ISSN:2320-0790

An International Journal of Advanced Computer Technology

An Efficient Annotation of Search Results Based on Feature Ranking Approach from Web Databases

A.Jebha[1], R.Tamilselvi[2]
[1]Research Scholar, Dr. SNS.Rajalakshmi College of Arts and Science, Coimbatore, India
[2]Assistant Professor, Department of Computer Science, Dr. SNS.Rajalakshmi College of Arts and Science, Coimbatore, India

Abstract: With the increased number of web databases, major part of deep web is one of the bases of database. In several search engines, encoded data in the returned resultant pages from the web often comes from structured databases which are referred as Web databases (WDB). A result page returned from WDB has multiple search records (SRR).Data units obtained from these databases are encoded into the dynamic resultant pages for manual processing. In order to make these units to be machine process able, relevant information are extracted and labels of data are assigned meaningfully. In this paper, feature ranking is proposed to extract the relevant information of extracted feature from WDB. Feature ranking is practical to enhance ideas of data and identify relevant features. This research explores the performance of feature ranking process by using the linear support vector machines with various feature of WDB database for annotation of relevant results. Experimental result of proposed system provides better result when compared with the earlier methods.

Keywords: web databases; data units; structured databases; semantic data; multi annotator method; schema value annotator; support vector machine; data alignment algorithm.

I. INTRODUCTION

Major part of the deep web is based on database that is for various search engines, data prearranged in the returned result pages arrive from the fundamental structured databases. Those types of search engines are frequently defined as Web databases (WDB). A usual result page returned from a WDB contains multiple search result records (SRRs) in which each SRR holds multiple data units every data units illustrates one part of a real-world entity. The following Fig. 1 depicts three search result records (SRRs) on a result page from a book Web databases (WDB). Each SRR illustrates one book with quite a few data units, for example, the first book record in Figure. 1(a) has data units "Talking Back to the Machine: Computers and Human Aspiration," "Peter J. Denning," etc. and Figure 1(b) contains the Simplified HTML source for the first SRR.

Figure 1: (a) Original Html Page of Search Results from Bookpool.Com

<FORM><A>Talking Back to the Machine: Computers and Human Aspiration</A>
 Peter J. Denning / <FONT> <I>Springer-Verlag / 1999 / 0387984135/0.06667</I> </FONT>
Our Price <B>$17.50</B> ~ <FONT>You Save $9.50 (35% Off)</FONT>
 <I>Out-Of-Stock</I></FORM>

Figure 1: (b) Simplified HTML source for the first SRR

A data unit is a portion of text that semantically symbolizes one idea of an entity. It responds to the value of a evidence under an attribute. This is not similar to the text node which depicts to a series of text bounded by a couple of HTML tags. There is a large command for gathering data of notice from multiple WDBs. For instance, when a book comparison shopping system gathers multiple result records from various book sites, it desires to establish whether any two SRRs transfer to the same book. Then the ISBNs can be matched to attain this. Once ISBNs are not accessible, their authors and titles might be compared. In addition the system wants to file the prices obtainable by each site. Hence, the system desires to recognize the

semantic of each data unit. In contrast, the data units are frequently not offered in result pages. For example in Fig. 1, semantic labels for the values of author, publisher, title, etc are not given. With the semantic labels for data units is not only significant for the obtained record linkage job, but also for managing gathered SRRs into a table of database for further analysis. Many applications necessitate great human labours to annotate data units manually, which greatly limits the scalability.

In this paper, feature raking is proposed for improving the search results from the Web database. To perform data unit level annotation, the relationship between data units and text nodes are analysed. After that data units are aligned into different groups by using clustering-based shifting technique in which similar semantic is achieved or data units of similar group. The data alignment is done based on considering significant features shared among data units namely Presentation styles (PT),data types(DT),Adjacency information(AD) and data contents(DC).Then feature ranking using linear support vector machine is preformed to select relevant information from the searched result. Feature ranking performed on the weights of each features by linear SVM. After performing ranking, annotation is done based on six basic annotators in which each annotator assign labels to data units independently based on certain features. Finally annotation wrapper is constructed for a given WDB in which wrapper efficiently annotate the SRR retrieved from WDB with new type of applied queries.

