Functional Classification of Websites

Najlah Gali
University of Eastern Finland
P.O. Box 111
Finland
najlaa@cs.uef.fi

Radu Mariescu Istodor
University of Eastern Finland
P.O. Box 111
Finland
radum@cs.uef.fi

Pasi Fränti
University of Eastern Finland
P.O. Box 111
Finland
franti@cs.uef.fi

ABSTRACT

We propose a novel method to classify websites based on their functional purpose. A website is classified either as single service, brand or service directory. We utilize a number of features that are derived from the link of the website, the postal addresses found in the website, the size of the website, and the text of the anchor element in the Document Object Model tree. We utilize two models to perform the classification task: decision tree and clustering-based models. Our method is fully automated and does not require extensive training data or user interaction. The proposed website classifier improves the baseline by 2 percentage points in case of single service, 33 percentage points in case of brand and 18 percentage points in case of service directory.[1]

CCS CONCEPTS

• **Information systems** → **Data management systems**; *Data model extensions*; Semi-structured data • **Information retrieval** → Retrieval tasks and goals; Clustering and classification

KEYWORDS

Website classification, information extraction, web data extraction, web mining

ACM Reference Format:
N. Gali, R. Mariescu Istodor, and P. Fränti. 2017. Functional Classification of Websites. In SoICT '17: *Eighth International Symposium on Information and Communication Technology, December 7–8, 2017, Nha Trang City, Viet Nam*. ACM, New York, NY, USA, 8 pages.
https://doi.org/10.1145/3155133.3155178

1 INTRODUCTION

Location-based web search, which searches for a business or a place of interest that is tied to a specific geographical location [1], is an example of applications where information extraction methods are required. It requires both identification of geographical data and automatic information extraction from websites. Using the location of the user and a set of keywords, the location-based search detects and validates locations, identifies service information and presents a ranked list of results consisting of the following: short text summary (title), link, image thumbnail, address, and distance. Unlike conventional web search, which is website oriented, location-based search is service oriented and it is relevant especially for mobile users because it offers nearby results in a brief and informative manner. By service, we mean a place of supplying a public need such as *restaurant*, *café*, *hotel*, and *car repair*.

Search engine optimization (SEO) aims at showing the most relevant webpages on the top of the results list. W*ebpage classification* can be used to improve SEO by identifying webpages that are relevant to the user's query [2, 3]. For example, when a user searches for a known service such as *Café Manta*, the homepage of the service is more relevant than a service directory page. However, when the user inputs a more general query such as *café*, then service directory webpages are more relevant because they provide several alternatives and sometimes even ratings, which are more useful than browsing through each service website separately. By using webpage classification, search engines can distinguish between homepages of the individual services and service directories, and adjust the search results based on how general the search query is. Webpage classification is also essential in focused web crawling [4, 5], which selectively collect webpages that satisfy certain properties such as language, topic or purpose [6], because it evaluates the relevance of the webpages to the purpose of crawling.

Webpage classification has been divided into subject classification, functional classification, and sentiment classification among several others in [7]. Subject classification aims at identifying the topic of the webpage such as whether the page is about *sport*, *business, news* or *art*. Functional classification aims at finding the purpose of the website for example whether the webpage of a school is a *course*, *a staff* or *a student* page. Sentiment classification aims at detecting author's attitude about some particular topic. Other types of classification include genre classification [8], and identifying spam in search engine [9]. In this study, we focus on functional classification and try to solve whether the webpage is a *single service*, *brand* or *service directory*. We define single service website as a website that contains a service located at one physical location such as Deli China restaurant (*www.deli-china.fi*) and Restaurant Kerubi (*www.kerubi.fi*). A brand website is a website that contains services located at different physical locations but belong to the same owner such as OP (*www.op.fi/*), Hesburger (*www.hesburger.fi/etusivu*), and Starbucks (*www.starbucks.com*)

ACM ISBN 978-1-4503-5328-1/17/12…$15.00
https://doi.org/10.1145/3155133.3155178

A service directory website is a website that lists services offered by other organizations according to the goods or services they offer such as Mihin.fi (*www.mihin.fi*), fonecta.fi (*www.fonecta.fi*) and, yellowpages.com (*www.yellowpages.com*).

