ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Web Database Search Result Annotation

M.Manivasakam[1], N.Yuvraj[2]

PG student, Park College of Engineering and Technology, Coimbatore, India[1]

Assistant Professor, Park College of Engineering and Technology, Coimbatore, India [2]

Abstract: An increasing number of databases have become web accessible through HTML form-based search interfaces. The data units returned from the underlying database are usually encoded into the result pages dynamically for human browsing. For the encoded data units to be machine process able, which is essential for many applications such as deep web data collection and Internet comparison shopping, they need to be extracted out and assigned meaningful labels. Search result presents an automatic annotation approach that 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, for each group annotate it from different aspects and aggregate the different annotations to predict a final annotation label for it. An annotation wrapper for the search site is automatically constructed and can be used to annotate new result pages from the same web database. As per the system they proposed will makes the data extraction in a best manner is the main theme of paper.

I. INTRODUCTION

A large portion of the deep web is database-based, i.e., the data encoded in the result pages returned by search engines come from the underlying structured databases (e.g., relational databases). Such type of search engines is usually referred to as Web databases. A typical result page returned from a Web database consists of multiple search result records (SRRs). Usually, one SRR contains multiple data units each of which describes one aspect of the corresponding entity. For example, Figure 1 depicts a portion of a sample result page returned from a book search engine with three records on it. Each record represents one book and consists of several data units (title, author, etc.).

Search result has the following contributions: (1) We propose a hierarchical clustering based approach to align data units into different groups. Instead of using only the DOM tree or other HTML tag tree structures of the result records to align the data units like most current methods do, our approach also considers other important features shared among data units, such as their data types, text contents, presentation formats, and adjacency information. (2) We utilize the integrated schema of the search interfaces of multiple Web databases in the same domain to enhance the label assignment process. The integrated schema can be automatically obtained and our experiments show the added benefits of using it. (3) We propose a multi-annotator approach to tackle the annotation problem with each basic annotator exploiting a different type of features. This approach is highly flexible as existing basic annotators may be modified and new annotators can be added easily without affecting the operation of others. (4) Our approach constructs an annotation wrapper for any given Web database. The wrapper can be applied to efficiently annotating the SRRs retrieved from the same Web database with new queries.

II. RELATED WORK

Wrapper induction (e.g., [17, 18]) is a semi-automatic technique to extract the desired information from Web pages. It needs users to label desired data to extract, and then the wrapper induction system induces the rules to construct the wrapper for extracting the corresponding data. [1, 5] are two efforts to automatically construct (extraction) wrappers, but they do not annotate the extracted data. Embley et al. [7] utilize ontologies together with several heuristics to automatically

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014
Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)
Organized by
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

extract data in multi-record documents and label them. However, ontologies for different domains must be constructed manually. [19] exploits the presentation styles and the spatial locality of semantically related items, but its learning process for annotation is domain-dependent. Moreover, a seed of instances of semantic concepts in a set of HTML documents has to be hand-labelled.

Data alignment is an effective step in achieving good annotation accuracy and it is also used in [19, 22]. However, existing automatic data alignment techniques (including the one described in [24]) are based on HTML tag tree structures only. The assumption is that the sub-trees corresponding to two data units in different SRRs but with the same concept usually have the same tag structure. However, this assumption is not always correct because the tag tree is very sensitive to even minor differences, which may be caused by the need to emphasize certain data unit, artificial effect of the page, or erroneous coding. Our data alignment approach differs from the previous works in two main aspects: (1) we utilize more features (see Section 3.1); (2) we employ a clustering approach to perform alignment, which has not been used by others for data alignment as far as we are aware.

To enable fully automatic annotation, the SRRs need to be automatically extracted from the result pages. We employ the ViNTs system [23] to perform this task. Each SRR is stored in a tree structure with a single root and each node in the tree corresponds to an HTML tag or a piece of text in the original page. With this structure, it becomes easy to locate each node in the original HTML page so that no information is lost. The physical position information of each node on the rendered page, including its coordinates and area size, can also be obtained through this system.

