An Automatic Annotation Technique for Web Search Results

Rosamma K S
Dept of Computer Science
College of Engineering Poonjar
Kottayam, India

Jiby J Puthiyidam
Dept of Computer Science
College of Engineering Poonjar
Kottayam, India

ABSTRACT
The uses of web search engines are very frequent and common worldwide over the internet by end users for different purposes. A web search engine takes the query request from the end user and executes that query on relational database used to store the information on behalf of that web search engine. Based on input queries the dynamic response is generated by search engine, in the form of HTML based pages. Such pages are supported with the web databases. Every web page generated contains many results to display for particular query, called as Search Result Records (SRRs). Sometimes it becomes troublesome to extract relevant data from diverse sources. The SRRs generated may contain data units that are relevant to one common semantic. These SRRs are further required to be assigned with proper labels. The manual methods for record extraction and labeling have a worse scalability. Thus automatic annotation based method is needed to improve the accuracy as well as scalability of web search engines. This paper presents an automatic annotation technique for web search results. The proposed approach first aligns the data units on a result page into different groups such that the data in the same group have the same semantic. Then, each group is annotated from different aspects and aggregates the different annotations to predict a final annotation label for it. The annotation wrapper generated for the search site is automatically constructed and can be used to annotate new result pages from the same web database. Experiments indicate that the proposed approach is highly effective.

General Terms
Search Result Annotation

Keywords
Web Database, Annotation, Data alignment, Annotation Wrapper

1. INTRODUCTION
Web information extraction and annotation are two important research areas in recent years. Large portion of the deep web is database based. The data encoded in the returned result pages come from the underlying structured databases, called Web databases (WDB)[9,10]. A typical search result page may contain multiple search result records (SRRs). Each SRR contains multiple data units each of which describes one aspect of a real-world entity. It corresponds to the value of a record under an attribute. Collecting data of interest from multiple WDBs[9,10], always have a high demand. Having semantic labels for data units is important for the record linkage task and for storing collected SRRs into a database table for later analysis.

Annotation of web pages is an area of research which is getting lot of attention as the count of websites of specific topics and as a whole is increasing very fast. All the databases are accessible over the web through HTML representations and data extraction over web is becoming more and more dynamic. Such type of data is huge and for applications such as article collection, online shopping comparison, etc.

Most of the online shopping sites contain the databases in an unorganized manner, so it takes more time to get the search results. To overcome this drawback automatic annotation approach is proposed. Annotation of such collected information leads to several advantages including fast decision making, elimination of older searches, relevant information visiting, historical data management and to reduce the time of futile searches.

The proposed annotation technique is an enhancement over the existing multi annotator approach. The newly proposed wrapper generation algorithm seems to both efficient and promising.

2. RELATED WORKS
Web information extraction and annotation area received a fast growth in the recent years. Many of the systems developed are based on, marking the specified areas in the sample page and then use a set of rules to extract the information. There are many systems that have higher extraction accuracy through the use of supervised learning techniques.

There exists many systems that use vision based approaches [1,2,3] for web data extraction. One of them is ViDE[1], which is based on some common visual features of the deep web. It first builds a visual block tree using the VIPS algorithm. Using the Visual Block tree, data record extraction and data item extraction are carried out based on the proposed visual features. Then a visual wrapper generated is to improve the efficiency of both data record extraction and data item extraction. There is an another system that uses enhanced co-citation algorithm[3]. Apart from other systems that develop a new set of APIs for the extraction of visual information, this algorithm retrieves the visual information of the deep web pages directly from the web database. So it consumes less time for web page extraction. The algorithm needs to improved, for achieving speed and to avoid noise in the extracted data.

An algorithm named EXALG[4] is proposed for automatically extracting the database values from web pages generated using a common template. It discovers the sets of tokens associated with the same type constructor in the (unknown) template used to create the input pages. And then uses the above sets to deduce the template and it is then used to extract the values encoded in the pages. EXALG is extremely good in extracting the data from the web pages, generated from a common template. But there exists a chance of information loss when naive key word indexing and searching is used.