We introduce a novel functional classification that identifies the purpose of the website. We utilize a number of features that are derived from the Uniform Resource Locator (URL) of the webpage, the number of postal addresses found in the website, the size of the website, and the text of the lists in the Document Object Model (DOM) tree. We parse the webpage hypertext markup language (HTML) source code to build the DOM tree and apply a set of criteria to extract and analyze the specified lists in the tree. We define three types of lists, which are *menu bar*, *minor lists,* and *nested lists*. We extract their text and apply string matching and term frequency rules to conclude the purpose of the website. We investigate two models to perform the classification task: decision tree and clustering-based models. Our method is implemented in the framework of *MOPSI* (*cs.uef.fi/mopsi*).

2 CLASSIFICATIONS OF WEBSITES

Several methods have studied the classification of websites by their topic [10-12]. However, classifying websites based on their functionality has been less studied [13].

Kraaij et al. [14] propose a method to find entry webpages (home page) of organizations. The structure of the webpage link, the length of the webpage, and the number of inlinks are used as features to train a Naïve Bayes model. Results have shown that the structure of the webpage link has the highest impact on the overall accuracy of the system.

Elgersma and De Rijke [15] propose a binary classification method to determine if a given webpage is a blog or not. Forty-six numerical and binary attributes are extracted from the DOM tree of the webpage, such as number of posts, average post length, minimum post length, maximum post length, and from the link of the webpage such as domain name. Different machine learning models have been used such as Naïve Bayes, support vector machines (SVM), decision tree and rule-based decision table. Results have shown that often, these models provide rather similar performance.

Lindemann and Littig L [16] investigated whether the structural properties of a website such as the size, the organization, the composition of webpage link, and the link structure of web sites reflect its functionality. A Naïve Bayes model has been used to classify the webpages into five categories: *Academic, Blog*, *Corporation*, *Personal*, and *Shop*. The work has been extended in [2] to classify the webpages into *Academic*, *Blog*, *Community*, *Corporate*, *Information*, *Nonprofit*, *Personal*, and *Shop*. Thirty features that describe the structure of the website are extracted, such as average external site outdegree, page count, the fraction of .pdf and .ps documents in the page, the fraction of the pages that contain JavaScript, and the average number of digits within the link path. A Naïve Bayes classifier has been used for this task and the results have shown that it is hard to differentiate between some type of webpages such as between *Academic*, *Information*, *Blog*, *Community*, and *Shop*, and between *Corporate*, *Nonprofit*, and *Personal* due to the functional relation between them.

Kenekayoro et al. [17] classify a university website into different categories such as *about*, *business and innovation*, *discussion*, *support*, *research*, *staff*, *student life* and *study webpage*. Decision tree induction and SVM classifiers have been trained using five hundred features that are derived from the keywords of the link and the title of the webpage.

Saraç and Özel [18] classify the webpages into *course*, *project*, *student and faculty*. Ant colony optimization algorithm has been used to select the best features from a large set of features that are derived from all keywords from the *<title>, <h1>, <h2>, <h3>, <a>, <b>, <i>, , , <p>, * tags and the link of the webpage. In every iteration, a subset of features is selected by each ant, and the webpages are classified using C4.5 classifier [19] and then features that provide higher F-measure are more likely to be selected in the next iteration. After a pre-defined number of iterations, the subset of features that has the best F-measure value is selected for classifying new webpages.

We introduce a new classification that depends on the purpose of the website and utilize new features such as postal addresses and text of navigation menus. We further developed a new classification model that depends on clustering.

3 PROPOSED METHOD

The overall process of the proposed method is shown in Fig. 1. We conduct a simple filtering step to clean the input from irrelevant webpages that do not contain any services. These are mostly blogs, reviews, news or social networking webpages. Similarly to [15], we first check the link of the webpage for well-known domains such as *facebook*, *twitter*, *wikipedia* and *linkedin* and a bag of keywords such as *blogs*, *news*, *journal*, *books*, and *author* as indicator. If this fails, a search is done for meta tags, which are created for news, blogs, review, story or personal webpages such as:

<meta name = newskeywords>, *<meta name = article:id>*,
<meta name = article_section>, *<meta name = article:type>*,
<meta property = og:channel>, *<meta content = blogger>*,
and *<meta content = profile>*. A webpage that satisfies any of these conditions is considered as a non-service webpage. After they have been discarded, we classify the remaining websites either as single, brand or service directory using the proposed websites classifier.

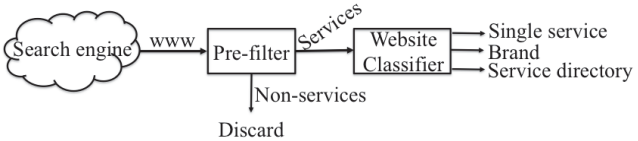


Figure 1: Workflow for website classification.

