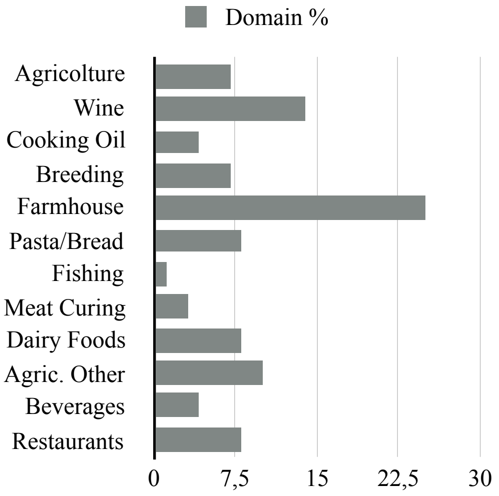

Figure 3: Agrifood distribution for .it registered domains

5 FUTURE WORK

This work is the base of a follow-up activity we will be carrying on. The plan is to compare the results we have obtained classifying web sites using our tools, with the data that is present in records of the Chamber of Commerce. In essence we want to correlate classification results with those we find in official company records. This is to verify what are the areas of “digital divide” for Italian companies in term of business sector, as well to understand if the official company records match with the information those companies put on the corporate web sites.

A work-in progress activity we are carrying on is the extraction of FaceBook data using the API they provide. In fact we want to verify how many companies use non-.it domain names for their activities, and how many use just FaceBook without having a registered domain name. This activity will also allow us to map agrifood business to regions, as FaceBook APIs offer location-based search.

Finally another future activity, is to apply the methods we developed for classifying agrifood sites to all sectors. This to generalise the tools we developed and also fully classify the .it web.

6 CONCLUSIONS

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 outcome is that we can correctly assign a category to domain names with a overall F1 score of over 80% that is great step ahead with respect to commercial classification services that produce poor results as reported in Table 1; this using broader categories, and thus easing the classification task, with respect to this work where we have used very specific categories. This work has been used in the context of the Universal Expo Expo2015 to classify the agribusiness sites active on .it, and divide them into sub-categories. While the system is operational since some months, we are extending it to user it for categorising non-agrifood domains.

In terms of original contributions, our system is a step forward with respect to commercial classification systems that fall short when classifying non-English or not-so-popular web sites. All the software is based on freely available tools and libraries, and its internals have been explained in this paper making the system open and extensible, contrary to commercial systems that do not explain how/how often they classify sites.

ACKNOWLEDGEMENTS

Our thanks to Prof. Giuseppe Attardi <attardi@di.unipi.it> for his help and suggestions thought this work. In addition we would like to thank our colleagues that have assisted us during the manual domain classification.

REFERENCES

BrightCloud Inc., 2014. BrightCloud Web Classification Service, http://www.brightcloud.com/pdf/BCSS-WCS-DS-us-021814-F.pdf

SimilarWeb Inc, 2014. Our Data & Methodology, http://www.similarweb.com/downloads/our-data-methodology.pdf.

AOL Inc, 2015. Open Directory Project (ODP), http://dmoz.org.

Blocksi SAS, 2014. Blocksi Manager for Cloud Filtering, http://www.blocksi.net.

zvelo Inc., 2014. Website Classification, https://zvelo.com/website-classification/.

Sun, A., Lim, E., 2002. Web classification using support vector machine, Proc. of the 4th international workshop on Web information and data management (WIDM '02).

Dumais, S., Chen, H., 2000. Hierarchical classification of Web content, Proc. of the 23rd ACM SIGIR conference on Research and development in information retrieval (SIGIR '00).

Zhang Zhang, Y., Zincir-Heywood, N., and Milios, E., 2003. Summarizing web sites automatically, Proc. of AI'03.

Soumen, C., Van den Berg, M., and Dom, B., 1999. Focused crawling: a new approach to topic-specific Web resource discovery, Computer Networks 31.11 (1999): 1623-1640.

Chandra, C., et al., 1997. Web search using automatic classification, Proc. of the Sixth International Conference on the World Wide Web.

Jung-Jin, L., et al., 2009. Novel web page classification techniques in contextual advertising, Proc. of the eleventh international workshop on Web information and data management. ACM.

Xiaoguang, Q., and Davison, B. D., 2009. Web page classification: Features and algorithms, ACM Computing Surveys (CSUR) 41.2 (2009):12.

Dou, S., et al., 2004. Web-page classification through summarization, Proc. of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM.

Hwanjo, Y., Han, J., and Chen-Chuan Chang, K., 2002. PEBL: positive example based learning for web page classification using SVM, Proc. of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.

Ji-bin, Z., et al., 2010. A Web Site Classification Approach Based On Its Topological Structure, Int. J. of Asian Lang. Proc. 20.2 (2010):75-86.

Soumen, C., Dom, B., and Indyk, P., 1998. Enhanced hypertext categorization using hyperlinks, ACM SIGMOD Record. Vol. 27. No. 2. ACM.

Attardi, G., Gulli, A., and Sebastiani, F., 1999. Automatic Web page categorization by link and context analysis, Proc. of THAI. Vol. 99. No. 99.

Vapnik, V., 1998. Probabilistic learning theory. Adaptive and learning systems for signal processing, communications, and control, John Wiley & Sons.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. Springer Berlin Heidelberg.

Sun, A., Ee-Peng, L., and Wee-Keong, N., 2002. Web classification using support vector machine, Proc. of the 4th international workshop on Web information and data management, ACM.

Weimin, X., et al., 2006. Web Page Classification Based on SVM, Proc. of WCICA 2006, IEEE.

Fernandez, V. F., et al., 2006. Naive Bayes Web Page Classification with HTML Mark-Up Enrichment, Proc. of ICCGI ’06.

Vinu, D., Gylson, T., Ford, E., 2011. Naive Bayes Approach for Website Classification, In Communications in Computer and Information Science, Vol 147.

Attardi, G., Dei Rossi, S., and Simi, M., 2010. The tanl pipeline, Proc. of Workshop on Web Services and Processing Pipelines in HLT, co-located LREC.

Rajaraman A., and Ullman, J., 2011. Mining of Massive Datasets, Cambridge University Press.

Mikolov, T., et al., 2013. Efficient estimation of word representations in vector space, Cornell University.

Min-Yen, K., and Oanh Nguyen Thi, H., 2005. Fast webpage classification using URL features, Proc. of the 14th ACM international conference on Information and knowledge management, ACM.

Powers, D. M., 2011. Evaluation: From precision, recall and F-factor to ROC, informedness, markedness & correlation (Tech. Rep.)., Journal of Machine Learning Technologies 2 (1): 37–63.

Goutte, C., and Gaussier, E., 2005. A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation, Proc. of ECIR ’05, 2005.

Marill, J. L. , Boyko, A., Ashenfelder, M., and Graham, L., 2004. Tools and techniques for harvesting the world wide web. In Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries (JCDL '04).