III. DATA ALIGNMENT

3.1. Features of data units

Data units belonging to the same concept from different SRRs usually share many common features. Five features are utilized in our approach.

1. Data Content (DC). The data units with the same concept often share certain keywords. This is true for several reasons. First, the data units corresponding to the search field where the user enters a search condition usually contain the search keywords. For example, in Figure 1, the sample result page is returned for the search on the title field with keyword "machine". We can see that all the titles have this keyword. Second, the web designers like to put some leading labels in front of certain data units to make it easier for users to understand the data. Such data units of the same concept usually have the same leading label. For example, in Figure 1, the price of every book has the leading words "Our Price".

2. Presentation Style (PS). This feature describes how a data unit is displayed on a web page. It consists of 6 style features: font face, font size, font color, font weight, text decoration (underline, strike, etc.), and whether it is italic. Data units of the same concept in different SRRs are usually displayed in the same style. For example, in Figure 1, all the availability information is displayed in the same font and in italic.

3. Data Type (DT). Each data unit has its own semantic type although it is just a text string in the HTML code. Seven basic data types are considered in our approach: Date, Time, Currency, Integer, Decimal, Percentage, and Ordinary String. Each type except

Ordinary String has certain pattern(s) so that it can be easily identified. Text not one of the first 6 types is treated as an Ordinary String. Usually the data units of the same concept have the same data type.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

4. Tag Path (TP). A tag path of a data unit is a sequence of tags traversing from the root of the record to the corresponding node in the tag tree. An observation is made that the tag paths of the data units with the same concept have very similar tag paths, though in many cases, not exactly the same.

5. Adjacency (AD). Consider two data units d_1 and d_2 from two different SRRs r_1 and r_2, respectively. Let p_i and s_i be the data units that precede and succeed d_i in r_i, respectively, i = 1, 2. It can be observed that if p_1 and p_2 belong to the same concept and/or s_1 and s_2 belong to the same concept, then it is more likely that d_1 and d_2 also belong to the same concept.

3.2. Text nodes and data units

Each SRR extracted by ViNTs has a tag structure that determines how its contents are displayed on the Web browser. Each node in such a tag structure is either a tag node or a text node. It is obvious that all data units are located in the text nodes. However, there does not exist a 1:1 correspondence between text nodes and data units. There are two cases. First, a text node contains exactly one data unit. In other words, the text of this node contains the value of a single attribute. We refer such kind of text nodes as "atomic text nodes". An atomic text node is the same as a data unit. The second case is that multiple data units are encoded in one text node. For example, in Figure 1, part of the second line of each SRR (e.g., "Springer-Verlag / 1999 / 0387984135" in the first record) is a single text node. It is clear that it consists of three semantic data units: the publisher, the publication date, and ISBN. Since the text of such kind of nodes can be considered as a composition of the texts of multiple data units, we call them "composite text nodes". Before aligning data units into different semantic groups, we need to deal with the problem of identifying composite text nodes and extracting data units from such nodes.

By analyzing a large number of result pages in real applications, the following observations can be made: if the data units of attributes A_1 … A_k in one SRR are encoded as a composite text node, then (1) it is highly likely that the data units of the same attributes in other SRRs are also encoded as composite text nodes; and

(2) they are usually located at the same place in the SRRs. These observations are valid because generally the SRRs are generated by template programs.

3.3. Data alignment algorithm

It is not difficult to see that all the features described in Section 3.1 are applicable to text nodes, including composite text nodes. Our data alignment algorithm consists of three steps:

1. Align text nodes. This step places text nodes with the same concept (for atomic nodes) or the same set of concepts (for composite nodes) into the same group.

2. Split (composite) text nodes. This step aims to split the "values" in composite text nodes into individual data units. This step is carried out based on the text nodes in the same group holistically. A group whose "values" are split is called a composite group.

3. Align data units. For each composite group, this step places the data units corresponding to the same concept into the same group.

