ONTOLOGY BASED DATA INTEGRATION TO IMPROVE DATA QUALITY WITH CACHE

[1]B. JAIBARATHI, [2]L. SOWMYA DEVI, [3]M.S.HEMA

[1,3]Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India
[2]Kumaraguru College of Technology, Coimbatore, Tamil Nadu, India
E-mail: [1]mail2jaibarathi@gmail.com, [2]sowmyalogaswamy@gmail.com, [3]hema.ms.cse@kct.ac.in

Abstract- In today's world the amount of data is increasing tremendously. In order to analyze data and make decisions, data residing at different sources are integrated. Data integration is the process of integrating data from different data sources. Data federation is a data integration strategy used to create integrated virtual view. The data integration process involves schema matching, duplicate detection and data fusion. The semantic heterogeneity is resolved using ontology. The data conflicts that occur during the data integration are resolved using the Enhanced Markov Logic Network (EMLN) to improve the quality of the data . To improve the performance of system cache is implemented. Enhanced LRU with Frequency (ELRLF) algorithm is used for page replacement in cache. This cache technique used to reduce number of times scanning of local ontology. Virtual table is created to populate the result of integration service. A new cache optimization algorithm, Enhanced LRU with frequency is used to improve the response time and recall rate. Enhanced LRU with frequency uses the hash map with skip list data structure to perform efficient searching of data item in cache. Ontology based data integration using cache support decision making for disaster management application.

Keywords- Data integration, Ontology, Semantic heterogeneity, Data quality, Cache optimization.

I. INTRODUCTION

In order to remain competitive, the integration of data source is an important for many large organizations. Applications that originally have been developed independently are now required to interoperate to support new or different functions of the enterprise. Integrating and querying data from heterogeneous sources is a hot research topic in database research field. The main advantage of data integration is to provide a uniform interface to distributed and heterogeneous sources. There are different techniques used for integrating data sources. The goal of data integration is to gather data from different sources combine it and present it as unified view. There are there important data integration processes are i) Data consolidation ii) Data propagation and iii) Data Federation. One of the most important problems with data integration is the semantic heterogeneity, which analyzes the meaning of attributes. This semantic heterogeneity can be overcome using ontology based data integration.

The process of data integration involves three steps is shown in figure 1. The three steps are schema matching, duplicate detection and data fusion. Schema matching is done in concern with resolution of schematic and semantic conflicts. To resolve the semantic conflicts richer semantics are provided and that is done by ontology. Ontology is a formal explicit specification of the shared conceptualization. Conceptualization refers to an abstract model of a domain which identifies the relevant concepts and their relationship. Explicit means that the used concepts are unique and their usage is formally confined. Formal specification denotes machine readability with computational semantics. Shared indicated that an ontology is indicated by a group of people and used corporately. In data federation to resolve the semantic heterogeneities ontology is used. The two components of ontology in data federation are names for important concepts in a domain and back ground knowledge or constraints on domains such as attributes, classification and constraints. In, Duplicate detection; object level conflicts are resolved using the duplicate detection technique. Two records with the same real world object are detected as duplicates. Data resolution refers to the task of finding records in a data set that refer to the same entity across different data sources. Entity resolution consists of two steps are Comparison to binary relation and field comparison. In comparison to binary relation, the resultant records from different data sources that are to be reduplicated are converted to binary relations. Let consider the following example of Registration database in disaster management application.

Registration (name, age, address)

The binary relation of the Registration database is as follows:

HasName (person, name)

HasAge (person, age)

HasAddress(person, address)

In Field comparison, the next process is the field comparison where each field in a data base to be reduplicated is a string composed of one or more words. The predicate HasWord(field, word) which is defined as true when field contains word. Applied to this predicate, reverse predicate equivalence states that are

$$\forall x1, x2, y1, y2\ HasWord(x1, y1) \wedge HasWord(x2, y2)$$

$$\wedge \quad y1 = y2 \Rightarrow x1 = x2$$

It means that fields that have a word in common are more likely to be the same. Data fusion is the process of combining the semantically equivalent data objects coming from different sources. The data fusion process is performed to determine consistent representations. Suppose if data sources report the conflicting values for the same data item, the fusion uses the mapping rules and heuristics to remove the data conflicts. The integration process at the data fusion level may be very difficult to identify data objects or to decide which data value is correct.

