International Journal of Information Retrieval Research
Volume 13 • Issue 1

Automated Selection of Web Form Text Field Values Based on Bayesian Inferences

Diksha Malhotra, Punjab Engineering College (Deemed), Chandigarh, India*
Rajesh Bhatia, Punjab Engineering College (Deemed), Chandigarh, India
Manish Kumar, Punjab Engineering College (Deemed), Chandigarh, India

ABSTRACT

The deep web is comprised of a large corpus of information hidden behind the searchable web interfaces. Accessing content through searchable interfaces is somehow a challenging task. One of the challenges in accessing the deep web is automatically filling the searchable web forms for retrieving the maximum number of records by a minimum number of submissions. The paper proposes a methodology to improve the existing method of getting informative data behind searchable forms by automatically submitting web forms. The form text field values are obtained through Bayesian inferences. Using Bayesian networks, the authors aim to infer the values of text fields using the existing values in the label value set (LVS) table. Various experiments have been conducted to measure the accuracy and computation time taken by the proposed value selection method. It proves to be highly accurate and takes less computation time than the existing term frequency-inverse document frequency (TF-IDF) method, hence increasing the performance of the crawler.

KEYWORDS

Automatic Form Filling, Bayesian Inference, Deep Web, Hidden Web, Information Retrieval, Instance Templates

1. INTRODUCTION

The content on Internet is growing at a breakneck pace, as more and more people are connecting to it. Publically Indexable Web (PIW) or surface web consists of a very small part of the Internet, which can be accessed by traversing through hyperlinks. Traditional web crawlers use different approaches to access only PIW. Whereas, its counterpart, hidden web, consists of information generated dynamically where the user needs to fill a searchable form to access data. However, a number of recent studies have shown that a significant amount of data lies outside the PIW. A commercial vendor, BrightPlanet.com, claims that the size of the deep web is 500 times greater than the publically indexable web (Bergman 2001). The hidden web data is very important for various stakeholders. Hence, deep web crawlers use numerous approaches to access hidden web data. However, the deep web can be entered only after filling the search forms and hereby, accessing databases. Whenever a user fills up a search form in

DOI: 10.4018/IJIRR.318399

*Corresponding Author

order to access hidden data, a dynamic webpage is generated. A query is shot to the database and the required results from the database are shown. The results from the database may contain diverse content types such as *Dynamic Data, Unlinked Content, and Non-Text Content.* Dynamic data can only be accessed through the supported query interfaces. These interfaces consist of input elements, and a user query includes providing values for these elements. However, unlinked content cannot be accessed by going through links, and non-Text content consists of various PDF, multimedia files, and non-HTML documents.Following are the main four key phases of the working of deep web crawler:

- Discovery of the entry points to the hidden web *i.e.,* searchable interfaces as these allow searching online databases (Lage et al. 2004; Onihunwa et al. 2017).
- Label extraction (Wang and Lochovsky 2003; Nguyen et al. 2008).
- Updating the LVS table and automatically filling the hidden web forms.
- Response analysis *i.e.* classification into valid and invalid responses.

Figure 1 shows the above-explained key phases of the working of a deep web crawler. Each phase has its approaches and challenges associated with it. While designing a deep web crawler, a designer can face the following challenges:

- *Determining the searchable interface of hidden web* (Wu et al. 2006; Moraes et al. 2013; Liu and Li 2016): As the hidden web crawler needs a searchable query form in order to access a hidden web page, hence, a hidden web crawler must be able to identify the query forms as an entry point to the hidden database.
- Extracting form labels(An et al. 2007; Nguyen et al. 2008): As the labels of a form are not at a specified position in web forms; hence, it is a challenging task to extract form labels. The form labels help to fill form fields automatically.
- *Automatically filling forms*: It requires filling form fields with efficient and most suitable words for the field.

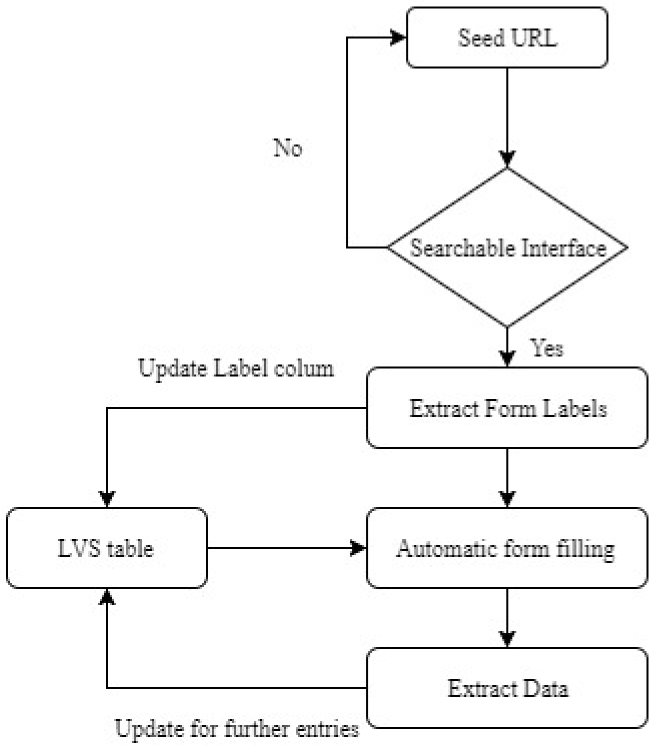
Considering the above challenges, the paper focuses on the challenge associated with process of automatic form filling (Álvarez et al. 2007). It requires the selection of appropriate values for the form fields so that with a minimum number of submissions, maximum records of data can be extracted. In order to assist in values selection, the paper focuses on the automatic filling of searchable web forms (excluding login forms) by generating informative instance templates (explained in the following sections) using fields of the form and selecting values for the fields using Bayesian inferences. The Bayesian inferences provide an automatic and effective way to help filling the searchable forms by creating a network structure, and calculating the joint probability.