Web data extraction based on partial tree alignment[5] studies the problem of extracting data record, then segments these records, and finally put them into a database table. Partial alignment takes only those data fields in a pair of data records that can be aligned (or matched) with certainty, and it doesn't depend on the rest of the data fields. This method enables very accurate alignment of multiple data records. The partial tree alignment method is able to align data items in nested records; it fails to handle data records that are relatively rare in record lists.

There exists another method that uses the structural semantic entropy[6] measure to locates and extracts the data of interest from web pages across different sites. The algorithm is efficient in data extraction based on the requirements that the web pages share the similar template as well as dissimilar template. It is independent of modifications in web-page format which enable to detect false positive rate in associating the attributes of records with their respective values. In order to make this approach better, efficient method would be proposed to handle computational overhead, memory consumption, fast processing etc. There exists another method, which performs automatic semantic annotation of data units in semantic manner and extract features from SRRs features such as text and data unit feature using Particle Swarm Optimization (PSO)[7] methods. In this method similarity of data units and text units nodes In the pages considered as the attributes of elements for better grouping of similar concepts which is additional second-hand to conclude the similarities among the data units based on their characteristic standards. The Method [11] also proposes a multi annotator approach which has many advantages over the above specified systems. The proposed method takes some of its features and trying a new wrapper generation algorithm

A large number of techniques have been proposed in this area, but most of them have some inherent limitations. In this paper an automatic annotation technique for web search results is proposed. It is carried out in three phases. First phase is the alignment phase. It identifies all data units in the SRRs and then organizes them into different groups with each group corresponding to a different concept. In Phase 2, called annotation phase, we introduce multiple basic annotators with each exploiting one type of features. All basic annotator is used to produce a label for the units within their group holistically, and then a probability model is adopted to determine the most appropriate label for each group. In third phase, the wrapper generation phase, generate an annotation rule that describes how to extract the data units of this concept in the result page and what the appropriate semantic label should be. Here we are introducing a new wrapper generation algorithm.

3. PROPOSED METHOD

In this approach first user gives the query to the search engine. In search result page containing multiple SRR, the data unit corresponding to the same concept often share special common features. After the feature selection data alignment is done. The aim of data alignment is to put the data units of the same concept into one group so that they can be annotated holistically.

The SRR extracted has a tag structure that determines how the contents of the SRRs are displayed on a web browser. Based on how many data units a text node may contain, we select the following four types of relationships between data unit (U) and text node (T):

- *One-to-One Relationship:* Each text node contains exactly one data unit ;the text of this node contains the value of a single attribute.
- *One-to-Many Relationship:* Many data units are encoded in one text node.
- *Many-to-One Relationship:* More than one text nodes together form a data unit
- *One-To-Nothing Relationship:* The text nodes belonging to this category are not part of any data unit inside SRRs

Data alignment is done to put the data units of the same concept into one group so that they can be annotated holistically. The similarity of two data units belong to the same concept is determined by how similar they are based on certain features. The features considered are:

- *Data Content (DC)*

Data units or text nodes with the same concept often share certain keywords. Because, the data units corresponding to the search field where the user enters a search condition usually contain the search keywords. And text nodes that contain data units of the same concept usually have the same leading label.

- *Presentation Style (PS)*

It describes how a data unit is displayed on a webpage. Commonly it consists of six style features: font size, font face, font color, font weight, text decoration and whether it is italic. Same concept data units in different SRRs are usually displayed in the same style

- *Data Type (DT)*

Every data unit has its own semantic type although it is just a text string in the HTML code. The basic data types considered in our approach are: Integer, Time, Currency, Date, Percentage, Decimal String, and Symbol

- *Tag Path (TP)*

The tag path of a text node is defined as a sequence of tags traversing from the root of the SRR to the corresponding node in the tag tree.

- *Adjacency (AD)*

Let d_p and d_s denote the data units immediately before and after d in the SRR, respectively. Let d_p and d_s be the preceding and succeeding data units of d. Consider two data units d_1 and d_2 from two separate SRRs. It can be observed that if d_{p1} and d_{p2} belong to the same concept and/or d_{s1} and d_{s2} belong to the same concept, which then leads to a conclusion that it is more likely that d_1 and d_2 also belong to the same concept.