3.1 Website Classification

3.1.1 Feature extraction. We consider two content-independent and two content-dependent sources for feature extraction:

- The depth and length of the web link;
- Size of the website;
- The number of address elements;
- The text inside anchor elements <*a*> from specified lists in the website.

A. Content-independent features

Using the link of the webpage has been widely studied in the field of website classification and it was found to be a good source of information [20-22]. It consists of a domain name, a directory path, and filename. We hypothesize that a webpage whose link has just the domain name is the homepage of a service. Service directories have usually more complex structure and their web link typically includes hierarchy (eg. *http://www.yellowpages.ca/bus/Quebec/Montreal/RobinSquare/8184329.html*). We further hypothesize that the probability of a website being a service directory is proportional to the length of the link, and to the depth of the path.

The size of the website is another indicator of its functional purpose. It is determined by counting the number of its known pages (page count). For example, the number of webpages of a service directory website is relatively high in comparison with a single service website. We use the pygoogle (*https://code.google.com/p/pygoogle*) module for counting the pages.

B. Content-dependent features

Simple content-independent features are not enough; therefore, content-related information needs to be analyzed as well. The first feature is the number of postal addresses contained in the website. It is used as an indicator to distinguish between single and brand websites. We hypothesize that brand contains multiple addresses, one per location, whereas single service contains usually a single one. To detect the postal addresses, we use the algorithm described in [23].

Studies such as [18, 24] have shown that certain HTML tags such as <*a*>, <*h*>, and <*li*> contain more valuable text than other tags such as <*small*> or <*span*>. We download the HTML source of the webpage and parse it as DOM tree. DOM is an interface allowing scripts and programs to dynamically access and handle all the elements such as content, structure and style of webpages. After the tree has been built, specified lists of links are extracted. We define three types of lists, which are: *menu bar*, *minor lists*, and *nested lists*. We consider these types of lists in our method because their text summarizes the content of the website and gives useful hints on the purpose it serves. For example, the texts of the horizontal bar in Fig. 2 give hints about different categories contained in the website such as *accommodation, gas station, restaurant, travel agency, and traffic*. This kind of variation in the categories of services is usually not found in brand or single service websites because they are focusing on one or few categories but is common to service directories.

Figure 2: Example of horizontal menu bar.

The menu bar (see Fig. 3) is a user interface within a webpage that contains links to other parts of the website. A website menu bar is generally displayed as a horizontal list of links at the top of each webpage. In some cases, it is placed vertically on the left side of each webpage and it is called a sidebar. Some websites have both a horizontal menu bar at the top and a vertical navigation bar on the left side with different content. The minor lists (see Fig. 3) are distributed over the page but in most cases they are located below the menu bar or at the footer of the website. They contain information about the topics or sub-topics the website serves and in some cases; they are more informative for users and our classification method than the menu bar. The nested lists (see Fig. 3) are the lists inside the menu bar or side bar and they are known as sub lists or drop down lists. The texts of nested lists are more specific to the service type. For example, if a website serves as a directory for restaurants, the nested list usually contains the names of these restaurants.

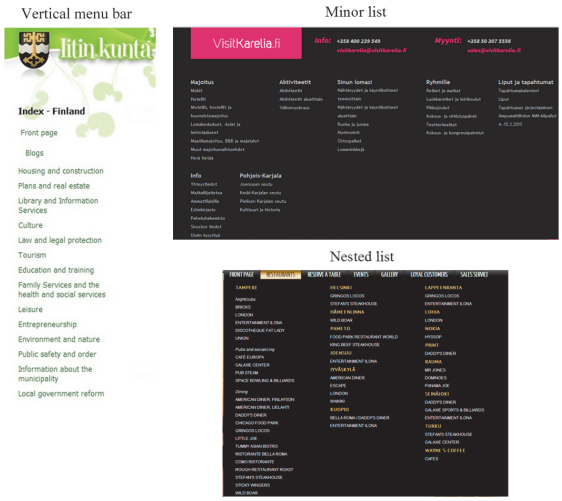


Figure 3: Example of three types of lists (menu, minor, nested).