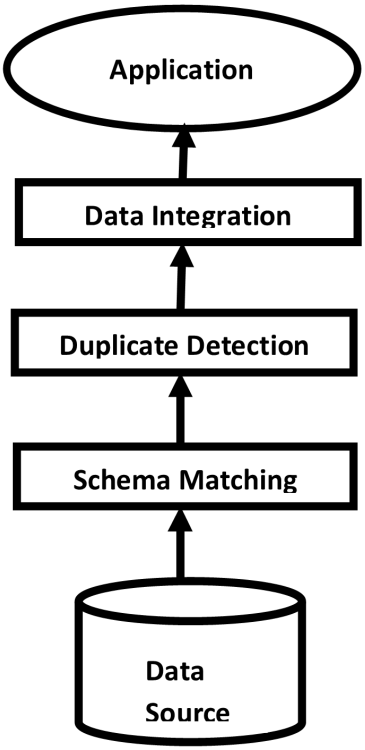

Figure.1. Data Integration Process

MLN is first order logic framework for combining description logic with statistical analysis. The important process involves in feature extraction and rule setting. Features are extracted from dataset using four important aspects are source, entities, entity attributes, facts and relationship. Enhanced MLN handles uncertainty and tolerate certain imperfection in data.

In this paper, an ontology based data integration to improve quality with cache deal with data quality along with cache optimization. When intergration more semantic features sematic conflict and duplicte record may occure; which are resolved using Markov Logic Network . Cache optimization part uses the Enhanced LRLF is used to improve the speed of data access. Enhanced LRLF techniques uses hash map with skip list data structure which stores both recency and frequency value for data prediction.

Cache replacement palys very important role in cache optimization. Cache replacement policies are classified based on some factors are given in table 1. To perform efficient searching differentt data structures are used in cache replacemnt algorithms.

Table 1 Classification of cache page replacement algorithms

S.No	Types	Description	Algorithm
1	Recency based	Recently used objects are considered	LRU
2	Frequency based	Based on Frequency count Objects are replaced	LFU
3	Size based	Size of the objects are considered	SIZE
4	Function based	Each object in cache has utility value	GD-SIZE

There are differernt types of data structure used to represent the nodes. The selection of data structure is based on cache characteristic . Most commonly used data structures are

1. Partitioned Array
2. B-tree
3. Hash table
4. Queue

Hash table is also called as Hash map which is data structure used to implement array. Hash key plays very important role in indexing. Hash table performance is based on the efficient has function. Care should be taken when considering hash function. Hash values are uniformly distributed over the hash table. Collisions can be avoided by using perfect hash function .The running time complexity and searching of hash is O (1).

There are some of the real time existing systems which have similar characteristic to this project are

A) KRAFT (Knowledge Reuse and Fusion / Transformation)

KRAFT is multisite research project conducted at University of Aberdeen, Cardiff and Liverpool in collaboration with BT in UK. It is just extension of Carnot Project to make legacy database system. The main objective is locating and extracting knowledge from multiple heterogeneous online sources and transform into union of knowledge. Local ontologies

allow the communication between heterogeneous resources that can maintain the intrinsic heterogeneity. It follow the two methods for execution there are building of shared ontologies and extracting of source ontologies It also detects set of ontology mismatch and establishing mapping between the shared ontology.

B) COIN (Context Interchange)

COIN was initiated in 1991 with goal of achieving semantic interoperability among heterogeneous information source. This framework uses temporal contextual knowledge representation and reasoning capability to allow retrieval of data from multiple sources. Ontology used as formal knowledge for representing context knowledge and a mediation service to dynamically detect and reconcile semantic conflict. It offer more flexible and scaling when compare to other system.

II. RELATED WORK

Many recent researches on ontology based data integration deal with handling of semantic heterogenity problem. Many approaches had been proposed to improve the heterogenity problem. Gagnon M [2] gave newer approach that offer new solution to interoperability. In that, deal with mapping between local and global ontology but failed to detect duplicate records and error; which are not considered during the integration process.

XL Dong [3] discussed data integration systems to resolve semantic conflicts. Data fusion plays important role in data integration which helps to detect and remove dirty data and increase accuracy of data. The problems of data fusion systems are openly discussed. Advance techniques like correctness of source, freshness of source and dependencies between sources are used to resolve semantic conflicts.

