iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** V **Month of publication:** May 2015

DOI:

www.ijraset.com Volume 3 Issue V, May 2015
IC Value: 13.98 ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Advanced Annotation Creator for Search Results from Web Databases

Gayathri Thangavel[1], C.Menaka. [2] MCA.,M.Phil
Master of Computer Applications, Anna University, Chennai.

Abstract— A large portion of the deep web is database based, i.e., for many search engines, data encoded in the returned result pages come from the underlying structured data-bases. Such type of search engines is often referred as Web databases (WDB). 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 processable, 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. In this paper, we present 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 we 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. Our experiments indicate that the proposed approach is highly effective. The application is designed using Microsoft Visual Studio .Net 2005 as front end. The coding language used is Visual C# .Net. MS-SQL Server 2000 is used as back end database.

Index Terms—Data alignment, data annotation, web database, wrapper generation

I. INTRODUCTION

A large portion of the deep web is database based, i.e., for many search engines, data encoded in the returned result pages come from the underlying structured data- bases. Such type of search engines is often referred as Web databases(WDB). A typical result page returned from a WDB has multiple search result records (SRRs). Each SRR contains multiple data units each of which describes one aspect of a real-world entity. Fig. 1 shows three SRRs on a result page from a book WDB. Each SRR represents one book with several data units, e.g., the first book record in Fig. 1 has data units "Talking Back to the Machine: Computers and Human Aspiration," "Peter J. Denning," etc.

In this paper, a data unit is a piece of text that semantically represents one concept of an entity. It corresponds to the value of a record under an attribute. It is different from a text node which refers to a sequence of text surrounded by a pair of HTML tags. Section 3.1 describes the relationships between text nodes and data units in detail. In this paper, we perform data unit level annotation. Grouping data units of the same semantic can help identify the common patterns and features among these data units. These common features are the basis of our annotators. In Phase 2 (the annotation phase), we introduce multiple basic annota- tors with each exploiting one type of features. Every basic annotator is used to produce a label for the units within their group holistically, and a probability model is adopted to determine the most appropriate label for each group.

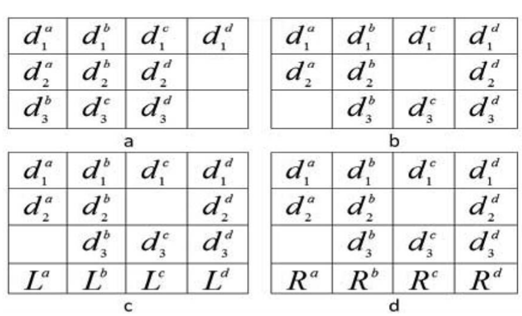


Fig. 2. Illustration of our three-phase annotation solution.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

This paper has the following contributions:

A. While most existing approaches simply assign labels to each HTML text node, we thoroughly analyze the relationships between text nodes and data units. We perform data unit level annotation.

B. We propose a clustering-based shifting technique to align data units into different groups so that the data units inside the same group have the same semantic. Instead of using only the DOM tree or other HTML tag tree structures of the SRRs 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 (DT), data contents (DC), presentation styles (PS), and adjacency (AD) information.

C. We utilize the integrated interface schema (IIS) over multiple WDBs in the same domain to enhance data unit annotation. To the best of our knowledge, we are the first to utilize IIS for annotating SRRs.

D. We construct an annotation wrapper for any given WDB. The wrapper can be applied to efficiently annotating the SRRs retrieved from the same WDB with new queries.

II. RELATED WORK

Web information extraction and annotation has been an active research area in recent years. Many systems [18], [20] rely on human users to mark the desired information on sample pages and label the marked data at the same time, and then the system can induce a series of rules (wrapper) to extract the same set of information on webpages from the same source. These systems are often referred as a wrapper induction system. Because of the supervised training and learning process, these systems can usually achieve high extraction accuracy. However, they suffer from poor scalability and are not suitable for applications [24], [31] that need to extract information from a large number of web sources.

This method has limited applicability because many WDBs do not encode data units with their labels on result pages. In ODE system [28], ontologies are first constructed using query interfaces and result pages from WDBs in the same domain. The domain ontology is then used to assign labels to each data unit on result page. After labeling, the data values with the same label are naturally aligned. The approach in [36] performs attributes extraction and labeling simulta- neously. However, the label set is predefined and contains only a small number of values.