The remainder of this paper is organized as follows. In Section 2, we present a brief literature review of the techniques in the existing literature for automatic filling of web form. Section 3 presents the terminologies related to the proposed approach of this paper. Section 4 describes the proposed approach for automatic form filling using Bayesian inferences in detail. Section 5 analyses the mathematical concepts related to the proposed approach, discusses the experimental setup and explains the result followed by conclusion.

2. RELATED WORK

In this section, we present a brief literature review of the existing techniques and solutions for Automatic filling of searchable interfaces. The automatic filling of web forms includes the automatic assignment of corresponding values to the fields and submission of web form to extract records from the hidden web.

Figure 1. Basic working of hidden web crawler



H. Belgacem et al. (Belgacem et al. 2022) have proposed a learning-based automated approach to fill few categorical fields. The authors build a Bayesian based model by learning the field dependencies from a set of input values. The technique then employs local modeling to efficiently extract the local dependencies of fields in the input instances cluster. Authors ***Troiano et al.***(Troiano et al. 2017) describe a prediction model which using Bayesian inferences which predicts values for the next fields. It predicts on the values entered so far in a single text field or values entered in the above fields in a multi-attribute form. Hence, it exploits the relation between different fields in a multi-attribute form. The network structure in the Bayesian network and joint probability distributions are calculated using Necessary Path Condition (NPC) and Expectation Maximization (EM). In the prediction model, first the entries entered by the user are used to develop a prediction schema, then using values entered in the above fields, the value for the current field is predicted. For prediction, a threshold value is kept, and if the value is less than the threshold value, then it can be used as suggestions for auto-completion. The corpus of evidence has been divided into two parts: (1) CWA (closed world assumption), and OWA (open world assumption). The CWA knows how the possible values can be combined, whereas, the OWA does not have the past knowledge about values. The main challenging part is the OWA.

For the experimentation purpose, Poste Italiane's services for desktop and mobile applications were considered. Many volunteers between ages 24-45 were enrolled and experiments were conducted at the Research and Development Centre of Poste Italiane. The system, however, proved to be 100% Accurate. The proposed model does not allow re-using the acquired data from distinct forms.

Kantorski et al.(Kantorski et al. 2013) presents the concept of Instance templates based on the concept of templates provided by Madhavan et al.(Madhavan et al. 2008). It describes Instance template as the values assigned with the fields of the form. If there are n fields in the form, then it will form 2n-1 templates. The instance template is denoted by an attribute-value pair. As there are many values corresponding to a field (infinite in case of text fields and finite in case of select fields), hence it uses the concept of informativeness introduced by Madhavan et al. Initially, templates with single dimension are considered i.e. templates of single field each. If the template proves to be non-informative, then further the templates of the higher dimension consisting of same field are not considered. The proposed framework uses the method of Instance Template Pruning (ITP) for generating and submitting instance templates and uses the method of Filling Text Fields (FTF) for selecting values to fill in the fields of the forms. The ITP method adds the non-informative templates into the pruning list and the FTF method uses the TF-IDF method to select values. Domains of Jobs, Movies, Books, and Food were used for experimentation purposes. It has maximum Coverage (>90%), with higher execution efficiency (>90%). However, the FTF method can be improved and can use prediction for the same.

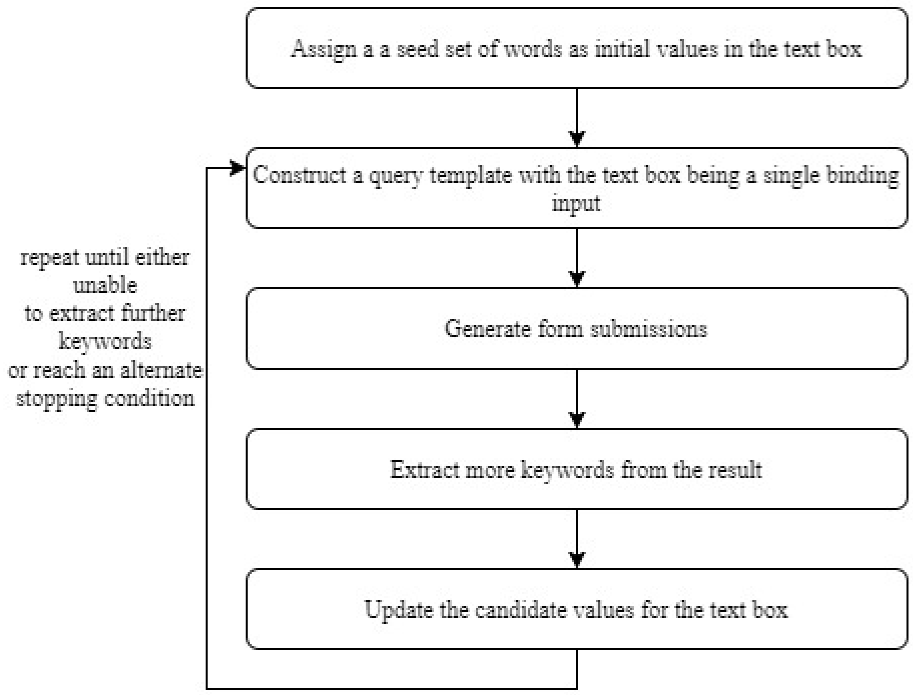
Toda et al.(Toda et al. 2010) presents an approach, iForm, based on Bayesian inferences in which the system extracts values from the input provided by the user and automatically fills the form using the extracted values. The iForm approach considers various machine learning algorithms such as SVM, Genetic Programming, linear combination of values and Bayesian learning as an alternative to find the probability of a field. However, Bayesian learning proves to have advantages over the rest of the approaches(Toda et al. 2009). The form filling is done by soft TF-IDF technique which finds the value which is very much related to the extracted value. The proposed approach has been implemented on various single and multi-attribute web forms such as websites of Short Movie Reviews, Car offers, Cell phone offers, and Book offers. It proves to have f-measure (>0.62) with higher Precision (>0.70), recall (>0.55). The approach requires human intervention and it does not properly detail how to deal with negative indications.

Madhavan et al.(Madhavan et al. 2008) introduces the surfacing problem i.e. finding a suitable set of query templates and suitable the binding input values. Selecting a suitable set of form submissions is basically the problem of surfacing. It defines the query template as one which tags some fields of the form as the “Binding inputs” while others as the “Free inputs”. The dimension of the query template is the number of binding inputs. Finding suitable values of the binding inputs is the main task of the proposed approach as the free inputs are filled with default values. It explains the concept of informative query templates i.e. based on the content similarity; the templates are distinguished to be informative or non-informative. First, the query templates with dimension one are considered, and if they prove to be informative, then only they are upgraded to the next dimension. In figure 2 shows the iterative probing approach for the free text boxes explained in the paper. It uses TF-IDF method for candidate selection. It has been experimented on 700 domains and proves out to have high precision (>78%). It may be noted that the number of templates generated based on their Cartesian products are huge when the number of fields is increased and there are no specific details on implementation on typed text boxes.

Ntoulas et al.(Ntoulas et al. 2005) cites two main challenges for a deep web crawler, given a searchable interface: (1) modeling and understanding the interface i.e. extracting form labels, and (2) issuing related queries after understanding the form interface. It formalizes the query selection problem as a set covering problem. As set covering problem is a NP Hard problem, hence they devise an approximation algorithm which is able to find the solution that is near to an optimal solution. The algorithm also facilitates to predict the number of pages returned in the result page. For selecting keywords for the query, they have suggested the following operations:

- *Random*: random keywords from the English dictionary.
- *Generic-frequency*: based on the frequency of keywords from the general documents available on the web.
- *Adaptive*: Keyword having maximum frequency from the previous submissions.

Figure 2. Iterative probing approach(Madhavan et al. 2008)


Assign a a seed set of words as initial values in the text box
Construct a query template with the text box being a single binding input
Generate form submissions
Extract more keywords from the result
Update the candidate values for the text box
repeat until either unable to extract further keywords or reach an alternate stopping condition

The main goal is to download maximum number of documents by issuing lesser queries. The Open Directory Project, Amazon, and the PubMed Medical Library websites were considered for the experimentation and evaluation purposes. It can be noted that the Adaptive method provides maximum coverage (>99%), and the random method provides the least coverage (~36.7%).

Liddle et al.(Liddle et al. 2002) terms the data extraction from the underlying database as the Common Gateway Interface (CGI) request to the server which is then converted to database queries. It addresses some of the difficulties encountered by the crawler which automatic processing of forms:

- Unrecognizable error messages which make the system difficult to automate.
- If the data is extracted piece by piece, then it may contain duplicate data as well.
- Some searchable interfaces can result in extracting another searchable interface which needs to be filled.
- Crawler may be unable to analyze and understand random scripts automatically.

The proposed tool deals only with the Hypertext Markup Language (HTML) forms (not even with Javascript embedded in it) and it considers only GET request. It converts the HTML forms into a parse tree and extracts useful information from the same. It initially fills each field with default values and automatically extracts values. In order to recognize duplicate values, it separates the values into sentences and finds their hash values using the copy detection system (Campbell et al. 2000). Table 1 summarizes the existing literature for automatic filling of web forms.

3. TERMINOLOGIES RELATED TO PROPOSED METHODOLOGY

The section describes some terms related to the proposed methodology for extracting data from deep web.

3.1 Query Templates

In this work, we employed the concept of Query templates as defined by ***Madhavan et al.*** (Madhavan et al. 2008) where a query template (Barua et al. 2015) is explained as a subset of the inputs of a form, after considering each input as a Binding input in a searchable form (f_e). By associating different values with Binding inputs, multiple form submissions can be generated. The dimension of a query template can be denoted by the number of binding inputs. If a form has *'k'* fields, then the number of query templates is2^k-1. For instance, if there are 5 fields in a web form namely,*V, W, X, Y, Z*, then there are 31 possible templates.

3.2 Instance Query Templates

The concept of query templates was enhanced by ***Kantorski et al.***(Kantorski et al. 2013) which introduced instance query template as an attribute-value pair <*t, v*> where *t* be a template in set of templates and *v* is an element fromthe set of values for fields in the form, V $\{v_1,v_2,v_3,\ldots,v_n\}$. The dimension of an instance query template is defined as the number of fields in the template. For instance, an instance template for the template *X&Y&Z* can be represented as *X=x1&Y=y1&Z=z1* and it can be represented as the pair (X&Y&Z, x1&y1&z1). The dimension of the stated template is, *therefore*, three. The basis for evaluation of a query template is the similarity/distinctness of the web page content with other query templates submissions. Based on the similarity, the templates are clustered and informativeness is checked. A template is defined as informative if the generated pages are acceptably distinct, and is *otherwise*, uninformative.

3.3 Value Selection using Bayesian Inferences

In order to select the values of the text fields, we have employed the method of using Bayesian inferences/ predictions in the forms by ***Troiano et al.*** (Troiano et al. 2017). The finite domain fields are filled by their default values. The main challenge of automatic form filling is to select values for infinite domain fields. Here, each form field is mapped to a node in the network after which the model is structured from the training data using structure learning. Further,predictions are made using parameter learning.

4. PROPOSED METHODOLOGY

In this section, we present the two main sections of our proposed methodology *i.e.* selecting values using Bayesian inferences and generating Instance templates for extracting data from Searchable web forms so that maximum data can be extracted with minimum number of submissions. Figure 3 depicts the proposed methodology explained in this section. The searchable web forms consist of both *finite* domain elements (select, radio buttons, etc.) as well as *infinite* domain elements (text boxes). Considering these fields, candidate templates are generated. After the generation of candidate templates, values for each field of the form are selected using Bayesian inferences and Instance templates are generated and checked for informativeness until all the instance templates are evaluated.

4.1 Selecting Values using Bayesian Inferences

For selecting values for fields of the web form, the proposed approach uses *Bayesian-based inferences* as the Bayesian network aids in efficient predictions. Using Bayesian networks, it is easy to create networks that are readable in nature. Moreover, it aids in predicting the missing values in the training data. We propose to employ Bayesian inferences as the performance is better than the existing TF-IDF method for value selection. As the data size increases, the existing TF-IDF method's (Kantorski et al. 2013) performance starts decreasing as it begins taking more computational time. However, Bayesian

Table 1. Review of Literature for automatic filling of web forms

S.no.	Title	Authors	Findings	Shortcomings	Metrics	Tool/Algo. proposed	Language	Subject System	Need of user support
1.	A Machine Learning Approach for Automated Filling of Categorical Fields in Data Entry Forms	Belgacem, Hichem and Li, Xiaochen and Bianculli, Domenico and Briand, Lionel	A learning-based approach to fill categorical fields in forms.	The proposed approach may provide some incorrect suggestions	Accuracy	Bayesian Inference	-	-	Yes
2.	Modeling and predicting the user next input by Bayesian reasoning	Troiano, Luigi Birtolo, Cosimo Armenise, Roberto	Provides a method for value selection by prediction using Bayesian inference.	The proposed model does not allow to reuse the acquired data from distinct forms.	Accuracy	Model to predict values using Bayesian Reasoning.	-	Poste Italiane	Yes
3.	Choosing Values for Text Fields in Web Forms	Kantorski, Gustavo Zanini Moraes, Tiago Guimaraes Moreira, Viviane Pereira Heuser, Carlos Alberto	Provides a method for value selection, Instance template generation and checking for informativeness of templates.	The TF-IDF method used in the paper for value selection is less efficient and does not provide any technique of machine learning to find field values in templates with greater than one order.	Coverage (>90%), execution efficiency (>90%)	ITP and FTF model.	-	Domains of Jobs, Movies,Books, and Food	No
4.	Selecting queries from sample to crawl deep web data sources	Wang, Yan Lu, Jianguo Liang, Jie Chen, Jessica Liu, Jiming	Provides a method to select a set of queries at minimal cost using a very small sample of documents.	Unable to fill forms with complex and multi-attributes.	Coverage (>90%)	Method based on sampling	-	Reuters, Gov2 (TREC test data Sets)	No
5.	Crawling the Hidden Web: An Approach to Dynamic Web Indexing	Soulemane, Moumie Rafiuzzaman, Mohammad Mahmud, Hasan	Provides a method to fill form by initially finding values from the form page and then values for each field (dom f1) are extracted from the retrieved result pages.	Deals only with defined format forms and single-attribute forms.	-	New framework to index dynamic data.	Java	Datasets from www.dmoz.org	No
6.	Carbon: Domain-independent automatic web form filling	Araujo, Samur Gao, Qi Leonardi, Erwin Houben, Geert Jan	Proposes a naïve framework to autofill forms independent of their domains while determining connection with their previous entries of forms.	There is no information on which classifier would be efficient for integration with the Carbon architecture.	Reduction rate (>42%), Completeness (~74%), Precision (~0.73), Recall (~0.53)	Carbon	Javascript (Greasemonkey)	TEL-8 dataset	Yes
7.	A Probabilistic Approach for Automatically Filling Form-Based Web Interfaces	Toda, Guilherme A. Cortez, Eli Silva, Altigran S. Da Moura, Edleno De da Silva, Altigran S. de Moura, Edleno	Provides a Bayesian based framework to fill remaining fields a form using previous submissions of the user by asking the user to input a field value.	Does not properly detail on how to deal with negative indications.	Precision (>0.70), recall (>0.55), f-measure (>0.62)	iForm	-	Short Movie Reviews, Car offers, Cellphone offers and Book offers.	Yes

continued on following page

Table 1. Continued

S.no.	Title	Authors	Findings	Shortcomings	Metrics	Tool/Algo. proposed	Language	Subject System	Need of user support
8.	Efficient deep web crawling using reinforcement learning	Jiang, Lu Wu, Zhaohui Feng, Qian Liu, Jun Zheng, Qinghua	Proposes a framework to deal with the surfacing problem using reinforcement learning and develops a Q-value approximation algo. to deal with myopia problem.	No specific details on implementation on typed text boxes.	Coverage (>80%),	Adaptive RL surfacing algorithm (A Q-value approximation algo.)	-	AbeBooks, Wikicfp, Yahoo movie, Google music, Baidu Baike	No
9.	Automatically filling form-based web interfaces with free text inputs	Toda, Guilherme a. Cortez, Eli Mesquita, Filipe Silva, Altigran S. Da Moura, Edleno Neubert, Marden	Proposes an approach to fill fields by taking a free input from the user and then using previous submissions automatically fill the remaining fields.	Lacks to provide information on how to deal with complex fields.	F-measure (>0.81), Precision (>77%), Recall (>77%)	iForm	-	Jobs postings (RISE Jobs collection), submissions of Car ads/offers and Mobile Phones offers.	Yes
10.	A framework of deep Web crawler	Peisu, Xiang Ke, Tian Qinzhen, Huang	Introduces the concept of LVS table, describes the mechanism to fill the LVS table.	Is not automatic technique as the user needs to enter some keywords to extract information from a website.	Ranking function (>60%),	-	-	news.cn.yahoo.com, news.sina.com.cn, news.163.com, news.sohu.com	Yes
11.	Google's Deep Web crawl	Madhavan, Jayant Ko, David Kot, Lucja Ganapathy, Vignesh Rasmussen, Alex Halevy, Alon	Introduces the concept of Surfacing, Informativeness of templates based on distinction showed by the content, Iterative probing technique for finding suitable values for text boxes.	The number of templates generated based on their Cartesian products are huge when the number of fields are increased, No specific details on implementation on typed text boxes.	High precision (>78%)	ISIT algorithm	-	700 domains	Yes
12.	Crawling the content hidden behind web forms	Álvarez, Manuel Raposo, Juan Pan, Alberto Cacheda, Fidel Bellas, Fernando Carneiro, Víctor	The proposed system DeepBot, is domain specific and it automatically finds the domain relevant forms on which it executes queries to extract data.	It fails for the fields with finite domain which do not have any default value in the form.	Precision (>0.94), Recall (> 0.79) and F-measure (>0.76)	DeepBot	-	Books Shopping, Music Shopping and Movies Shopping	Yes
13.	Downloading textual hidden web content through keyword queries	Ntoulas, a. Pzerfos, P. Cho, Junghoo Cho Junghoo	Formalises the query selection problem as a set covering problem. Approximation algorithm is able to find the solution that is near to optimal solution and facilitates to predict the number of pages returned in the result page.	Multi-attribute forms have not been taken into account.	Cost, Coverage (>72%),	approximation algorithm	-	PubMed Medical Library, Amazon, and the Open Directory Project	Yes

continued on following page

Table 1. Continued

S.no.	Title	Authors	Findings	Shortcomings	Metrics	Tool/Algo. proposed	Language	Subject System	Need of user support
14.	Siphoning Hidden-Web Data through Keyword-Based Interfaces	Barbosa, Luciano Freire, Juliana	Focusses on retrieving data from the Keyword based interfaces.	Only deals with keyword-based interfaces.	Coverage (>75%), Effectiveness	Sampling based algorithms	-	nwfusion. com,apsa.org, georgetown. edu	Yes
15.	Automatic generation of agents for collecting hidden Web pages for data extraction	Lage, Juliano Palmieri Da Silva, Altigran S. Golgher, Paulo B. Laender, Alberto H.F	Provides a method to learn to fill forms by finding mapping between labels and attributes of repository.	Navigation patterns based on mechanisms like applets, Javascripts, which are used to extract from hidden web have not been mentioned.	Accuracy (>80%), Precision (100%) and Recall (100%)	Method using agents.	Javascript and Java	30 Web sites from three different domains: books, CDs, and software	No
16.	Extracting Data Behind Web Forms	Liddle, Stephen Embley, David Scott, Del Yau, Sai Ho	Proposes a tool which tries all the possible set of queries after getting the results from the target website.	Does not deal with the form submission using Http POST method, and the number of queries can be very large at time.	Accuracy (>80%)	-	-	13 websites	Yes
17.	Crawling the Hidden Web	Raghavan, Sriram Garcia-molina, Hector	Proposes a method to find form labels using pruning/ prune tree method which sheds the images and other styling details.	Unable to determine relations between form labels.	Accuracy (93%), higher efficiency	LITE	-	IEEE Spectrum, Semiconductor Online, Business News, Yahoo News, Total News, Semiconductor Intl., Solid State Technology Intl. Magazine, CNN Financial News, Technology News.	Yes

inferences include predicting values by training Bayesian network with the training data. The training of the Bayesian network is divided into two phases, (a) *Creating of the network structure*, and (b) *Calculating the joint probability*(Troiano et al. 2017). Bayesian inferences consider each variable in the data as a random variable. Each random variable is considered as a node while creating a network structure. After the fields/Random variables are mapped to the nodes in the network, structure learning is used to create links between the nodes based on their dependencies in the training data. Afterwards, the joint/conditional probability is calculated using parameter learning from the data for each value corresponding to each node. Finally, the values of the fields in the form are predicted.

4.2 Generating Instance Templates for Extracting Data from Searchable Web Forms

On the basis of concepts of Informativeness and Instance templates introduced by Kantorski et al. (Kantorski et al. 2013), we propose our approach to generate Instance templates in order to extract maximum data present behind the searchable web forms with minimum number of submissions. As for infinite domain fields such as text boxes, the number of corresponding values can be numerous, hence, the number of submissions can also be large. Therefore, in order to limit the number of submissions, the concept of informativeness, which groups together the similar instance templates, is used. The process initiates with one-dimensional instance templates. Data is extracted from the hidden web by submitting these instance templates. If the data extracted from the instance template is sufficiently different from the other instance templates' extracted data, then it is considered informative and hence, considered for higher dimensions. Otherwise, it is added to the pruning/discarded list and is not considered for higher dimensions. It has been noted that if an instance template *i* of lower order proves to be uninformative, then the higher dimension instance templates which include *i* also prove to be uninformative and hence, are pruned.

5. EXPERIMENTAL RESULTS AND DISCUSSION

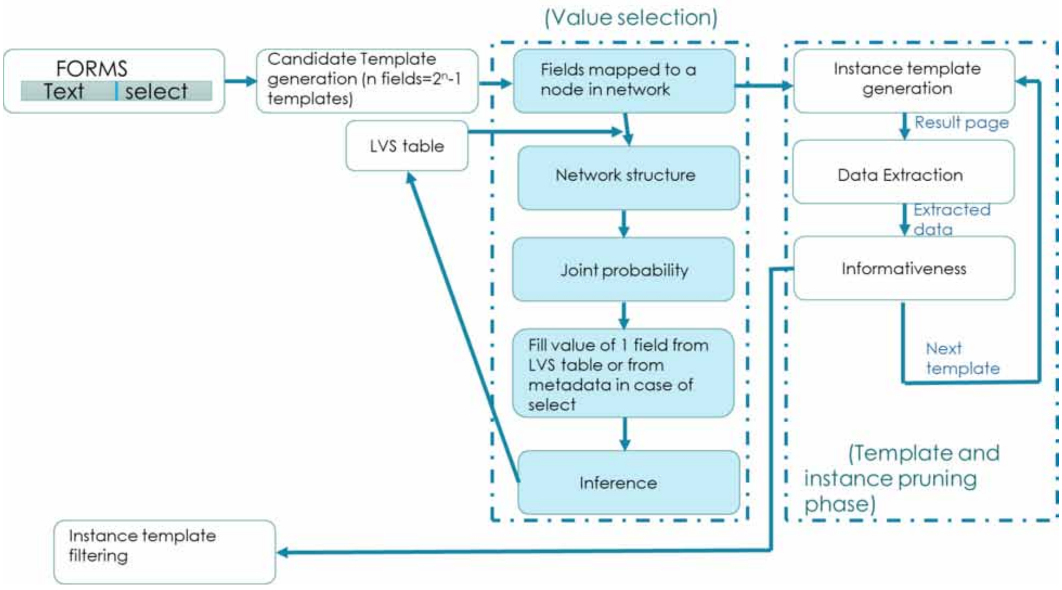
The section discusses the results of the proposed methodology based on experimental setup explained. The results are discussed based on various evaluation parameters such as accuracy, distinction fraction and performance.

5.1 Experimental Setup

The aim of the proposed methodology is to extract data present behind the searchable web forms by automatically filling the web form fields. As the proposed methodology works efficiently for forms having more than one field; hence, it is necessary to locate searchable web forms with more than one field. The *Runeberg.org* website's search interface is used for experimental purposes as it serves the purpose of having more than one field. This interface contains search elements as *First name, Surname, Born, Dead,* and *Profession*. Corresponding to these elements, the crawler automatically fires an appropriate value in the search field and the form is submitted which provides web content related to that element. The implementation of the proposed approach is done in *Python* using libraries such as *BeautifulSoup* and *itertools*.

The predictions for value selection using Bayesian network are done using the *Bayes server* software, which aids in creating Bayesian network using the training data by structure learning and helps to predict the values for text fields by parameter learning. It also aids in calculating the accuracy of the network using testing data. Further, *Spreadsheet Compare* (Excel compare) by Microsoft is used to perform a cell-by-cell comparison of worksheets within the same or different workbooks. As for informativeness of the instance templates, we need to compare the results of the instance

Figure 3. Proposed Methodology


(Value selection)
FORMS
Text
select
Candidate Template generation (n fields=2ⁿ-1 templates)
Fields mapped to a node in network
Instance template generation
LVS table
Result page
Network structure
Data Extraction
Extracted data
Joint probability
Informativeness
Fill value of 1 field from LVS table or from metadata in case of select
Next template
Inference
(Template and instance pruning phase)
Instance template filtering

template with the results of the existing instance templates; hence, the package aids in calculating the distinction function used for informativeness.

5.2 Accuracy

After parameter learning of the Bayesian network for inference, the network is tested using the testing data with some missing values as well. This process is known as *Batch Query*. So,the overall accuracy of the network is calculated after the prediction for one of the random variable in the testing data.

The Overall accuracy is calculated as:

$$\text{Overall Accuracy} = \frac{\#\text{accepted predictions}}{\#\,\text{total predictions}}$$

The prediction of the proposed Bayesian network for the given dataset proves to be accurate. The Bayesian network was experimented for accuracy using training datasets of different sizes. Table 2 shows the comparison of the accuracy of prediction of the Bayesian network for training datasets of different sizes. It depicts that with the increase in the size of the dataset, the predictions made by the Bayesian network for the proposed approach becomes more and more accurate (figure 4).

5.3 Distinction Fraction

After comparing the Comma Separated Values (CSV) files of the instance templates using Spreadsheet compare, and storing the result of the distinction in an excel file, Distinction fraction is calculated to check informativeness.

The Distinction fraction is calculated as:

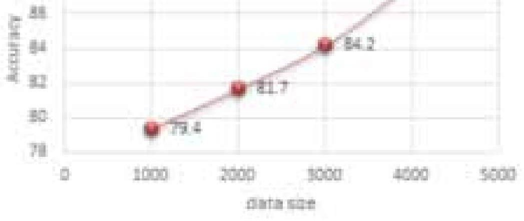
$$\text{Distinction fraction} = \frac{\text{number of distinct rows}}{\text{Total no. of rows}}$$

Note: The instance templates with distinction fraction greater than 0.2 only will be considered as informative.

Table 2. Comparison of accuracy for different sizes of dataset

No. of rows in Dataset	1000	2000	3000	4000
Accuracy	79.4	81.7	84.2	87.8

Figure 4. Effect on Accuracy with increase in data size



Each instance template is compared for informativeness in the dataset used. Out of 31 instance templates, 16 instance templates prove to be informative and the rest of 15 instance templates do not return sufficiently distinct data. Due to the distinction fraction, data is extracted from only 16 instance templates that lead to decrease in computation time taken by the deep web crawler. This aids in achieving our aim of extracting maximum data hidden behind the searchable web forms by minimum number of submissions.

5.4 Performance

Performance refers to the average time taken by the proposed approach. The performance of the value selection module of the proposed approach is calculated and compared with the existing method for value selection in (Kantorski et al. 2013). In terms of the performance of the value selection module, the proposed approach proves to be less time consuming as compared to the existing method *i.e.* TF-IDF for value selection by 35-40%. The Bayesian prediction method proves to be efficient in terms of performance.

6. CONCLUSION

As the hidden data can only be accessed through searchable web forms, therefore, it is essential to fill the web forms and submit them to extract relevant data. The main challenge in designing a hidden web crawler is automatic filling of web forms. In this paper, we have proposed an approach for automatic form filling using Bayesian predictions. The existing approach for value selection, which uses TF-IDF approach, takes more time for value selection and increases the time taken by the crawler to extract data. As the proposed approach incorporates the Bayesian inference method for value selection; hence, the crawler takes less time to extract the data. It may be noted that with the increase in data size of the training dataset, the accuracy of prediction of Bayesian network increases. In case of complex web forms with multiple text fields, a large number of candidate templates are generated. Hence, the informativeness of the instance templates aids in reducing the number of submissions. In the proposed approach, the computation time decreases by 35-40%.

As the Bayesian network exploits the interdependency of random variables/nodes of the network; hence, the proposed approach works well for more than one text fields in the form. Hence, in future, we plan to extend our work for even one text field.

REFERENCES

Álvarez, M., Raposo, J., & Pan, A. (2007). Crawling the content hidden behind web forms. *Springer Int Conf Comput Sci Its Appl*, 322–333. 10.1007/978-3-540-74477-1_31

An, Y., Geller, J., Wu, Y., & Chun, S. (2007) Semantic deep web: automatic attribute extraction from the deep web data sources. *Proc 2007 ACM Symp Appl Comput., 1667*–1672. doi:10.1145/1244002.1244355

Barua, J., Patel, D., & Goyal, V. (2015). TiDE: Template-Independent Discourse Data Extraction. *Big Data Analytics and Knowledge Discovery: 17th International Conference, DaWaK 2015, Valencia, Spain, September 1-4, 2015. Proceedings*, *17*, 149–162.

Belgacem, H., Li, X., Bianculli, D., & Briand, L. (2022). A Machine Learning Approach for Automated Filling of Categorical Fields in Data Entry Forms. *ACM Transactions on Software Engineering and Methodology*, 3533021. doi:10.1145/3533021

Bergman, M. K. (2001). White Paper: The Deep Web: Surfacing Hidden Value. *J Electron Publ, 7*. 10.3998/3336451.0007.104

Campbell, D. M., Chen, W. R., & Smith, R. D. (2000). Copy Detection Systems for Digital Documents. *Adv Digit Libr Conf IEEE*, *0*, 78. 10.1109ADL.2000.848372

Kantorski, G.Z., Moraes, T.G., Moreira, V.P., & Heuser, C.A. (2013). Choosing values for text fields in web forms. *Adv Intell Syst Comput, 186*, 125–136. 10.1007/978-3-642-32741-4_12

Lage, J. P., Da Silva, A. S., Golgher, P. B., & Laender, A. H. F. (2004). Automatic generation of agents for collecting hidden Web pages for data extraction. *Data & Knowledge Engineering*, *49*(2), 177–196. doi:10.1016/j.datak.2003.10.003

Liddle, S., Embley, D., Scott, D., & Yau, S. H. (2002). Extracting Data Behind Web Forms. *Proc 28th VLDB Conf, 402*–413. doi:10.1007/978-3-540-45275-1_35

Liu, B., & Li, Z. (2016). A deep web query interface discovery method. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015*, 1340–1344.

Madhavan, J., Ko, D., & Kot, L. (2008) Google's Deep Web crawl. *Proc VLDB Endow, 1*, 1241–1252. doi:<ALIGNMENT.qj></ALIGNMENT>10.1145/1454159.1454163

Moraes, M. C., Heuser, C. A., Moreira, V. P., & Barbosa, D. (2013). Prequery discovery of domain-specific query forms: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *25*(8), 1830–1848. doi:10.1109/TKDE.2012.111

Nguyen, H., Kang, E. Y., & Freire, J. (2008). Automatically extracting form labels. *Proceedings - International Conference on Data Engineering*, *1498–1500*, 1498–1500. Advance online publication. doi:10.1109/ICDE.2008.4497602

Ntoulas, A., Pzerfos, P., & Cho, J. C. J. (2005). Downloading textual hidden web content through keyword queries. *Proc 5th ACM/IEEE-CS Jt Conf Digit Libr*, 100–109. doi:10.1145/1065385.1065407

Onihunwa, J., Onifade, O., Ariyo, I., Omotugba, S., & Joshua, D. (2017). Scalable Framework for Locating Deep Web Entry Points. *IOSR Journal of Computer Engineering*, *19*(02), 45–55. doi:10.9790/0661-1902034555

Toda, G. A., Cortez, E., & Da Silva, A. S. (2010). A Probabilistic Approach for Automatically Filling Form-Based Web Interfaces. *Proc VLDB Endow, 4*, 151–160. doi:10.14778/1929861.1929862

Toda, G. A., Cortez, E., & Mesquita, F. (2009). Automatically filling form-based web interfaces with free text inputs. *Proc 18th Int Conf World wide web, 1163*. doi:10.1145/1526709.1526908

Troiano, L., Birtolo, C., & Armenise, R. (2017). Modeling and predicting the user next input by Bayesian reasoning. *Springer J Soft Comput, 21*(6), 1583–1600. 10.1007/s00500-015-1870-7

Wang, J., & Lochovsky, F. H. (2003) Data extraction and label assignment for web databases. *Proc Twelfth Int Conf World Wide Web, 187*. doi:10.1145/775152.775179

Wu, P., Wen, J. R., Liu, H., & Wei-Ying, M. (2006). Query selection techniques for efficient crawling of structured Web sources. *Int Conf Data Eng*, *47*, 47. Advance online publication. doi:10.1109/ICDE.2006.124