We use XPath (*www.w3.org/TR/xpath20*), which is a query language for addressing parts of an XML document, to identify the three types of the lists and extract their text values. Every node in the tree has its own XPath value referring to its location in the tree. XPath is used because of its short processing time; there is no need to traverse every node in the tree. We apply the following criteria to extract the lists and analyze their content:

Menu bar list:

The menu bar is defined by the following criteria:

- Formatted using specific HTML tags, namely <ul>, <ol>, <dl>, and <h1>-<h6> tags. Each item in <ul> and <ol> list is followed by an <il> tag and each item in <dl> list is followed by a <dt> tag. We observe that these HTML-list patterns are mostly used for menu items representation, which was also concluded in [25];

- If no list is detected with <ul>, <ol>, <dl>, and <h1>-<h6> tags, we also look for tags that are followed by <a> tags and their number of children ≥ 4;
- The leaf nodes should have hyperlinks, therefore the anchor element <a> must exist;
- The leaf text length should be less than a threshold *t*, as also concluded in [26];
- The size of the list is between four and eight elements in case of horizontal bar;
- The list is located at the top of the webpage (visually below the logo and header but above the main content); or it at the left side of the page. We calculate the position of the list relative to the top left corner and we compare it with the page size;
- The leaf nodes should have similar property to each other. For example, in vertical lists all leaf elements have the same value of Y coordinate.

After we have identified the list, we extract the text of each leaf node. The text summarizes the topics (categories) of the website. Next, we match these topics with the main categories in the database. Our database consists of 27 main categories and 387 sub-categories (*cs.uef.fi/mopsi/titleextraction*). For this, we use edit distance [27], which is the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one word into the other. The reason of using edit distance is that there are expected differences in the writing style between the website categories and the database categories. For example, *And* in some websites is written as *&* like *Food and Drink* and *Food & Drink*.

Minor lists

The minor lists are defined by the following criteria:

- They are formatted using specific HTML tags <ul>, <ol>, <dl>, , and <form> followed by <select>, <option> tags;
- Their leaf nodes should have hyperlinks, therefore the anchor element <a> must exist;
- Their size vary;
- They are distributed over the page but in most cases they are located below the horizontal menu bar, at the sides or at the footer of the website.

We extract the text value of each leaf node in the lists by XPath. Then, we apply string comparison to find the similarity between the values of text nodes and the categories in the database in the same way we performed for the menu bar.

Nested lists

Nested list are usually formatted in HTML code using <li> tag with a child being <ul> tag, and the <a> tag must exist. The advantage of this type of lists is that their text has one frequent term that represent a service of one type. For example, restaurants service directory will have terms like *food*, *restaurant* or *café*, which appear more frequent in the list. For this reason, we use term frequency (TF) to find the most frequent term in the list:

$$TF(t) = N_{t,l} \quad (1)$$

Here N is the number of occurrences of term t in list l.

3.1.2 The models. We consider two alternative models: *decision tree* and *clustering-based*. The advantage of the decision tree is its simplicity, and no training data or very limited data will be needed. Clustering-based model, on the other hand, is a supervised classifier but can also be built based on limited training data. It is a simplified version of Gaussian mixture model, in which only the centroids of the mixtures are used. In the following, we describe these two models in more detail. Same set of features is used in both models.

A. Decision tree

We construct a decision tree based on rules that use the features described in the previous section. We identify a single service website based on the depth of the webpage link and the number of postal addresses detected in the website (see Fig. 4). If the link of the webpage is in its domain format (eg. *http://www.example.com/* or *http://www.example.com/index.html*), then we further check the number of postal addresses contained in the website. If it contains at most one postal address, we assign it as single service; otherwise, we conclude that the website is a Brand website.

If the link of the webpage is not in its domain format (eg. *http://www.example.com/path*or*http://www.example.com/path/file name*), then we check the size of the website. If the page count is at least 150,000 then we conclude that the website is a service directory. Otherwise, we analyze the number of categories and sub-categories detected in the website. If the categories of the website match at least 3 main categories from our database, we conclude that the website is a service directory. However, some directories use more service specific categories and cannot be detected by a single threshold. Therefore, we also match the categories of the website with the sub-categories of the services in the database. If the categories of the website match at least 4 sub-categories, we conclude that the website is a service directory. If not, then we proceed further by checking the text of the minor lists. If more than 3 categories are discovered in the website, we conclude that the website is a service directory. Otherwise, we proceed with the final rule, which checks the text of the nested lists. We conclude that the website is a service directory, if *TF(t) > threshold 4,* and *t* matches any of the sub-categories in the database. If none of the above-mentioned rules indicate that the webpage is a service directory, then it is classified as a brand website (see Fig. 5). The above-mentioned thresholds were optimized using brute-force search for a small set of 30 webpages, except the page count, for which the threshold was manually determined.