We align data units and annotate the ones within the same semantic group holistically. Data alignment is an important step in achieving accurate annotation and it is also used in [25] and [30]. Most existing automatic data alignment techniques are based on one or very few features. The most frequently used feature is HTML tag paths (TP) [33]. The assumption is that the subtrees corresponding to two data units in different SRRs but with the same concept usually have the same tag structure. ViDIE [21] uses visual features on result pages to perform alignment and it also generates an alignment wrapper. But its alignment is only at text node level, not data unit level. The method in [7] first splits each SRR into text segments. The most common number of segments is determined to be the number of aligned columns (attributes). Our data alignment approach differs from the previous works in the following aspects.

Specifically, among the six basic annotators in our method, two (i.e., schema value annotator (SA) and frequency-based annotator (FA)) are new (i.e., not used is DeLa), three (table annotator (TA), query-based annotator (QA) and common knowledge annotator (CA)) have better implementations than the corresponding annotation heuristics in DeLa, and one (in-text prefix/suffix annotator (IA)) is the same as a heuristic in DeLa. We employ ViNTs [34] to extract SRRs from result pages in this work

This paper is an extension of our previous work [22]. The following summarizes the main improvements of this paper over [22]. First, a significantly more comprehensive discus- sion about the relationships between text nodes and data units is provided. Specifically, this paper identifies four relationship types and provides analysis of each type, while only two of the four types (i.e., one-to-one and one-to-many) were very briefly mentioned in [22]. Second, the alignment algorithm is significantly improved. With these two improvements, the new alignment algorithm takes all four types of relationships into consideration. Third, the experiment section (Section 7) is significantly different from the previous version. The data set used for experiments has been expanded by one domain (from six to seven) and by 22 WDBs (from 91 to 112). Moreover, the experiments on alignment and annotation have been redone based on the new data set and the improved alignment algorithm. Fourth, several related papers that were published recently have been reviewed and compared in this paper.

III. FUNDAMENTALS

A. Data Unit and Text Node

Each SRR extracted by ViNTs has a tag structure that determines how the contents of the SRRs are displayed on a web browser. Each node in such a tag structure is either a tag node or a text node. A tag node corresponds to an HTML tag surrounded by "<" and ">" in HTML source, while a text node is the text outside the "<" and ">." Text nodes are the visible elements on the webpage and data units are located in the text nodes. However, as we can see from Fig. 1, text nodes are not always identical to data units. Since our annotation is at the data unit level, we need to identify data units from text nodes.

Depending on how many data units a text node may contain, we identify the following four types of relationships between data unit (U) and text node (T): One-to-One Relationship (denoted as T ¼ U). In this type, each text node contains exactly one data unit, i.e., the text of this node contains the value of a single attribute. This is the most frequently seen case.

B. Data Unit and Text Node Features

We identify and use five common features shared by the data units belonging to the same concept across all SRRs, and all of them can be automatically obtained. It is not difficult to see that all these features are applicable to text nodes, including composite text nodes involving the same set of concepts, and template text nodes.

C. Data Content (DC)

The data units or text nodes with the same concept often share certain keywords. This is true for two reasons. First, the data units corresponding to the search field where the user enters a search condition usually contain the search key- words. For example, in Fig. 1, the sample result page is returned for the search on the title field with keyword "machine." Text nodes that contain data units of the same concept usually have the same leading label.

D. Presentation Style (PS)

This feature describes how a data unit is displayed on a webpage. It consists of six 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 Fig. 1, all the availability information is displayed in the exactly same presentation style.

E. Tag Path (TP)

A tag path of a text node is a sequence of tags traversing from the root of the SRR to the corresponding node in the tag tree. Since we use ViNTs for SRR extraction, we adopt the same tag path expression as in [34]. Text node is simply represented as <#TEXT>. For example, in Fig. 1b, the tag path of the text node "Springer-Verlag/1999/0387984135/0.06667" is <FORM>C<A>C
S<#TEXT>S
<FONT>C<T>C. An observation is that the tag paths of the text nodes with the same set of concepts have very similar tag paths, though in many cases, not exactly the same.

IV. DATA ALIGNMENT

A. Data Unit Similarity

The purpose of data alignment is to put the data units of the same concept into one group so that they can be nnotated holistically.

Our alignment algorithm also needs the similarity between two data unit groups where each group is a collection of data units. We define the similarity between groups G_1 and G_2 to be the average of the similarities between every data unit in G_1 and every data unit in G_2.