II. Related Work

In [1], [2] methods presented by Krushmerick et.al and Liu et.al relied on human users to point out the required information on sample pages and label the pointed data at the same time, and after that the system made a series of rules or wrapper to inherit the identical set of information on web pages from the similar source. Still, the difficulty arises from reduced scalability and is not appropriate for applications in [3], [4] that require extracting information from a huge amount of web sources. In [5] Embley et al. make use of ontologist jointly with some heuristics to repeatedly extract data in multi record documents and for labelling them. Conversely, ontologist for dissimilar domains must be created manually. In [6] Mukherjee et al. develop the spatial locality and presentation styles of semantically linked items, except its learning process for annotation is domain dependent. In [7] Wang et.al presented primary use of HTML tags to align data units by loading them into a table by a regular expression based data tree algorithm. After that, it utilizes four heuristics to pick a label for each aligned table column. Next in [8] Zhu et.al achieves attributes extraction and labelling at the same time. Yet, the label set is predefined and holds only a little amount of values. In [9] Liu et.al uses ViDIE which is visual features on result pages to carry out alignment and in addition it produces an alignment wrapper. Although its alignment is simply at text node level, not achieved in data unit level. In [10] Elmeleegy et.al presented a method which initially divides each SRR into text segments. Generally ordinary number of segments is evaluated to be the amount of aligned columns or attributes. The SRR with extra segments are then again divided using the frequent number.

III. Proposed Work

The following diagram in figure 2 shows the architecture of proposed system is as follows:

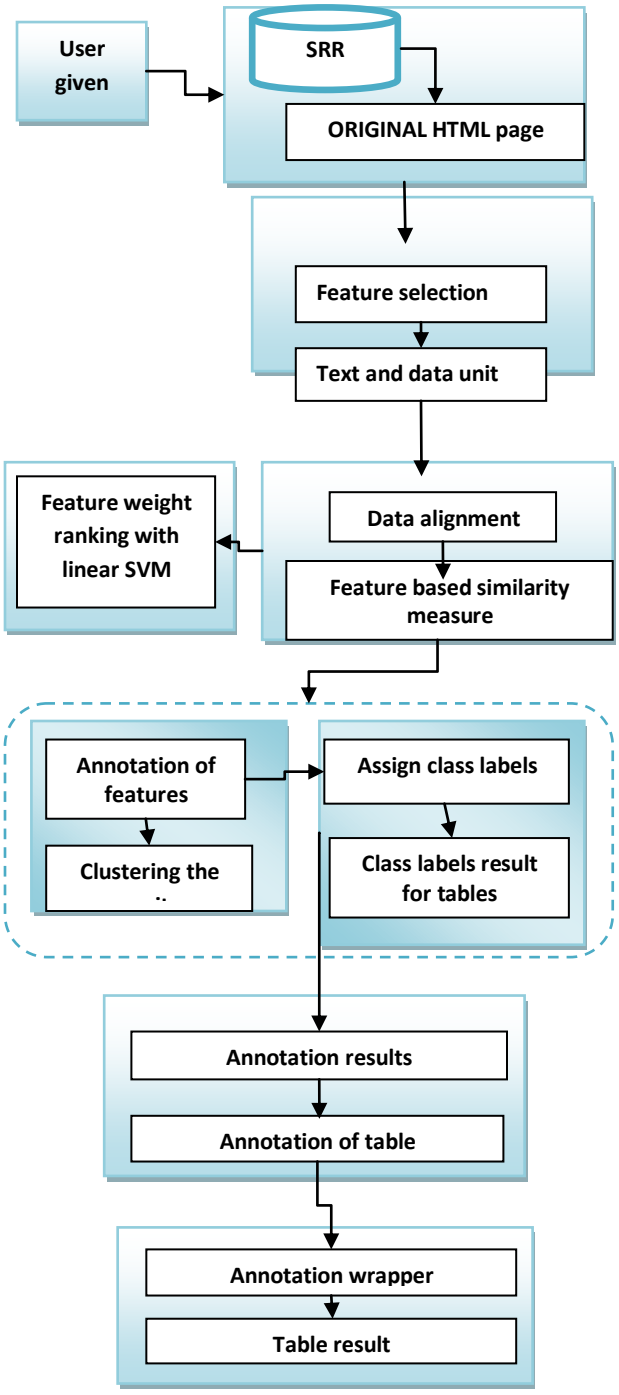

Figure 2: Proposed System Architecture

A. *Analysis of Search Result Records(SRR)*

One-to-One Relationship (denoted as T= U).In this type, each text node holds precisely one data unit that is the text of this node encloses the value of a single attribute. Each

text node is enclosed by the pair of tags <A> and </A> which refers to is a cost of the Title attribute. This can be referred to those types of text nodes known as atomic text nodes which are equal to the data units.

One-to-Many Relationship (denoted as T $\supset$ U). In this type of relationship, compound data units are instructed in one text node. It contains four semantic data units namely Date, ISBN, Publisher Relevance Score and Publication. As the text of those kinds of nodes can be regard as a composition of the texts of several data units, and can be called it as composite text node. The significant observation that can be done is: when the attributes data units A1 ...A_k in one SRR are fixed as a complex text node, it is frequently accurate that the data units of the identical attributes in extra SRRs revisited by the same WDB are also fixed as complex text nodes, and such embedded data units constantly emerge in the similar order. Generally this examination is suitable for the reason that SRRs are produced by template programs. Finally each complex text node is divided to get real data units and annotate them.

Many-to-One Relationship (denoted as T $\subset$ U). In this type of relationship, multiple nodes of text jointly form a data unit. Author attribute value is composed with multiple nodes of text with each embedded contained by a distinct pair of (<A>, </A>) HTML tags. In general the webpage designers employ particular HTML tags to decorate definite information. This kind of tags is called as decorative tags since they are utilized primarily for varying the appearance of part of the text nodes. For this reason of extraction and annotation, tags inside SRRs are identified and removed in order that the completeness of each split data unit can be re-established. The initial phase of data alignment algorithm holds this case particularly.

One-To-Nothing Relationship (denoted as T $\neq$ U). In this type of relationship, the text nodes based on to this group are not included of any data unit inside SRRs. In addition, its examinations point out that these text nodes are frequently exhibited in a definite pattern across every SRRs. Hence, this is called as template text nodes. This identifies template text nodes by utilizing frequency-based annotator.

B. Data Alignment Algorithms

Data alignment algorithm is based on the hypothesis that attributes emerge in the similar order across every SRRs on the similar result page, even though the SRRs might hold dissimilar sets of attributes. In general, this is considered as true for the reason that the SRRs from the similar WDB are usually produced by the similar template program. Accordingly, the SRRs on a result page is conceptually considered in a table arrangement where each row symbolizes one SRR and each cell contains a data unit.
Data alignment method is performed based on following steps. The detail of each step will be provided later.

Step 1: Text nodes merging: This step identifies and eliminates decorative tags from each SRR to permit the text nodes equivalent to the same attribute to be merged into a single text node.

Step 2: Text nodes Alignment: This step aligns text nodes into clusters or groups in order that ultimately each group holds the text nodes with the similar concept or the same set of concepts.

Step 3: Text nodes Splitting: This step goal is to split the "values" in text nodes of composite into distinct data units. This step is performed based on the text nodes in the similar group accordingly.

Step 4: Data units Alignment: This step divides each composite group into group of multiple aligned in which each holding the data units of the similar concept.

