J Intell Inf Syst (2013) 40:85–108
DOI 10.1007/s10844-012-0217-4

Automatic discovery of Web Query Interfaces using machine learning techniques

Heidy M. Marin-Castro · Victor J. Sosa-Sosa · Jose F. Martinez-Trinidad · Ivan Lopez-Arevalo

Received: 21 December 2011 / Revised: 8 May 2012 / Accepted: 1 August 2012 /
Published online: 23 August 2012

Abstract The amount of information contained in databases available on the Web has grown explosively in the last years. This information, known as the Deep Web, is heterogeneous and dynamically generated by querying these back-end (relational) databases through Web Query Interfaces (WQIs) that are a special type of HTML forms. The problem of accessing to the information of Deep Web is a great challenge because the information existing usually is not indexed by general-purpose search engines. Therefore, it is necessary to create efficient mechanisms to access, extract and integrate information contained in the Deep Web. Since WQIs are the only means to access to the Deep Web, the automatic identification of WQIs plays an important role. It facilitates traditional search engines to increase the coverage and the access to interesting information not available on the indexable Web. The accurate identification of Deep Web data sources are key issues in the information retrieval process. In this paper we propose a new strategy for automatic discovery of WQIs. This novel proposal makes an adequate selection of HTML elements extracted from HTML forms, which are used in a set of heuristic rules that help to identify WQIs. The proposed strategy uses machine learning algorithms for classification of searchable (WQIs) and non-searchable (non-WQI) HTML forms

H. M. Marin-Castro (✉) · V. J. Sosa-Sosa · I. Lopez-Arevalo
Center of Research and Advanced Studies of the National Polytechnic Institute,
Information Technology Laboratory,
Victoria City, Tamaulipas, Mexico
e-mail: hmarin@tamps.cinvestav.mx

V. J. Sosa-Sosa
e-mail: vjsosa@tamps.cinvestav.mx

I. Lopez-Arevalo
e-mail: ilopez@tamps.cinvestav.mx

J. F. Martinez-Trinidad
National Institute for Astrophysics, Optics and Electronics Tonantzintla,
Puebla, San Andrés Cholula, Mexico
e-mail: fmartine@inaoep.mx

using a prototypes selection algorithm that allows to remove irrelevant or redundant data in the training set. The internal content of Web Query Interfaces was analyzed with the objective of identifying only those HTML elements that are frequently appearing provide relevant information for the WQIs identification. For testing, we use three groups of datasets, two available at the UIUC repository and a new dataset that we created using a generic crawler supported by human experts that includes advanced and simple query interfaces. The experimental results show that the proposed strategy outperforms others previously reported works.

Keywords Deep Web · Hidden-Web databases · Web Query Interfaces · supervised classification

1 Introduction

The explosive growth of the Internet has turned the Web into one of the most important sources of information containing a large number of available databases. Recent studies have found that more than 60 % of the information available on the Web is located in Hidden-Web databases that are accessed by special HTML forms (Lu and Li 2010). This information is known as the Deep Web. The HTML forms that allow accessing these databases are called Web Query Interfaces (WQIs). A WQI has HTML elements such as selection list, text input box, radio button and checkbox, etc., as well as attributes and labels. Unlike the surface Web, where data are available through URL links, data from Deep Web are stored in Hidden-Web databases that only produce results dynamically in response to a direct request made through WQIs, being these the only means for accessing that information (Bergman 2001).

Figure 1 conceptually illustrates the Deep Web, where there are numerous Hidden-Web databases partially distributed storing information related to different topics. The major information resources stored in the Deep Web are normally in non-textual format and content-rich databases. Further, the information in the Deep Web such as news, job postings, flight schedules, accommodation reservation, etc., is usually new, dynamically created and its content changes frequently.

Given the dynamic nature of the Web, new Web sites are added frequently and old Web sites are removed or modified constantly. This makes that the discovery of Hidden-Web databases containing specific domain information becomes a great challenge.

The identification of Web sites containing WQIs is useful for users that want to access information related to a specific topic. Consider a user with a specific need, for example “buy a book”. The topic in this case is “books”. Although the user can access a known and popular site as amazon.com to satisfy its needs, it would be more useful for the user to have a set of Web sites containing WQIs that allow him to query back-end databases in order to get the desired information, for example, the price and the delivery time. In this case, the user will not only query amazon.com but several other sites, increasing the probability to get the best price. As another example, consider a user searching Web sites for house rental without any idea about where to start. It would be useful for the user to find a web site with a list of WQIs representing hidden databases that belong to different agencies. Another application

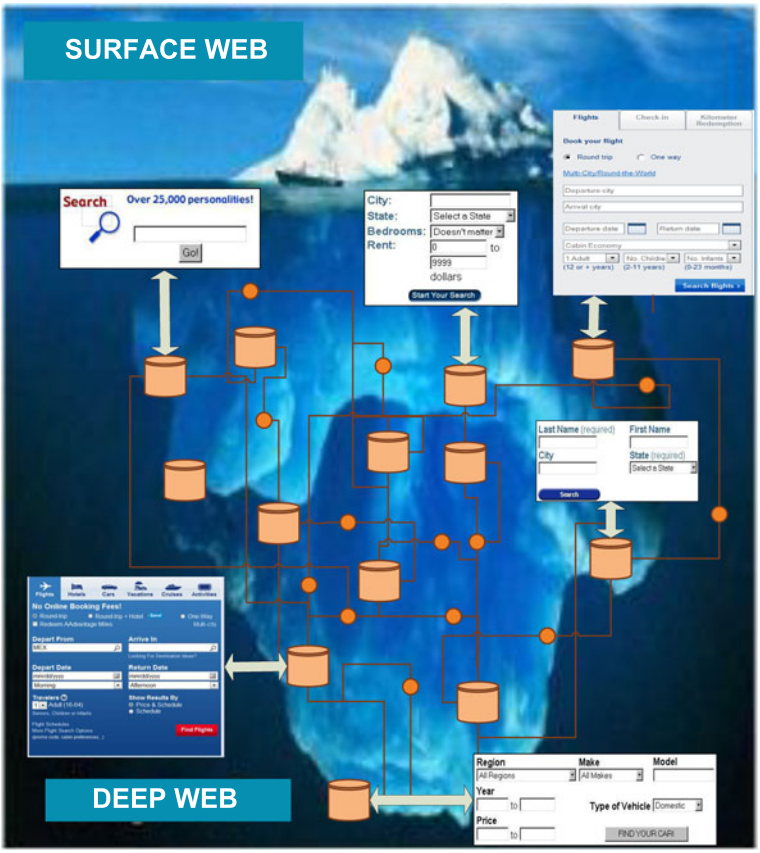

Fig. 1 Conceptual illustration of the Deep Web

of having a list of WQIs is to facilitate the construction of specific domain Deep Web meta-searchers. They could take this list as entry point in order to query and filter the information obtained from these databases. In this sense, it is clear that the first step for obtaining useful information from the Deep Web is to find the right WQI, situation that motives us to face the problem of the automatic detection of specific-domain WQIs.

There are several problems related to the access and the information retrieval from the Deep Web. One of the problems faced here is that Hidden-Web databases are very scattered distributed over the Web, even within specific domains, which makes difficult their location. Another problem is that most of the HTML forms contained in Web pages are not WQIs, such as HTML forms for discussion groups, logging, mailing list subscriptions, among others, which make more difficult the correct identification of WQIs. In addition, many of the traditional search engines cannot locate the content of the Deep Web. Even the most powerful as Google that exceeds one billion of Web pages in a query only access a small part of the great world of the Web.

In a Web page, when an HTML form is submitted, the Web browser sends to the HTTP server a request that includes the input information by using one of the following methods: *GET* or *POST* (Madhavan et al. 2008). With *GET*, the input data are included as part of the URL in the HTTP request. With *POST*, the input data are sent in the body of the HTTP request. Hence, the URLs obtained from forms that use *GET* are unique (and dependent on submitted values), while the ones obtained with *POST* are not. For a search engine to identify Web pages obtained from a *GET* method is a problem because these are not directly indexable.

Another challenging problem is that the WQIs are not created neither designed to be used and understood by computer applications. A simple task (from the position of an user) as to identify the content and to fill in a query form becomes a hard problem computationally (Shestakov 2008).