In our approach the similarity between two data units (or two text nodes) d1 and d2 is defined as the weighted sum of the similarities of the five features between them

$$Sim(d_1, d_2) = w_1 * SimC(d_1, d_2) + w_2 * SimP(d_1, d_2) + w_3 * SimD(d_1, d_2) + w_4 * SimT(d_1, d_2) + w_5 * SimA(d_1, d_2)$$

The measure of similarity for each individual feature is defined as follows:

- Data content similarity (SimC).

It is defined as the Cosine similarity between the term frequency vectors of d_1 and d_2

$$SimC(d_1, d_2) = \frac{V_{d_1} \cdot V_{d_2}}{\|V_{d_1}\| * \|V_{d_2}\|}$$

- Presentation style similarity (SimP)

Presentation style similarity is the average of the style feature scores (FS) over all six presentation style features (F) between d_1 and d_2:

$$SimP(d_1, d_2) = \sum_{i=1}^{6} FS_i / 6$$

- Data type similarity (SimD).

It is determined by the common sequence of the component data types between two data units. The longest common sequence (LCS) cannot be longer than the number of component data types in these two data units. Let t_1 and t_2 be the sequences of the data types ofd_1 and d_2, and TLen(t) represent the number of component types of data type t, then similarity between data units d_1 and d_2 can be defined as

$$SimD(d_1, d_2) = \frac{LCS(t_1, t_2)}{Max(T_{len}(t_1), T_{len}(t_2))}$$

- Tag path similarity (SimT).

This is the edit distance (EDT) between the tag paths of two data units. Edit distance here refers to the number of insertions and deletions of tags needed to transform one tag path into the other. The maximum number of possible operations needed is the total number of tags in the two tag paths. Consider p_1 and p_2 as the tag paths of d_1 and d_2, and PLen(p) denote the number of tags in tag path p, then the tag path similarity between d_1 and d_2 is defined as

$$SimT(d_1, d_2) = 1 - \frac{EDT(p_1, p_2)}{P_{len}(p_1) + P_{len}(p_2)}$$

- Adjacency similarity (SimA).

The adjacency similarity between two data units d_1 and d_2 is the average of the similarity between d^p_1 and d^p_2 and the similarity between d^s_1 and d^s_2, where d^p denote the preceding data unit and d^s denote the succeeding data unit $SimA(d_1, d_2) = Sim'(d_1^p, d_2^p) + Sim'(d_1^s, d_2^s)$

3.1 Alignment Algorithm

The data alignment algorithm is based on the assumption that attributes appear in the same order across all SRRs on the same result page, even the SRRs may contain different sets of attributes. The aim of alignment is to move the data units in the table so that every alignment group is well aligned, while the order of the data units within every SRR is preserved. The alignment method can be concluded in the following four steps

Step 1: Merge text nodes. Here decorative tags are removed from each SRR to allow the text nodes corresponding to the same attribute to be merged into a single text node.

Step 2: Align text nodes. This step aligns text nodes into groups so that eventually each group contains the text nodes with the same concept (for atomic nodes) or the same set of concepts (for composite nodes).

Step 3: Split (composite) text nodes. This step aims to split the "values" in composite text nodes into individual data units.

Step 4: Align data units. This step is to separate each composite group into multiple aligned groups with each containing the data units of the same concept.

The alignment algorithm is given as follows:

```
ALIGN(SRRs)
  J←1
while true
   //create alignment groups
 For i←1 to number of SRRs
  G_j←SRR[i][j]
 If G_j is empty
  Exit//break the loop
 V←CLUSTERING(G)
 If |V|>1
  //collect all data units in groups following j.
  S←Ø
 For x←1 to number of SRRs
     For y←j+1 to SRR[i].length
        S←SRR[x][y]
 //find cluster c least similar to following groups
 V[c]=min_{k=1to|v|}(Sim(V[k],S))
//Shifting
For k←1 to |v| and k≠c
    For each SRR[x][j] in V[k]
    Insert NIL at position j in SRR[x]
j←j+1 //move to next group
CLUSTERING(G)
V←all data units in G
While |V|>1
   best←0
   L←NIL;R←NIL
   For each A in V
     For each B in V
      If(A!=B) and (Sim(A,B)>best ))
        best←Sim(A,B)
        L←A
        R←B
If best>T
 Remove L from V
 Remove R from V
 Add L U R to V
Else break loop
Return V
```