1: j← 1;
2: while true
//create alignment groups
3: for $i \leftarrow 1$ to number of SRRS
4:$G_j \leftarrow SRR[i][j]$
5: If G_j is empty
6: exit;
//break the loop
7: V $\leftarrow$ CLUSTERING (G)
8: IF $|V| > 1$
9:$S \leftarrow \emptyset$;
10: for $x \leftarrow 1$ to number of SRRS
11: for y ← $j + 1$ to SRR[i].length
12: S ← $SRR[x][y]$;
13: $V[c] = \min_{k=1\ to\ V}(sim(V[k], s))$;
14: for k← $1\ to\ |V|$ and k≠ c
15: for each SRR[i][x] in V[K]
16: insert NIL at position j in SRR[x];
17: $j \leftarrow j + 1$;
// move to next group
CLUSTERING (G)
1: V← $all\ data\ units\ in\ G$;
2: while $|V| > 1$
3: best ← 0;
4: L← NIL; R← NIL;
5: for each A in V
6: for each B in V
7: If $(A! = B)$ and (sim(A,B)>best)
8: best ← $sim(A, B)$;
9: L ← A;
10: $R \leftarrow B$;
11: If best >T
12: Remove L from V;
13: Remove R from V;
14: add $L \cup R\ to\ V$;
15: Else break loop;
16: Return V;

C. Data Unit and Text Node Features Extraction

Features such as Presentation Style (PS), Adjacency (AD), Data Content (DC), Tag Path (TP), and Data Type (DT) are extracted as follows.

Presentation Style (PS): This feature depicts how a data unit is showed on a webpage. It contains of six style features: font face, weight, text decoration (underline, strike, etc.), font size, font color, and whether it is italic. Data units of the similar idea in dissimilar SRRs are typically displayed in the similar style.

Adjacency (AD): This feature represents that for a known data unit in an SRR and data units immediately before and after data units in the SRR, correspondingly. Assume two data units are derived from two separate SRRs. This can be observed that two such data units belong to the similar concept.

Data Content (DC): The text nodes or data units with the similar concept frequently share definite keywords. This is factual for two causes. First, the data unit's equivalent to the search field where the user prompts a search term typically holds the search keywords.

Tag Path (TP): This feature considers a tag path of a text node which contains a series of tags traversing from the origin of the SRR to the analogous node in the tag tree. As ViNTs is utilized for SRR extraction, the similar tag path expression is adopted. Each node in the expression holds two parts, one is the name of the tag name and another is the direction representing whether the subsequent node is the next sibling or the first child.

Data Type (DT): Each data unit possesses semantic type even though it is just a text string in the HTML code. The subsequent basic data types are presently incorporated in this approach namely Integer, Percentage, Symbol, Decimal, String, Date, Time and Currency.

D. Feature Ranking using Linear SVM

In this section feature ranking using linear SVM is presented for ranking the weights of features. In general Support vector machines (SVMs) are mainly used for data classification. SVM determines a separating hyper plane with the maximal margin among two classes of data. Consider SVM works by getting a set of instance-label pairs (x_i, y_i) where $x_i \epsilon R^n$, $y_i \in \{1,-1\}$ i= 1,……,l,

The following unconstraint optimization problem is solved by SVM is as follows:

$$\mathbf{min_{w,b}} \frac{1}{2}\mathbf{w^T w + C\sum_{i=1}^{l} \xi(w, b; x_i, y_i)} \rightarrow \mathbf{(1)}$$

Where $\xi(w, b; x_i, y_i)$ are referred as loss function, and $C \leq 0$ is a penalty parameter on the training error. Two common loss functions are as follows:

Max $(1\text{-}\mathbf{y_i}(\mathbf{w^T}\ \Phi\ (\mathbf{x_i}) + b), 0)$ and max $(\mathbf{1 - y_i}(\mathbf{w^T}\ \mathbf{\Phi}(\mathbf{x_i}) + \mathbf{b}), \mathbf{0})^{\mathbf{2}} \rightarrow (2)$

Where $\mathbf{\Phi}$ referred as a function that plotted training data into higher dimensional space. The first one is called L1-loss SVM, and the second one is called as L2-loss SVM. While taking part in the challenge, the L2-loss function has been chosen.

For any testing instance x, the decision function or predictor is as follows:

$F(x) = sgn(\mathbf{w^T}\ \Phi(x)+b) \rightarrow (3)$

In practical, a kernel function $K(x_i, x_j) = \Phi(x_i)^T\ \Phi(x_j)$ is used train the SVM.

