Venkatesh.B *et al*, International Journal of Advances in Agricultural Science and Technology,

Vol.1 Issue. 1, December- 2013, pg. 16-26 **ISSN: 2348-1358**

Web Image Interpretation Using Graph Grammar

Venkatesh.B[1], Prakash.P[2]

[1]K.S.R College of Engineering/Computer Science and Engineering, Thiruchengode, India
Email: Venkyq1@gmail.com

[2]K.S.R College of Engineering/Computer Science and Engineering, Thiruchengode, India
Email: prakaash@aol.com

Abstract

Web knowledge extraction becomes a hot topic once the invention of World Wide Web, as a result of the massive quantity of information on the online makes it difficult to retrieve helpful information. Thanks to the varied styles and show of information on completely different websites, it's onerous to implement a general idea to extract knowledge completely different websites. This paper presents a unique technique supported graph grammar to extract the constant sort of data from completely different Web sites without the requirement of coaching or adjustment. Our approach formalizes a standard Web pattern as a graph grammar. Then, based on the visual layout and HTML DOM structure, a Web page is abstracted as a spatial graph that highlights the essential spatial relations between data objects. According to the defined graph grammar, a spatial parsing is performed on the spatial graph to extract structured records. We've evaluated our approach on twenty completely different Web sites, and achieved the Final Score as 97.47% which shows promising performance.

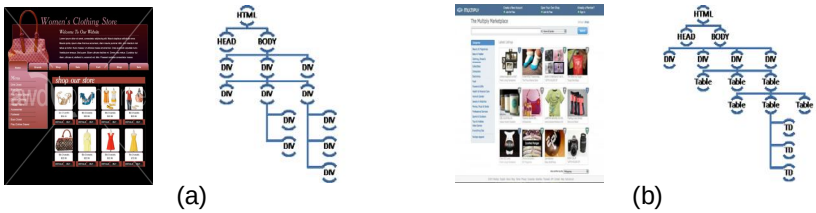
Keywords: Web knowledge Extraction; Graph Grammar; Wrapper; HTML DOM; Spatial Parsing

1. Introduction

Exploring helpful information becomes progressively tough as the volume and variety of obtainable info apace grow. To expeditiously discover information from the huge quantity of heterogeneous knowledge on the Web, it's vital to extract meaningful contents from sites and organize extracted information in a very structured format, i.e. *Web knowledge extraction.*

HTML DOM structures can be terribly various among different Websites. As an example, some Website designers could use *table* to represent tabular information whereas others use *table* to divide space into totally different grids for layout purpose. Therefore, even if two Web content have similar layouts, their HTML supply codes may be utterly totally different. As an example, Figure 1 presents two Web content that have similar layouts however totally different DOM structures. Because of the variety, it's difficult to create a wrapper applicable to totally different websites of constant class. In addition, a DOM structure is complex. As an example, even the DOM structure of Google homepage includes one hundred ten HTML tags. The complexity of DOM structures might any further increase the diversity and cut back the accuracy. Recently, layout primarily based analysis receives a lot of and a lot of attention [5, 16, 11]. In order to provide economical net browsing, Web content that embrace similar data generally area unit given with consistent layouts albeit those pages is also enforced completely otherwise. Therefore, layout primarily based analysis addresses the variety issue to an exact degree. However, existing layout-based approaches have restricted relevancy. Some approaches [11] would like training before they're applied to a Web site. Some approaches area unit

optimized for a selected sort of domains, and will not be simply adjusted to a different domain. For example, the Visual Wrapper [5] is powerful to extract news stories whereas it's not applicable to different domains.



(a) (b)

Figure1. The same kind web page layout with different DOM structures

This paper presents a completely unique approach that mixes layout and DOM structure analysis. Our approach extracts structured records by analyzing the screenshot of a Website that's coherent with human's psychological features of beholding. In our approach, the screenshot of a Website is start abstracted as a spatial graph, during which a node is an info object (such as text or image) and an edge indicates a close semantic relation between two info objects. Rather than victimization machine vision technique to recognize pictures and texts, our approach recognizes info objects supported the DOM structure. Though a DOM-structure primarily based recognition isn't as powerful as the machine vision technique, it's efficient and sparse to recognize texts, links and pictures that are helpful in information extraction. Supported the spatial graph, we're the graph parsing technique to extract structured records.