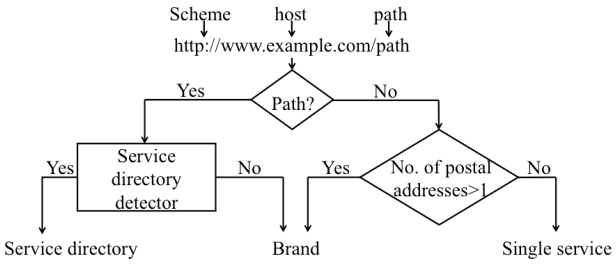


Figure 4: Decision tree to identify single service website.

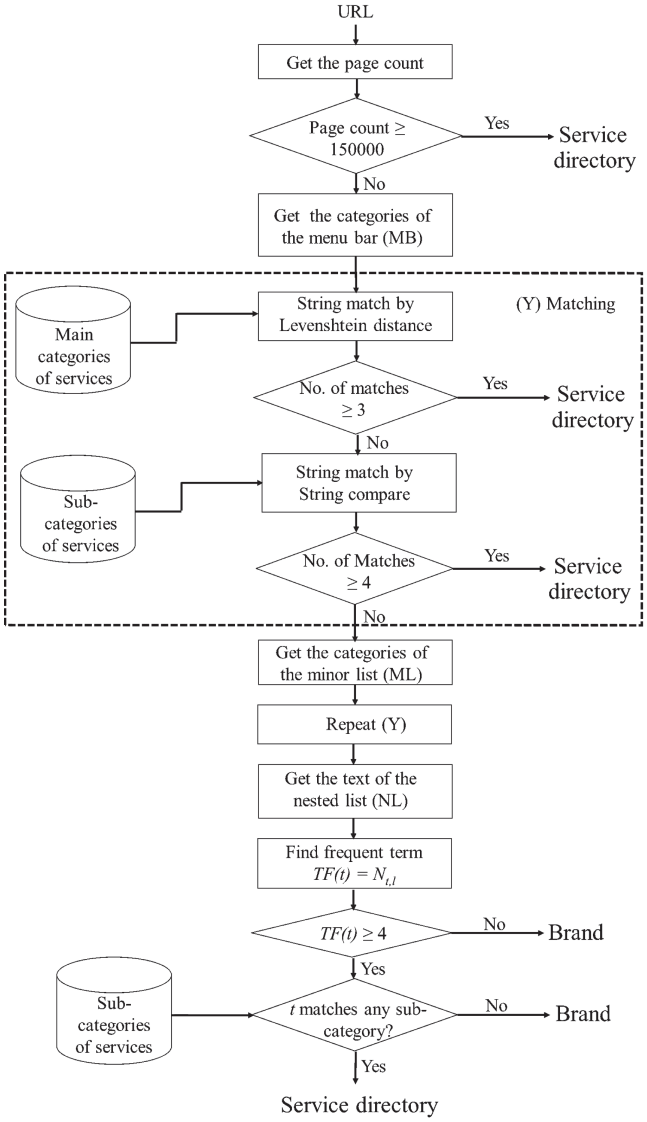


Figure 5: The algorithm to identify service directory website.

B. Clustering-based

The clustering-based model consists of two phases: training and classification (see Fig. 6). We first calculate a feature vector for each website using the features described in 3.1.1. In the training phase, we manually label the websites as single service, brand, or service directory and cluster their feature vectors using random swap algorithm [28]. Then, we determine the dominant label of each cluster and use this label to all features that will be mapped to this cluster. For example, in Fig. 7, the websites are grouped in 6 clusters. Cluster 1 contains 154 service directories, 97 brand and 59 single service websites. It is therefore labeled as a representative for service directory websites. In the classification phase, we compute the feature vector for each new website and map it to the nearest representative cluster using Euclidean distance in the feature space. The website is then classified according to the label of that cluster.