We use the agglomerative clustering algorithm[8] to cluster the text nodes inside the group (CLUSTERING). First, each text node forms a separate cluster of its own. We then repeatedly merge two clusters that have the highest similarity value until no two clusters have similarity above a threshold T. After clustering, we obtain a set of clusters V and each cluster contains the elements of the same concept only.

3.2 Annotation

There are six basic annotators [1] to label data units, each of them considering a special type of patterns. Once the data units on a result page have been annotated, and use these data units to construct an annotation wrapper for the WDBs. Then the new SRRs retrieved from the same WDB can be annotated using this wrapper quickly without reapplying the entire process.

Alignment algorithm is to move the data units in the table so that every alignment group is well aligned; the order of the data units within every SRR is preserved and also needs the

similarity between two data unit groups. In this first create the alignment groups then the clustering is done. When clustering is done choose the annotation method. There are six annotation methods are there.

- *Table annotator:* Table annotators first identify the column header. Then for each SRR takes a data unit, select the column header with maximum overlap. At last a unit is assigned and labeled.
- *Query based Annotator:* Given a query with a set of query terms submitted against an attribute A on the local search interface, first find the group that has the largest total occurrences of these query terms and then assign a the label to the group.
- *Schema Annotator:* The schema value annotator first identifies the attribute that has the highest matching score among all attributes and then uses annotate the group.
- *Frequency-Based Annotator:* The frequency-based annotator intends to find common preceding units shared by all the data units of the group Gi
- *In-Text Prefix/Suffix Annotator:* This annotator checks whether all data units in the aligned group share the same suffix or prefix. If the same prefix is confirmed and it is not a delimiter. It can be removed from all the data units in the group and can used as the label to annotate values following it. If the same suffix is identified. And if the number of data units having the same suffix match the number of data units inside the next group, then it can be used to annotate the data units inside the next group
- *Common Knowledge Annotator:* Common knowledge annotator uses some predefined common concepts. Given a group of data units from the alignment step, if all the data units match the pattern or value of a concept, the label of this concept is assigned to the data units of this group.

No single annotator is capable of fully labeling all the data units on different result pages. These annotators are fairly independent from each other since each exploits an independent feature.

3.3 Wrapper Generation

We use the tree alignment methods to calculate the similarity between input web pages and build a wrapper on tree alignment results. The tree alignment method is also used to calculate the similarity between wrapper and the input web results. The input trees are merged into one union tree whose nodes record the statistical information such as the times a node has been aligned, the text length of the node. A heuristic method is employed to find the most probable content block. The alignment algorithm is utilized again to detect the repeating patterns on the union tree. The wrapper is generated based on the most probable content block and the repeating patterns.

A similarity series was built by calculating the similarity between the input web pages and the current wrapper using the tree alignment algorithm. The similarity series is in the order of the input web pages' timestamp.

Change point detection and wrapper regeneration. A log likelihood ratio test is utilized to detect the change points on the similarity series. The wrapper generation method is applied again to generate a wrapper once a change point is detected. In this work, we focus on setting the weight (cost) of different node mapping (tag-matching). One of the major contributions of our work is a kind of linear regression method for getting the weight of different tag-matching.

The main problem of the previous method is that they did not consider about employing different weights for various tag-matching.

For example, the block elements are elements that usually, but not always, contain other elements. They normally act as containers of some sort. The inline elements normally mark up the semantic meaning of something. Furthermore, the level of the different nodes should also be considered. The higher-level nodes should have higher weight as the higher-level nodes usually act as bigger structure block. Different weight should be assigned to different type of tag-matching.