Qing-zhong, L [4] proposed Markov Logic Network for resolving data conflict; make use of multi angel feature and knowledge .MLN hands tolerant imperfect, uncertainty among data and knowledge contradiction. The steps involved in MLN are feature extraction, rule setting and MLN training.

Zheng H [5] proposed agent and ontology oriented knowledge base for ontology based semantic cache. To overcome the performance problem semantic cache is used. Semantic cache keeps the result of previously asked queries. Cache implementation makes use of Least Recently Used (LRU) algorithm to perform cache optimization

Mengdong Yang [6] proposed efficient approach to debug ontologies based on set pattern. Ontology debugging helps to understand the unsatisfiable concept in ontology by finding minimal unsatisfiability preserving sub ontologies (MUPS).

Kavar [7] discussed different page replacement in memory management. There are many page replacement algorithms such as Least Recently Used, Least Frequently Used, Most Recently used, FIFO and Non Recently Used. Each algorithm perform following operations

A) Search
B) Delete
C) Insert

Mainly focuses on performance of LRU algorithm with different data structure. LRU combine with different data structures like double linked list, splay tree, skip list. LRU perform efficiently with skip list. Skip list consist of page address, frequency value and two pointers. Insertion is done in head position and deletion is done in tail position.

Abdelfattah [8] proposed LR+5LF algorithm which combine the concept of Least Recently Used with Frequency. Development process contains three steps are

A) LRU and LFU weight assigning
B) Joining LRU and LFU
C) Replacement of line

Many approaches separately handle semantic heterogeneity, semantic conflict and cache. MLN is proposed for resolving semantic heterogeneity. Ontology is used to provide efficient handling of semantic. Sources are ordered, features are extracted and under gone MLN training to improve the quality. Cache implementation makes use of LRU algorithm which store query of received input. Cache is mainly implemented to improve the access rate.

III. PROPOSED WORK

This paper proposes OBDI – QC architecture for data integration. This architecture consist of three layer are a) User Interface Layer b) Mediator layer c) Database layer are in figure 2. In data integration process each layers are mutually exclusive to each other

A) USER INTERFACE LAYER

User post query and get the required result through user Interface layer. User Interface layer consist of two modules i) Query Input, ii) Query Output. Query Input module accept query from user and validate it for processing to Mediator layer. Query Output module display the integrated quality data that corresponds to user query. First, user query pass on to cached result database of mediator layer.

If cache database contains the result of the posted query; then the results are communicated to user; else queries are send to Mediator layer for processing.

B) MEDITORE LAYER

Mediator layer consist of six modules are i) Local ontology and Global ontology are mapped using schema mapping, ii) Query processing, iii) source ordering in done to improve the correctness of data, iv) Duplicate detection to remove unwanted records; help to avoid redundancy problem, v) Markov Logic Network is used to resolve data conflict at the time of data integration vi) cache implementation to improve the recall rate and avoid unwanted scanning of local database.

A) Schema mapping: Ontology is used to provide unified view to different data sources and to resolve semantic heterogeneity among the data sources. Local ontology is created for each data sources. Global ontology provides global view of data sources. Schema mapping is done between local ontology and global ontology. Each local data source schemas are analyzed to construct local ontology. An ontology web language (OWL) is used to describe every ontologies. OWL formally describes the semantic of classes and properties used. The Local As View is constructed using Protégé which help to combine the concept and attributes with same meaning. LAV is used when GAV (Global As View) the global schema is described in terms of local schema.

