Record Matching Over Query Results Using Fuzzy Ontological Document Clustering

V.Vijayaraja
Computer Science and Engineering
Jaya Engineering College
Chennai, India

R.Prasanna Kumar
Research Scholar
Anna University of Technology
Coimbatore, India

M.A.Mukunthan
Computer Science and Engineering
Jaya Engineering College
Chennai, India

G.Bharathi Mohan
Research Scholar
Anna University of Technology
Chennai, India

***Abstract*—Record matching is an essential step in duplicate detection as it identifies records representing same real-world entity. Supervised record matching methods require users to provide training data and therefore cannot be applied for web databases where query results are generated on-the-fly. To overcome the problem, a new record matching method named Unsupervised Duplicate Elimination (UDE) is proposed for identifying and eliminating duplicates among records in dynamic query results. The idea of this paper is to adjust the weights of record fields in calculating similarities among records. Two classifiers namely weight component similarity summing classifier, support vector machine classifier are iteratively employed with UDE where the first classifier utilizes the weights set to match records from different data sources. With the matched records as positive dataset and non duplicate records as negative set, the second classifier identifies new duplicates. Then, a new methodology to automatically interpret and cluster knowledge documents using an ontology schema is presented. Moreover, a fuzzy logic control approach is used to match suitable document cluster(s) for given patents based on their derived ontological semantic webs. Thus, this paper takes advantage of similarity among records from web databases and solves the online duplicate detection problem.**

Keywords- Ontology schema,record matching, query results, SVM, UDE, duplicate detection

I. INTRODUCTION

In the field of patent knowledge management, patent clustering plays a critical role to help define future research and development directions. However, current research on patent clustering depends on statistical methodologies which use keywords and phrases that do not adequately represent the knowledge contained in the patent documents. To provide a better solution to patent knowledge clustering, this correspondence adopts the technique of ontological knowledge representation and fuzzy logic control. Ontological knowledge representation enables domain experts to define knowledge in a consistent way and to improve the efficiency of knowledge interchange using a standard format (such as XML, resource description framework (RDF), or OWL). Fuzzy logic is then used on the linguistic expressions to derive the similarity measures among patent

documents for clustering. With the support of these two techniques, a deeper knowledge of a patent's meaning can be derived and the similarity among patents can be reliably defined.

Record matching can be done by supervised learning where training dataset is required beforehand. In the web databases, the result records are obtained through online queries. They are query dependant and thus, supervised learning is inappropriate. The representative training set in supervised learning cannot be applicable for the web results that are generated on-the-fly. Hence, we define a unsupervised technique named Unsupervised Duplicate Elimination (UDE) which uses three classifiers for record matching and duplicate detection. This eliminates the user preference problem in supervised learning. The UDE method is based on adjusting the weights set for the records fields. It does not employ any labeled training examples. The record matching is initiated by forming a universal data containing record pairs from different data sources. This dataset with no redundancy is considered as negative set. Based on the dissimilarity among these records, field's weight is set and record matching is done by the first classifier. These results i.e., the matched records form the duplicate or positive set. The second classifier uses both duplicate and the non duplicate sets to identify the duplicate record pairs. Then, the fuzzy ontological knowledge document clustering method is proposed.

II. RECORD MATCHING OVER QUERY RESULTS

A. *First Problem Definition*

Our focus is to find the matching status among the records and to retain the non duplicate records. Then, the goal is to cluster the matched records using fuzzy ontological document clustering.

B. *Element Identification*

Supervised learning methods use only some of the fields in a record for identification. This is the reason for query results obtained using supervised learning to contain duplicate records. Unsupervised Duplicate Elimination (UDE) does not suffer from these types of user reference problems. A preprocessing step called exact matching is used for matching relevant records. It requires the data format of the records to be the same. So, the exact matching method is applicable only for the records from the same data source. Element identification thus merges the records that are exactly the same in relevant matching fields.

C. *Ontology matching*

The term *Ontology* is derived from the Greek words '*onto'* which means *being* and '*logia'* which means *written* or *spoken disclosure.* In short, it refers to a specification of a conceptualization.