The graph parsing assumes that Website designers basically follow some pointers or patterns to represent information on the Website. This is often a valid assumption in practice since a consistent layout vogue will give effective browsing for final users and ease the efforts of development and maintenance. This assumption has been valid by our analysis on different internet sites and also by different researchers [17]. Those common pointers are said to as patterns that are visually specified through graph grammars [9] in our approach.

Since the input of knowledge extraction could be a Web content that's abstracted as a spatial graph and also output could be a tree, the method of knowledge extraction is through of as remodeling from one graph to another one which will naturally mere through the graph grammar technology. Graph grammars offer a solid theoretic foundation to outline computing in an exceedingly two-dimensional space. In our approach, a graph grammar visually nevertheless formally defines a Website pattern, so the knowledge extraction is implemented as a method of graph parsing that searches in an exceedingly spatial graph the sub-structures in keeping with the outlined pattern. So as to attenuate the manual effort of coming up with a graph grammar, we have a tendency to enforced a graphical interactive tool to facilitate the design of graph grammars. We have evaluated our approach on 18 Websites that extract product data. The results are promising and also the performance of our approach measured in terms of Final Score (See Section V for additional detail) is high

2. Related Work

With a transparent structure to specify the layout of a Website page, the hypertext markup language source codes are basically analyzed to extract structured knowledge records [1, 2, 3, 6, 7, 8, 10, 11, 12, 13, 14, 18, 19]. Many approaches [8, 10, 13] use the machine learning technique to mechanically derive a wrapper supported a collection of manually labeled training knowledge. Although the above approaches apply totally different technologies to derive a wrapper, they all require a collection of training knowledge, which are manually labeled by human consultants. Many approaches [3, 6, 7, 14] mechanically derive a example from sample Web contents and use the extracted template to find structured records. These approaches do not need manually labeled knowledge, which greatly reduces the manual effort within the knowledge extraction method. However, they require that Web content being analyzed should follow the constant template because the sample Web content. MDR [12], e.g., *table*, *form*, *tr*, *td*, and etc. This hypertext markup language tag tree considerably reduces the complexness of the original internet page. Supported the hypertext markup language tag tree, they use the string comparison technique to divide an internet page into different regions. In every region, it identifies knowledge records by shrewd similarity between tag strings. Zhai *et. al.* [19] rendered an internet page and allowed users to pick out info objects in the screen shot to outline an information pattern. Different from our approach, this knowledge pattern is outlined on the DOM structure, not on the visual layout. Zheng *et. al.* [11] used some applied mathematics analysis to investigate the DOM tree parts

and establish the relevant info objects. All of the higher than ways use HTML DOM structure because the main source for knowledge extraction. Instead, our approach i.e. *Visual Grammar Based Extractor -VGE*, implements knowledge extraction supported the knowledge presentation. The visual analysis will address the problem of complexity and various usages of hypertext markup language DOM structures to a certain degree. By really rendering an internet page, our approach supports dynamic info objects which are generated at run time.

Recently, the visual perception technique has been applied to extract structured knowledge since it's freelance from the detailed implementation underlying an internet page. These approaches [5, 16,11] primarily calculate the visual similarity among completely different sites to cluster semantically connected information. [5, 11] are restricted to extract news stories, and are not applicable to alternative domains. ViPER [16] is enforced on statistical models that emphasize on extracting repetitive knowledge records.

The Hybrid technique takes advantages from both DOM Structure and Visual Perception, and combines them along. ViNTs [18] mechanically acknowledges completely different content shapes supported the visual position of data objects. Afterward, a wrapper is generated supported an HTML structure which represents each shape. This approach still extracts data supported HTML DOM structures, although the wrapper spring from visual analysis. Instead, our approach specifies extraction rules from both the layout and therefore DOM structure. ViNTs is restricted to search results, while our approach is general to completely different domains.