The content of sites in the Deep Web usually has implicit structures or schemes referred as semistructured Web pages (Madhavan et al. 2008). These pages present an unknown structure, which is deduced from the data contained in the Web pages. This structure can change frequently without notification. Another problem with the semi-structured information is that the schema of the database is not known by the accessing or storing mode. That is, the representation format does not contain information about the schema or the semantic of its contents. Moreover, the query capabilities of the WQIs can be limited due to the amount of data received by the user as well as the number of attributes contained in the interface itself. Such intermediate position of semi-structured data has motivated that many challenges of the Deep Web are currently tackled by the database community that is working with structured data and the information retrieval community dealing with unstructured content.

This paper introduces a new strategy for automatic discovering of WQIs. Different to previous approaches, the proposed strategy exploits the internal components of WQIs and the use of machine learning techniques for obtaining a best accuracy in the identification of WQIs. Several works reported in the literature for identification of WQIs (Cope et al. 2003; Wu et al. 2004; Zhang et al. 2004) have not provided a detailed study about the design, internal structure, number and type of HTML elements in a WQI that can be taken as a reference for its identification.

In Wang et al. (2011), the authors built some rules to identify WQIs, for example, they stated that a HTML form is considered a WQI if it has more than three attributes. However, this is not always true because an HTML form could be a simple query interface (SQI) having only one or two attributes. In some works as Barbosa and Freire (2007a), Cope et al. (2003), Wu et al. (2004) and Zhang et al. (2004) the identified WQIs are not validated, that is, they do not ensure if the identified HTML form allows to query a Hidden-Web database. In addition, the majority of these related works try to identify WQIs for fixed domains, which limits the application of those strategies to different contexts.

In this work we use features contained in HTML forms, like HTML elements, and their corresponding fields, to form *characteristic vectors* used in the classification task. These features are extracted without considering a specific domain of the application. Moreover, the majority of WQIs are designed with HTML, which does not express data structures and semantics.

The contributions of this work are:

– A novel strategy for automatic discovery of WQIs based on an adequate selection of components taken from HTML forms and the use of machine learning

techniques for their classification. The proposed strategy correctly work with a reduced training set that represents the relevant objects of a WQI independently of the domain. We emphasize that the proposed strategy can be used with any type of WQIs (advance and simple query).
- Eight heuristic rules to discriminate non-searchable (non-WQI) and searchable (WQI) HTML forms. These rules are based on the analysis of code segments in an HTML form.
- A new repository built manually by human experts. This repository contains Advanced and Simple WQIs related to five different domains (books, car rental, hotels, jobs and movies) and a dataset with negative examples (non-WQIs). The goal of this repository is to provide datasets for supporting research focused on exploring and integrating Hidden-Web databases. This repository is available for download at http://www.tamps.cinvestav.mx/∼hmarin/Repository

The rest of the paper is organized as follows. The next section discusses the related work for identification of WQIs. Section 3 describes some concepts about the study area. Sections 4 and 5 give a brief description of classification task and the prototype selection algorithm used in this work. Section 6 introduces our strategy for automatic identification of WQIs. Section 7 describes and discusses the experimental results and finally, Section 8 presents the conclusions of this work.

2 Approaches for automatic identification of WQIs

Most of the proposals to solve the problem of automatic identification of WQIs can be categorized into two approaches: Pre-Query and Post-Query (Ru and Horowitz 2005). The Pre-Query approach exploits the internal content provided by HTML forms such as control elements, attributes, values of attributes and user's tags. This approach only relies on visible features of HTML forms. On the contrary, the Post-Query approach is based on the identification of WQIs from the resulting pages retrieved by submitting probing queries through HTML forms. This approach depends on the correct construction of queries for a given domain. The Post-Query approach is difficult to apply to HTML forms with multi-attributes because they require to be automatically filled out with valid values. Sections 2.1 and 2.2 describe some of the reported works related to the discovery of WQIs using Pre and Post-Query approaches previously mentioned.

2.1 Pre-query approach

The automatic identification of WQIs has been mainly faced with Pre-Query techniques, (Barbosa and Freire 2005, 2007a; Barbosa et al. 2010; Wang et al. 2008, 2011; Zhang et al. 2010; Li et al. 2010; Kabisch et al. 2009; Zhang et al. 2004), because the Post-Query techniques can not be easily adapted for multi-attribute forms. In Barbosa and Freire (2005), the authors propose a new strategy called Hierarchical Form Identification (HIFI). This strategy is based on the decomposition of the feature space for the HTML forms and uses the best suited learning classifiers for this kind of application. It uses a Form-Focused Crawler (FFC) that extracts the characteristics of the Web pages that FFC identifies as WQIs in order to focus the search on a specific topic. The crawler uses two classifiers to guide its search: a generic

classifier and a specialized classifier. The generic classifier allows to eliminate HTML forms that not generate any query for Hidden-Web databases. The specific classifier identifies the domain of the HTML forms selected by the generic classifier. The decomposition of the feature space uses a hierarchy of form types through the selected HTML forms, followed by an analysis of WQIs related to a specific domain. The authors used structural patterns to determine whether an HTML document is or not a WQI. They empirically observed that the structural characteristics of an HTML form can determine whether the form is or not a WQI. However, this strategy has some limitations: it requires substantial manual tuning and the form set retrieved by FFC is very heterogeneous. In Barbosa and Freire (2007a, b), the authors present a new Adaptive Focused Crawling (ACHE) for locating Deep Web entry points, which classifies online databases based on features that can be extracted from HTML forms. Their specialized classifier uses the textual content of an HTML form to determine its domain. For this task, they use the C4.5 and Support Vector Machine (SVM) classification algorithms. Unfortunately, the ACHE framework can not efficiently handle very sparse domains. Moreover, the ACHE framework is complex to implement.

Wang et al. (2011) present a framework for locating Hidden-Web databases based on a focused crawler assisted by an ontology. Their framework is composed by three classifiers: a Web Page Classifier (WPC), a Form Structure Classifier (FSC) and a Form Content Classifier (FCC). The WPC searches interesting Web pages by means of analyzing their features. FSC determines whether these pages contain searchable forms by analyzing their structural characteristics and the FCC identifies whether searchable forms belong to a given domain. The authors implemented an ontology which contains concepts and relations of DOCM (Domain Ontology Concept Model) from searchable forms and resulting pages. To locate interesting Web pages that may contain searchable forms belonging to a given domain, the crawler follows two strategies: The crawler starts from a Web page which is classified as belonging to a topic. From that Web page the crawler follows the hyperlinks found on that page until a specified level of depth is reached. During the search of interesting Web pages the crawler builds two vectors, a page feature vector and a topic vector to find similarity between the page found and the source page. For the FSC the authors built a decision tree from which they obtained a rules set to classify HTML documents as WQI or non-WQI. However, in the FSC the rules derived from the decision tree do not always keep true, for example, in FSC the rule 3 establishes that if there exists a $<form>$ tag and the number of attributes is less than 3, then this form is non-searchable, which is not always true, because simple query forms are characterized by at least one attribute.

In Wang et al. (2008) the authors propose a three-step framework to automatically identify domain-specific Deep Web entries. They use three classifiers working in a hierarchical fashion to guide the Deep Web Crawler. The three classifiers are: a form structure classifier, a form text classifier and a page text classifier. To identify whether a form is a domain-specific searchable form or not, the authors use structural and textual form features. To extract textual features from HTML forms they use two methods, the FT method, which uses the HTML code of the form and splits it using non-alphanumeric strings, and the PT method, which extracts those textual features from the HTML form perceived by one human at first time as well as the action attribute of the form, which allows to send the form data to the Web server. Eight classifiers are used for each domain obtaining an accuracy of 0.88

for the eight representative domains. Unfortunately, the authors do not make an adequate selection of HTML components and their three-step framework can not handle different domains efficiently.

Cope et al. (2003) use an automatic feature generation technique to depict candidate HTML forms and a C4.5 decision tree to classify them. In their two testbeds (ANU collection and random Web collection), they get an accuracy above 85 and 87 % respectively. However, the proposed technique in that work can not completely identify if a Web page is or not a WQI. In Zhang et al. (2004), Zhang et al. hypothesize the existence of a hidden syntax that guides the creation of query interfaces from different sources. Such hypothesis allows to transform query interfaces into a visual language. In that work, authors stated that in the automatic extraction task is essential to understand the content of a WQI. This task is rather "heuristic" in nature so it is difficult to group pairs of the closest elements by spatial proximity or semantic labeling in HTML forms. One proposed solution for this problem is the creation of a 2P grammar, which allows to identify not only patterns in the WQIs but also their precedence. However, this grammar is over sized with more that 80 productions than were manually derived from a corpora with 150 interfaces.