Ontology basically refers to the set of concepts such as things, events and relations that are specified in some way in order to create an agreed-upon vocabulary for exchanging information. Ontologies can be represented in textual or graphical formats. Usually, graphical formats are preferred for easy understandability. Ontologies with large knowledge bases[5] can be represented in different forms such as hierarchical trees, expandable hierarchical trees, hyperbolic trees, etc. In the expandable hierarchical tree format, the user has the freedom to expand only the node of interest and leave the rest in a collapsed state [2]. If necessary, the entire tree can be expanded to get the complete knowledge base. This type of format can be used only when there are a large number of hierarchical relationships. Ontology matching is used for finding the matching status of the record pairs by matching the record attributes.

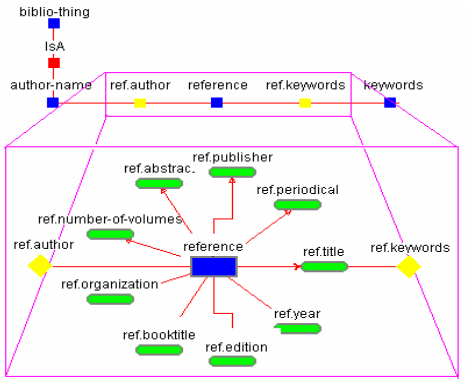

Figure1. An abstract view of ontology and zoomed-in view of an entity showing its attributes.

III. SYSTEM METHODOLOGY

A. Unsupervised Duplicate Elimination

UDE employs a similarity function to find field similarity. We use similarity vector to represent a pair of records.

Input: Potential duplicate vector set P

Non-duplicate vector set N

Output: Duplicate vector set D

C_1 : a classification algorithm with adjustable parameters W that identifies duplicate vector pairs from P

C_2 : a supervised classifier, SVM

Algorithm:

1. D = φ
2. Set the parameters W of C_1 according to N
3. Use C_1 to get a set of duplicate vector pairs d_1 and f from P and N
4. P = P- d_1
5. while | d_1 |≠ 0
6. N' = N - f
7. D = D + d_1 + f
8. Train C_2 using D and N'
9. Classify P using C_2 and get a set of newly identified duplicate vector pairs d_2
10. P = P - d_2
11. D =D + d_2
12. Adjust the parameters W of C_1 according to N' and D
13. Use C_1 to get a new set of duplicate vector pairs d_1 and f from P and N
14. N=N'
15. Return D

Figure 2. UDE Algorithm

The similarity vector V_i is denoted as 1(i.e. V_i=1) if the i*th* fields of the two records in the record pair are equal. If the i*th* fields of the two records are different, the similarity vector, V_i=0.Initially, two vector sets namely non duplicate vector set N and potential duplicate vector set P are built. UDE classifies the result data into two sets. The similarity vector set formed by duplicate record pairs is referred as duplicate vector set or positive set. The similarity vector set formed by non duplicate record pairs is referred to as non duplicate vector set or negative set.

The two classifiers C_1 and C_2 in the above algorithm refer to Weighted Component Similarity Summing (WCSS) classifier and Support Vector Machine (SVM) classifier respectively [3]. These two classifiers are iteratively used. Then, One-class SVM (OSVM) is employed to the result dataset. The assumptions that are used in UDE algorithm are

- The weights used are adjusted dynamically.
- A similarity function is implemented.
- Wrapper generation is done for extracting the result and inserting them into relational database according to the global schema.
- A global schema is defined.
- The records from the same data source have the same format.

Database schema matching is very essential step in data integration. It is the process of finding mappings between attributes of two schemas that semantically correspond to each other [6]. In UDE, a global schema for specific type of records is predefined and each database's individual query result schema has been matched to the global schema

1) Weight component similarity summing classifier

This classifier is used to identify some duplicate vectors when there are no positive examples available. An intuitive method to identify duplicate vectors is to assume that two records are same if most of the fields under consideration are similar. If the corresponding fields of the two records are dissimilar, then the two records are assumed to be non duplicates [1].

The similarity between two records will be in (0, 1).The similarity between two duplicate records should be close to 1.The similarity for two non duplicate records should be close to 0.The similarity threshold should be calculated for all the records[4].The sum of all component weights is equal to 1.The weight is assigned such a way to indicate the importance of the component fields. The component similarity value is given as

$$p_i = \sum_{v \in D} v_i$$

Where p_i is the accumulated i^{th} component similarity value for all duplicate vectors in D.