3. A Grammar Based Approach

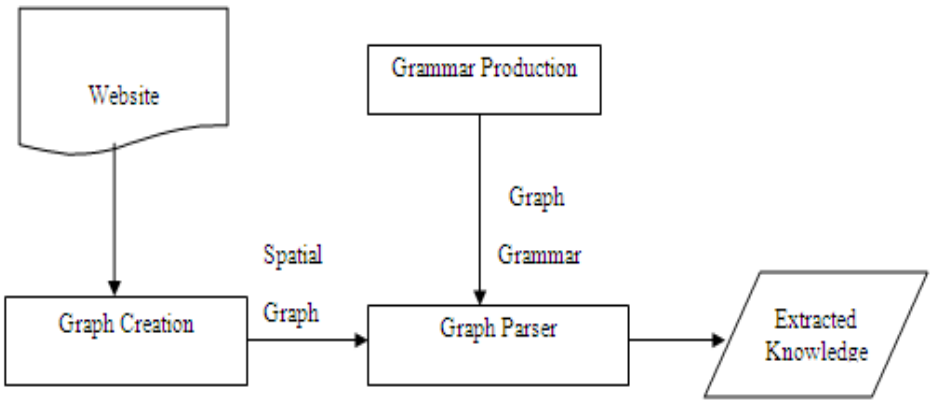


Figure 2. A Graph Grammar supported Extractor

This paper presents a completely unique and sturdy approach, i.e. ***V**isual **G**rammar based **E**xtractor - VGE*, to extracting structured information. Our approach consists of three parts as shown in Figure 2. The graph generation element abstracts a Web page as a spatial graph that simplifies the original internet page and highlights necessary semantic relations between recognized data objects. The graph generation income in the following steps: (1) render an internet page on the screen, (2) recognize data objects and divide an internet page into different regions in step with its DOM structure, (3) calculate semantic relations between recognized data objects based on the layout data, and eventually (4) optimize the spatial graph. Supported the generated spatial graph, the data extraction is enforced as a graph parsing method that searches within the spatial graph sub-graphs satisfying bound spatial properties. Those spatial properties are visually outlined through a graph grammar. The grammar generation element provides associate interactive grammar generate tool that permits end users to outline a graph grammar by directly manipulating the screenshot of an internet page. This interactive grammar design tool eases the method of developing a graph grammar and improves the usability of our approach.

3.1 Graph Generation

HTML could be a terrible versatile language. Info with the same presentation can be enforced in many different ways. Being coherent with the HCI principle that consistent presentations will improve the usability of an interface [15], Web designers across completely different internet sites unremarkably use similar layouts to provide the equivalent kind of information. Therefore, our approach extracts structured records by analyzing the layout of an internet page. The visual analysis will address the variety of HTML usages and build our approach applicable to completely different internet sites. The method of graph generation could be a important step in our approach since it simplifies original Web content and eliminates variations among completely different Web pages. The simplification solely preserves essential information objects. Especially, the graph generation method removes (1) vogue and layout parts, which don't embrace any real content, (2) advertisements and (3) menus within the border areas. The simplification effectively reduces the complexity of HTML pages and removes potential noises within the data extraction. The graph generation method returns in three steps: Website rendering, node and edge generation, and graph optimization.

The first step within the graph generation is to render an internet page. The visual layout of an internet page is decided by three variables, i.e. (1) the particular hypertext mark-up language source code that specifies the DOM structure of the page, (2) knowledge things like text and picture and (3) vogue sheets and consumer side scripts which are executed by a browser at run time. We will access all hypertext mark-up language elements, particularly dynamic parts which are generated on the fly, solely by actually rendering an internet page. Also, the page rendering determines the position and size of every component. Based on the dynamic and static hypertext markup language components and their spatial properties, the second step generates a spatial graph in which a node represents a data object for information extraction and an edge indicates a close semantic relation between the pair of connecting nodes. Contents are stored in three varieties of nodes, i.e. image, text and link. The contents enclosed within the <img> or <a> tags are recognized as a picture node or a link node, respectively. However, it is difficult to identify a text node since one complete sentence is also separated by many HTML tags and it is necessary to consolidate those data items along. For instance, inside the text block of a sentence, format and styling tags (such as <b>,
, <font>,) will divide the sentence into many items. Within the graph generation, all those format and styling tags are removed and adjacent contents are consolidated together single text node.