The above described methods are based on the use of a framework with three-steps or three classifiers. However, most of them present some limitations. These works do not exploit completely the internal content of HTML forms to obtain a better identification of WQIs without discarding some of them or including some that are not WQIs.

Several works have focused on different aspects of information retrieval and integration of Deep Web such as: search and identification of Web Query interfaces (WQIs), database schema matching, integration of data, building a unified scheme, among other aspects. Our work focuses on the automatic identification of WQIs, which represents an important and challenging task in the retrieval and integration of Deep Web. Table 1 shows a summary of related works that face the WQI automatic identification problem using the pre-query approach. These works are based on a

Table 1 Different approaches to WQIs identification

Ref.	Technique	Dataset	Accuracy (%)
Barbosa and Freire (2005)	Hierarchical decomposition of characteristic space (HIFI)	TEL-8 dataset of the UIUC repository	90
Cope et al. (2003)	Automatic generation of features based on a limited set of HTML tags	ANU collection	85
Kabisch et al. (2009)	Based on nine domain-independent general rules	TEL-8 and ICQ datasets of the UIUC repository	87.5
Wang et al. (2011)	Framework WFF based on a focused crawler assisted by an ontology	TEL-8 dataset of the UIUC repository	94
Zhang et al. (2004)	Grammar 2P and tree parse	TEL-8 and ICQ of the UIUC repository	85

visual analysis of characteristics of the HTML documents that were tested and heuristic techniques based on textual properties, such as the number of words, similarity between words, etc., as well as layout properties such as position of a component, distance among components, etc.

In Table 1, the second column indicates the technique used for identification of WQIs, in the third column the name of dataset used to experimental evaluation and finally, in the last column, accuracy represents the percentage of correctly identified WQIs over all HTML forms that were included in the testing dataset. Table 1 can be used as a good reference but not as a fair comparison because the presented works are not using the same testing scenario. The information describing the datasets included in this table was extracted from the following references:

- In Barbosa and Freire (2005) the authors extracted 216 searchable forms (WQIs) from the UIUC repository (UIU 2003), and they manually gathered 259 non-searchable forms (non-WQIs) from the Web as negative examples.
- Kabisch et al. (2009) used 100 WQIs from the ICQ dataset and only 50 % from the TEL-8 dataset because the other WQIs referred to no longer existing Web servers. Both datasets belong to the UIUC repository.
- In Wang et al. (2011) the authors selected four domains from the TEL-8 dataset of UIUC repository: Airfare, Jobs, Hotels and Movies.
- In Zhang et al. (2004) the authors tested their approach from the TEL-8 dataset of the UIUC Repository using 150 sources in three domains (i.e., Airfares, Automobiles and Books.)

Most of the works described in Table 1 used the datasets of the UIUC repository (UIU 2003), in particular the TEL-8 dataset, except the work developed by Cope et al. (2003), where the authors used a repository of the Australian National University with 149 WQIs and 70 non-WQIs. However, these works used a different number of instances from the UIUC repository because some Web servers were no longer available or the proposed identification method was not able to work with the internal structure of all WQIs included in this repository. The TEL-8 and ICQ datasets from the UIUC repository were also used to evaluate and compare our proposed method for WQIs identification. An additional and recently created dataset was used in order to enrich our validations. Some disadvantages of the previous presented works are:

- The human intervention is constantly required to perform the identification of WQIs.
- The objective of most of them is to determine the domain of the WQIs without performing the automatic identification of WQIs
- Lack of a clear, precise and defined scheme for automatic identification of WQIs

2.2 Post-query approach

Lin and Zhou (2009) propose a method for a systematic schema identification of Web databases with simple query interfaces (SQIs). They design a probing strategy by analyzing the hit rate of the query words and the reappearance word frequency in the result pages. The problem of the interface schema identification is defined as regenerating the full condition query which has only one text input field, usually labeled with simple words like “keywords” or “search”. In other works, as in Jiang

et al. (2010), the authors propose a novel Deep Web crawling framework based on reinforcement learning, where the crawler is regarded as an agent and the Hidden-Web databases as the environment. The agent perceives its current state and selects an action (query) to submit to the environment according to a value.

Jiang et al. (2009) propose a novel Deep Web crawling method. They argue that the key for crawling the Deep Web is to submit promising keywords to the query forms and retrieve Deep Web content efficiently. These keywords are encoded as a tuple by their linguistic, statistics and HTML features so that a harvest rate evaluation model can be learned from the issued keywords. In Liu et al. (2004) the authors introduce the notion of dynamic probing and study its effectiveness under a probabilistic framework, which allows them to contact a few web databases.

The pre-query approach is able to work with a great variety of heterogeneous forms. This approach is more used in the identification of WQIs than the post-query approach. Therefore the identification, characterization and classification of WQIs continue being a challenging research topic.

In the next section we describe with detail some concepts related to the study area.

3 Preliminaries

Definition 1 A Deep Web site is a collection of Web pages $W = \{P_i | 2 \leq i \leq n\}$ where P_i denotes a Web Page and each Web Page is formed by a 4-tuple $< U, R, F, H >$ where U represents the URL address of the Web page; R is the set of information resources as textual information, non-textual information (static and animated images, audio, video) and interactive information (interactive text and illustration); and F is a set of forms that provides interaction with the server-side database and H represents the set of hiperlinks to any other Web site.

Definition 2 A Hidden-Web database is an organized and logical collection of data that is available for online query. Each Web database is composed by a 4-tuple $< N, D, WQI, RQI >$. N denotes the name of the Web database, D makes reference to the domain that the database belongs to. WQI and RQI represent the Web Query Interfaces and the Web pages generated in response to the submitted queries.

Definition 3 An HTML form is a segment of HTML code in a Web page P delimited by $< form >$ and $< /form >$ tags. This segment contains a set of control elements $C = \{c_i | 1 \leq i \leq n\}$ that can be described using the 3-tuple $< name, label, domain >$, where *name* corresponds to the name of the field associated to this element, *label* is a string that describes this element(human-readable) and the *domain* is the set of valid values that the field can take. Every control element could contain a set of attributes $A = \{a_i | 1 \leq i \leq m\}$. Some examples of HTML control elements are $C = \{input, button, select, option, option\ group$ and $textarea\}$ and possible attributes are $A = \{type, name, value, checked, alt, src, size, readonly$ and $maxlenght\}$. The attributes are associated to the control elements and these are associated to an HTML form.

Definition 4 A Web Query Interface (WQI) is defined as a HTML searchable form that is intended for users that want to query a Hidden-Web database. WQIs have

a heterogeneous schema of design, semantic and value; they may change frequently and without notice. Moreover, each WQI may have a limited query capability.

Definition 5 A Simple Query Interface (SQI) is defined as a WQI that accepts keywords for doing queries. A SQI is composed by one or two attributes which not always describe the domain of the SQI.

Definition 6 An Advanced Query Interface (AQI) is defined as a WQI that is composed of multi-attributes making reference to a given domain.

Next, we describe three classification algorithms used in the identification of WQIs.

4 Classification

We face the WQIs identification problem, where the WQIs design, semantic and value are heterogeneous. Specifically, we consider SQIs and AQIs both composed of HTML elements, attributes and labels related with a given subject. To divide the set of HTML forms into searchable forms (WQIs) and non-searchable forms (non-WQIs) we use classification algorithms.

In recent years, the classification techniques have gained great popularity due to its effectiveness to solve different problems such as drug discovery, in banking to predict the behavior of client accounts, medical diagnosis, prediction of climate, fraud detection, character recognition, detection of abnormality in chromosomes, among others.

In the WQIs classification task, we define a learning function $L : X \longrightarrow Y$, called classifier. X represents a set of input feature vectors (in this context, HTML forms) $X = x_1, x_2, x_3, ..., x_n$ where $x_i = a_1, a_2, a_3, ..., a_k$ and a_j represents a single feature (an HTML control element). Y represents a set of labels or classes (WQI or Non-WQI). In the x_i input feature vector, a_j contains the number of occurrences of the j-th element in a HTML Form (x_i). Some HTML control elements considered in the input vector are textboxes, checkboxes, buttons, texts, images, selects, among others.

In our scenario the learning set contains feature vectors obtained from HTML forms and the labels (searchable and non-searchable) determine if a form is or not a WQI. We selected three classification algorithms Naive Bayes (Mitchell 1997), J48 (Quinlan 1986) and SVM (Platt 1998) due to their high accuracy rate.