2) Support vector machine classifier

The second classifier used in UDE should be insensitive to the relative size of the positive and negative examples because the size of the negative examples is usually much bigger than the size of the positive examples. The classifier should work well given limited training examples. Support vector machine classifier is the classifier that satisfies the requirements mentioned previously. So, we have implemented SVM as C_2 in UDE.

B. Fuzzy Ontological Document Clustering

The methodology for fuzzy ontological document clustering (FODC) is described as follows [7]. Initially, domain experts define the domain ontology using a knowledge ontology building and RDF editing tool called Protégé, and the words and phrases (e.g., speech, chunks, and lemmas) of the patent documents are mapped to the corresponding domain ontology concepts. The experts also create a training set of patents using a free and easy-to-use natural language processing and tagging tool called MontyLingua. Afterward, the probabilities of the concepts in given document chunks are computed. The concept probabilities calculated in any given patent document are then used for clustering the patents with fuzzy logic inferences. Hence, the hierarchical clustering algorithm is refined by adapting fuzzy logic to the process of ontological concept derivation. The detailed FODC method is described step-by-step in the following sections.

1) Building a Patent Ontology

The first step of the FODC methodology requires the use of a knowledge-based RDF editing tool called Protégé. The tool assists the domain experts in defining an ontology schema using a graphical interface. Protégé is a free open-source ontology editor and a knowledge acquisition system. Similar to Eclipse, Protégé is a framework on which various other software plug-ins can easily be added and linked. Protégé is considered a suitable computer-aided tool for developing the ontology. The ontological web can be automatically transformed into standard data formats (XML, RDF, or OWL) for further manipulation and interpretation for knowledge analysis and synthesis.

2) Natural Language Processing and Terminology Training

In order to measure the knowledge contained in patent documents with respect to the defined ontology schema, the system is trained using a set of patent documents. The sentences from the training documents are tagged to extract the parts of speech, chunks, and lemmas using the MontyLingua natural language processing tool. The definitions for the parts of speech [8] are listed in Table II.

TABLE 1. PARTS OF SPEECH USING THE PENN TREE BANK TAG SET

Pos tag	Description
NN	Noun, singular or mass
NNS	Noun, plural
NP	Proper noun, singular
NPS	Proper noun, plural

Afterward, knowledge engineers map the extracted words to the concepts of the ontology. By using the example sentence "A chemical mechanical polishing apparatus and method for polishing semiconductor wafers. . .," the phrase chemical mechanical polishing apparatus and method maps to the concept CMP_method (n.), polishing represents the concept polish (v.), and semiconductor wafers represents the concept substrate (n.) in the ontology schema. The system records the probabilities of the concepts that a word implies in the patent.

The conditional probability, P(The patent concept | The word W in chunk C of the corpora), is derived during the training session. For example, we have ten training patents that contain the word polishing, and the chunk of polishing is NX in these data. To map polishing to the ontological concept, consider that the CMP_method concept is referred to in five patents, and the polish_pad concept is referred to in another five patents. Thus, P (The concept is CMP_method | The word polishing is in the NX corpora chunk) = 0.5 and P (The concept is polish_pad | The word polishing is in the NX corpora chunk) = 0.5.

To maintain the completeness of the FODC system, the research also includes an iterative relearning mechanism to include new words that are not part of the current terminology database. When a new term is detected, it is first stored in the terminology database. Afterward, the system manager assigns a corresponding ontological concept to this term to enable the system to automatically recalculate and update the terminology-ontological concept knowledgebase.

3) Terminology Analyzer

After natural language processing and terminology training, all of the sentence concepts are inferred. Hence, the probabilities of the concepts for each chunk are computed. Then, the probabilities of deriving concepts are derived.