B. Alignment Algorithm

Our data alignment algorithm is based on the assumption that attributes appear in the same order across all SRRs on the same result page, although the SRRs may contain different sets of attributes (due to missing values). If an alignment group contains all the data units of one concept and no data unit from other concepts, we call this group well- aligned. The goal 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.

Our data alignment method consists of the following four steps. The detail of each step will be provided later.

Step 1: Merge text nodes. This step detects and removes decorative tags from each SRR to allow the text nodes corresponding to the same attribute (separated by decorative tags) 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. This step is carried out based on the text nodes in the same group holistically. A group whose "values" need to be split is called a composite group.

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.

As we discussed in Section 3.1, the Many-to-One relationship between text nodes and data units usuallyoccurs because of the decorative tags. We need to remove them to restore the integrity of data unit. In Step 1, we use a modified method in [35] to detect the decorative tags. For every HTML tag, its statistical scores of a set of predefined features are collected across all SRRs, including the distance to its leaf descendants, the number of occurrences, and the first and last occurring positions in every SRRs, etc.

In Step 2, as shown in ALIGN in Fig. 4, text nodes are initially aligned into alignment groups based on the positions within SRRs so that group G_j contains the jth text node from each SRR (lines 3-4). Since a particular SRR may have no value(s) for certain attribute(s) (e.g., a book would not have discount price if it is not on sale), G_j may contain the elements of different concepts. We apply the agglomerative clustering algorithm [17] to cluster the text nodes inside this group (line 7 and CLUSTERING).

The data units in a composite group are not always aligned after splitting because some attributes may have missing values in the composite text node. Thus, in this case, these two features are not used for calculating similarity for aligning data units. Their feature weights are proportionally distributed to the three features used.

V. ASSIGNING LABELS

A. Local versus Integrated Interface Schemas

For a WDB, its search interface often contains some attributes of the underlying data. We denote a LIS as S_i ¼ fA_1 ; A_2 ; A_k g, 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 ¼ fa_1 ; a_2 ; . . . ; a_n g, where each a_j (j ¼ 1 . . . n) is an attribute to be discovered. The schema of the retrieved data and the LIS usually share a significant number of attributes.

Another potential problem associated with using LISs for annotation is the inconsistent label problem, i.e., different labels are assigned to semantically identical data units returned from different WDBs because different LISs may give different names to the same attribute. This can cause problem when using the annotated data collected from different WDBs, e.g., for data integration applications.

In our approach, for each used domain, we use WISE- Integrator [14]) to build an IIS over multiple WDBs in that domain. The generated IIS combines all the attributes of the LISs. For matched attributes from different LISs, their values in the local interfaces (e.g., values in selection list) are combined as the values of the integrated global attribute [14]. Each global attribute has a unique global name and an attribute-mapping table is created to establish the mapping between the name of each LIS attribute and its correspond- ing name in the IIS. In this paper, for attribute A in an LIS, we use gn(A) to denote the name of A's corresponding attribute (i.e., the global attribute) in the IIS.

B. Basic Annotators

In a returned result page containing multiple SRRs, the data units corresponding to the same concept (attribute) often share special common features. And such common features are usually associated with the data units on the result page in certain patterns. Based on this observation, we define six basic annotators to label data units, with each of them considering a special type of patterns/features.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Manufacture	Model	Class	Year	City	State	Price
HONDA	accord LX	4 DOOR	1998	playa del rey	CA	$11,500
HONDA	ACCORD LX	4 DOOR	1994	Spokane	WA	$ 7,500
HONDA	Accord Lx	4 DOOR	1997	Winona ake	ID	$ 8,700
HONDA	Accord LX	4 DOOR	1994	Cave Creek	AZ	$ 5,999
HONDA	Accord	4 DOOR	1999	Pomona	CA	$17,500

Fig. 6. SRRs in table format.

C. Table Annotator (TA)

Many WDBs use a table to organize the returned SRRs. In the table, each row represents an SRR. The table header, which indicates the meaning of each column, is usually located at the top of the table. However, DeLa only relies on HTML tag <TH> and <THEAD> for this purpose. But many HTML tables do not use <TH> or <THEAD> to encode their headers, which limits the applicability of DeLa's approach. In the test data set we collected, 11 WDBs have SRRs in table format, but only three use <TH> or <THEAD>. In contrast, our table annotator does not have this limitation.