5 Prototypes selection

The supervised classification task is a process for assigning a class to unseen objects or prototypes. In order to assign the class to a new prototype, a previously assessed prototype set is provided to the classifiers, known as the training set T. However, not all information in a training set T is useful (that is, superfluous prototypes could be noisy or redundant), therefore it is possible to discard some irrelevant prototypes. This process is known as prototypes selection. We use prototypes selection with the objective of removing redundant information and working with representative information, in our case, with feature vectors corresponding to examples of HTML forms.

The main goal of a prototype selection method is to obtain a set $A \subset T$ such that A does not contain superfluous prototypes and $P(A) \cong P(T)$, where $P(X)$ is the classification accuracy obtained using X as the training set. Through prototype selection, the training set size is reduced, which could be useful for reducing the run times in the classification process, particularly for instance-based classifiers.

5.1 Prototype Selection by Clustering (PSC)

We used the prototype selection method named PSC (Prototype Selection by Clustering) (Olvera-Lopez et al. 2007). This method is based on clusters that are created from a training set, these can be either homogeneous (all the prototypes belong to the same class) or non homogeneous (the prototypes belong to two or more classes). The PSC works as follows: PSC starts creating n clusters from a T training set, after that, PSC works over each cluster. If the cluster C_j is homogeneous then the objects in C_j are central objects, that is, they do not lie on critical regions therefore they could be discarded from T without affecting the classification accuracy. PSC finds the nearest object to the mean (mean object) and discards the remaining objects from C_j, in this way C_j is represented by the mean object. If C_j is non-homogeneous then there are in C_j some objects located at critical regions, that is, border objects. In order to find the border regions, PSC finds the minority class objects (objects that do not belong to the most frequent class in C_j) since these objects are nearest to a border delimited by the most frequent class in C_j. Once the minority class objects have been found, the border objects in the most frequent class are the nearest objects to each minority class object, and by analogy, the border objects in the minority class are the nearest objects to each border object in the majority class. Finally, the prototypes selected by PSC are the mean objects from each homogeneous cluster and the border objects from each non homogeneous cluster (Olvera-Lopez et al. 2007).

In PSC was used the C-means (García-Serrano and Martinez-Trinidad 1999) clustering algorithm for its simplicity and effectiveness. However, any clustering algorithm could be used. PSC requires, as a parameter, the value of C (number of clusters) for the C-means algorithm. The C-means algorithm starts with an initial partition and then, it tries all possible moving or swapping of data from one group to another iteratively to optimize the objective measurement function. The objects must be described in terms of features in such a way that a metric can be applied for the distance evaluated. The C-means algorithm proposes a distance function to handle quantitative and qualitative features. The distance between two objects is computed evaluating the distance between quantitative features (with an Euclidean distance) plus the distance between qualitative features, using the chi-square distance, where each value of a qualitative feature is coded as a binary feature (García-Serrano and Martinez-Trinidad 1999).

6 A novel approach for automatic discovery of WQIs

This section introduces the proposed strategy for automatic discovery of WQIs based on the automatic extraction of HTML elements from HTML forms, the use of heuristic rules and machine learning techniques. We first carry out an analysis, manipulation and representation of an HTML document by an automatic extractor.

In the analysis phase, we apply eight rules to filter static Web pages and Web pages than do not contain searchable forms. Next, we represent the content of a HTML form building a characteristic vector that contains the number of occurrence of HTML elements that are inside HTML segments delimited by the $<form><$ $/form>$ tags.

Accordingly, we first describe the features of a WQI (also called searchable form) independently of its domain and how to filter non-searchable forms with the use of eight rules. We also describe what is the importance of adequate selection of HTML elements that allow to better represent a WQI. Finally, we use a classification algorithm to classify HTML forms in WQI or non-WQIs. For all of the illustrations and descriptions of this section we considered AQIs and SQIs from the UIUC repository (UIU 2003) that was used for the evaluation of our method; see Section 7 for details.

6.1 WQIs features

A Web page is an information resource that can be accessed through a Web browser. This information is usually in HTML or XHTML format, and may provide navigation

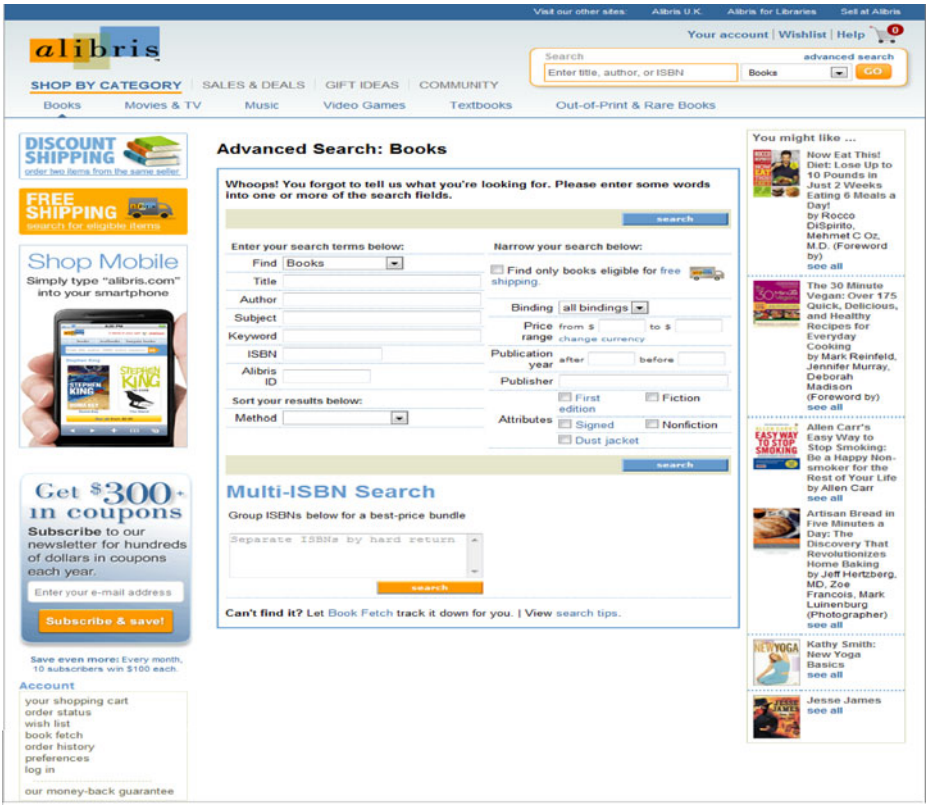

Fig. 2 Web page of Alibris library

to other Web pages via hypertext links. Moreover, the Web pages frequently are formed by more than one WQIs corresponding to a given domain as it is showed in Fig. 2, where there are three WQIs, one AQI and two SQIs related to the topic of books.

The WQIs are characterized for presenting different heterogeneous schemes of design. Most of the time, WQIs manage a small vocabulary and some common features that belong to the same domain. A WQI can frequently change and without notice. Their query capabilities may be limited depending on their design. A main property that characterizes a Web page that contains WQIs is its semi-structured content as XML files, unstructured content (videos, images, text, files), hypertext links to other Web pages and WQIs related to a topic.

For example, let us consider the WQIs shown in Fig. 3 related to the airline flights topic. Each one of them has a different design and query capabilities according to their design. In these WQIs the predominant HTML control elements are *selection lists*, *checkboxes* and the existence of one single button to submit a flight search. This button can contain a string related to the string "search" such as Find Flights, Search Flights, Show Flights, Search, Go, among others. The four WQIs present related