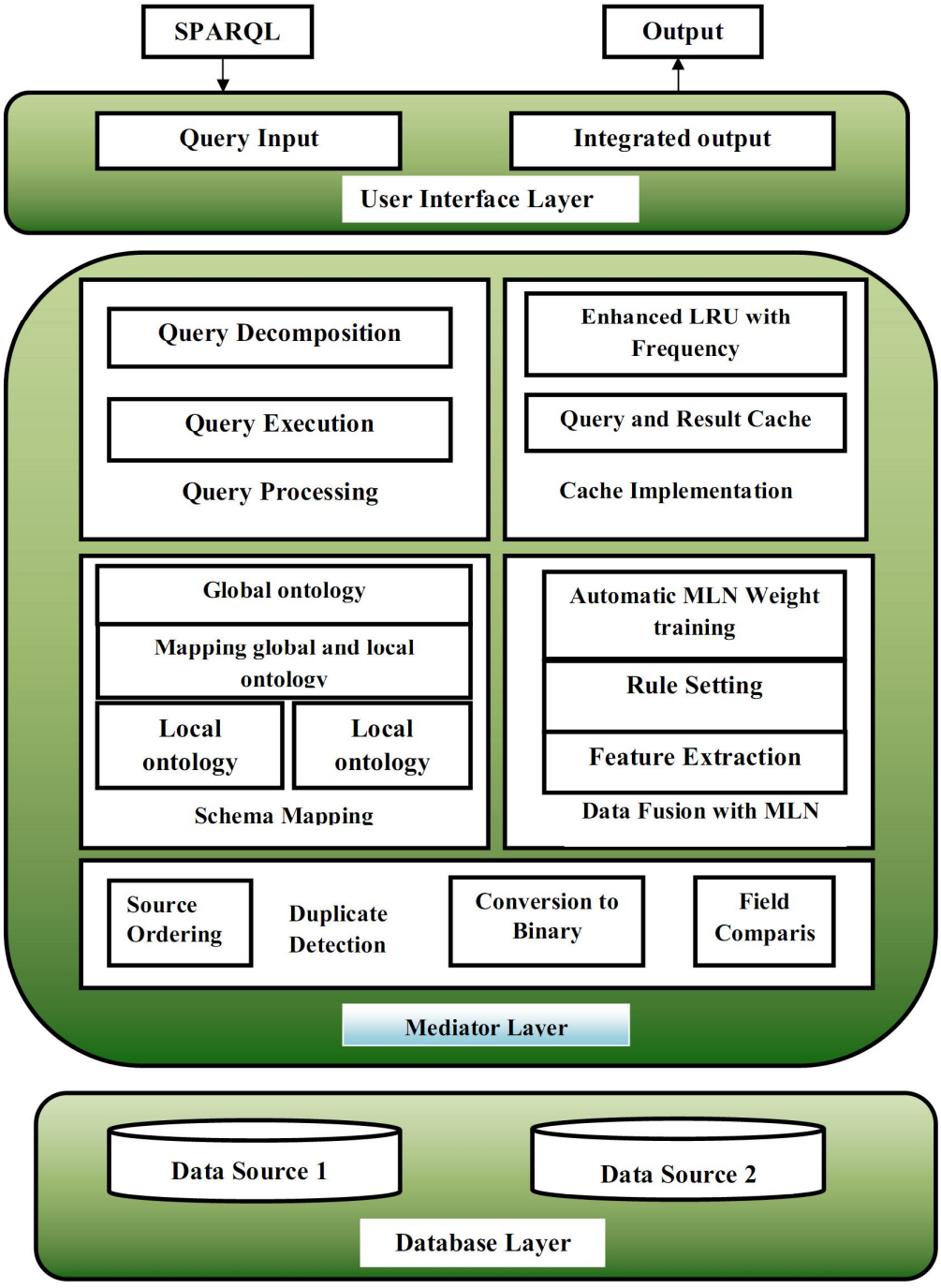

Figure.2. Proposed Architecture

New sources can be added easily in LAV. Query processing can be done easily in GAV. The main process involves in analyzing local ontology, global ontology conceiving and mapping between local ontology and global ontology by defining attributes. Semantically equivalent classes of different ontologies are combined to form global ontology with one class. A simple example is illustrated in the table 1 in which it gives the view of global ontology.

Global Schema	PID	Pname	PAge
Data Source 1	P_id	P_Name	P_age
Data Source 2	P_no	Person_Name	Age
Data Source 3	PRegno	Name	Person age
Data Source 4	PersonID	C_name	C_age

Table1. Mapping between local and global ontology

B) Source Ordering: Data sources must be ordered to resolve conflict between different data. The higher priority is given to the trust worthy data sources; that is the data source has higher accuracy. Simple diagram is illustrated in figure 3, in which data sources are given along with accuracy value. The accuracy value is analyzed on history of previously delivered data. In figure data source S3 gives more accurate result and treated as trust worthy data sources.

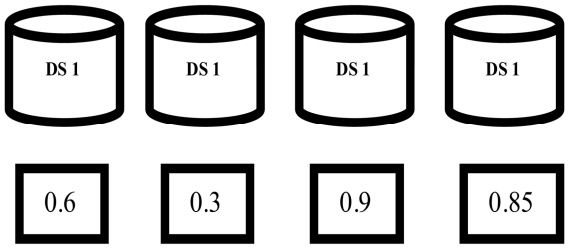

Figure 3: Database with true value

C) Query Processing: There are two modules in query processing are query decomposition and query execution. In query processing queries are decomposed and converted into database acceptable form.