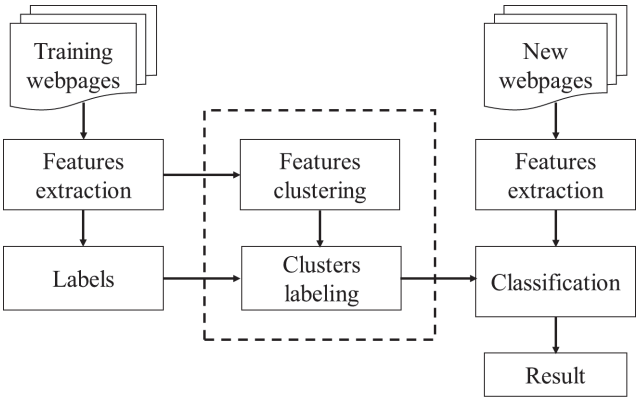
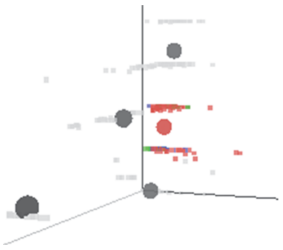


Figure 6: Architecture of the clustering-based model.



clus.	sd	brand	single	total
1	**154**	97	59	310
2	**125**	21	4	150
3	0	140	**686**	826
4	**46**	2	0	48
5	**57**	0	0	57
6	**18**	0	0	18

Figure 7: Visualization of data with six clusters (centroids) and a service directory label.

4 EXPERIMENTS

4.1 Data Sets

The test set was collected during 18-31 July 2014 and 19-23 April 2015, by choosing different type of websites from different regions of the world, in order to have a reasonable geographical diversity. This set contains 1,761 webpages from 1,552 websites in eight categories: *Food & Drinks*, *Home & Garden, Hotels and Accommodation*, *Shopping*, *Arts & Entertainment*, *Hobbies & Leisure*, *Sport*, and *Health & Social care,* and 200 non-service webpages, collected from Google and Google maps search results using queries such as *bar*, *restaurant*, *café, Pizza*, *Radisson blue hotel*, *H&M shop*, *Play bar*, *Cavalier pub*, *Rosso restaurant*, *Intersport shop*, *sauna*, *swimming pool* and *bowling alley.*

We manually tagged the websites either as single service, brand, service directory or non-service. In the following experiments,

this data is used as a ground truth to measure the accuracy of our website classification.

4.2 Evaluation Measure

For the websites in a given class (single, brand, or service directory), we count the following classification results:

tp = the number of websites correctly labeled as belonging to this class.

fp = the number of websites incorrectly labeled as belonging to this class but they belong to another class.

fn = the number of websites incorrectly labeled as belonging to another class but they belong to this class.

We measure the performance of our website classification methods using precision, recall, and F-measure, which are widely used to evaluate information extraction systems.

$$Precision = \frac{tp}{tp + fp} \quad (2)$$

$$Recall = \frac{tp}{tp + fn} \quad (3)$$

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (4)$$

4.3 Methods Evaluated

We compare the following methods:

- Link-based (link) (baseline)
- Decision tree (DT) (proposed)
- Clustering-based (CLUS) (proposed)

As a baseline, we use the type of the webpage's link as proposed in [14] with few modifications explained below. These types are: *root*, *sub-root*, *path* and *file*. Root is the domain name such as *http://www.example.com*; Sub-root is a domain name followed by single directory such as *http://www.example.com/direcotryname*; Path is a domain name followed by such as *http://www.example.com/directoryname/path*. All these can be optionally followed by index.html. The fourth type is file, which is any link with a file name other than index.html such as *http://www.example.com/filename.html*. We assign root and sub-root types to single service category because 95% of the links of single service webpages from the dataset are of these formats. We assign path to service directory because 75% of the links of the service directory webpages from the dataset are of this format. We assign filename type to brand webpages.

For CLUS model, we conducted five-fold cross-validation using the test set. We utilized 1409 vectors for training and 352 vectors for testing when filtering was applied, and 1569 vectors for training and 392 vectors for testing when filtering was not applied. All the results reported here are averaged over five trials. For DT we used the thresholds optimized as discussed in subsection 3.1.2 (A) on the entire test data. We experimented with different number of clusters from 1 to 200 as shown in Fig. 8. When only one cluster is used, the model labels all websites as single service (majority of training data is of this type). When the number of clusters exceeds 100, the problem of over fitting is raised; and the overall accuracy goes down again. As a compromise, we select 60 clusters for all the further experiments.

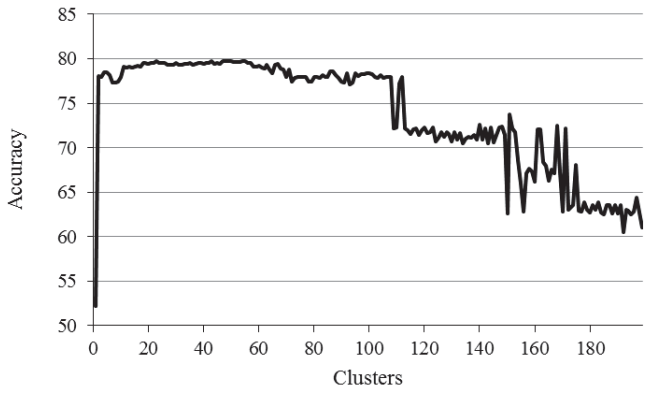


Figure 8: Classification accuracy of the filtered dataset with different number of clusters.

4.4 Results

Tables 1, 2 and 3 show the classification accuracy for the models with different versions of the data:

- All websites included (no filtering);
- Detected non-service pages removed (filtering applied);
- Clean data with manually removing non-service pages (oracle).

Table 1: Classification accuracy for decision tree model

Type of websites	Number of websites	Filtering non-service webpages		
		No	Proposed	Oracle
Single	919	92%	91%	92%
Brand	326	39%	38%	39%
Service directory	516	89%	89%	89%
Non-service	200	0%	72%	100%
Total	1,961	73%	80%	83%

Table 2: Classification accuracy for clustering-based model

Type of websites	Number of websites	Filtering non-service webpages		
		No	Proposed	Oracle
Single	919	92%	91%	92%
Brand	326	19%	19%	19%
Service directory	516	97%	97%	97%
Non-service	200	0%	72%	100%
Total	1,961	72%	79%	82%

Table 3: Comparison of website classification models using the filtering

Type of website	Link			DT			CLUS		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Single	76%	91%	85%	83%	91%	87%	83%	91%	87%
Brand	12%	8%	9%	46%	38%	42%	44%	19%	26%
Service directory	76%	71%	73%	88%	89%	91%	76%	97%	88%

Table 3 summarizes the comparative results between all the models (with the applied filtering) using precision, recall, and F-measure. We observe that DT and CLUS provide good precision (83%) and recall (91%) values for detecting single service website. The values for detecting service directory websites (88% and 89%) and (76% and 97%) respectively are also good, considering that this is a challenging task due to their heterogeneous structures. For brand websites, our classifiers provide less satisfactory results as only (38%) of them are detected by DT and (19%) of them are detected by CLUS. These results are due to the fact that brand websites can be as simple as single websites such as *Cocoa Bar website (www.cocoabarnyc.com*), or as difficult as service directories such as *Best Western Hotels website* (*www.bestwestern.fi/hotels/best-western-hotel-savonia-kuopio-910831)*. Our classifiers misclassify more than half of them. The clustering-based model has an advantage that it classifies service directory webpages with high accuracy (97%) in comparison to the DT model (89%). This would be beneficial in applications that need central place of information rather than searching every website individually.

The experiments show that the precision, recall, and F-measures are affected by the facts that some websites use Google maps as a part of their templates, which makes it harder to discover the type of these websites, as they do not provide any informative lists, or keywords that could be used for analysis such as *http://yossa.fi/bar-play*. Famous Brand websites especially those that are classified under *Hotels and Accommodations* category such as *Radisson blue hotels* and *Marriott hotels* provide rich information about tourism, travel, and accommodation, and are therefore incorrectly classified as service directory. Comparing these results to the Link model, we observe that the structure of the webpage link acts as a good indicator for classifying single service websites by providing (91%) accuracy. However, it fails to classify brand websites, providing only (8%) correct classification in comparison with DT (38%) and CLUS (19%). Satisfactory results are achieved when classifying service directories by providing 71% accuracy.

5 CONCLUSIONS

In this paper, we propose a fully automated method to identify the type of the websites without extensive needs of training data or user interaction. The proposed classification is useful for search engines and web crawling where the purpose of the website may be useful for the search result. Our method is integrated with the framework of MOPSI [29], which is a platform that implements various location-based services and applications such as mobile search engines, data collection, user tracking, and route recording. It has applications integrated both on web and in mobile phones.

We conducted various experiments to evaluate the performance of our method. The results show that our classifier outperforms the baseline by 2 percentage points in case of single service, 33 percentage points in case of brand, and 18 percentage points in case of service directory.

Although the developed method is tailored for this particular task, the idea behind it can potentially be extended to classify web pages in general; for example, blogs and news pages.

REFERENCES

[1] Amin A, Townsend S, Van Ossenbruggen J, Hardman L (2009) Fancy a drink in canary wharf?: A user study on location-based mobile search. In Human-Computer Interaction–INTERACT 2009. Springer, Berlin, Heidelberg, pp 736-749

[2] Lindemann C, and Littig L (2007). Classifying web sites. In Proceedings of the 16th international conference on World Wide Web, ACM, pp. 1143-1144.

[3] Qi X (2012). Webpage classification and hierarchy adaptation (Doctoral dissertation, Lehigh University).

[4] Chakrabarti, S., Van den Berg, M., & Dom, B. (1999). Focused crawling: a new approach to topic-specific Web resource discovery. Computer Networks, 31(11), 1623-1640

[5] Ozel S. A, and Saraç E, (2008). Focused crawler for finding professional events based on user interests. In Computer and Information Sciences, ISCIS'08, pp. 1-4.

[6] Hernández I, Rivero C. R, Ruiz D., Corchuelo R (2014). CALA: An unsupervised URL-based webpage classification system. Knowledge-Based Systems, 57, pp.168-180.

[7] Qi X, and Davison B. D (2009). Webpage classification: Features and algorithms. ACM Computing Surveys (CSUR), 41(2), 12

[8] Zu Eissen, S M, Stein B (2004). Genre classification of webpages. In KI: Advances in artificial intelligence. Springer Berlin Heidelberg, pp. 256-269

[9] Castillo C, Donato D, Gionis A, Murdock V, Silvestri F (2007). Know your neighbors: Web spam detection using the web topology. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 423-430

[10] Riboni D (2002) Feature selection for webpage classification

[11] Zhong S, Zou D (2011) Webpage Classification using an ensemble of support vector machine classifiers. Journal of Networks, 6(11), pp1625-1630

[12] Qu H, La Pietra A, Poon S (2006). Automated Blog Classification: Challenges and Pitfalls. In AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, pp. 184-186

[13] Järvstråt, L. (2013). Functionality Classification Filter for Websites (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-93702

[14] Kraaij W, Westerveld T, Hiemstra D (2002). The importance of prior probabilities for entry page search. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 27-34

[15] Elgersma E, and De Rijke M (2006). Learning to recognize blogs: A preliminary exploration. NEW TEXT Wikis and blogs and other dynamic text sources, 24

[16] Lindemann C, and Littig L (2006). Coarse-grained classification of web sites by their structural properties. In Proceedings of the 8th annual ACM

international workshop on Web information and data management, pp. 35-42

[17] Kenekayoro P, Buckley K, Thelwall M (2014). Automatic classification of academic webpage types. Scientometrics, 101(2), pp.1015-1026.

[18] Saraç E, and Özel S. A (2014). An Ant Colony Optimization Based Feature Selection for Webpage Classification. The Scientific World Journal, 2014

[19] Quinlan J. R (1993). C4.5: programs for machine learning. San Francisco: Morgan Kaufmann Publishers Inc

[20] Kan M. Y, and Thi H. O. N (2005). Fast webpage classification using URL features. In Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 325-326

[21] Baykan E, Henzinger M, Marian L, Weber I (2009). Purely url-based topic classification. In Proceedings of the 18th international conference on World Wide Web, ACM, pp. 1109-1110

[22] Baykan E, Henzinger M, Marian L, Weber I (2011). A comprehensive study of features and algorithms for url-based topic classification. ACM Transactions on the Web (TWEB), 5(3), 15

[23] Tabarcea A, Hautamäki V, Fränti P (2011) Ad-hoc georeferencing of web-pages using street-name prefix trees. In Web Information Systems and Technologies, Springer, Berlin, pp 259-271

[24] Kim S, and Zhang B. T (2003). Genetic mining of HTML structures for effective web-document retrieval. Applied Intelligence, 18(3), pp. 243-256

[25] Weninger T, Fumarola F, Barber R, Han J, Malerba D (2011) Unexpected results in automatic list extraction on the web. ACM SIGKDD Explorations Newsletter, 12(2), pp 26-30

[26] Chen J, Zhou B, Shi J, Zhang H, Fengwu Q (2001) Function-based object model towards website adaptation, International conference on World Wide Web, ACM, pp 587-596

[27] Levenshtein V. I (1966) Binary codes capable of correcting deletions, insertions and reversals. In Soviet physics doklady vol. 10, pp 707-710

[28] Fränti P, and Kivijärvi J, (2000) Randomized local search algorithm for the clustering problem. Pattern Analysis and Applications, 3 (4), pp 358-369

[29] Fränti P, Chen J, Tabarcea A (2011) Four Aspects of Relevance in Sharing Location-based Media: Content, Time, Location and Network. In WebIST, pp 413-417