The same algorithm is used in the first and the third steps above, with the only difference being that for the former text nodes are considered while for the latter data units are considered. To ease discussion, we refer both text nodes and data units as data units from now on when there is no confusion. A clustering algorithm is utilized to place similar data units

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

into the same group and dissimilar data units into different groups.

The similarity between two data units is defined as follows. If they are from the same SRR, their similarity is 0. Otherwise the similarity is defined as the aggregated sum of the similarities of the 5 features between the two data units. More specifically, the similarity between data units d1 and d2 is:

$$\begin{aligned} Sim(d1, d2) &= w1*SimC(d1, d2) \\ &+ w2*SimP(d1, d2) + w3*SimD(d1, d2) \\ &+ w4*SimT(d1, d2) + w5*SimA(d1, d2). \end{aligned}$$

The feature similarities are defined as follows:

Data content similarity (SimC): It is the Cosine similarity [26] between texts of the two data units.
Presentation style similarity (SimP): It is the ratio of the number of style features the two data units match over the six style features used in our approach.

Data type similarity (SimD): If two data units have the same data type, the similarity is 1; otherwise, 0.
Tag path similarity (SimT): This is the edit distance between the tag paths of d1 and d2. The edit distance (EDT) refers to the number of insertions and deletions needed to transform one tag path into the other. Obviously, the maximum number of operations needed is the total number of tags in the two tag paths. Let t1 and t2 be the tag paths of d1 and d2 respectively,

$SimT(d1,d2) = 1 – EDT(t1, t2) / (Len(t1)+Len(t2))$, where Len(t) denotes the number of tags in tag path t.
Adjacency similarity (SimA): This is the average similarity between the preceding data units and between the succeeding units of d1 and d2. Only the first 4 features are used in this computation.

We apply the agglomerative clustering algorithm [16] to cluster the data units. Initially, each data unit forms a separate group of its own. We then repeatedly merge two groups that have the highest similarity value until no two groups have similarity above a threshold T. Each remaining group then contains the data units of the same concept. The similarity between two groups C1 and C2 is defined to be the average of the similarities between every data unit in C1 and every data unit in C2.

After the splitting is completed for a composite group , the data units in this group are not aligned yet. Using the separators to generate the alignment directly may be problematic because the “values” in the composite text nodes often do not have a uniform format, for example, some data units may be missing in some of these “values” if the corresponding SRRs do not have information for some attributes. Our solution to the data unit alignment problem is to apply the above agglomerative clustering method to the data units in each composite group.

IV. ASSIGNING LABELS

4.1 Local vs. integrated interface schemas

For a Web database, its local search interface often contains some attributes of the underlying data. We denote a local search interface schema (LIS) as $S_i = \{A_1, A_2, \ldots, A_k\}$, where each A_j is an attribute. When a query is submitted against the search interface, the entities in the returned results also have a certain “hidden” schema, denoted as $S_e = \{a_1, a_2, \ldots, a_n\}$, where each a_j ($j = 1,\ldots,n$) is an attribute to be discovered. The schema of the retrieved data and the interface schema

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

usually share a significant number of attributes [21]. This observation provides the basis for some of our basic annotators (see Section 4.2). If an attribute a_t in the search results does have a matched attribute A_t in the LIS, all the data units identified with a_t can be labeled by the name of A_t.

In our approach, for each used domain, we use WISE-Integrator [12, 13] as the basic tool to automatically build an integrated interface schema (IIS) over multiple Web databases in that domain. The generated integrated interface combines all the attributes of the local interface schemas. For the attributes of the same concept, their values are also merged. Each global attribute has a unique global name and an attribute-mapping table is created to establish the mapping between the name of each local interface schema attribute and its corresponding name in the integrated interface schema.