These converted queries are executed in respective data sources using query execution module. The queries are decomposed using mapping rules. Global query Q is decomposed into Q_1, Q_2,….Q_n and send to respective data sources. User post query in SPARQL language which is; resource description framework language; specifically used query language for ontology based data integration. Global query on global ontology is decomposed and send to local ontology as a sub queries. The data sources that receive sub queries return a set of results R_1, R_2, …. R_n. The result of integrated data is shown in Equation1.

Result = R1 (DS1) U R_2 (DS2) U …….. R_n (DS

D) Duplicate Detection: Object level conflicts are resolved using the duplicate detection technique. Two records with the same real world object are detected as duplicates. Data resolution refers to the task of finding records in a data set that refer to the same entity across different data sources. Entity resolution consists of two steps comparison to binary relation and field comparison. In Comparison to binary relation; the resultant records from different data sources that are to be reduplicated is converted to binary relations. Consider the following example of Registration database in disaster management application.

Registration (name, age, address)

The binary relation of the Registration database is as follows:

HasName (person, name)

HasAge (person, age)

HasAddress(person, address)

Field comparison; this process involves in field comparison where each field in a data base to be reduplicated is a string composed of one or more words. The predicate HasWord(field, word) which is defined as true when field contains word. Applied to this predicate, reverse predicate equivalence states that

$$\forall x1, x2, y1, y2\ HasWord(x1, y1) \wedge HasWord(x2, y2) \wedge y1 = y2 \Rightarrow x1 = x2$$

It means that fields that have a word in common are more likely to be the same.

E) Data Integration: In previous layer duplicate records are detected which help to reduce the redundancy. Markov Logic Network (MLN) is used to improve the quality of data. MLN is first order logic framework for combining description logic with statistical analysis. The important process involves in feature extraction and rule setting. Features are extracted from dataset using four important aspects are source, entities, entity attributes, facts and relationship. The other important features are like interdependency, mutual dependency between sources and facts, mutual implication between facts are also considered. In rule setting; different rules are framed to know the true values which are given along with MLN.

Rule 1: First rule is voting rule; used to identify the true value of conflict facts. Usually, most frequently used fact given high priority and treated as a accurate entity attribute.

MaxFreq (ea,fact) $\Longrightarrow$ IsAccurate(fact)

Rule2: Second rule is trustworthy rule; it provides trustworthy data source which treated as accurate.

IsAccurate(fact)^Provide(source,fact)
⇒ IsTrustworthy(Source)

Rule 3: If two or more data sources having same facts for many entity attributes, then there exists mutual dependency.

InterDepend(Sorce1,Source2)^about (fact1,pa) ^ about(fact2, pa)^ provide(source1, fact1) ^ provide(source 2,fact2)
⇒ IsAccurate(fact1) ⇔ IsAccurate(fact2)

In MLN weight training process model is trained and automatically learn weight for each formula. MLN uses, voted perception algorithm; which is gradient descent algorithm that firm set all weights to zero. It will update the training set ; on each iteration and check whether the predicted value is matches with true value. The output contains only record with true values without duplication and data conflict.

F) Cache implementation: Cache implementation contains two modules are query, result cache and Enhanced LRU with frequency algorithm. Query and result cache is used to store query along with data in repository. Enhanced LRU with frequency algorithm uses hash map with skip list data structure for efficient store and search. In Enhanced LRU with frequency calculate both recency and frequency value of data.

When a new object enters it check least time and priority of frequency value. If the priority is greater than frequency value then look for the second smallest time stamp value and check for frequency again. In the same time; time stamp value also considered; if the data item time stamp value is greater than time stamp value; then remove the first least data item .This process is continued until the data find the location.

This new algorithm, Enhanced LRU with frequency uses hash map with skip list to improve the running time of search. This helps to increase processing speed and improved recall rate. The time taken to display the records to user will be reduced. Unwanted scanning of local data sources is also avoided. The structure of skip list node is given in the figure 4. Skip list node contain four sections that are used for different purpose.