4) Knowledge Extraction

After analyzing the terminology, we compute the concept probabilities for each chunk. The chunks implying concepts as predicates are the first to enter into the ontology. The next step is to select chunks that imply the concepts as the subject in the ontology from the previous sentence to the next sentence.The same process is used to determine the object candidates. If there are ten candidates for subject, two candidates for predicate, and ten candidates for object, then there are 200 ($10 * 2 * 10$) candidates for the statement. Statements that do not exist in ontology are eliminated. Finally, the output is generated using the probability derived from the following equation:

$$\text{Max}(\text{ for all statements based on chunk 5}) \times [\text{prob(subject)} + \text{prob(predicate)} + \text{prob(object)}]/3$$

The process described earlier is used for chunks that imply the concepts of the predicate in the document ontology. Thus, a document is transformed into a set of statements in the ontology. These statements are viewed as indices of the document and are the basis of similarity comparisons with other documents.

5) Patent Similarity Match

In order to compute the similarity between patent documents, fuzzy logic is used to derive the similarity measure. First, the contents of patent documents are partitioned into the set of main concepts and the set of details. Before input to the inference model, the patent documents are translated into an ontological format including main concepts and details. The main concepts consist of higher triples, and the details consist of the lower triples

$$X = ST/TT$$

Where

X similarity measure of document 1 and document 2;
ST the same triples in document 1 and document 2;
TT sum of triples in document 1 and document 2.

The Mamdani fuzzy inference model applies legacy if–else rules to fuzzify the input and output. The ease of formulating the model, the simple calculation, and the clarity in presenting human linguistics support the selection of this approach. Thus, the Mamdani fuzzy inference model using a min–min–max operation considering two rules is adopted and modified. The original Mamdani min–min–max operation only considers a two-rule approach, but this correspondence considers nine rules simultaneously. The steps for the procedure are as follows.

- Calculate the similarity of the documents matched in main concepts (*X*mc) and the similarity of the documents matched in detailed descriptions (*X*dd).
- Evaluate *X*mc and *X*dd using the rules to derive the corresponding memberships.
- Compare the memberships and select the minimum membership from these two sets to represent the membership of the corresponding concept (high similarity, medium similarity, and low similarity) for each rule.
- Collect memberships which represent the same concept in one set.
- Derive the maximum membership for each set, and compute the final inference result.

C. Evaluation Metric

The overall performance can be found using precision and recall where

$$\text{Precision} = \frac{\text{Number of correctly identified duplicate pairs}}{\text{Number of all identified duplicate pairs}}$$

$$\text{Recall} = \frac{\text{Number of correctly identified duplicate pairs}}{\text{Number of true duplicate pairs}}$$

The classification quality is evaluated using F-measure which is the harmonic mean of precision and recall

$$\text{F-measure} = \frac{2(\text{precision})(\text{recall})}{\text{Precision} + \text{recall}}$$

IV. RESULT

Ontology matching helps in finding the relevant records based on the user queries by considering all the record attributes information. Thus, the exact matching and the ontology matching are employed for merging the relevant record pairs. The exact matching and ontology matching is found to reduce the duplicates by 82% when investigating 50 websites randomly (Table 2).

TABLE 2.DUPLICATE REDUCTION USING EXACT MATCHING AND ONTOLOGY MATCHING

Domain	Number of user specified fields	Number of websites	Duplicate Ratio	Duplicate pair reduction ratio
Audio record	3	50	2.5%	87%
Electronic store	3	50	4.2%	85%
Restaurant	3	50	3.5%	79%
Shopping mall	2	50	6.4%	76%
Book	2	50	7.2%	87%
Average	-	50	4.8%	82.8%