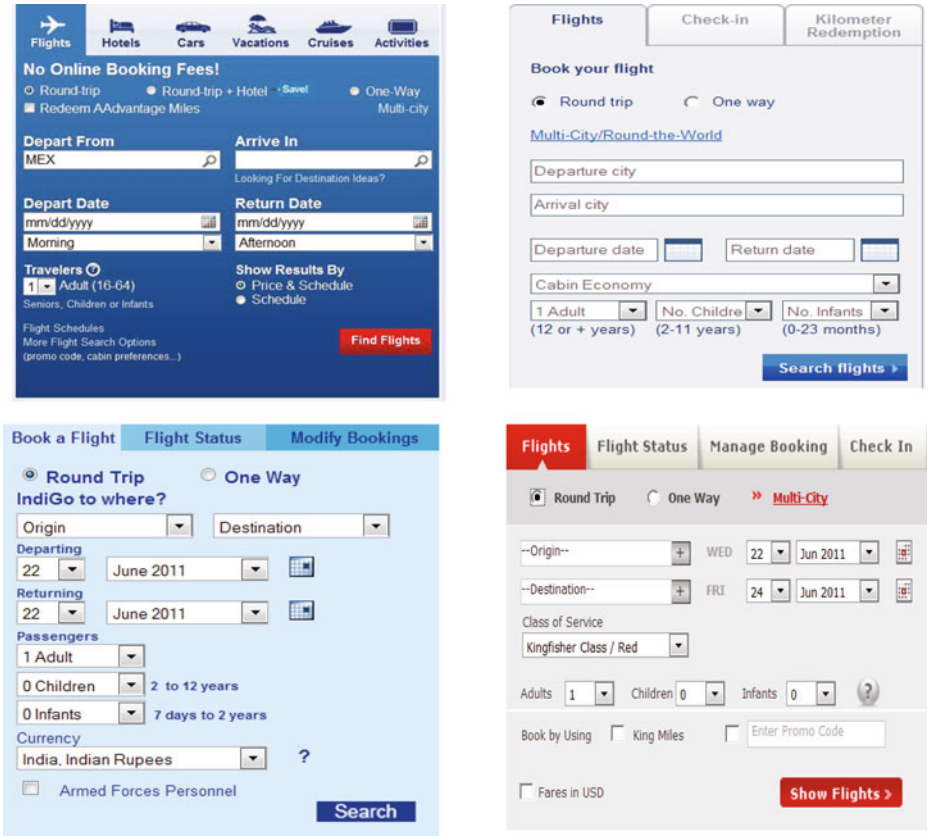

Fig. 3 Example of WQIs of airlines-flights

vocabulary with different semantics as the strings *Depart from*, *Departure city* and *origin* among others.

6.2 Filtering of HTML documents

We created eight heuristic rules that are the first step for filtering the documents without HTML forms or with non-searchable forms (non-WQI). In this context, the rules are taking into account the fact that the input dataset could have HTML documents that do not include HTML forms. These rules were created from an extensive heuristic analysis of the content of HTML documents collected by a generic crawler. The obtained WQIs were identified and extracted manually by human experts in order to create a fresh and more extended WQI testing dataset. The features obtained from these WQIs were analyzed and compared with those found in the dataset included in the UIUC repository to determine new designs and structures.

According to our study, it is still valid to say that the first basic feature that allows the identification of a WQI is the existence of the $<form></form>$ tags. Some simple design characteristics were also found in the HTML documents that help to determine if there is a WQI embedded in the document. These characteristics were used to define the set of heuristic rules that will filter some HTML forms that are not WQIs before the classification process. The rules identify the segments of HTML code delimited by the $<form>$ and $</form>$tags, and analyze every HTML control element c_i to verify if some common values $V = \{$"*search*", "*find*", "*go*", "*buy*"$\}$ have been assigned to any attribute a_i (e.g. "value", "class", "name", "id") that is included in c_i. The heuristic rules applied to each HTML document in a dataset are the following:

- **Rule 1:** If a $<form>$ tag does not exist in the HTML document then this document does not have a searchable form (WQI).
- **Rule 2:** If an HTML segment delimited by the $<form>$ and $</form>$ tags exists and contains a control element that includes the attribute $type =$ "*password*", then this form is a non-searchable form (non-WQI).
- **Rule 3:** If an HTML segment delimited by the $<form>$ and $</form>$ tags exists and contains a control element that includes the attribute with $type =$ "*mail*", then this form is not a searchable form.
- **Rule 4:** If an HTML segment delimited by the $<form>$ and $</form>$ tags exists and contains an "*input*" element with the attribute $type =$ "*button*" and any attribute $a_i = v_j$, where v_j is any value taken from the set of common values V, then this form is a searchable form.
- **Rule 5:** If an HTML segment delimited by the $<form>$ and $</form>$ tags exists and contains an "*input*" element with the attribute $type =$ "*image*" and any attribute $a_i = v_j$, where v_j is any value taken from the set of common values V, then this form is a searchable form.
- **Rule 6:** If an HTML segment delimited by the $<form>$ and $</form>$ tags exists and contains an "*input*" element with the attribute $type =$ "*submit*" and any attribute $a_i = v_j$, where v_j is any value taken from the set of common values V, then this form is a searchable form.
- **Rule 7:** If an HTML segment delimited by the $<form>$ and $</form>$ tags exists and contains a "*button*" element with any attribute $a_i = v_j$, where v_j is any

value taken from the set of common values V, then this form is a searchable form.

– **Rule 8:** If an HTML segment delimited by the $<form>$ and $</form>$ tags exists and contains an "*img*" element with any attribute $a_i = v_j$, where v_j is any value taken from the set of common values V, then this form is a searchable form.

6.3 Selection of HTML elements

An important part in the recognition of WQIs is to characterize the HTML forms found on the HTML documents. This characterization is done through the selection of certain HTML control elements, such as *textb oxes*, *checb oxes*, among others. The basic idea for an adequate selection of elements in a form f_j is to determine which HTML control elements occur in both $AQIs$ and $SQIs$ more times. By using this characterization, we can determine when the identified forms are non-searchable forms, such as discussion group forms, logging forms, mailing list subscription forms, among others.

Initially we identify the existence of a form f_j in the Web page W_i by the identification of the $<form></form>$ tags. Next, we create a feature vector $X_i = \{a_1, a_2, a_3, ..., a_n\}$ that represents f_j, being a_i the number of occurrences of the $i-th$ HTML control element in f_j. Once the tags $<form></form>$ are identified, we build a table $H =< tag, occurrence >$ to store the HTML control elements ($b utton$, $text$, $image$, $select$, $etc.$) inside the f_j HTML form, together with their number of occurrences.

In Fig. 4 we can observe the most common HTML control elements that occur at least once in a HTML form f_j. In that figure, each bar represents a WQI and

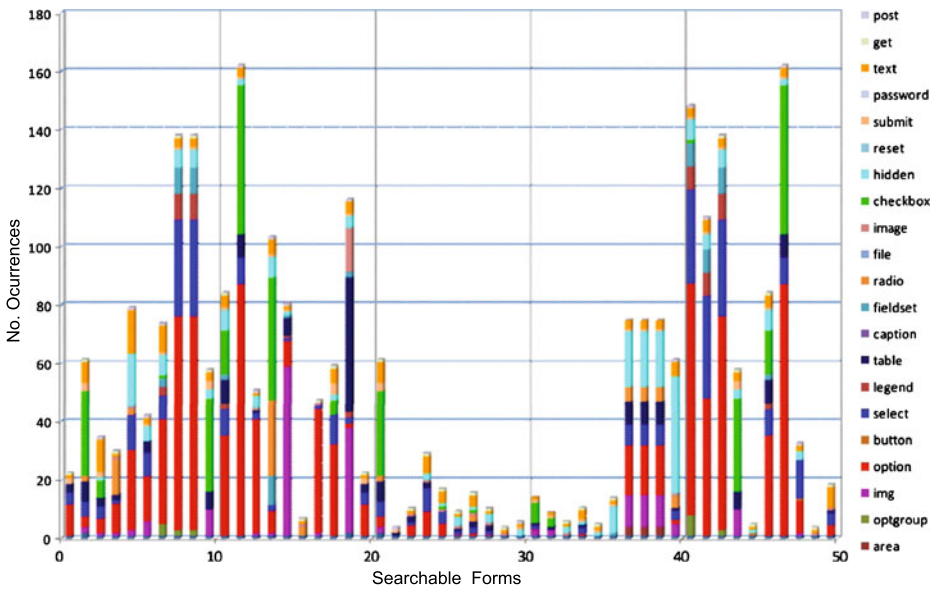

Fig. 4 Number of occurrences of HTML elements in independent-domain WQIs

each different color in the bar represents a HTML element of the WQI. The most predominant color in the bar corresponds to the control element with the highest number of occurrences in the WQI. Note that most of the AQIs are formed by tags such as $<option>$, $<checkbox>$ and $<select>$, that occur much more than other elements such as the elements of type *text*, *hidden*, *img* that appear at least once. SQIs are formed by three basic tags: $<select>$, $<text>$ and $<hidden>$. These WQIs show lack of $<option>$ and $<checkbox>$ tags. From this study, it can be infered that the majority of SQIs and AQIs are formed by the tags $<select>$, $<option>$, $<checkbox>$ and $<text>$. So we establish that when an HTML form includes these four tags, this form most likely is a WQI.

On the other hand, in Fig. 5 we can observe that in non-searchable forms the tags with the highest number of occurrences are the tags $<hidden>$, $<post>$ and $<text>$. In these kind of HTML forms, it is notorious the absence of the $<option>$ and $<checkbox>$ tags.