After distinctive atomic info objects as nodes, it is critical to calculate semantic relations between info objects and use an edge to connect two nodes that are closely related in semantics. In an exceedingly two dimensional space, an information object will have an arbitrary spatial relation with adjacent nodes. A whole spatial parsing that analyzes different spatial properties in an exceedingly graph might be time overwhelming. Our approach initial derives the semantic relation between adjacent nodes, and every close semantic relation is delineated as an edge in the spatial graph. Supported the derived semantic relations, we will limit the spatial parsing to objects that have semantic relations and therefore cut back the search space to speed up the parsing method. We have got extensively investigated totally different Web sites and located that a low distance powerfully indicates a close semantic relation between two objects. This observation is per the Human Computer Interaction principle that closely connected objects ought to be classified along and placed in proximity [15]. Consequently, we tend to derive the semantic relation by calculating the space between two objects. Also, an HTML DOM structure provides valuable hints for deriving semantic relations. Website designers cluster related objects together by employing a container, like *table* or *div*. In general, two objects belonging to two containers don't seem to be related. For example, in Figure 3, though *text* objects 4 and 5 are placed in proximity, they are not semantically connected since they belong to two totally different containers. Our approach uses HTML DOM structures to recognize the containers, and semantic relations are restricted to info objects that have one common (ancestor) container. So as to accommodate totally different image sizes and variations in Websites, we tend to propose a completely unique approach to calculating distance and deriving semantic relations. The size of an info object *a* is extended to a certain degree. If the extended object *a* is overlapping with at least two corners of another info object *b*, *a* has a semantic relation with *b*.

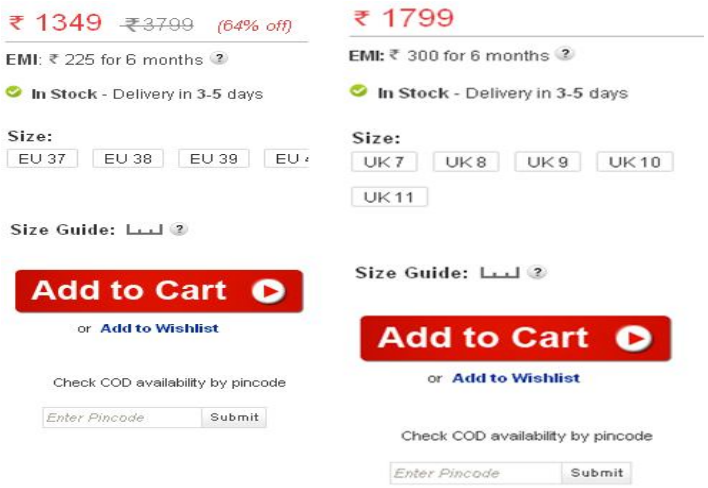


Figure 3. Different containers in a Website

The last step within the graph generation is to optimize the generated spatial graph. In a spatial graph, some nodes is also considered as noises (such as advertisements and menus), which do not contribute to the information extraction method. Since those objects in generally are placed in the border areas of a Website page, we are able to take away them according to their position. Another type of noise is little repetitive images, like the "*Add to Cart"* icon in Figure 3.

3.2 Grammar Generation and Parser

The graph generation component generates a spatial graph, and the knowledge extraction is performed on a spatial graph to search for data objects having certain spatial relations. In order to support efficient browsing, the constant variety of information in generally is bestowed equally across totally different Web sites. Those consistent spatial features among data objects are summarized as an internet pattern that is visually specified through a graph grammar. A graph grammar defines computation in a multi-dimensional fashion supported a set of rewriting rules, i.e. *productions*. Since the input of knowledge extraction could be a graph and therefore the output could be a tree structure that represents structured records, the information extraction is essentially a process of graph transformation which will be naturally specified through graph grammars. Furthermore, a graph grammar is powerful to handle the variations among instances of a website pattern. This paper selects the Spatial Graph Grammar (SGG) [9] because the specifying formalism. With the potential of spatial specification within the abstract syntax, SGG provides the flexibility to outline a pattern from both the edges (i.e. close semantic relations) and spatial features (e.g. directions).

Instead of coming up a graph grammar from scratch, we designed an associate interactive design tool that enables users to design a graph grammar visually and intuitively. This interactive tool renders a sample internet page on the screen, and highlights recognized info objects within the internet page. Users can directly choose one or additional information objects within the internet page to make a production. This tool supports a immediate manipulation interaction on the grammar design that reduces the gap between a concrete Web pattern and associated an abstract graph grammar. With the help of this tool, even users while not a lot of training in graph grammars could design a graph grammar.