In this study, a kind of linear regression method is employed to get the weight of different tag-matching. First, we found a collection of similar web pages belong to the same "class". It's feasible to get this kind of web pages collection automatically. Next, we will use this web pages collection for getting the optimal weighting schema.

Let w_i be the weight of tag-matching and $w_i > w_j$ for $i < j$.
Let D_{mn} be the sum of the gains in the best alignment between the trees T_m and T_n.

$$D_{mn} = \sum_i w_i \, t_i^{mn}$$

(1) Where t_i^{mn} is the number of w_i occur in the alignment procedure.
(2) The sum of the gains in the collection is:

$$f = \sum_{m,n} D_{mn} = \sum_{m,n} \sum_i w_i \, t_i^{mn} = \sum_i w_i \sum_{m,n} t_i^{mn}$$

Because the collection is the similar web pages belonging to the same “class”, a set of w_i is selected which makes the maximum f.
To get $argmax_w \sum_i w_i \sum_{m,n} t_i^{mn}$,a constraint $\sum_i w_i^2 = 1$ is added
The group of equations is rewritten as:

$$f = \sum_i w_i \, C_i + \lambda\left(\sum_i w_i^2 - 1\right)$$

Where, $C_i = \sum_{m,n} t_i^{mn}$ and $\sum_i w_i^2 = 1$
The solution of the above equations is used as the weight of each type of tag-matching (w_i).

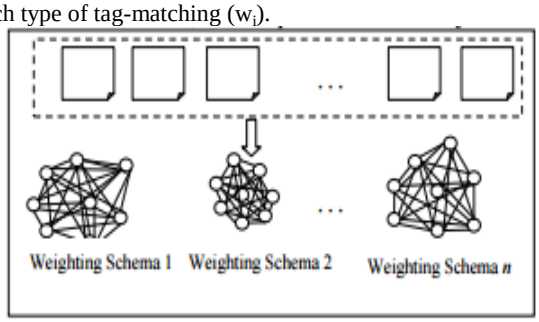

Fig 1: Example-Schema Matching

Figure 1illustrates an example of the weight setting method. For one collection of similar web pages belong to the same “class”, we calculate the sum of the alignment gains (or the similarity) for each weighting schema. The best weighting schema is the one maximize the sum of the gains. That means to find a set of w_i that output the maximum f in the equation.

4. EXPERIMENTAL RESULTS

We have made experiments data from various domains with respect to six annotators only. The data set includes a set of values from an electronic shopping site. The values are taken from mobile, laptop and camera domains. The annotators used include table annotator, schema value annotator, prefix suffix annotator, frequency annotator common knowledge annotator and query based annotator. Both the annotators are supported by the prototype application and it is extensible so as to support more annotators in future. The performance of data alignment and annotation are presented in Table 1.

Table1.Performance Evaluation

Domain	Data Alignment performance		Annotation Performance	
	Precision	Recall	Precision	Recall
Mobile	97.2%	97.4%	93.5%	90.2%
Laptop	97.1%	97.2%	92.6%	91.3%
Camera	96.8%	97.0%	92.6%	91.3%

For evaluating the performance of the method, we adopt the precision and recall measures from information retrieval. For alignment ,the precision is defined as the percentage of the correctly aligned data units over all the aligned units by the system; recall is the percentage of the data units that are correctly aligned by the system over all manually aligned data units by the expert. As presented in the above table, it is evident that more than 90 percentage precision and recall were recorded for both the performances such as data alignment and annotations. Performance of data annotation with wrappers for three domains is capable of producing annotations automatically given search results of Google .The performance of the application is encouraging and the application can be used in the real world applications.

The results are presented in the following graphs. It shows the precision and recall measure for both data alignment and data annotation in all the three domains.