Linear SVM has $\Phi(x)=x$, then the kernel function is $K(x_i, x_j)= x_i^T x_j$

After performing a linear SVM model, the relevance of each feature weights is decided by $w \epsilon R^n$ in (1).The larger $|w_j|$ is that the j^{th} weight of feature describes a more significant role in the decision function described in (3). Thus ranking is performed based on $|w_j|$. Each phase of working is described in the following algorithm:

Algorithm: Feature weight ranking

Input: Training set, (x_i, y_i), i=1,….,l

Output: Sorted feature weight ranking list

Steps:

1. Find the best parameter of C by using grid search
2. With the best value of C ,train the L2-loss of linear SVM model
3. Sort the weights of features based on the absolute values of weights obtained in the model.
4. Return the sorted feature weight results.

E. Basic Annotators

Table Annotator (TA): Several WDBs exploits a table to arrange the returned SRRs. Each row in a table depicts an SRR. The table header, which describes the sense of each column, is typically positioned at the top of the table. Position information of each data unit is attained through SRR extraction; then the information is utilized to connect each data unit with its corresponding header. The presented Table Annotator works as follows: Initially, it recognizes every column headers of the table. Then, for each SRR, it holds a data unit in a cell and picks the column header whose area has the maximum vertical overlap with the cell.

Query-Based Annotator (QA): The fundamental scheme of this annotator is that the returned SRRs from a WDB are constantly connected to the particular query. Particularly, the query concepts provided in the search attributes will probably emerge in various retrieved SRRs. Generally query concepts next to an attribute can be provided to a textbox or selected from a local search interface of selection list.

Schema Value Annotator (SA): Several attributes on a search interface contains user defined standards on the interface. For instance, the attribute Publishers might have a set of defined values in its selection list. Additional attributes in the IIS be inclined to have already defined values and such attributes are probably contains more values than those in LISs, since those attributes from multiple interfaces are incorporated, their values are also mutually shared.

Frequency-Based Annotator (FA): Adjacent units contains dissimilar incidence Frequencies the data units with the advanced frequency are probable to be attribute names, as piece of the template program for discovering records,

whereas lower frequency data units probably arrive from databases the same as embedded values.

In-Text Prefix/Suffix Annotator (IA): In particular, a part of data is programmed with its label to figure a single unit without any understandable separator among the value and the label except it holds both the value and label. Those nodes might appear in every multiple SRRs. Subsequent to data alignment, every nodes could be aligned together to form a group.

Common Knowledge Annotator (CA): Various data units present on the result page are self-descriptive as of the general idea are shared by human beings. For instance, "in stock" and "out of stock" appear in various SRRs from e-commerce sites. Users realize that it is concerning the ease of use of the product since this is general idea. As a result general idea annotator tries to develop this condition by means of several predefined general ideas.

F. Annotation Wrapper

In this section annotation of data units are done on a result page by means of using these annotated data units to build an annotation wrapper for the Web databases in order that the new SRRs retrieved from the similar WDB may be annotated employing this wrapper rapidly without reapplying the complete annotation method. Each annotated data unit groups responds to an attribute in the SRRs. Next the data unit groups are annotated; they are prearranged founded by organizing its data units in the novel SRRs. Assume the i^{th} group is G_i. All SRR contains a tag-node sequence that contains of simply HTML tag texts and names. For each data unit in group G_i, the sequence can be scanned from both forward and backward to attain the data units prefix and suffixes. The scan terminates when a determined unit is a suitable data unit with a significant label assigned. After that, the prefixes of all the data units in Group G_i is compared to get hold of the general prefix common by these data units.

IV. Experimental Analysis

In this section the proposed system is experimentally evaluated based on precision and recall measures.

Precision and recall measures are used for information retrieval to evaluate the performance of our presented methods. 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.