4. System Implementation

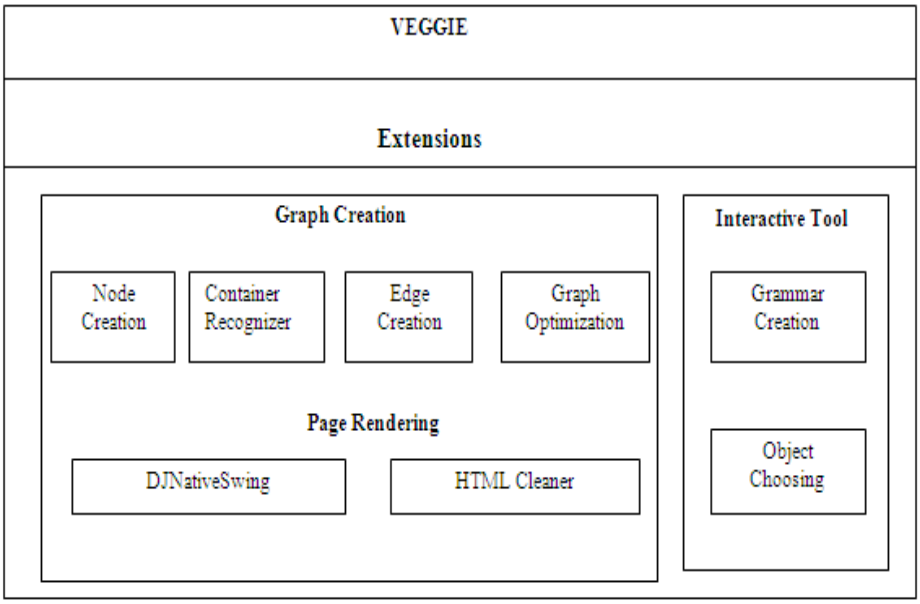


Figure 4. The VGE system structure

We have enforced a model for our approach. Our model is built based on VEGGIE - **V**isual **E**nvironment for **G**raph **G**rammars: **I**nduction and **E**ngineering [4]. VEGGIE is a general visual programming environment, and supports the Spatial Graph Grammar specification and parsing. VEGGIE mainly consists of three free-lance editors (i.e., the Type Editor, the Grammar Editor, and therefore the Graph Editor) and an SGG parser. The three editors give GUIs for designers to visually design a graph grammar, and are closely related and seamlessly working along in VEGGIE. Grammar designers will visually create visual objects, i.e. node types, within the Type Editor, or import existing node types from a file from go into the shape of GraphML. Then, supported these outlined nodes, the designer will outline productions within the Grammar Editor. The designer will visually draw or import a host graph to be analyzed by the SGG parser. As shown in Figure 4, we have extended VEGGIE with two subsystems, each enforced in Java. The primary sub-system is responsible to generate a spatial graph from an internet page. The generated spatial graph is fed to the SGG parser for a spatial parsing. The second sub-system provides associate interactive graphic tool to design a graph grammar.

The Graph Generation sub-system has many components. The *Page Rendering* part renders a Web page, extracts size/position data and passes the output to the *Node Generation* to come up with nodes. Containment relations are identified by the *Containment Recognizer*. The *Edge Generation* derives semantic relations supported containers and spatial properties. The *Graph Optimization* part optimizes a generated spatial graph by removing noises.

The *Page Rendering* component renders a Web page primarily based on the DJNativeSwing Browser .The HTML Cleaner component solves syntactical issues (e.g., unclosed tags and markup errors). The HTML Cleaner returns a clean and well-structured HTML DOM tree that has all static and dynamic elements. Supported this DOM tree, the graph generation component, as well as *node generation*, *container recognizer*, *edge generation* and *graph optimizer*, generates an optimized spatial graph for information extraction. The second subsystem eases the method of designing a graph grammar. It consists of two components, i.e. *object selection* and *grammar generation*.