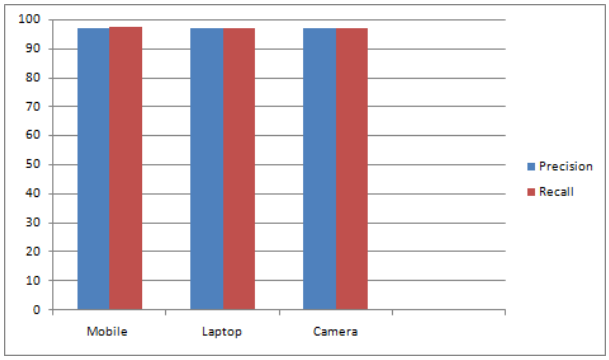

Fig 2: Alignment Performance

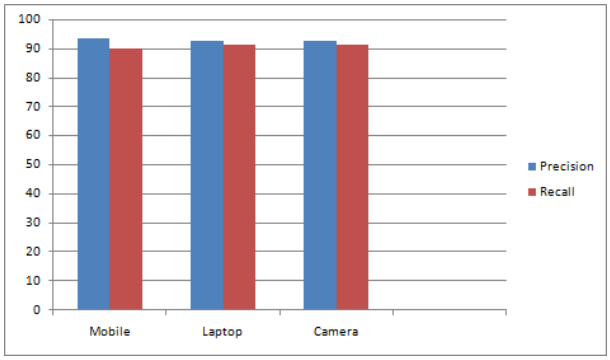

Fig3: Annotation Performance

Our experimental results show that each of the annotators is useful and they together are capable of generating high quality annotation.

5. CONCLUSION

The automatic annotation approach considers several types of data unit and text node features and makes annotation scalable and automatic. A large number of techniques have been proposed in this area, but most of them have some inherent limitations. Here we proposed a multi annotator approach to automatically constructing an annotation wrapper for annotating the search result records retrieved from any given web database. This approach consists of six basic annotators and a probabilistic method to combine the basic annotators. Each of these annotators exploits one type of features for annotation and our experimental results show that each of the annotators is useful and they together are capable of generating high quality annotation. The alignment method is a clustering based shifting method utilizing richer yet automatically obtainable features. We use the tree alignment methods to calculate the similarity between input web pages and build a wrapper on tree alignment results. The newly proposed wrapper generation algorithm is both efficient and promising. The precision and recall are of both data alignment and annotations are recorded above 90 percentage. The performance of the application is encouraging and the application can be used in the real world applications.

There is a space for improvement. The model is highly flexible in the sense that when an existing annotator is modified or a new annotator is added It is possible to try using different machine learning techniques and using more sample pages from each training site to obtain the feature weights so that we can identify the best technique to the data alignment problem.

6. ACKNOWLEDGMENTS

I sincerely express my gratitude to everyone who supported me to prepare this paper. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the work.

7. REFERENCES

[1] Wei Liu, Xiaofeng Meng, Member, IEEE, and Weiyi Meng, Member, IEEE, ViDE: A Vision-Based Approach for Deep Web Data Extraction

[2] D. Raghu, V. Sridhar Reddy, Ch. Raja Jacob, Dynamic Vision-Based Approach in Web Data Extraction

[3] R.Vijay1, Dr. K. Prasadh, A Vision Based Approach for Web Data Extraction using Enhanced Co-citation Algorithm

[4] Arvind Arasu, Hector Garcia-Molina, Extracting Structured Data from Web Pages, Stanford University

[5] Yanhong Zhai, Bing Liu, Department of Computer Science University of Illinois at Chicago, Web Data Extraction Based on Partial Tree Alignment

[6] P.V.Praveen Sundar1 1Research Scholar, Hindusthan College of Arts &Science, Coimbatore, India, Towards Automatic Data Extraction Using Tag and Value Similarity Based on Structural -Semantic Entropy

[7] T.Seeniselvi1, N.Thangamani.2 1Associate Professor, Hindusthan college of Arts and Science, India, Semantic Similarity Based Data Alignment and Best Feature Extraction using PSO for Annotating Search Results from Web Databases

[8] L. Kaufman and P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990.

[9] Hongjun Lu, Integrating Database and World Wide Web Technologies, National University of Singapore

[10] Jungwha Hong, How to Build a Web Database: A Case Study, The University of Texas at Austin.

[11] Yiyao Lu, Hai He, Hongkun Zhao, Weiyi Meng, Member, IEEE, and Clement Yu, Senior Member, IEEE, Annotating Search Results from Web Databases