The implementation of our proposed strategy is described in Algorithm 1. We begin with a set W of HTML documents (containing WQIs) from the UIUC repository (UIU 2003) and a set N of HTML documents (without WQIs) that were recovered by a generic crawler. For each HTML document, to identify HTML forms and count the number of occurrences for each element inside that form. The aim is to create characteristic vectors that describe the HTML forms in the HTML document and classify them into WQI or non-WQIs. The result of this characterization process is a file containing the characteristic vectors for the HTML forms identified in the HTML document. A characteristic vector includes information such as number of occurrences of the HTML control elements and the values of true or false in relation to the existence of the strings *get*, *post* and *search* in each set. This file serves as input to the classifiers (Naive Bayes, J48 or SVM), which determine the class of each URL, in this case if it contains or not a WQI.

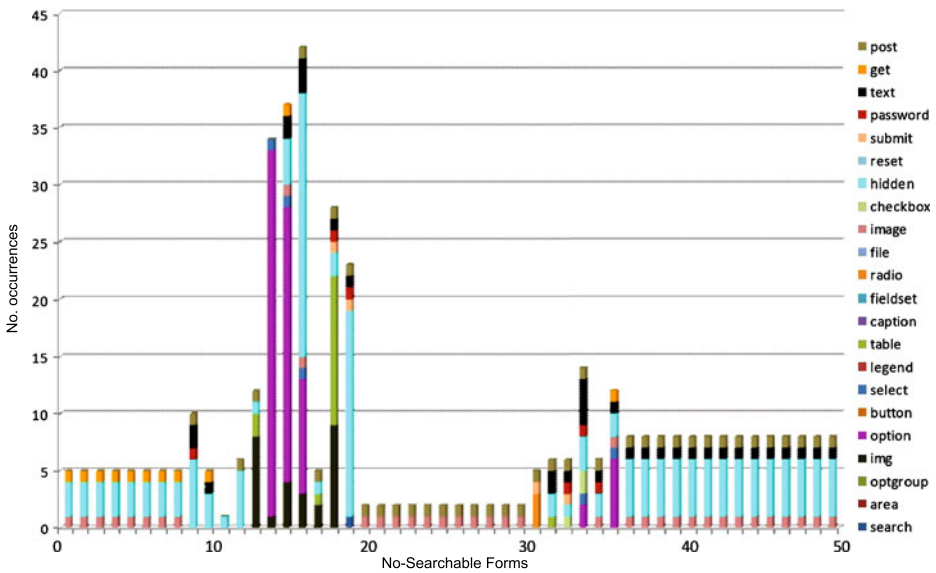

Fig. 5 Number of occurrences of HTML elements in non-searchable forms

Algorithm 1 Automatic Identification of WQIs

Require: W: HTML documents (WQIs), N: HTML documents (non-WQIs), *Extractor*: Jericho, *classifier*: Naive Bayes, J48 and SVM,
Ensure: Output: Instances classified as WQIs or non-WQIs

```
Search keywords: <form> </form> in W and N
if EXIST(keywords) then
    Table = label < String, Integer >
    Definition of HTML labels:
    "select", "button", "text",
    "checkbox", "hidden", "radio", "file",
    "image", "submit", "password", "reset"
    "search", "post", "get"
    for each HTML segment from <form> to </form> do
        Call to labels = Extractor(HTML segment)
        if (label = definite label) then
            Table = Table(label, counter + 1)
        end if
    end for
    file =< Table(label, counter), tag >
end if
Classify(classifier, file)
```

The task of automatic extraction of HTML control elements inside a HTML form is based on the use of the Jericho HTML Parser (Jericho HTML Parser 2010), which analyze and manipulate parts of an HTML document.

Fig. 6 Example of the execution of the HTML extractor on a single form

```
**********************************************************

URL : "http://jobsearch.monstertrak.monster.com/"

 SEARCHING HTML FORMS

HTML FORM: 1

----------------------------------------------------------
----------------------------------------------------------

select       : 1
button       : 0

----------------------------------------------------------

FILE         : 0
HIDDEN       : 0
RESET        : 0
SUBMIT       : 1
TEXT         : 1
RADIO        : 0
CHECKBOX     : 0
search       : 0
PASSWORD     : 0
IMAGE        : 0

----------------------------------------------------------

get  : 0
post         : 1
```

A partial result of the output in the extractor module is shown in Fig. 6. The extractor displays the URL being analyzed and the statistics of HTML elements located in the forms that are identified in the current URL.

7 Experimental results

In this section, the experiments that were carried out to evaluate the proposed strategy and their results are described. The main goal was to demonstrate how the adequate selection of HTML elements and the use of heuristic rules can improve the accuracy and performance of several well known machine learning algorithms in the WQI identification process.

In our experiments we used the WQIs contained in TEL-8 and ICQ, two datasets from the University of Illinois at Urbana-Champaign (UIUC) repository, as positive samples. HTML forms that do not generate queries to Hidden-Web databases, such as logging forms, discussion interfaces groups, HTML subscription mail lists, etc., were used as negative samples. These HTML forms were manually collected from the Web, independently of their designs and topic. Additionally, a second repository that includes positive and negative examples was created by human experts using a generic crawler. Both repositories were used to prove the effectiveness of the proposed strategy for WQIs identification.

We carry out three experiments with the aim to evaluate different aspects of our strategy: (1) Evaluation of the accuracy of the WQIs classification task using the UIUC repository, our heuristic rules and the classification algorithms Naive Bayes, J48 and SVM; (2) Calculation of the effectiveness of our WQIs identification strategy targeting four specific domains, *jobs*, *hotels*, *movies* and *airfares*, using the decision three J48 algorithm with a reduced training set that was obtained by the prototypes selection algorithm (PSC); (3) Evaluation of the accuracy of our proposed strategy using a recent dataset repository.

7.1 Experiment 1

The objective of this experiment is to show the advantages of using a training set that includes the best attributes that describe WQIs. The attributes selection was based on the study discussed in the Section 6. In this experiment we compared our results against the ones reported in Barbosa and Freire (2005), using the same pre-query approach, the same dataset, the same classifiers and the same cross validation. By doing this, we were able to state that if our experiment showed better results in the classification process (accuracy or error) was due to the better feature selection in the training set.

In the experiment 1, we considered one corpus of HTML forms called *corpus 1*, that includes positive and negative examples. This corpus was formed with 216 positive examples (WQIs) taken from the dataset TEL-8 and 259 negative examples (non-WQIs) manually extracted by human experts from a set of Web pages that were retrieved using a generic crawler. The corpus 1 was also used, in the analysis mentioned in Section 6.3, in the features selection process. The following are the 14 selected features: the number of images, buttons, input files, *select* labels, *submit* labels, *hidden* labels, *resets* labels, *radio* labels, *textboxes*, *checkboxes* and the presence of

the following strings: *password*, *get*, *post* and *search*. The accuracy rate was calculated with (1), using the Naive Bayes, J48 and SVM algorithms (using the Sequential Minimal Optimization -SMO- algorithm at various degrees of complexity) to classify HTML forms in WQIs and non-WQIs. The accuracy represents the percentage of correctly identified WQIs over all the identified forms. In (1), TP represents the number of True Positives, TN the number of True Negatives, FP the number of False Positives and FN False Negatives.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

During the training and testing task, the predictive model was evaluated using the *k-fold cross validation* technique (Witten et al. 2000), which divides randomly the original sample of data into k sub-sets of (approximately) the same size. From the k sub-sets a single subset is kept for testing the model and the remaining $k-1$ sub-sets are used as training data. The *cross-validation* process is repeated k times (the folds), where each of the k sub-sets is used exactly once as testing data. The average of results in the k folds is obtained to produce a single estimate. The advantage of cross validation is that all the observations are used for training and testing. In our experiments, we used different values for k in order to have 50 different tests.

Table 2 shows the accuracy obtained by our proposed strategy, using the corpora mentioned above. The results are compared with the work presented in Barbosa and Freire (2005) for WQIs automatic identification.

We used three of the best algorithms for classification with high accuracy (Hall et al. 2009): Naive Bayes, J48 and SVM. Also, in Table 2, the results of a second corpus called *corpus 2* were included. This was formed with positive examples of ICQ dataset from UIUC repository and negative examples manually collected by human experts with the only aim to show the performance of our proposed strategy in WQIs identification using a different dataset under the same classification algorithms. As it can be seen in Table 2, the classification algorithm J48 obtains the best accuracy compared with the other two algorithms because J48 works fine in presence of not relevant attributes.