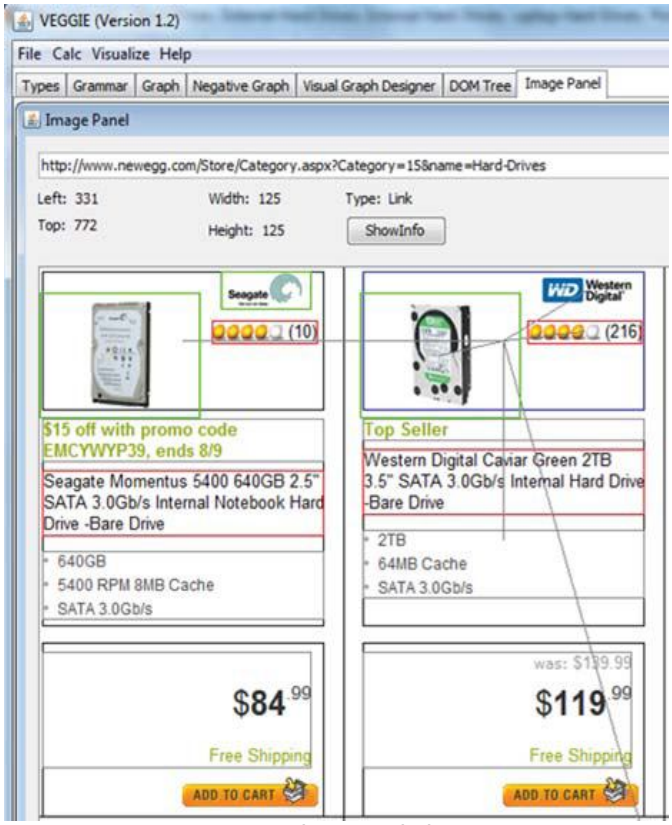


Figure 5. Browsing a Website

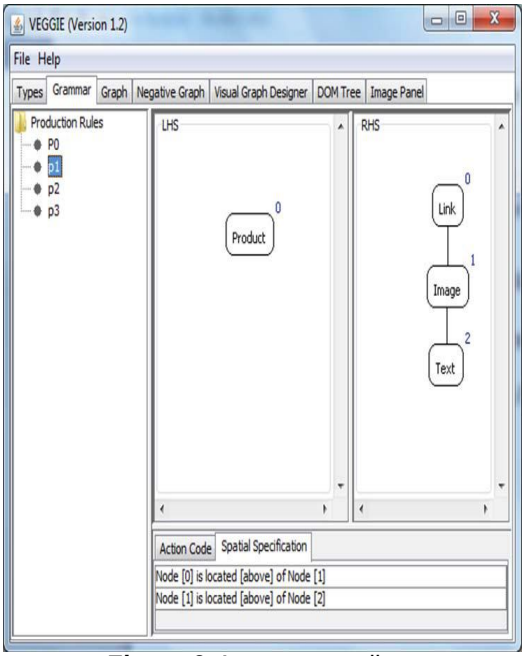


Figure 6. A grammar editor

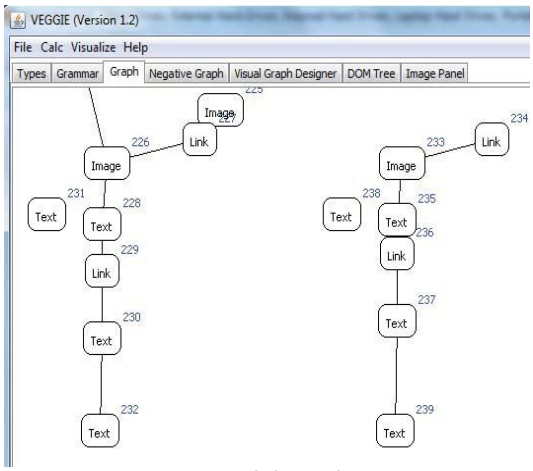


Figure 7. A spatial graph

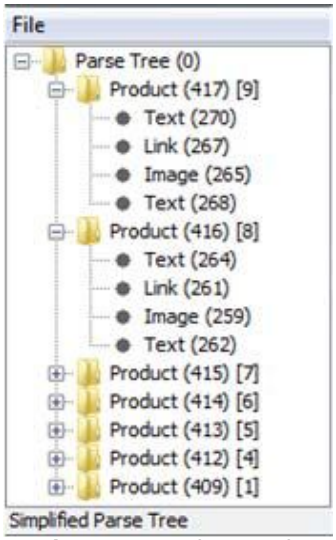


Figure 8. Parsing result

The prototype provides a complete system to support information extraction, from the specification of extraction information (i.e. defining a graph grammar) to the image of extraction results. At first, a user uses the *Image Panel*, as conferred in Figure 5, to show an internet page. In Figure 5, recognized data objects are highlighted with rectangles, and edges indicate close semantic relations. Supported the screenshot of an internet page, a user will choose related information objects to intuitively design a graph grammar. Once a graph grammar is outlined, the user will move to the VEGGIE Type Editor or Grammar Editor (as presented in Figure 6) to elaborate the designed graph grammar. Once a graph grammar is finalized, a user will use the prototype to extract structured records that are consistent with the defined graph grammar. A user initial inputs the URL of an internet page in the image panel. Then, the related spatial graph is automatically generated and might be retrieved within the graph panel, as conferred in Figure 7. By applying the outlined graph grammar to the spatial graph, a low popup window shows up to present the extracted records, as exposed in Figure 8.

5. Experiment

This section discusses the first experiment on VGE. We first discuss the design of the experiment, and then present the results.

5.1 Setup

- **Experiment web pages:** We have evaluated our approach on 18 online shopping internet sites, which embrace accepted internet sites, such as ebay.com, lycos.com, amazon.com, compusa.com, and etc.
- **Measurement:** We measured the performance with the standard metrics:

$$recall = \frac{Ecorrect}{Ntotal}; precision = \frac{Ecorrect}{Etotal};$$

Where *Ntotal* is that the number of information records contained in an internet page; *Ecorrect* indicates the total number of properly extracted information records; and *Etotal* denotes the total number of knowledge records extracted from an internet page. We have a tendency to conjointly calculated F1- Score, which is the harmonic mean of *precision* and *recall* and is outlined as

$$\frac{2 \times recall \times precision}{recall + precision}$$

The F1-Score has been commonly used as a metric to gauge the general performance in several approaches [5, 11].

- **Execution Platform:** We have got evaluated our approach on a desktop with a Core 2 Duo CPU 2.82 GHz and 2 GB RAM, running Windows XP Professional.

5.2 Evaluation

- **Precision/Recall/Final-Score:** The analysis results are presented in Table 1. The recall of our approach is 99.2%. The high recall rate in our approach indicates that graph-grammar based visual analysis is powerful to recognize structured records. The precision of our approach is 95.8%. Our approach may incorrectly recognize some records, which are primarily caused by noise. As an example, if a commercial is placed within the central area and its overall layout is analogous to our designated pattern (e.g. including a link, a image and several lines of textual description that are displayed vertically); this advertisement is also recognized as a product record. In order to improve the precision, it is vital to enhance the graph generation method by removing potential noise. Final Score shows the overall performance. Our approach encompasses a high Final Score of 97.47%. In summary, the results indicate our approach has a sensible performance in terms of both precision and recall.

Table 1. Evaluation Results

Website Domain Name	Number of Structured Records	Our Results	
		Exact Records	Founded Records
www.snapdeal.com	16	16	16
www.yebhi.com	13	13	14
www.homeshop18.com	20	18	18
www.flipkart.com	46	46	46
www.futurebazaar.com	29	30	31
shopping.rediff.com	5	5	7
www.futurebazaar.com	18	18	21
www.indiaplaza.com	17	15	16
shopping.indiatimes.com	20	20	20
www.tradus.com	30	30	30
www.pepperfry.com	28	28	28
www.shoppingmantra.com	21	21	21
www.myntra.com	21	21	27
www.lotofstock.com	23	23	23
www.shop.irctc.co.in	19	19	19
www.americanswan.com	51	51	52
www.yepme.com	13	13	14
www.ebay.com	9	9	10
Total	399	396	413
Recall/Precision		99.2%/95.8%	
Final Score		97.47%	

6. Conclusion

This paper presents a unique and general solution to extract data across totally different internet sites. Our methodology works supported graph grammars to extract the constant type of data from different Web sites while not necessary of training and adjustment for different internet sites. Our approach utilizes both the visual features of a rendered Web page and also the HTML DOM structure to extract structured records. We have enforced a prototype and tested the model on twenty one internet sites. The evaluation shows promising results. Our approach encompasses a high Final Score as 97.47%. The analysis results indicate our approach encompasses a sensible performance in terms of both precision and recall. The important advantage of our approach lies in its ability to distinguish the most important contents from less important and noisy information and to convert the complicated HTML DOM structure to a simple spatial graph. The generated spatial graph considerably reduces the complexity of the original Website. Supported the simplified spatial graph, our approach is efficient to extract structured records through a graph parsing.

In the future work, we will identify additional spatial relations between data objects, and optimize the graph generation algorithm. These optimizations might increase the quality of generated spatial graphs, which can affect both the precision and recall.

References

[1] F. Ashraf and R. Alhajjt, "ClusTex: Information extraction from HTML pages," in *Proc. 21st Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2007, pp. 355–360.

[2] K. Ates, J. Kukluk, L. Holder, D. Cook, and K. Zhang, "Graph grammar induction on structural data for visual programming," in *Proc. IEEE 18th Int. Conf. Tools Artif. Intell.*, Nov. 2006, pp. 232–242.

[3] Arasu and H. Garcia-Molina, "Extracting structured data from web pages," in *Proc. Special Interest Group Manage. Data Conf.*, 2003, pp. 337–348.
[4] K. Ates and K. Zhang, "Constructing VEGGIE: Machine learning for context-sensitive graph grammars," in *Proc. IEEE 19th Int. Conf. Tools Artif. Intell.*, Oct. 2007, pp. 456–463.
[5] J. Chen and K. Xiao, "Perception-oriented online news extraction," in *Proc. 8th ACM/IEEE-CS Joint Conf. Digital Libraries*, 2008, pp. 363–366.
[6] S.Chuang and J.Y.Hsu, "Tree-structured template generation for Web pages," In *Proceedings of the 2004 IEEE/WIC/ACM international Conference on Web intelligence*. IEEE Computer Society, Washington, DC, 327-333.
[7] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: Towards automatic data extraction from large web sites," in *Proc. Very Large Data Bases Conf.*, 2001, pp. 109–118.
[8] C.Hsu, and D.Dung, 1998. "Generating finite-state transducers for semi-structured data extraction from the Web," *Inf. Syst.* 23, 9, 521-538.
[9] J. Kong, K. Zhang, and X. Q. Zeng, "Spatial graph grammar for graphic user interfaces," *ACM Trans. Human-Comput. Interaction*, vol. 13, no. 2, pp. 268–307, 2006.
[10] N. Kushmerick, D. S. Weld, and R. B. Doorenbos, "Wrapper induction for information extraction," in *Proc. Int. Joint Conf. Artif. Intell.*, 1997, pp. 729–737.
[11] S. Zheng, R. Song, and J. Wen, "Template-independent news extraction based on visual consistency," in *Proc. 22nd Nat. Conf. Artif. Intell.*, 2007, vol. 2, pp. 1507–1512.
[12] B.Liu, R.Grossman , and Y.Zhai, "Mining data records in Web pages," In *Proceedings of the Ninth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining.*, ACM,2003, New York, NY, 601-606.
[13] N. Kushmerick, D. S. Weld, and R. B. Doorenbos, "Wrapper induction for information extraction," in *Proc. Int. Joint Conf. Artif. Intell.*, 1997, pp. 729–737.
[14] D.C.Reis, P.B.Golgher, A.S. Silva,and A.F.Laender, "Automatic web news extraction using tree edit distance," In *Proceedings of the 13th international Conference on World Wide Web*. ACM, 2004, New York, NY, 502-511.
[15] B.Shneiderman, "Designing the User Interface: Strategies for Effective Human-Computer Interaction," Addison-Wesley Longman Publishing Co., 2003, Inc.
[16] K.Simon, and G.Lausen, "ViPER: augmenting automatic information extraction with visual perceptions," In *Proceedings of the 14th ACM international Conference on information and Knowledge Management*. ACM, 2005, New York, NY, 381-388.
[17] Z.Zhang, B.He, and K.C.Chang, "Understanding Web query interfaces: best-effort parsing with hidden syntax. In *Proceedings of 2004 ACM SIGMOD International Conference on Management of Data*, 107-118.
[18] Y.Zhai, and B.Liu , " Web data extraction based on partial tree alignment," In *Proceedings of the 14th international Conference on World Wide Web*. ACM , 2005, New York, NY, 76-85.
[19] Y. Zhai, and B.Liu, "Extracting Web data using instance based learning," *World Wide Web* 10, 2, 2007,113-132.