Precision: value is calculated is based on the retrieval of information at true positive prediction, false positive .In healthcare data precision is calculated the percentage of positive results returned that are relevant.
Precision =TP/ (TP+FP)

Recall: value is calculated is based on the retrieval of information at true positive prediction, false negative. In healthcare data precision is calculated the percentage of positive results returned that are Recall in this context is also referred to as the True Positive Rate. Recall is the fraction of relevant instances that are retrieved.
Recall =TP/ (TP+FN)

The following comparison graph shows the precision and recall measures of existing and proposed work of web database search results annotations as follows:

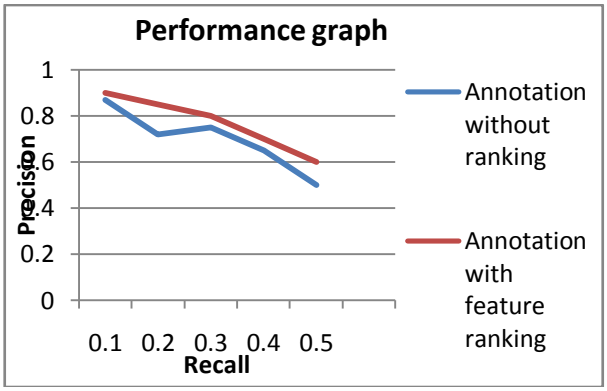

Figure 3: Performance comparison graph

In the above graph measure the performance of the annotation of the search result records with the feature ranking by using linear SVM against existing results based feature search result records alignment phase in the results shows the best precision and recall result in the results respectively.

V. Conclusion

The present work proposes feature ranking method by using linear SVM for annotating the web search results. The proposed method handles the data annotation problem efficiently and presented a multi annotator method by automatically generating an annotation wrapper for annotating the search result records (SRR) retrieved from any specified web database (WDB).The proposed feature ranking provides improved performance of search results. Experimental result of proposed result efficiently annotates the search results and shows that a feature ranking employing weights from linear SVM models yields better performances, when compared with the earlier system of work respectively.

References

[1] N. Krushmerick, D. Weld, and R. Doorenbos, "Wrapper Induction for Information Extraction," Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI), 1997.

[2] Bizarro, L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources," Proc. IEEE 16th Int'l Conf. Data Eng. (ICDE), 2001.

[3] W. Meng, C. Yu, and K. Liu, "Building Efficient and Effective Meta search Engines," ACM Computing Surveys, vol. 34, no. 1, pp. 48-89, 2002.

[4] Z. Wu et al., "Towards Automatic Incorporation of Search Engines into a Large-Scale Metasearch Engine," Proc. IEEE/WIC Int'l Conf. Web Intelligence (WI '03), 2003.

[5] D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y. Ng, and R. Smith, "Conceptual-Model-Based Data Extraction from Multiple-Record Web Pages," Data and Knowledge Eng., vol. 31, no. 3, pp. 227-251, 1999.

[6] S. Mukherjee, I.V. Ramakrishnan, and A. Singh, "Bootstrapping Semantic Annotation for Content-Rich HTML Documents," Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2005.

[7] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. 12th Int'l Conf. World Wide Web (WWW), 2003.

[8] J. Zhu, Z. Nie, J. Wen, B. Zhang, and W.-Y. Ma, "Simultaneous Record Detection and Attribute Labeling in Web Data Extraction, Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2006.

[9] W. Liu, X. Meng, and W. Meng, "ViDE: A Vision-Based Approach for Deep Web Data Extraction," IEEE Trans. Knowledge and Data Eng., vol. 22, no. 3, pp. 447-460, Mar. 2010.

[10] H. Elmeleegy, J. Madhavan, and A. Halevy, "Harvesting Relational Tables from Lists on the Web," Proc. Very Large Databases (VLDB) Conf., 2009.

[11] Spaccapietra.S and Parent.C, "A step forward in solving structural conflicts," IEEE Transactions on Knowledge 5and Data Engineering, vol. 6, no. 2, 1998.

[12] Selinger, P.G., Astrahan, M.M., Chamberlin, D.D., Lorie, R.A.,Price T.G. Access Path Selection in a Relational Database System. In Readings in Database Systems. Morgan Kaufman.