In Barbosa and Freire (2005), the authors used the classification algorithms Naive Bayes, J48, Multilayer perceptron and SVM to classify WQIs. However, the accuracy rate that they obtained for the identification of WQIs is lower compared to the results that we achieved. It is because they did not carry out a detailed study of the HTML control elements that provide more information on the characterization of WQIs, as we have done in this work.

Table 2 Comparison of accuracy obtained by the Naive Bayes, J48 and SVM classifiers

Works	Dataset	No. examples	Naive Bayes (%)	J48 (%)	SVM (degree=1) (%)	SVM (degree=2) (%)	SVM (degree=3) (%)
Barbosa and Freire (2005)	TEL-8	216 WQIs 259 not WQIs	76	90.95	85.1	83.8	85.1
Corpus 1	TEL-8	216 WQIs 259 not WQIs	88.84	97.26	89.96	90.57	87.84
Corpus 2	Manually collected	125 WQIs 130 not WQIs	98.82	98.5	98.82	95.30	99

Table 3 shows the results of a test to proof the statistical significance in our experiment comparing two WQI identification approaches, one using a training dataset obtained by our heuristic strategy and another using the approach in Barbosa and Freire (2005). In this table, M, Dev, V and Avg represent the mean, standard deviation, variance and average respectively. A *P-value* less than 0.05 determines a statistical significance (SS) that is denoted by a "+" symbol in the last column.

We have reproduced the experiments reported in Barbosa and Freire (2005) using the same dataset TEL-8 from UIUC repository with the same conditions, considering the same number of instances and attributes. In this test, a total of 50 means were obtained by 100 independent executions of the *k-cross-validation* method. The *D'Agostino-Pearson's omnibus* K^2 test (D'Agostino et al. 1990) was used to determine if data exhibit a normal distribution. We found that the values obtained were not normally distributed, so we adopted the non-parametric Kruskal–Wallis test considering a significance level of $\alpha = 0.05$.

The null hypothesis states that there is not any improvement in the WQI identification task between our approach and the work presented in Barbosa and Freire (2005). This hypothesis has been rejected because the $P - values$ are less than α and hence it is concluded that our results are said to be statistically significant, particularly using the J48 classifier.

7.2 Experiment 2

The aim of this experiment was to demonstrate that the accuracy rate in the WQIs identification process is not substantially reduced when the Prototype Selection by Clustering (PSC) is used. It means that in cases when the number of instances is very large, it is possible to improve the runtime execution of the evaluation process by eliminating irrelevant or redundant instances without having an important variation in the accuracy. In this experiment we used the same reference domains and the same number of instances example reported in Wang et al. (2011), so a comparison against that work was possible. For classification, we used the J48 algorithm.

This second experiment is intended to compute the accuracy of automatic identification of WQIs for four reference domains: jobs, hotels, movies and airfares using the decision tree J48 algorithm and a prototypes selection algorithm. We used the decision tree J48 algorithm with two training sets. The first one includes the complete training set and the second one is a reduced version of the training set that was obtained by the PSC algorithm (explained in Section 5.1).

Figure 7 shows the results obtained for different domains with the decision tree J48 algorithm after using both training sets, the original and the reduced training set.

Table 3 Statistical test between the training set used in the proposed strategy against the training set proposed by Barbosa and Freire (2005), using the TEL-8 dataset and the classifiers NaiveBayes, J48 and SVM

	Proposed strategy			Barbosa and Freire (2005)				
Algorithm	*M*	*Dev.*	*V*	*M*	*Dev.*	*V*	*P-value*	*SS*
Naive Bayes	88.589	0.751	0.564	77.417	1.599	2.557	6.84E-16	+
J48	96.906	0.261	0.068	92.8	0.875	0.766	8.24E-17	+
SVM	89.768	1.493	2.28	85.885	0.477	0.228	3.62E-16	+
Avg.	90.168	1.245	2.684	85.367	0.984	1.184		

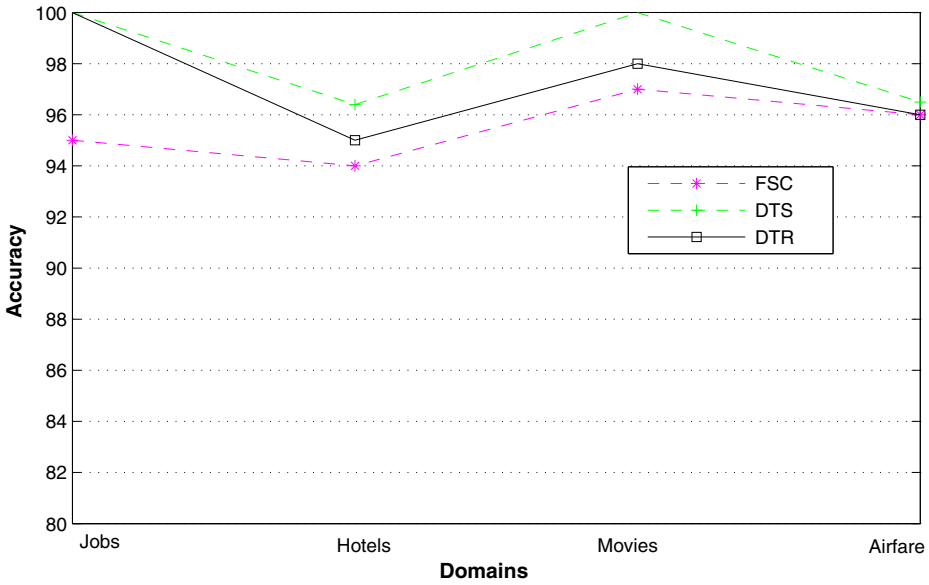

Fig. 7 Accuracy obtained by three strategies for the automatic identification of WQIs using the J48 algorithm

In this figure, *FSC* (Form Structure Classifier) represents the accuracy reported in the original work developed by Wang et al. (2011), *DTS* (Decision Tree Source) represents the accuracy of the decision tree algorithm using a complete training set whose features are those selected by our strategy, and finally, *DTR* (Decision Tree Reduced) represents the accuracy of the decision tree algorithm using a reduced version of this training set after applying the PSC algorithm. The evaluation metric for these three classifiers is the accuracy, that is, the percentage of searchable forms correctly identified over all the identified forms. *FSC*, *DTS* and *DTR* are based on a domain-independent decision tree.

The best results were obtained with the complete training set, which contains all feature vectors built with the features extracted from positive and negatives examples of HTML forms using our heuristic strategy. However, we can see that the accuracy is almost the same in some domains (Airfare) using a reduced training set that in some cases decreases to a quarter of its original size. In this sense, when the number of instances is too large it would be feasible to apply the prototype selection algorithm and to obtain encouraging results.

The results in Table 4 show the number of instances used by *FSC*, *DTS* and *DTR* in this experiment.

Table 4 Number of instances used as examples in the execution of the strategies FSC, DTS and DTR

Works	Domains			
	Jobs	Hotels	Movies	Airfares
FSC (Wang et al. 2011)	55	42	162	72
DTS	44	46	103	65
DTR	12	10	22	12

Table 5 Comparison of accuracy obtained by the Naive Bayes, J48 and SVM classifiers using recent datasets

Domain	No. instances	No. positives instances	No. negatives instances	Naive Bayes (%)	J48 (%)	SVM (%)
Books	90	43	47	98.88	99.3	99
CarsRental	19	10	9	57.89	99	99.1
Hotels	53	31	22	84.9	99.3	98.11
Movies	87	52	35	95.4	99.1	99
Jobs	49	19	30	91.83	99	100
Advanced Query Interfaces	247	126	121	93.52	98.78	98.38
Simple Query Interfaces	97	40	56	94.84	99	100

7.3 Experiment 3

In this last experiment, we evaluated the accuracy of our proposed strategy using a new dataset with new technology of designs of WQIs (XHTM, HTML5). This was done because the UIUC repository was created in 2003 and the WQIs included in the TEL-8 dataset may not correspond to the current designs of HTML forms. With this experiment, we determine how the accuracy rate varies in the identification process of WQIs using the updated dataset. In this experiment, we use a repository of WQIs that was built manually from a set of Web pages retrieved by a generic crawler. The identification of the positives and negatives examples included in the new dataset was carried out by human experts.

In this experiment, we used again the Naive Bayes, J48 and SVM classifiers. Table 5 shows the results obtained for the three classification algorithms. We can see that the use of recent HTML forms are not producing important changes in the accuracy obtained for these algorithms. This situation demonstrates that our feature selection and WQIs identification strategy is still valid for current HTML forms.