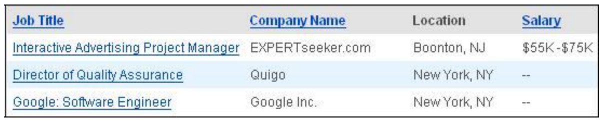
For each Web database in a given domain, our approach uses both the LIS of the database and the IIS of the domain to annotate the retrieved data units. Using IIS has two major advantages. First, it has the potential to increase the annotation recall. Since the integrated interface contains the attributes on all the LISs, it has a better chance that an attribute discovered from the returned results has a matching attribute in the IIS even though it has no matching attribute on the local interface. Second, when an annotator discovers a label for a group of data units, the label will be replaced with its corresponding global attribute name (if any) on the IIS by looking up the attribute-mapping table so that the data units of the same concept across different Web databases will have the same label.

4.2. Basic annotators

The data units belonging to the same concept (attribute) often share special common features, which are usually displayed in certain patterns. Based on this observation, we define 6 basic annotators to label data units, with each of them considering a special type of patterns/features.

Table Annotator (TA)

Many Web databases use a table to organize the returned SRRs, which visually has multiple rows and columns with each row representing an SRR. The table header, which indicates the meaning of each column, is usually located at the top of the table. Figure 2 is an example of such a Web database. Usually, the data units of the same concepts are well aligned with its corresponding column header. This special feature of the table layout can be utilized to annotate the SRRs.



Job Title	Company Name	Location	Salary
Interactive Advertising Project Manager	EXPERTseeker.com	Boonton, NJ	$55K-$75K
Director of Quality Assurance	Quigo	New York, NY	--
Google: Software Engineer	Google Inc.	New York, NY	--

Figure 2. Result page with table format

Our Table Annotator uses the physical position information of each data unit obtained at the extraction step and it works as follows. First, it identifies all the column headers of the table. Second, for each SRR, it takes a data unit in a cell and selects the column header whose area (determined by coordinates) has the maximum vertical overlap (i.e., based on the x- axis) with the cell. This unit is then assigned with this column header and labeled by the header text. The remaining data units are processed similarly. In case that the table header is not provided or is not successfully extracted by the ViNTs extractor [23], the Table Annotator will not be applied.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Query-based Annotator (QA)

The basic idea of this annotator is that the returned SRRs from a Web database are always related to the specified query. Specifically, the query terms entered through the search attributes on the local search interface of the Web database will most likely appear in some retrieved SRRs. For example, in Figure 1, query term "machine" is submitted through the "Title" field and all three titles of the returned SRRs contain this query term. Thus, the search attribute name "Title" can be used to annotate the title values of these SRRs.

Given a query with a set of query terms submitted against an attribute A_j on the local search interface, the query-based annotator finds the group that has the largest total occurrences of these query terms and we collect 2 sample result pages using different queries to form data sets, DS3 and DS4. Each of them contains one page from every site. DS3 is used to test the performance of our alignment and annotation methods based on the parameter values and statistics obtained from DS1 and DS2. At the same time, the annotation wrapper for each site will be generated. DS4 is used to test the quality of the generated wrappers. For each result page in the four data sets, the data units are manually extracted, aligned in groups, and assigned labels by a human expert, and the information will be used as the ground truth for comparison purpose.

We adopt the precision and recall to measure the performance of our methods. For alignment, the precision is defined as the percentage of the correctly aligned data units over all the units aligned by the system; recall is the percentage of the data units correctly aligned by the system over all data units aligned by the expert. For annotation, precision is the percentage of the correctly annotated units over all the data units annotated by the system; recall is the percentage of the data units correctly annotated by the system over all the manually annotated units.

The optimal APV obtained through our genetic training method is {0.30, 0.67, 1.08, 0.49, 0.14, 0.86}. The alignment precision and recall yielded using this APV are both 97.6% on average over the 14 pages in DS1. This result indicates that the data type and the presentation style are the most important features.