D. Frequency-Based Annotator (FA)

The adjacent units have different occurrence frequencies. As argued in [1], the data units with the higher frequency are likely to be attribute names, as part of the template program for generating records, while the data units with the lower frequency most probably come from databases as embedded values. Consider a group G_i whose data units have a lower frequency. This can be easily conducted by following their preceding chains recursively until the encountered data units are different. All found preceding units are concatenated to form the label for the group G_i.

E. In-Text Prefix/Suffix Annotator (IA)

In some cases, a piece of data is encoded with its label to form a single unit without any obvious separator between the label and the value, but it contains both the label and the value. Such nodes may occur in all or multiple SRRs. After data alignment, all such nodes would be aligned together to form a group. For example, in Fig. 1, after alignment, one group may contain three data units, {"You Save $9.50," "You Save $11.04," "You Save $4.45"}. The in-text prefix/suffix annotator checks whether all data units in the aligned group share the same prefix or suffix. If the same prefix is confirmed and it is not a delimiter, then it is removed from all the data units in the group and is used as the label to annotate values following itIn the above example, the label "You save" will be assigned to the group of prices. Any group whose data unit texts are completely identical is not considered by this annotator.Some data units on the result page are self-explanatory because of the common knowledge shared by human beings. For example, "in stock" and "out of stock" occur in many SRRs from e-commerce sites. However, it only considers certain patterns. Our Common knowledge annotator considers both patterns and certain value sets such as the set of countries. First, our common concepts are domain independent. Second, they can be obtained from existing information resources with little additional human effort TABLE 1 Applicabilities and Success Rates of Annotators The applicability of an annotator is the percentage of the attributes to which the annotator can be applied. One advantage of this model is its high flexibility in the sense that when an existing annotator is modified or a new annotator is added in, all we need is to obtain the applicability and success rate of this new/revised annotator while keeping all remaining annotators unchanged. We also note that no domain-specific training is needed to obtain the applicability and success rate of each annotator.

VI. ANNOTATION WRAPPER

Once the data units on a result page have been annotated, we use these annotated data units to construct an annota- tion wrapper for the WDB so that the new SRRs retrieved from the same WDB can be annotated using this wrapper quickly without reapplying the entire annotation process. Each annotated group of data units corresponds to an attribute in the SRRs. The annotation wrapper is a descrip- tion of the annotation rules for all the attributes on the result page. To use the wrapper to annotate a new result page, for each data unit in an SRR, the

Volume 3 Issue V, May 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

annotation rules are applied on it one by one based on the order they appear in the wrapper. If this data unit has the same prefix and suffix as specified in the rule, the rule is matched and the unit is labeled with the given label in the rule.

VII. EXPERIMENTS

A. Data Sets and Performance Measure

Two testing data sets DS2 and DS3 are generated by collecting two sample result pages from each testing site using different queries. We note that we largely recollected the result pages from WDBs used in our previous study. We use a genetic algorithm based method [10] to obtain the best combination of feature weights and clustering threshold T that leads to the best performance over the training data set.

DS2 is used to test the performance of our alignment and annotation methods based on the parameter values and statistics obtained from DS1. A data unit is said to be correctly annotated if its system-assigned label has the same meaning as its manually assigned label.

B. Experimental Results

The optimal feature weights obtained through our genetic training method (See Section 4) over DS1 are {0.64, 0.81, 1.0, 0.48, 0.56} for SimC, SimP, SimD, SimT, and SimA, respectively, and 0.59 for clustering threshold T. The average alignment precision and recall are converged at about 97 percent. The learning result shows that the data type and the presentation style are the most important features in our alignment method. Then, we apply our annotation method on DS1 to determine the success rate of each annotator.

The performance is consistent with that obtained over the training set. The errors usually happen in the following cases. First, some composite text nodes failed to be split into correct data units when no explicit separators can be identified. For example, the data units in some composite text nodes are separated by blank spaces created by consecutive HTML entities like " " or some format- ting HTML tags such as <SPAN>. Second, the data units of the same attribute across different SRRs may sometimes vary a lot in terms of appearance or layout. For example, the promotion price information often has color or font type different from that for the regular price information. Note that in this case, such two price data units have low similarity on content, presentation style, and the tag path.