8 Conclusions

Web Query Interfaces (WQIs) allow to access Hidden-Web databases and retrieve useful information that is not reachable by traditional search engines. The automatic identification of WQIs is a complex task given the heterogeneity in their design, style, semantic content among others. This work presented a new strategy for automatic identification and classification of WQIs, its performance was evaluated by carrying out three experiments using different corpora of HTML forms. The proposed strategy makes an adequate selection of HTML elements and extracts them from HTML forms that were filtered by the application of eight rules. These control elements are used to characterize the HTML forms and create characteristic vectors that are used by supervised classification algorithms that will help to determine if the web form is or not a WQI. We used a prototypes selection algorithm by clustering (PSC) for removing irrelevant or redundant data in the training set. The supervised

classifier uses characteristic vectors that contain information about control elements of HTML forms contained in Web pages such as the number of *textb oxes*, command *b uttons*, *lab els*, the presence of keywords as *"submit", "search","get"*, or "*post*", etc. We obtained better level of accuracy compared with previously reported works. This improvement was mainly due to an adequate selection of contro elements together with the eight rules defined to filter non-WQIs. The selection of elements allowed a high level of characterization of WQIs providing only those HTML elements that offer relevant information for their identification. The rules allowed discarding HTML forms that do not emit queries to Hidden-Web databases.

The results reported in this work give evidence of the effectiveness and usefulness of the proposed strategy. Future work will involve the classification of WQIs for specific domains.

References

Barbosa, L., & Freire, J. (2005). Searching for hidden-web databases. In *Proceedings of the 8th ACM SIGMOD international workshop on web and databases* (pp. 1–6). Baltimore, Maryland.

Barbosa, L., & Freire, J. (2007a). Combining classifiers to identify online databases. In *Proceedings of the 16th international conference on World Wide Web, WWW '07* (pp. 431–440). New York: ACM. ISBN 978-1-59593-654-7. doi:10.1145/1242572.1242631.

Barbosa, L., & Freire, J. (2007b). An adaptive crawler for locating hidden-web entry points. In *Proceedings of the 16th international conference on World Wide Web, WWW '07* (pp. 441–450). New York: ACM. ISBN 978-1-59593-654-7. doi:10.1145/1242572.1242632.

Barbosa, L., Nguyen, H., Nguyen, T., Pinnamaneni, R., Freire, J. (2010). Creating and exploring web form repositories. In *Proceedings of the 2010 international conference on management of data, SIGMOD '10* (pp. 1175–1178). New York: ACM. ISBN 978-1-4503-0032-2. doi:10.1145/1807167.1807311.

Bergman, M.K. (2001). The deep web: surfacing hidden value (white paper). *Journal of Electronic Publishing, 7*(1), 4.

Cope, J., Craswell, N., Hawking, D. (2003). Automated discovery of search interfaces on the web. In *Proceedings of the 14th Australasian database conference, ADC '03* (Vol. 17, pp. 181–189). Darlinghurst: Australian Computer Society. Inc. ISBN 0-909-92595-X. URL: http://portal.acm.org/citation.cfm?id=820085.820120.

D'Agostino, R.B., Belanger, A., D'Agostino, R.B. Jr. (1990). A suggestion for using powerful and informative tests of normality. *The American Statistician, 44*(4), 316–321. ISSN 00031305. URL http://www.jstor.org/stable/2684359.

García-Serrano, J.R., & Martinez-Trinidad, J.F. (1999). Extension to c-means algorithm for the use of similarity functions. In *Proceedings of the 3rd European conference on principles of data mining and knowledge discovery, PKDD '99* (pp. 354–359). London: Springer-Verlag. ISBN 3-540-66490-4. URL: http://dl.acm.org/citation.cfm?id=645803.669654.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H. (2009). *The WEKA data mining software: an update* (Vol. 11, Issue 1). SIGKDD Explorations.

Jericho HTML Parser (2010). A Java Library for parsing HTML documents. Sourceforge Project, 2010. http://jericho.htmlparser.net/docs/index.html. Accessed 12 Dec 2011.

Jiang, L., Wu, Z., Zheng, Q., Liu, J. (2009). Learning deep web crawling with diverse features. In *Web intelligence* (pp. 572–575).

Jiang, L., Wu, Z., Feng, Q., Liu, J., Zheng, Q. (2010). Efficient deep web crawling using reinforcement learning. In *PAKDD (1)* (pp 428–439).

Kabisch, T., Dragut, E.C., Yu, C.T., Leser, U. (2009). A hierarchical approach to model web query interfaces for web source integration. *Proceedings, Very Large Data Bases, 2*(1), 325–336.

Li, Y., Nie, T., Shen, D., Yu, G. (2010). Domain-oriented deep web data sources' discovery and identification. In *Proceedings of the 2010 12th International Asia-Pacific Web Conference*, APWEB '10, pages 464–467, Washington, DC, USA. IEEE Computer Society. ISBN 978-0-7695-4012-2. doi:10.1109/APWeb.2010.54.

Lin, L., & Zhou, L. (2009). Web database schema identification through simple query interface. In *RED* (pp. 18–34).

Liu, V.Z., Luo, R.C., Cho, J., Chu, W.W. (2004). D-pro: A probabilistic approach for hidden web database selection using dynamic probing. In *Proceedings of the ICDE*. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.2525.

Lu, J., & Li, D. (2010). Estimating deep web data source size by capture-recapture method. *Information Retrieval, 13*(1), 70–95.

Madhavan, J., Ko, D., Kot, L., Ganapathy, V., Rasmussen, A., Halevy, A. (2008). Google's deep web crawl. *Very Large Data Bases, 1*, 1241–1252. ISSN 2150-8097. doi:10.1145/1454159.1454163.

Mitchell, T.M. (1997). *Machine learning*. New York: McGraw-Hill.

Olvera-Lopez, J.A., Martinez-Trinidad, J.F., Carrasco-Ochoa, J.A. (2007). Mixed data object selection based on clustering and border objects. In *CIARP* (pp. 674–683).

Platt, J.C. (1998). Sequential minimal optimization: a fast algorithm for training support vector machines. *Advances in Kernel MethodsSupport Vector Learning, 208*(MSR-TR-98-14), 1–21. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.560&rep=rep1&type=pdf.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106. doi:10.1007/BF00116251.

Ru, Y., & Horowitz, E. (2005). Indexing the invisible web: a survey. *Online Information Review, 29*(3), 249–265.

Shestakov, D. (2008). *Search interfaces on the web: Querying and characterizing*. PhD thesis, University of Turku Department of Information Technology.

The UIUC web integration repository (2003). Computer Science Department, University of Illinois at Urbana-Champaign. http://metaquerier.cs.uiuc.edu/repository.

Wang, H., Liu, Y.W., Zuo, W.L. (2008). Using classifiers to find domain-specific online databases automatically. *Journal of Software, 19*(2), 246–256. URL: http://www.jos.org.cn/1000-9825/19/246.htm.

Wang, Y., Li, H., Zuo, W., He, F., Wang, X., Chen, K. (2011). Research on discovering deep web entries. *Computer Science and Information Systems, 8*(3), 779–799.

Witten, I.H., Frank, E., Hall, M.A. (2000). *Data mining: Practical machine learning tools and techniques with java implementations*. USA: Academic Press. ISBN 1558605525.

Wu, W., Yu, C., Doan, A., Meng, W. (2004). An interactive clustering-based approach to integrating source query interfaces on the deep Web. In *Proceedings of the 2004 ACM SIGMOD international conference on management of data, SIGMOD '04* (pp. 95–106). New York: ACM. ISBN 1-58113-859-8. doi:10.1145/1007568.1007582.

Zhang, P., Qu, Y., Huang, C., Jaeger, P.T., Wells, J., Hayes, W.S., Hayes, J.E., Jin, X. (2010) Collaborative identification and annotation of government deep web resources: A hybrid approach. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia, HT '10* (pp. 285–286). New York: ACM. ISBN 978-1-4503-0041-4. doi:10.1145/1810617.1810677.

Zhang, Z., He, B., Chang, K.C.-C. (2004). Understanding Web query interfaces: Best-effort parsing with hidden syntax. In *Proceedings of the 2004 ACM SIGMOD international conference on management of data*, SIGMOD '04, pages 107–118, New York: ACM. ISBN 1-58113-859-8. doi:10.1145/1007568.1007583.