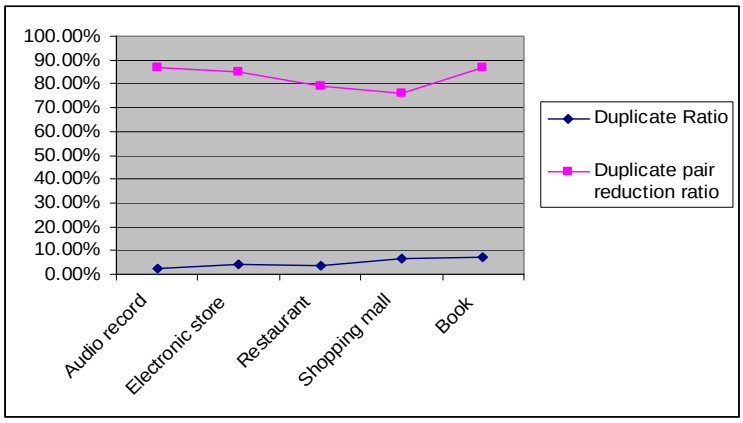

Figure 3. Graph Showing Duplicate Reduction

V. CONCLUSION

The essential steps in data integration are record matching, duplicate detection and clustering. An unsupervised online approach called Unsupervised Duplicate Elimination (UDE) is presented which uses two classifiers namely WCSS, SVM to find the duplicate records. UDE does not require any pre-labeled training examples. It is well suited for online record matching.A phrase can represent many meanings, and many different phrases can represent the same meanings. In this correspondence, we analyze the grammar of the sentences and derive the ontology of documents. Then, the relationships between documents are inferred, and the document similarities and differences are compared. A fuzzy ontology-based methodology for clustering knowledge documents (the FODC methodology) is presented which outperforms the K-means clustering approach.

REFERENCES

[1] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani, "Robust and Efficient Fuzzy Match for Online Data Cleaning," Proc. ACM SIGMOD, pp. 313-324, 2003.

[2] Kuhanandha Mahalingam and Michael N.Huhns, "Representing and using Ontologies",USC-CIT Technical Report 98-01.

[3] Weifeng Su, Jiying Wang, and Federick H.Lochovsky, " Record Matching over Query Results from Multiple Web Databases" IEEE transactions on Knowledge and Data Engineering, vol. 22, N0.4,2010.

[4] R. Ananthakrishna, S. Chaudhuri, and V. Ganti, "Eliminating Fuzzy Duplicates in Data Warehouses. VLDB", pages 586-597, 2002.

[5] Tetlow.P,Pan.J,Oberle.D,Wallace.E,Uschold.M,Kendall.E,"Ontology Driven Architectures and Potential Uses of the Semantic Web in Software Engineering",W3C,Semantic Web Best Practices and Deployment Working Group,Draft(2006).

[6] Ji-Rong Wen, Fred Lochovsky, Wei-Ying Ma, **"**Instance-based Schema Matching for Web Databases by Domain-specific Query Probing", Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

[7] Amy J.C.Trappey, Charles V.Trappey, Fu-Chiang Hsu,and David W.Hsiao, "A Fuzzy Ontological Knowledge Document Clustering Methodology",IEEE Transactions on Systems,Man,and Cybernetics-Part B:Cybernetics,Vol.39,No.3,june 2009.

[8] M. Marcus, B. Santorini, and M. Marcinkiewicz, "Building a large annotated corpus of english: The penn treeBank," *J. Comput. Linguist.*, vol. 19,no. 1, pp. 313–330, Jun. 1993.

AUTHORS PROFILE

V.Vijayaraja is a Assistant Professor, Department of Computer Science, Jaya Engineering College, Tamilnadu, India. He got his B.E. Degree in Electronics and Instrumentation Engineering from Annamalai University and M.Tech Degree in Computer science and Engineering from Dr.M.G.R University. He has about 14 years of teaching experience and 2 years research experience in the field of Wireless sensor networks. He is the life member of I.S.T.E.

R.Prasanna Kumar is a Research Scholar, Department of Computer Science, Anna University of Technology, Coimbatore, Tamilnadu, India. He got his B.E. Degree in Computer science and Engineering from Madras University and M.Tech Degree in Computer science and Engineering from Dr.M.G.R University. He has about 8 years of teaching experience and 2 years research experience in the area of Data Mining.

M.A Mukunthan is a Assistant Professor, Department of Computer Science, Jaya Engineering College, Tamilnadu, India. He got his B.E. Degree in Computer science and Engineering from Madras University and M.E Degree in Computer science and Engineering from Anna University. He has about 8 years of teaching experience and 2 years Industrial experience.

G.Bharathi Mohan is a Research Scholar, Department of Computer Science, Anna University of Technology, Chennai, Tamilnadu, India. He got his B.E. Degree in Electrical and Electronics Engineering from Madras University and M.Tech Degree in Information Technology from Anna University. He has about 5 years of teaching experience.