We can see that the overall precision and recall are very high, which shows that our annotation method is very effective. We also found that in a few cases some texts are not assigned labels by any of our basic annotators. One reason is that some texts are for cosmetic or navigating purposes. These texts do not represent any attributes of the real-world entity and they are not the labels of any data unit, which belong to our One-To-Nothing relationship type. It is also possible that some of these texts are indeed data units but none of our current basic annotators are applicable to them.

The reason is that wrapper based approach directly extracts all data units specified by the tag path(s) for each attribute and assigns the label specified in the rule to those data units. In contrast, the nonwrapper-based approach needs to go through some time-consuming steps such as result page rendering, data unit similarity matrix computation, etc., for each result page.

We also conducted experiments to evaluate the significance of each feature on the performance of our alignment algorithm. For this purpose, we compare the performance when a feature is used with that when it is not used. This result is consistent with our training result where the data type and the presentation style have the highest feature weights. The adjacency and tag path feature are less significant comparatively, but without either of them, the precision and recall drop more than 15 percentage points.

VIII. CONCLUSION

In this paper, we studied the data annotation problem and proposed a multiannotator approach to automatically con- structing 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. A special feature of our method is that, when annotating the results retrieved from a web database, it utilizes both the LIS of the web database and the IIS of multiple web databases in the same domain. We also explained how the use of the IIS can help alleviate the local interface schema inadequacy problem and the inconsistent label problem. Accurate

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

alignment is critical to achiev- ing holistic and accurate annotation. Our method is a clustering based shifting method utilizing richer yet auto- matically obtainable features. This method is capable of handling a variety of relationships between HTML text nodes and data units, including one-to-one, one-to-many, many-to-one, and one-to-nothing. Our experimental results show that the precision and recall of this method are both above 98 percent. For example, we need to enhance our method to split composite text node when there are no explicit separators. We would also like 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.

REFERENCES

[1] A. Arasu and H. Garcia-Molina, "Extracting Structured Data frombWeb Pages," Proc. SIGMOD Int'l Conf. Management of Data, 2003. [2] L. Arlotta, V. Crescenzi, G. Mecca, and P. Merialdo, "Automatic Annotation of Data Extracted from Large Web Sites," Proc. Sixth Int'l Workshop the Web and Databases (WebDB), 2003.

[3] P. Chan and S. Stolfo, "Experiments on Multistrategy Learning by Meta-Learning," Proc. Second Int'l Conf. Information and Knowledge Management (CIKM), 1993.

[4]W. Bruce Croft, "Combining Approaches for Information Retrie- val," Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval, Kluwer Academic, 2000.

[5] V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRUNNER: Towards Automatic Data Extraction from Large Web Sites," Proc. Very Large Data Bases (VLDB) Conf., 2001.

[6] S. Dill et al., "SemTag and Seeker: Bootstrapping the Semantic Web via Automated Semantic Annotation," Proc. 12th Int'l Conf. World Wide Web (WWW) Conf., 2003.

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

[8] 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.

[9] D. Freitag, "Multistrategy Learning for Information Extraction," Proc. 15th Int'l Conf. Machine Learning (ICML), 1998.

[10] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley, 1989.

[11] S. Handschuh, S. Staab, and R. Volz, "On Deep Annotation," Proc. 12th Int'l Conf. World Wide Web (WWW), 2003.

[12] S. Handschuh and S. Staab, "Authoring and Annotation of Web Pages in CREAM," Proc. 11th Int'l Conf. World Wide Web (WWW), 2003.

[13] B. He and K. Chang, "Statistical Schema Matching Across Web Query Interfaces," Proc. SIGMOD Int'l Conf. Management of Data, 2003.

[14] H. He, W. Meng, C. Yu, and Z. Wu, "Automatic Integration of Web Search Interfaces with WISE-Integrator," VLDB J., vol. 13, no. , pp. 256-273, Sept. 2004.

[15] H. He, W. Meng, C. Yu, and Z. Wu, "Constructing Interface Schemas for Search Interfaces of Web Databases," Proc. Web Information Systems Eng. (WISE) Conf., 2005.

[16] J. Heflin and J. Hendler, "Searching the Web with SHOE," Proc. AAAI Workshop, 2000. [17] L. Kaufman and P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, 1990.