The collected pages in DS3 are used to evaluate our alignment and annotation methods. The alignment performance is evaluated by comparing the system generated aligned groups with the manually identified groups. Table 1 shows the average precision and recall for each domain and the overall average precision and recall for all 65 pages. Our method achieves very high accuracy in all domains in terms of both precision and recall. The overall average performances are nearly the same as those obtained from training, which shows that our alignment method is robust. The errors usually happen when an attribute has multiple data units in the same SRR (e.g., multiple authors for a book).

We use a similar method to evaluate the significance of each annotator. Each time, one annotator is removed and the remaining annotators are used to annotate the pages in DS3. From Figure 4, we can see that first, omitting any annotator causes both precision and recall to drop, i.e., every annotator contributes positively to the overall performance. Among the 6 annotators, the query based annotator and the frequency based annotator are the most significant. Moreover, when an annotator is removed, the recall decreases more severely than precision. This indicates that each of our annotators is fairly independent in describing one aspect of the attribute which, to a large degree, is not applicable to other annotators.

Finally, we conducted experiments to study the effect of using LIS versus IIS in annotation. We run the annotation process on DS3 again but this time, instead of using the integrated interface built for each domain, we use the local interface of each Web database. From the result depicted in Table 3, we can see that using the LIS has little effect on precision, but significant effect on recall (the overall average recall is reduced by almost 7 percentage points) because of the local interface schema inadequacy problem (see Section 4.1). And it also proves that using IIS can indeed increase the annotation performance.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Table 1. Performance using IIS vs. LIS

Domain	II S		LI S	
	Precision	Recall	Precision	Recall
Auto	100%	95.8%	100%	99.4%
Book	97.2%	96.2%	97.0%	85.9%
Job	95.3%	92.6%	95.3%	95.3%
Movie	99.7%	97.7%	99.7%	93.1%
Music	93.9%	93.9%	92.2%	84.7%
Game	98.9%	98.8%	98.9%	93.0%
average	97.2%	95.9%	96.7%	89.1%

V. CONCLUSION

The data annotation problem and proposed a multi-annotator approach to automatically constructing an annotation wrapper for annotating the search result records retrieved from any given Web database. Each of these annotators exploits one type of special features for annotation and our experimental results indicate that each proposed annotator is useful and they together are capable of generating high quality annotation wrappers. We also illustrated how the use of the integrated interface schema can help alleviate the local interface schema inadequacy problem and the inconsistent label problem. In addition, a new data alignment technique using richer yet automatically obtainable features was proposed to cluster data units into different groups/concepts in support of more robust and holistic annotation. There is still room for improvement in several areas as mentioned in Section 6. For example, we need to enhance the ability to deal with multi-valued attributes that may have more than one value for some SRR (e.g., authors for books).

REFERENCES

[1] A. Arasu and H. Garcia-Molina. Extracting Structured Data from Web pages. SIGMOD Conference, 2003.
[2] L. Arlotta, V. Crescenzi, G. Mecca, and P. Merialdo. Automatic Annotation of Data Extracted from Large Web Sites. WebDB Workshop, 2003.
[3] P. Chan and S. Stolfo. Experiments on Multistrategy Learning by Meta-Learning. CIKM Conference, 1993.
[4] W. Bruce Croft. Combining approaches for information retrieval. In Advances in Inf. Retr.: Recent Research from the Center for Intel. Inf. Retr., Kluwer Academic, 2000.
[5] V. Crescenzi, G. Mecca, and P. Merialdo. RoadRUNNER: Towards Automatic Data Extraction from Large Web Sites. VLDB Conference, 2001.
[6] S. Dill, N. Eiron, D. Gibson, and et al. SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation. WWW Conference, 2003.
[7] D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y. Ng, R. Smith. Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages. Data and Knowledge Engineering, 31(3), 227-251, 1999.
[8] D. Freitag. Multistrategy Learning for Information Extraction. ICML, 1998.
[9] S. Handschuh, S. Staab, and R. Volz. On Deep Annotation. WWW Conf., 2003.
[10] S. Handschuh and S. Staab. Authoring and Annotation of Web Pages in CREAM. WWW Conference, 2003.