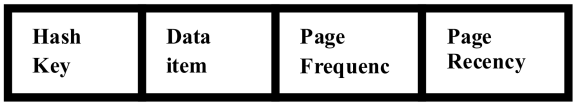

Figure 4 Skip List node Structure

Using query evaluation hash key is computed which is treated as recency value. Each page has its own recency value and frequency value. Data item associated with given query are cached. In page recency section stores data and time of last execution of query.

C) DATABASE LAYER

Data base layer consist of different data sources which are tend to form local ontology. In this project uses three different data sources are MySQL, Oracle and SQL server.

CONCLUSION

A method has been proposed to improve the data quality alon with cache optimization. Markov Logic Network is used to resolve semantic heterogenity and Enhanced LRU with Frequency is used for handling efficient cache replacement and search. Duplicate datas are removed and accurate data are found; which automatically improve the system quality. Cache implementation reduce the processing overhead and unwanted scanning of local ontology. On using cache in ontology based data integration results in improved recall rate. This project mainly concentrated on resolving semantic conflict . This work can be extended on considering syntax, description level conflict handling.

REFERENCES

[1] Zhao, Y., Zhang, S., & Yan, Z. (2009, June), "Ontology-based model for resolving the data-level and semantic-level conflicts", In Information and Automation, International Conference on 2009, (pp. 455-459), IEEE. (Zhao Y et al, 2009)

[2] Gagnon, M, "Ontology based integration of data sources", Information Fusion. In proceeding, 10th International conference, pp (1-8), E-ISBN: 978-0-662-45804-3, IEEE.

[3] Dong, X. L., & Naumann, F. (2009)," Data fusion: resolving data conflicts for integration", In Proceedings of the VLDB'09, 35th International Conference on Very Large Database Lyon, France, 2009, pp (1654-1655).

[4] Qing-zhong, L., Yong-xin, Z., & Li-zhen, C. "Data Conflict Resolution with Markov Logic Networks, 2011. (Qing L et al, 2011)

[5] Zheng, H., Lu, R., Jin, Z., & Hu, S. (2002), "Ontology-based semantic cache in AOKB", In Journal of Computer Science and Technology, 17(5), pp (657-664). (Zheng H et al, 2002)

[6] Yang, M., & Wu, G. (2012),"Semantic caching for semantic web applications", In The Semantic Web, series volume no: 7185, pp (192-209), 2012, Springer Berlin Heidelberg.

[7] Kavar, C. C., & Parmar, S. S. (2013, February) , "Performance Analysis of LRU Page Replacement Algorithm with Reference to different Data Structure", International Journal of Engineering Research and Applications (IJERA), Vol. 3, Issue 1, January - February 2013, pp (2070-2076).

[8] Abdelfattah, A., & Samra, A. A. (2012, January)," Least recently plus five least frequently replacement policy (LR+ 5LF)", In the international Arab journal of information technology, vol. 9, no. 1, January 2012, pp. (16-21).

[9] Rodrigo Fernandes Calhau, Ricardo de Almedia Falbo (2010),"An Ontology-based Aproach for Semantic Integration" , IEEE International Enterprise Distributed Object Computing Conference, EDOC 2010.

[10] Richard Y. Wang, Veda C. Storey, Christopher P. Firth (1995), 'A Framework for Analysis of Data Quality

Research', IEEE Transaction on Knowledge and Data Engineering, Vol. 7, No. 4, pp 623-639. (Richard Y et al, 1995)

[11] Laomo Zhang, Ying Ma, Guodong Wang (2009), 'An Extended Hybrid Ontology Approach to Data Integration', International Conference on Biomedical Engineering and Informatics, BMEI '09, pp 1-4. (Laomo Z et al, 2009)

[12] Nicolicin-Georgescu, V., Benatier, V., Lehn, R., & Briand, H. (2009, September), "An ontology-based autonomic system for improving data warehouses by cache allocation management",In FGWM 09 Workshop on Knowledge and Experience Management, 2009, (pp. 31-37). (Nicolicin G et al, 2009)

[13] D. Lee and W. W. Chu (1999), "Semantic Caching via Query Matching for Web Sources" , In Proc. of the 1999 ACM CIKM Int. Conf. on Information and Knowledge Management, Kansas City, Missouri, USA, pages 77–85. ACM, November 2–6 1999.(Lee D et al, 1999)

[14] D. Lee and W. W. Chu (Nov, 2001),"Towards Intelligent Semantic Caching for Web Sources", Journal of Intelligent Information Systems, 17(1):23–45, November 2001.

★★★