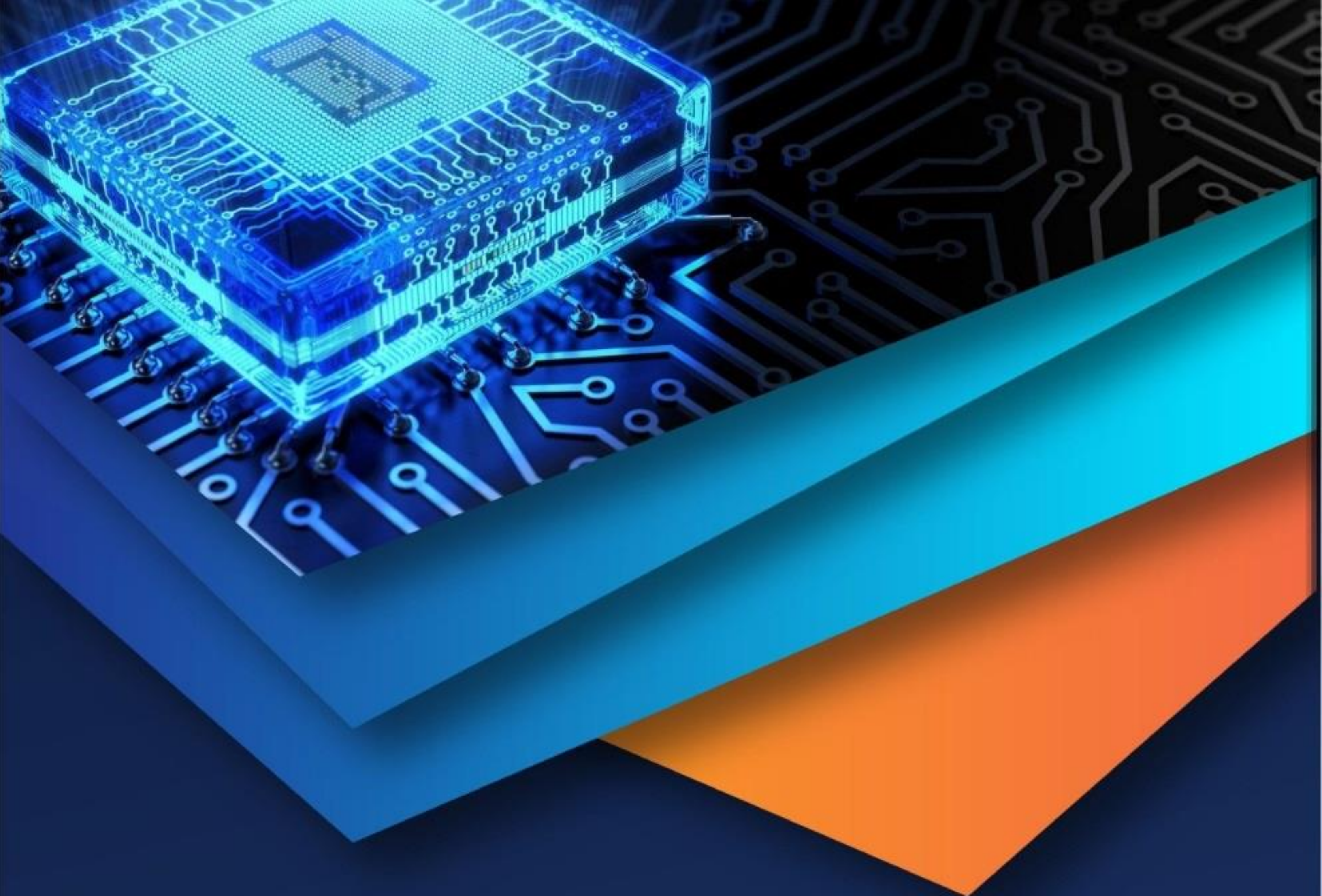
[18] N. Krushmerick, D. Weld, and R. Doorenbos, "Wrapper Induction for Information Extraction," Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI), 1997.

[19] J. Lee, "Analyses of Multiple Evidence Combination," Proc. 20th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, 1997.

[20] 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.

[21] 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.

[22] Y. Lu, H. He, H. Zhao, W. Meng, and C. Yu, "Annotating Structured Data of the Deep Web," Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE), 2007.



doi crossref
10.22214/IJRASET

INDEX COPERNICUS
45.98

JISRAIF
IMPACT FACTOR: 7.129

SJIFactor Scientific Journal Impact Factor
TOGETHER WE REACH THE GOAL
IMPACT FACTOR: 7.429

ISI

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY