A New Method Based on Tree Simplification and Schema Matching for Automatic Web Result Extraction and Matching

Yasar Gozudeli, Hacer Karacan, Oktay Yildiz, Mohammed R. Baker, Ali Minnet, Murat Kalender, Ozcan Ozay, M. Ali Akcayol

***Abstract*— In this paper, a new method proposed for extracting and matching the Search Result Record (SRR) data items from different search engines. The method first detects SRRs for a given Web search result. Afterwards, an SRR simplification algorithm is devised to deal with complexity of SRR Document Object Model (DOM) Trees. SRRs and their data items (or properties) are extracted after simplification. Data items are normalized in local and global domain as a last step. Experimental results show that the proposed methods are successful in extracting and merging the SRRs.**

***Index Terms*— Automatic Web extraction, meta-search engines, schema matching, SRR, tree similarity.**

I. Introduction

The Web can be categorized into two groups; Surface Web and Deep Web. The surface Web is conventional, same for all the users and it has been crawled by all search engines for about twenty five years. Deep Web is a relatively new area and defined as the Web pages that are accessible after filling some Web forms properly. Academic researches on deep Web have been expanded for last decade after the term “Deep Web” introduced at 2000 [1]. The number of Web databases was approximately calculated to be between 10 to 25 million pages in 2007 [2], [3].

In meta-search engines the result lists are gathered from different public Web search engines. Each item in a Web search engine result is named as a Search Result Record (SRR). All of the SRRs are formed of varying data items. An example of SRR and its data items are shown in Figure-1. Some of the data items exist in all SRRs and are called mandatory. Some other data items are optional and may not be presented by all SRRs. Finding the position of SRRs, extraction of SRRs and data items, and deciding their structures are the topics of automatic Web result extraction. After those, matching different structures in a global result screen is possible with schema matching techniques.

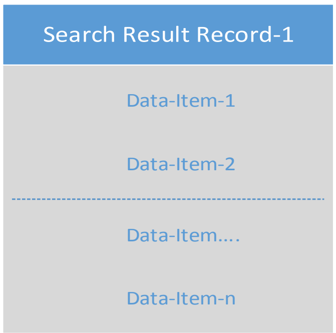


Figure 1. An example of SRR and its data items. Some of the data items are mandatory and some others are optional.

Automatic Web result extraction is a common resource area [4], [5], [6]. The methods of Web result extraction include wrapper generation or automatic template generation to resolve the repeating patterns.

In literature, there is no consensus on the definition of a wrapper. Simply a wrapper is defined as predefined code or symbolic definition used for Web result extraction [7]. Wrapper induction based approaches generally have three steps: (1) wrapper generation, (2) wrapper execution and (3) wrapper maintenance. A wrapper has to be defined and generated before its usage by a developer, a user or software in automatic systems. Then, Web SRR can be extracted by executing the wrapper [8]. The wrapper needs to be adapted as Web resource template alters in time.

Automatic template generation is another popular Web extraction method [9]. This method assumes that the Web result pages are generated in some loops by server side programs so that the results have common patterns. Automatic template generation approaches generally use tree structures to find repeated patterns [10]. While some

Manuscript received 29 December 2014. This work was supported by the Huawei Technologies Co., Ltd. and Ministry of Science, Industry and Technology of Turkey (Project Number 0431.STZ.2013-2).

Yasar Gozudeli. PhD candidate Computer Engineering Dept. of Gazi University, Maltepe, 06570, Ankara, Turkey (corresponding author: ygozudeli@gmail.com).

Hacer Karacan. Assistance Professor at Computer Engineering Dept. of Gazi University, Maltepe, 06570, Ankara, Turkey (hkaracan@gazi.edu.tr).

Oktay Yildiz. Instructor at Computer Engineering Dept. of Gazi University, Maltepe, 06570, Ankara, Turkey (oyildiz@gazi.edu.tr).

Mohammed R. Baker. PhD candidate Computer Engineering Dept. of Gazi University, Maltepe, 06570, Ankara, Turkey (mr_baker@live.com).

Ali Minnet. Huawei Technologies Co., Ltd., Istanbul, Turkey (ali.minnet@huwei.com).

Murat Kalender. Huawei Technologies Co., Ltd., Istanbul, Turkey (murat.kalender@huawei.com).

Ozcan Ozay. Huawei Technologies Co., Ltd., Istanbul, Turkey (ozcan.ozay@huawei.com).

M. Ali Akcayol. Professor at Computer Engineering Dept. of Gazi University, Maltepe, 06570, Ankara, Turkey (e-mail: akcayol@gazi.edu.tr).

researchers focus on visual items [11], [12] and use visual block trees, the others use HTML tag trees [13], [7].

After extracting an SRR, its data items, and re-structuring the Document Object Model (DOM) Tree of data items, a method is required to merge the SRRs from different search engines under a common schema definition. Automatic schema matching algorithms are the mostly preferred for this problem.

Schema matching algorithms are grouped as element level, structural level and instance level techniques with respect to matching level of data. Methods like string matching, linguistic similarity and constraint similarity compare the element level data. The methods that use schema and structure of schema are called structure level schema matching techniques. Some matching approaches may use the data instance inside a schema to match different schemas. These kinds of methods are called instance level schema mapping methods [14], [15].

In this study, a new model has been developed for SRR item extraction and merging without any user interaction or manual process.

Compared to existing approaches, this study has following contributions.

- A new approach has been developed for automatic SRR item extraction from a known data region.
- A practical new method is developed for SRR HTML DOM tree simplification.
- A new method is devised to convert a non-structured Web data into semi-structured data.
- The proposed methods and algorithms are evaluated on common Web databases and large variety of search results.

This paper is structured as follows. Related work is discussed in Section 2. In addition to describing the approach, Section 3 also discusses the developed algorithm. Section 4 presents the test environment and the last section concludes the study.

II. RELATED WORKS

There are many studies on Web search result extraction in the literature. [4], [5] and [6] present different methods and classification techniques for Web result extraction approaches.

The earliest studies have manual approaches based on wrapper induction. In these approaches, manually labeled Web page instances are used for learning Web result extraction.

On later researches, studies have focused on various semi-automatic and automatic approaches. Semi-automatic methods can be classified into two categories: string based techniques and tree-based techniques. Wien [16] and Stalker [17] proposed string based techniques. In these studies, Web results are assumed as a flat-sequence of strings and delimiter sections are determined by the support of manually labeled Web result documents. W4F [18] and Wrapper [19] parse Web documents into hierarchical (Document Object Model-DOM) trees instead of flat-string. Finally, a set of delimiter based rules is generated after manually labeled training instances, since, string based and tree-based semi-automatic methods required human interaction. Hence, these methods are not suitable for huge amount of Web data extraction processes of today.

Recent studies have suggested automatic approaches to deal with the scaling of deep Web [20] such as, IEPAD [21], MDR [22], RoadRunner [9], EXALG [23], DEPTA [24], Tag of Path [13]. While some automatic extraction approaches use more than one page to resolve the SRRs, some others use one page. Both IEPAD and MDR are focused on extracting Web results from only one Web page by generating rules. While IEPAD identifies repeated substrings as tokens, MDR uses similarity based aggressive approach to match two segments.

SRR items can be extracted by using visual tree processing approaches or HTML node processing techniques. In this paper, HTML node processing techniques have been used. IEPAD, MDR, Tag of Paths, Ranking XPaths and Content Density [8] are the newest studies in this area. Content Density suggests classifying SRRs as regular SRRs and Irregular SRRs before fully extracting them.

Schema matching is an old problem for matching data coming from different data sources. The matching algorithms are studied in three different groups depending on the level of information used by matching process [14], [15]. (1) Item level, (2) structural level and (3) instance level algorithms [25].

Item level algorithms use item level data like names or other properties of the sources and relevant destinations to find the best matching option [14]. The properties are generally string values. Many of the String matching algorithms are inherited from Information Extraction discipline [14] [15]. Levenshtein and other edit distance algorithms, Cosine similarity and its derivatives, stemming and n-gram are some of the algorithms in this level. For example PruSM [26] uses Cosine similarity to match two schemas. In spite of ontological rule dependency, linguistic methods also belong to element level algorithms [14]. Most of the linguistic methods are not directly applicable in all languages due to external ontological data requirements. However, tokenization and elimination can be used with some limitations.

Structure level algorithms use structural information in process of schema matching. Generally acyclic graphs and tree structures are used to match different schemas structurally. Many of graph-based mapping scheme studies use neighborhood similarity to identify similar individuals. Similarity flooding algorithm can be mentioned in this category [27].

Tree Edit Distance [28] and Graph Edit Distance algorithms check schema similarity by adding or removing some nodes to achieve the exact match. However, complicated structures make it harder to find a qualified solution of matching.

Instance level schema mapping algorithms infer schema data from a data instance in a schema or unknown schema to enhance schema information or re-generate the schema [29], [30].

There are various schema matching platforms, like Cupit, CLIO, COMA, OLA and S-Match [14], [15]. These platforms combine different algorithms to improve schema matching quality. COMA [31], one of these platforms, is actively developed and maintained.

III. Methodology

Main goal of this study is to create a new model to extract SRRs from a single search result Web page instance. Afterwards, a transformation algorithm is applied on SRR DOM trees to re-structure the complex HTML tree into a rectangular data schema.

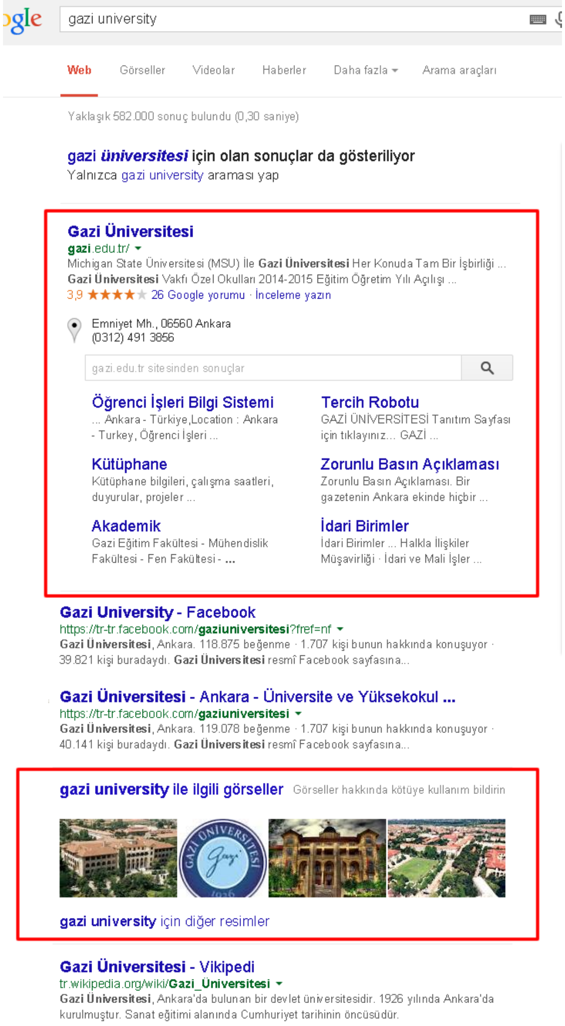
A. *The Algorithm*

Overall process of the proposed model is shown in Figure-2 in pseudo code format.

```
Algorithm overall follow
Input:test keyword, SE query url
Begin
call retrieveSearchResult(keyword);
call findSRR_Region();
foreach searchResults
call findMaxLeafLevelContentSegmentItem()
call detect dataTypeofSegment()
call collectLabelData()
call flattenHTML_DOM_Tree()
end

foreach SearchEngineSource
call weightedLocalTree()
end

call         generateGlobalTree(weightedLocalTree$_1$,
weightedLocalTree$_2$)
foreach weightedLocalTree$_i$
  rearrangeGlobalTree(weightedLocalTree$_i$)
end

end
```

Figure 2: The algorithm of overall process

Finding the position of SRRs is the first step and a very critical process in Web Result Extraction. Details of estimating the position of Web search results are explained in our previous study [8]. The SRRs are categorized as regular SRRs and irregular SRRs as seen in Figure-3 in the same study.



gazi university

Web Görseller Videolar Haberler Daha fazla Arama araçları

Yaklaşık 582.000 sonuç bulundu (0,30 saniye)

gazi *üniversitesi* **için olan sonuçlar da gösteriliyor**
Yalnızca gazi university araması yap

Gazi Üniversitesi
gazi.edu.tr/
Michigan State Üniversitesi (MSU) İle **Gazi Üniversitesi** Her Konuda Tam Bir İşbirliği ...
Gazi Üniversitesi Vakfı Özel Okulları 2014-2015 Eğitim Öğretim Yılı Açılışı ...
3,9 ★★★★☆ 26 Google yorumu · İnceleme yazın

Emniyet Mh., 06560 Ankara
(0312) 491 3856

gazi.edu.tr sitesinden sonuçlar

Öğrenci İşleri Bilgi Sistemi
... Ankara - Türkiye,Location : Ankara - Turkey, Öğrenci İşleri ...

Tercih Robotu
GAZİ ÜNİVERSİTESİ Tanıtım Sayfası için tıklayınız... GAZİ ...

Kütüphane
Kütüphane bilgileri, çalışma saatleri, duyurular, projeler ...

Zorunlu Basın Açıklaması
Zorunlu Basın Açıklaması. Bir gazetenin Ankara ekinde hiçbir ...

Akademik
Gazi Eğitim Fakültesi - Mühendislik Fakültesi - Fen Fakültesi - **...**

İdari Birimler
İdari Birimler ... Halkla İlişkiler Müşavirliği · İdari ve Mali İşler ...

Gazi University - Facebook
https://tr-tr.facebook.com/**gaziuniversitesi**?fref=nf
Gazi Üniversitesi, Ankara. 118.875 beğenme · 1.707 kişi bunun hakkında konuşuyor · 39.821 kişi buradaydı. **Gazi Üniversitesi** resmî Facebook sayfasına...

Gazi Üniversitesi - Ankara - Üniversite ve Yüksekokul ...
https://tr-tr.facebook.com/**gaziuniversitesi**
Gazi Üniversitesi, Ankara. 119.078 beğenme · 1.707 kişi bunun hakkında konuşuyor · 40.141 kişi buradaydı. **Gazi Üniversitesi** resmî Facebook sayfasına...

gazi university ile ilgili görseller Görseller hakkında kötüye kullanım bildirin

gazi university için diğer resimler

Gazi Üniversitesi - Vikipedi
tr.wikipedia.org/wiki/**Gazi_Üniversitesi**
Gazi Üniversitesi, Ankara'da bulunan bir devlet üniversitesidir. 1926 yılında Ankara'da kurulmuştur. Sanat eğitimi alanında Cumhuriyet tarihinin öncüsüdür.

Figure 3: Regular SRRs and irregular SRRs (inside the rectangles)

B. *HTML Flattening Algorithm*

Relational Databases are rectangular and easy to match the schemas. However, SRRs may consist of thousands of nodes, for this reason matching similar schemas in HTML is not as easy as in relational databases. Therefore, a method that identifies the segment of a data item is required to flatten an HTML complex tree into a rectangular data. Article Clipper [32] suggests an algorithm to extract the maximum text segment. Similarly, leaf content nodes of the HTML DOM tree are traversed and items are parsed into the format of flattened databases as in Figure-4 and Figure-5.

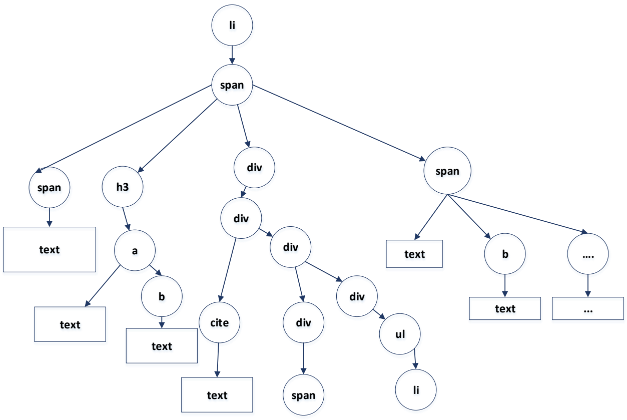


Figure 4: Tree diagram of an SRR DOM objects.

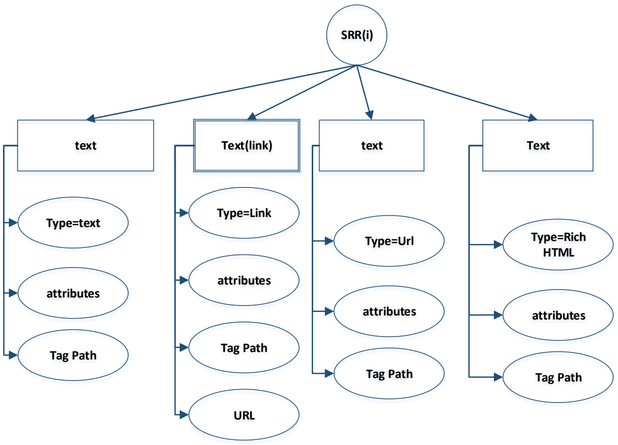


Figure 5: Flattened structure of an SRR HTML-DOM tree

C. *Type System for SRR Data Items*

A type system is needed to hold the transformed item data. For search results domain, a type system is suggested and used in this work as below.

```
SRR  Type  System  =  {URL,  Link,  Numerical  Value,
Date Time, Email, Price, Text, HTML Rich Text
}
```

In Figure-5, double framed rectangle shows Link typed data. Link type is specific to SRRs and it includes the link tag data in addition to link label.

D. *Evaluating Local Similarity and Finding Local normalized Schema*

Although irregular-SRRs are filtered with a certain approximation, still slightly different data items may exist in a single search engine's SRRs. Therefore, each new data item is added to the normalized SRR, the cardinality of each item is calculated for each result page and then frequency is associated with the related data item. After that, a cutoff value is used for filtering the rare data items. An example of frequency weighted item tree is shown in Figure-6.

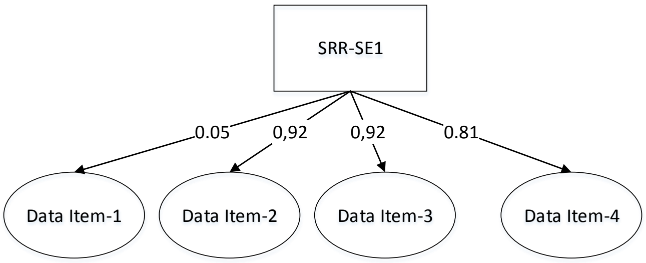


Figure 6. An example of generated frequency weighted item tree for a search engine result page

E. *Schema Matching Function*

It's assumed that the schema items are in 1:1 relation between the two local schemas. 4 different attributes are used to find the matching data items. These are,

- W_{CT} : Data types of the each date items
- W_{LT} : Label and Tag Path word similarity
- W_{CSF} : Content Size Factor [7] similarity
- W_{FS} : Item frequency similarity.

All items are checked for best matching similarity value consecutively according to formula (1).

$$W(A_i, B_j)= W_{CT} + W_{LT} + W_{CSF} + W_{FS} \qquad (1)$$

Where, A_i and B_j are local schema data item definitions from different search engines.

IV. Experimental Results

The proposed method has been evaluated on leading search engines in Turkey (Google, Bing and Yandex) with a keyword list described in [8]. The keyword list has been built manually from different domains.

F-score value is calculated by using following equations.

$$precision=\frac{|true\ positives|}{|true\ positives|+|\ false\ positives|} \qquad (2)$$

$$recall=\frac{|true\ positives|}{|true\ positives|+|\ false\, negatives|} \qquad (3)$$

$$f-score=2*\frac{precision*recall}{precision+recall} \qquad (4)$$

F-score graph depending on number of experimental results is shown Figure 7.

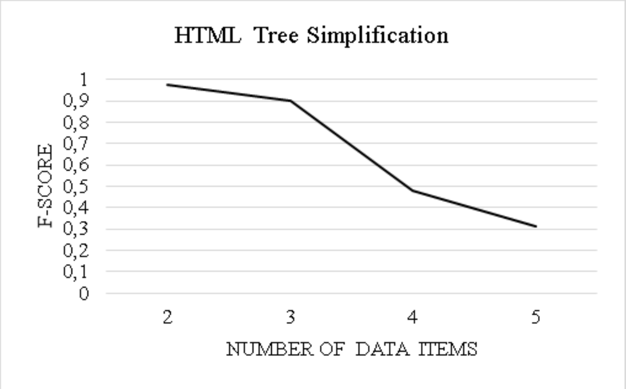


Figure 7. F-score graph depending on number of data items

The first three data items are mandatory in the performed experiments, while the others are optional. While the method can easily match mandatory data items, there is a significant quality reduction in the mapping of optional data items. This reduction, however, is not a deficiency; but rather related to decline in the number of data items within each data source.

TABLE I
COMPARISON OF COMA3.0 AND TREE SIMPLIFICATION ALGORITHMS

	Number of data items (avg of F-measures)				Total working time (min)
Methods	2	3	4	5	
COMA 3.0	0,96	0,93	0,45	0,28	107,10
HTML Tree Simplification	0,97	0,90	0,47	0,32	30,47

We compared our results with COMA 3.0 in terms of quality and performance. When compared for performance; total working time of the HTML tree simplification process consists of HTML simplification, frequency weighted item tree generation and schema matching steps. For COMA 3.0, the SRRs are converted to XML schemas and evaluated using COMA 3.0 algorithm. The process time of Coma 3.0 algorithm does not include XML conversion time. While the quality of the two methods is similar, the performance of the HTML tree simplification method is better than the COMA 3.0 method.

V. Conclusion

A new method based on tree simplification and schema matching for automatic Web result extraction and matching has been proposed to find and extract the SRRs and then match them within the same result list. First, SRRs are detected by processing HTML DOM Trees and then a transformation has been applied to simplify DOM trees in a rectangular form. After that, the schema matching algorithm is performed in order to match the results from different data sources. Finally, experimental results show that the proposed method is successful for extracting and merging the SRRs.

References

[1] M.K. Bergman, “The Deep Web: Surfacing Hidden Value”, BrightPlanet.com [Online].

[2] E. Ferrara, P. De Meo, G. Fiumara, R. Baumgartner, “Web Data Extraction, Applications and Techniques: A Survey”, arXiv preprint arXiv:1207.0246, 2012.

[3] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, A. Halevy, Google's Deep Web Crawl, Proceedings of the VLDB Endowment, 1(2), pp.1241-1252, 2008.

[4] A.H. Laender, B.A. Ribeiro-Neto, A.S. da Silva, J.S. Teixeira, “A Brief Survey of Web Data Extraction Tools”, ACM Sigmod Record, 31(2), pp.84-93, 2002.

[5] C.H. Chang, M. Kayed, M.R. Girgis, K.F. Shaalan, “A Survey of Web Information Extraction Systems”, IEEE Transactions on Knowledge and Data Engineering, 18(10), pp.1411-1428, 2006.

[6] E. Ferrara, P. De Meo, G. Fiumara, R. Baumgartner, R., “Web Data Extraction, Applications and Techniques: A Survey”, arXiv preprint arXiv: 1207.0246, 2012.

[7] R.B. Trieschnigg, K.T.T.E Tjin-Kam-Jet, D. Hiemstra, “Ranking XPaths for Extracting Search Result Records”, Technical Report TR-CTIT-12-08, Centre for Telematics and Information Technology, University of Twente, Enschede. ISSN 1381-3625, 2012.

[8] Y. Gozudeli, O. Yildiz, H. Karacan, M.R. Baker, A. Minnet, M. Kalender, O. Ozay, M.A. Akcayol, "Extraction of Automatic Search Result Records Using Content Density Algorithm Based on Node Similarity", The International Conference on Data Mining, Internet Computing, and Big Data (BigData2014), Asia Pacific University of Technology and Innovation (APU), Kuala Lumpur, Malaysia, 2014.

[9] V. Crescenzi, G. Mecca, and P. Merialdo, “RoadRunner: Towards Automatic Data Extraction from Large Web Sites”, VLDB Conference, pp.109-118, 2001.

[10] X. Yin, W.Tan, X. Li, X., Y.C. Tu, “Automatic Extraction of Clickable Structured Web Contents for Name Entity Queries”, In Proceedings of the 19th International Conference on World Wide Web, pp.991-1000, ACM, 2010.

[11] W. Liu, X. Meng, W. Meng, “Vide: A Vision-Based Approach for Deep Web Data Extraction”, IEEE Transactions on Knowledge and Data Engineering, 22 (3), pp.447-460, 2010.

[12] A. Banu, M. Chitra, “DWDE-IR: An Efficient Deep Web Data Extraction for Information Retrieval on Web Mining”, Journal of Emerging Technologies in Web Intelligence, 6(1), pp.133-141, 2014.

[13] G. Miao, J. Tatemura, W.P. Hsiung, A. Sawires, L.E. Moser, “Extracting Data Records From the Web Using Tag Path Clustering”, In Proceedings of the 18th International Conference on World Wide Web”, pp.981-990, 2009.

[14] P. Bernstein and E. Rahm, "A survey of approaches to automatic schema matching" The VLDB Journal, no. 10, p. 334–350, 2001.

[15] P Shvaiko, J Euzenat, "A survey of schema-based matching approaches." Journal on Data Semantics IV. Springer Berlin Heidelberg, 2005. 146-171.

[16] N. Kushmerick, Wrapper Induction: Efficiency and Expressiveness, Artificial Intelligence, 118(1), pp.15-68, 2000.

[17] I. Muslea, S. Minton, C.A. Knoblock, “Hierarchical Wrapper Induction for Semistructured Information Sources” Autonomous Agents and Multi-Agent Systems, 4(1-2), pp.93-114, 2001.

[18] A. Sahuguet, F. Azavant, “Building Intelligent Web Applications Using Lightweight Wrappers”, Data & Knowledge Engineering, 36(3), pp.283-316, 2001.

[19] L. Liu, C. Pu, W. Han, “XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources”, In Data Engineering, Proceedings 16th International Conference, pp. 611-621, 2000.

[20] H. Zhao, W. Meng, Z. Wu, V. Raghavan, C. Yu, ”Fully Automatic Wrapper Generation for Search Engines”, Proceedings of the 14th International Conference on World Wide Web, pp.66-75, 2005.

[21] C.H. Chang, S. C. Lui, “IEPAD: Information Extraction Based on Pattern Discovery”, In Proceedings of the 10th International Conference on World Wide Web, pp. 681-688, AC, 2001.

[22] B. Liu, R. Grossman, Y. Zhai, “Mining Data Records in Web Pages”, In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.601-606, ACM, 2003.

[23] A. Arasu, H. Garcia-Molina, “Extracting Structured Data From Web Pages”, In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp.337-348, ACM, 2003.

[24] Y. Zhai, B. Liu, “Web Data Extraction Based On Partial Tree Alignment”, In Proceedings of the 14th International Conference on World Wide Web, pp.76-85, ACM, 2005.

[25] F. Duchateau, Z. Bellahsene and M. Roche, "A context-based measure for discovering approximate semantic matching between schema elements" In RCIS , 2007.

[26] H. Nguyen, H. Nguyen ve J. Freire, “PruSM: A Prudent Schema Matching Approach for Web Forms”, CIKM’10, Toronto, 2010.

[27] P. Bernstein and E. Rahm, "A survey of approaches to automatic schema matching" The VLDB Journal, no. 10, p. 334–350, 2001.

[28] M. Pawlik, N. Augsten, “A Memory-Efficient Tree Edit Distance Algorithm”, In Database and Expert Systems Applications, pp.196-210, Springer International Publishing, 2014.

[29] L. Getoor and A. Machanavajjhala, "Entity Resolution: Theory, Practice & Open Challenges", VLDB2012, İstanbul, 2012.

[30] H. H. Do, S. Melnik and E. Rahm, "Comparison of schema matching evaluations", In Web, Web-services, and database systems workshop, 2002.

[31] S.Massmann, S. Raunich, D.Aumüller,P. Arnold, , E. Rahm,” Evolution of the coma match system”, Ontology Matching, 2, 2011

[32] J. Fan, P. Luo, S.H. Lim, S. Liu, J. Parag, J. Liu, "Article clipper: a system for Web article extraction", Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.

[33] J. Madhavan, S.R. Jeffery, S. Cohen, X.L. Dong, D. Ko, C. Yu, A. Halevy, “Web-Scale Data Integration: You Can Only Afford to Pay As You Go”, Proc. Conf. Innovative Data Systems Research (CIDR), pp. 342-350, 2007.