Multi-domain Schema Clustering and Hierarchical Mediated Schema Generation

Qizhen Huang, Chaoliang Zhong and Jun Zhang
Information Technology Lab, Fujitsu R&D Center Co., LTD.,
No.56 Dong Si Huan Zhong Rd, Chaoyang District, Beijing, P.R. China

Keywords: Data Integration, Hierarchical Mediated Schema, Schema Clustering.

Abstract: In data integration, users can access multiple data sources through a uniform interface. Yet it is still not easy to query from data sources where many domains coexist even if the data sources are clustered into several domains since users have to write different query clauses for each different domain. Previous researches have presented various data integration techniques, but nearly all of them require the schemas of data sources to be integrated belong to the same domain, or failed to address that some different domains may be the sub-domains of a high level domain in which case a more abstract query clause for upper domain can substitute several less abstract query clauses for lower domains. In this paper, we propose a graph-based approach for clustering schemas which would finally expose to users a hierarchical mediated schema forest, and a query forwarding mechanism to transform queries down along the schema forest. A set of experimental results demonstrate that our schema clustering algorithm is effective in clustering the data sources into hierarchical schemas, queries on the mediated schemas could achieve answers with good accuracy, and the cost of writing query clauses for users is reduced without losing query accuracy.

1 INTRODUCTION

Many application contexts involve a large amount of heterogeneous structural data sources, such as HTML tables, downloadable spreadsheets and tables from enterprise databases. If users need to obtain answers from the underlying data sources that meet certain conditions, it's troublesome to access each data source one by one. Data integration is an approach used to access such large numbers of heterogeneous structured data sources by providing users with a uniform interface to these data sources (Sarma et al., 2008; Dong et al., 2007; Aboulnaga and Gebaly, 2007). A data integration system consists typically of a mediated schema for one domain and semantic mappings between the schemas of the data sources and the mediated schema (Sarma et al., 2008). With data integration, the user can pose queries on the mediated schema so as not to pose queries on each table of different data sources to get the answers. It is now common that data sources where many domains coexist are maintained by one system. Yet it is not easy to query from such system even if the data sources are clustered into several domains and data integration are applied for each domain since users still have to choose the target domain or write different query clauses for different domains when querying.

Previous researches present various data integration techniques, but nearly all of them require the schemas of different data sources to be integrated belong to the same domain (Sarma et al., 2008; Dong et al., 2007; Aboulnaga and Gebaly, 2007). When dealing with schemas of multiple domains, data integration without the step of clustering schemas tends to produce semantically incoherent mediated schemas and incorrect schema-mappings. Clustering schemas has been taken into consideration in (Mahmoud and Aboulnaga, 2010). Their method clustered real schemas of databases into different domains, but in this case it is difficult for users to identify which mediated schema is their target schema because users still need to face too many mediated schemas. This observation led to the requirement of an obvious mark to represent a mediated schema and to reduce the amount of mediated schemas that user would visit so that the cost of accessing underlying data can be diminished.

This paper proposes an approach to generate hierarchical mediated schemas for different domains.

Huang Q., Zhong C. and Zhang J..
Multi-domain Schema Clustering and Hierarchical Mediated Schema Generation.
DOI: 10.5220/0004825601110118
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 111-118
ISBN: 978-989-758-027-7

Specifically, this paper (1) newly defines a similarity measurement between two schemas considering the different importance of schema name and attributes simultaneously, (2) designs a graph-based algorithm to hierarchically cluster schemas through two phase removals of edges and generate mediated schemas for each schema clusters adding mediated schema names, (3) and designs a query forwarding mechanism to transform queries down along the hierarchical mediated schemas. Mediated schema name is first proposed in this paper to represent a mediated schema which is helpful for the user to identify her target schemas. Hierarchical schemas are used to reduce the amount of mediated schemas which users have to visit. Finally, we design the query forwarding mechanism to conform to the hierarchical schemas framework.

The remainder of the paper is organized as follows. Section 2 presents the related work to our research. Section 3 gives an overview of our solution. In Section 4, we show the details of hierarchical mediated schema generation. Section 5 explains how the queries are executed in the database with hierarchical mediated schemas. Experimental results and analysis are presented in Section 6. Finally, we draw conclusions in Section 7.

2 RELATED WORK

Most of the previous work on structural data integration focused on automatically creating mediated schemas (Sarma et al., 2008) and schema-mapping creation (Sarma et al., 2008; Dong et al., 2007). Our approach automatically groups the schemas of different domains while traditional one-domain schema generation methods chose the schemas belonging to one domain to integrate. Previous work on schema-mapping creation (Sarma et al., 2008; Dong et al., 2007) mainly studied on how to compute a set of attribute correspondences between attributes in mediated schemas and attributes in source schemas. We create all the schema-mappings by assign a unique schema ID to each schema to avoid the computation of attribute correspondences. In (He et al., 2004), the authors take clustering schemas into domains into consideration. However, their approach assumes that for each domain there are some anchor attributes that do not appear except in that domain, but we do not require such prerequisite. Schema clustering is also mentioned in (Madhavan et al., 2007) as part of their pay-as-you-go architecture, but they did not describe the details such as similarity between schemas and the clustering process. The work closest to ours is in (Mahmoud and Aboulnaga, 2010) which employed hierarchical clustering algorithm (Murphy, 2012) to cluster schemas into domains and assigned schemas to domains using a probabilistic model. Compared with the multi-domain schema clustering method in (Mahmoud and Aboulnaga, 2010), the name of schema and attributes are all considered and given different importance in our work, while their method only considers the attributes when computing the similarity of two schemas. Furthermore, our approach produces a hierarchical mediated schema forest with the lowest level schemas being the real source schemas but other level schemas being the abstract mediated schemas. We produce mediated schema both for source schemas and mediated schemas, whereas, it (Mahmoud and Aboulnaga, 2010) only produces source schema clusters (mediated schemas of one level). The final difference between our approach and other data integration methods is that we add a name to mediated schema to help users understand a schema much more.

3 SOLUTION OVERVIEW

The objective of our research is to automate producing hierarchical mediated schemas for data sources of multiple domains. Providing a valid and complete representation of a *domain* is a non-trivial task in Conceptual modeling (Wand and Weber, 2002). Here we follow the definition of *domain* in (Mahmoud and Aboulnaga, 2010): a domain is a set of schemas with sufficiently large intra-domain similarity and sufficiently large inter-domain dissimilarity, according to some measure of similarity. A schema is defined as a set of attributes associated with the name of this schema in our paper. A database table with its name and without its data is such a schema accordingly.

The input of our system is a set of schemas that are extracted from structured data sources. The attributes and the schema names are the only information we require. Domains and their upper domains would be discovered from the available schemas. Then the discovered domains would be delivered to a schema mediation algorithm which has been widely studied. Consequently, a mediated schema forest which is the final output of our system will be exposed to users who want to query the underlying data sources. Schema forest in this paper is defined as hierarchical schemas where schemas in the lowest level are source schemas and other lower

level mediated schemas are sub-domains of their respective high level domains.

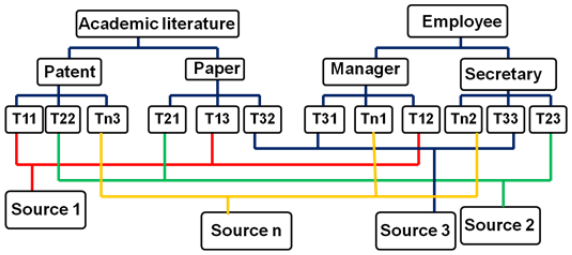

Figure 1: A scenario about domain and sub-domain of several data sources.

Figure 1 depicts the hierarchical relationship about schemas in different data sources. As we can see on the 2-nd level of Figure 1, the domain *Patent* contains three source schemas from different sources, i.e. T_{11} of *Source 1*, T_{22} of *Source 2* and T_{n3} of *Source n*. A mediated schema with attributes and *Patent* as its schema name represents domain *Patent* will be generated. On the 1-st level, the domain *Academic literature* owns two sub-domains, i.e. *Patent* and *Paper*. The mediated schemas reflecting all the domains are generated the same as the domain *Patent*.

Figure 2 shows the system diagram of the proposed approach. In traditional small-scale database scenario, one user only needs to pose a query on a table and get the answers she wants. Steps 1, 4 and 5 of Figure 2 are required in such case. When dealing with large-scale data space, step 2 is required to offer a uniform interface of all data sources and step 3 is used to give appropriate query formats conformed to mediated schemas. We will introduce the details of step 2 in Section 4 and step 3 in Section 5.

4 MEDIATED SCHEMA FOREST GENERATION

4.1 Schema Similarity

Before clustering the schemas, we need to determine how to measure the similarity between two schemas. Besides the attributes of a schema which are useful to represent a schema, the schema name is also valuable since different databases are likely to use similar and even the same titles to name the tables that may relate to the same object. For instance, a table in $database_1$ is named *staff*, while another table in $database_2$ called *employee* probably describes the same object. In such case, these two tables are likely to belong to the same domain. Thus, we define a new similarity measurement between two tables/schemas considering the different importance of schema name and attributes simultaneously. The definition is as equation (1).

$$\text{similarity}(s_i, s_j) = \theta \cdot \text{sim}(s_i.\text{name}, s_j.\text{name}) + (1+\theta)\frac{\sum_{a_x \in s_i.A}\sum_{a_y \in s_j.A}\text{sim}(a_x, a_y)}{|s_i.A| \cdot |s_j.A|}$$
$$\theta \in [0,1]. \quad (1)$$

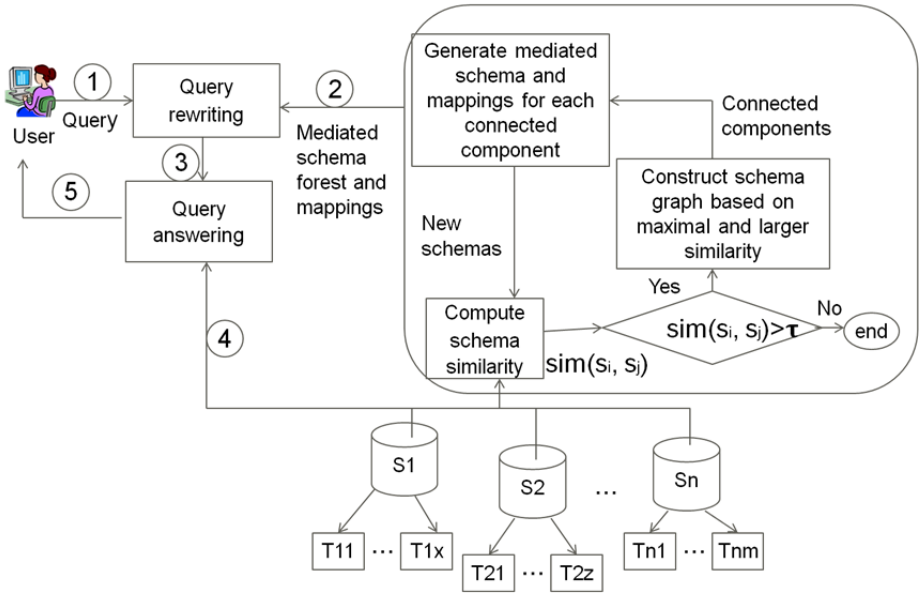

Figure 2: A query system with hierarchical mediated schemas.

The first term after "=" measures the syntactic or semantic similarity of two schema names. Similarity between two attribute sets of the respective schemas is evaluated in the second term. θ is a parameter which is used to estimate the importance of schema name. It is pre-defined and could be tuned in concrete application. s.name represents name of schema **s**. The capital letter "A" denotes the attribute set of a schema. Consequently, |s.A| represents the cardinality of attribute set A of schema **s**. sim(a,b) could be syntactic or semantic similarity of *a* and *b*.

4.2 Hierarchical Generation of Mediated Schemas

An algorithm of hierarchical generation of mediated schemas is proposed. The details are depicted in Algorithm 1.

Algorithm 1: Hierarchical mediated schemas generation.

```
0: Input: a set of source schemas S_all
1: Output: a set of mediated schemas T
2: Initialize α, τ, S= S_all, T=null
3: for each schema s_i ∈ S
4:   for each schema s_j ∈ S, s_i ≠ s_j
5:     Compute similarity(s_i, s_j) using equation (1)
6: if one of the similarities is greater than τ
7:   Construct a weighted graph G(V,E,W),
     where V=S, E={(s_i, s_j)}, W={(v_i, v_j)| (v_i, v_j)=
      similarity(s_i, s_j) }
8:   for each v_i ∈ V
9:     for each v_j ∈ V, v_i≠v_j
10:      if ((v_i, v_j) <(v_i, v_imax)
          where (v_i, v_imax) =max (v_i, v_k) (v_i≠v_k,
          v_k ∈ V ))
          && ((v_i, v_j) <(v_j, v_jmax)
          where (v_j, v_jmax) =max (v_j, v_k) (v_j≠v_k,
          v_k ∈V))
11:       remove (v_i, v_j)
12:   for each connected component c_i ⊆ G
13:     for each e_i, e_j ∈c_i.E, ω∈c_i.W
14:       if ωe_i /ωe_j >= α     remove e_j
15:       if ωe_i /ωe_j <= 1/ α  remove e_i
16:   S=null
17:   for each c_i ⊆G (where {s_i1, s_i2, …}=c_i.V )
18:     create a mediated schema med-s_i for c_i
19:     infer med-s_i.name by {s_i1.name,
         s_i2.name, …}
20:     add med-s_i to S
21:     med-s_i.child_set={ s_i1, s_i2, …}
22:     add med-s_i to T
23:   update α
24:   goto step 3
24: else return T
```

Steps 3-5 in Algorithm 1 compute the pair-wise schema similarity according to our newly defined similarity measurement. Step 7 constructs a weighted graph with schemas as vertices. If the similarity of two schemas is greater than 0, there is an edge (s_i, s_j) with weight similarity(s_i, s_j). Steps 8-11 are the first phase removal of edges so as to hold the maximal similarities. The second phase removal of edges based on larger similarities is taken in steps 12-15. Remove edges in a connected component if the weight of the edge is significantly less than other edges' weights. α is a threshold used to control whether remove an edge or not in a connected component. If the orders of magnitude of two edges' weights are larger than α or less than 1/α, remove the less weighted edge. α is updated for each level, the higher lever owns a lager α since the schemas in the same cluster of higher level are more dissimilar than lower level. α can be tuned for different schema sets.

In steps 17-22, we can use any mediated schema generation techniques (Mahmoud and Aboulnaga, 2010) to create mediated schemas since the schemas belong to one domain for a schema cluster after the schema clustering process. In this paper, the mediated schema of each domain is created as the method in (Sarma et al., 2008). We choose the most frequent attribute in one cluster as the mediated attribute of its mediated schema. The mediated attribute along with the attributes in its cluster are stored as a schema attribute mapping which will be described in Section 5.

Besides the mediated schema attributes, unlike other methods, we provide each schema a name as an obvious mark to a schema. Each mediated schema will be given a name according to the names of including schemas. Cluster labelling method using Wikipedia (Carmel et al., 2009) is used to infer the name of mediated schema. A search index was built from one of the available Wikipedia dump[1] using *Lucene*[2] search system. For the names of each including schemas, we search the generated index with the disjunction of such names. The titles and categories of the documents returned by the search are considered as the potential candidates of the target mediated schema name. Finally, the most frequently appeared term of the candidates as well as all including schema names is chose as the mediated schema name.

The iteration ends when all of the similarities are less than the threshold **τ**. Each schema graph corresponds to one level in the schema forest and

[1] http://dumps.wikimedia.org/enwiki/20130204/
[2] http://lucene.apache.org/

can produce a set of mediated schemas. The above procedures correspond to step 2 in Figure 2.

5 QUERY FORWARDING

At the schema clustering time, we produce a mediated schema forest of various data sources. At query phase, users can pose queries using the terminology of any of the mediated schemas although most of time the higher level schemas are used. The query is reformulated and delivered downward to the source schemas according to the mapping between the mediated schema and its lower schemas. A whole mapping called *schema mapping* in our system is designed to consist of two parts. First part is the one-to-many schema *name mappin*g. The mediated schema name is mapped downward to the schema names of the lower schemas. We give an ID to each schema which can uniquely identify the schema. The name mappings keep schema names along with their IDs. The IDs guarantee the schema mappings are correct mappings. Consider the next example of 3 source schemas.

Example 1:

(*Paper: Paper_No., Title, Presentation, Author_info, Nation, Contact, Company*)

(*Proceedings: Year, Conference, Title, Author1, Company1, Author2, Company2*)

(*Publication: Ref_No., Type, Title, Author_list, Reference, Volume, Issue_section, Pages, Year, Month, Day, Conference_notes*)

In the above example, schema name and schema attributes are separated by colon, and attributes are separated by comma. After processing by our system, we could obtain the following mediated schema.

Example 2:

(*Academic_publication: Title, No., Author, Conference, Year*)

The name mapping between the above mediated schema and source schemas is as follow.

Example 3:

((*4, Academic_publication*) → ((*1, Paper*), (*2, Proceedings*), (*3, Publication*)))

This mapping maps schema (*4, Academic_publication*) to three lower schemas (*1, Paper*), (*2, Proceedings*), and (*3, Publication*). The first element in (*4, Academic_publication*) is the schema ID and the other one is the schema name.

One-to-many attribute mappings are the other part of the *schema mapping*. The *attribute mapping* format is the same as the name mapping, but the second element within the parenthesis is an attribute such like "*Title*" in (*4, Title*) of Example 4. There would be several attribute mappings in one *schema mapping* since a mediated schema usually owns several mediated attributes. Consider the next mappings.

Example 4:

((*4, Title*) → ((*1, Title*), (*2, Title*), (*3, Title*)))

((*4, No.*) → ((*1, Paper_No.*), (*3, Ref_No.*)))

((*4, Author*) → ((*1, Author_info*), (*2, Author1*), (*2, Author2*), (*3, Author_list*)))

((*4, Conference*) → ((*2, Conference*), (*3, Conference_notes*)))

((*4, Year*) → ((*2, Year*), (*3, Year*)))

Example 4 is the attribute mappings between Example 1 and Example 2.

When users pose a query on the mediated schema in Example 2, the system will reformulate the query to the source schema according to the mappings in Example 3 and Example 4. Several queries would be generated matching the source schemas in Example 1. Finally the databases will return the required answers to users. To illustrate the forwarding process, consider the next queries. The query is posed on the mediated schema in Example 2.

SELECT Title, Author, Year
FROM Academic_publication
WHERE Year > 2006 AND Author = 'Strehl'

According to the mappings in Example 3 and 4, the next two queries will be posed on the corresponding data sources automatically.

(1) SELECT Year, Title, Author1, Author2
FROM Proceedings
WHERE Year > 2006 AND (Author1= 'Strehl' OR Author2 ='Strehl')

(2) SELECT Title, Author_list, Year
FROM Publication
WHERE Year > 2006 AND Author_list = 'Strehl'

6 EXPERIMENTS

Our algorithms were implemented in Java. We run the experiments on a Windows 7 machine, with 2.60GH Intel(R) i5 processor and 8GB memory. The goal of our experiments is to demonstrate that our schema clustering algorithm is effective in clustering the data sources of multiple domains, queries on the mediated schemas could achieve answers with good accuracy and the cost of writing query clauses for users is reduced without losing query accuracy.

For the purpose of our query evaluation, we used MySQL to store the data. Two string similarity measurements are utilized to compute the schema similarity since two strings may be semantically

similar or syntactically similar. For example, "*Film*" and "*Movie*" are semantically similar, whereas "*Author*" and "*Author_info*" would be syntactically similar. We use the SecondString tool[3] and WS4J[4] to compute the syntactic similarity and semantic similarity of pair-wise attribute strings or schema names. The function sim() in equation (1) is evaluated by the larger one of syntactic similarity and semantic similarity.

6.1 Data Used

Table 1: Number of schemas in each domain and each upper domain.

	#schema (the leaf layer)	#schema (the third layer)	#schema (the second layer)	#schema (the first layer)
Paper	23	2	1	1
Patent	16			
Employee	26	2	1	1
Student	15			
Car	12	3	1	1
Truck	16			
Bus	13			
Movie	25	3	1	1
TV serial	11			
Novel	17			
Baseball	15	2	3	1
Basketball	17			
Road running	13	2		
Adventure running	10			
Swing	16	2		
Diving	12			

We collect the schemas and their data from our enterprise databases and the downloadable spreadsheets that are obtained using Google's "search by file type" option with certain keywords related to different domains. For the purpose of generating hierarchical mediated schema forest, some domains we selected expose parallel relationship and we could infer the upper domains for them. For instance, "Employee" and "Student" belong to the upper domain "Person". In each spreadsheet, we use the strings in the headers of the columns as the attributes, and manually choose the title or topic of the spreadsheet as the schema name. For the tables extracted from our enterprise databases, the table names and their field names are used as schema names and attributes. We manually associated each source schema a label indicating the domain it belongs to. And schema name of a mediated schema indicates the domain it represents. Table1 shows some details of the extracted tables.

[3] http://secondstring.sourceforge.net/

[4] https://code.google.com/p/ws4j/

6.2 Schema Clustering Evaluation

In this part, we evaluate the effectiveness of the proposed schema clustering algorithm. Let the set $S = \{s_1, s_2, \cdots\}$ denote the schema set. $d = \{d_1, d_2, \cdots\}$ denotes the actual domain set associated to each source schema. Two functions indicating which domain a schema is contained in are defined as

$D_{before}(s)$: $\{ s_1, s_2, \cdots\} \rightarrow \{ d_1, d_2, \cdots\}$,

$D_{after}(s)$: $\{ s_1, s_2, \cdots\} \rightarrow \{ d_1, d_2, \cdots\}$.

Specifically, $D_{before}(s) = d_i$ $(d_i \in d)$ means the schema s are associated with domain d_i, and $D_{after}(s) = d_i$ means s are clustered into the cluster with d_i as its domain, i.e. the mediated schema name of the cluster.

We measure *precision*, *recall* and *NMI* (Normalized Mutual Information) (Strehl and Ghosh, 2003) of clustering results of the leaf layer schemas in the resulting mediated schema forest since the domain of each mediated schema cannot be assigned in advance and the effectiveness of schema clustering algorithm can be validated using the schemas in any layer.

Precision: We estimate the average precision as

$$\frac{1}{|d|}\sum_{d_i \in d}\frac{|TP_{d_i}|}{|TP_{d_i}| + |FP_{d_i}|}$$

where TP_{d_i} is the true positive set of domain d_i and FP_{d_i} is the false positive set of domain d_i. Therefore, $TP_{d_i} = \{s|\ D_{before}(s) = d_i \ \&\& \ D_{after}(s) = d_i\ \}$, and $FP_{d_i} = \{s|\ D_{before}(s) \neq d_i \ \&\& \ D_{after}(s) = d_i\ \}$.

Recall: We estimate the average recall as

$$\frac{1}{|d|}\sum_{d_i \in d}\frac{|TP_{d_i}|}{|TP_{d_i}| + |FN_{d_i}|}$$

where FN_{d_i} is the false negative set of domain d_i. Therefore $FN_{d_i} = \{s|\ D_{before}(s) = d_i \ \&\& \ D_{after}(s) \neq d_i\ \}$.

NMI: We estimate the NMI according to the definition in (Strehl and Ghosh, 2003).

In this paper, we emphasise that the schema names are important in clustering schemas. To evaluate such view, we run the experiments with different values of θ on three schema sets composed of the schemas showed in Table 1 to see how the clustering results change. We selected the schemas of different domains the schema names of which are clearly not similar to constitute schema set SS1, but in schema set SS2, some schema names in different

domains are somewhat similar to others. Schema set ALL is comprised of all the schemas in Table 1. The clustering results are showed in Figure 3. θ=0 means the schema names are not concerned in similarity measurement when clustering schemas. As θ increases, the three metrics increase, but they would not increase all the time. Each measurement reaches a maximal value for different schema sets, and then decreases. The parameter α influences the clustering results as well. If α is too small, there would be more small clusters or some schemas are isolated alone. In contrary, a too large α will lead less clusters, namely, the schemas are not sufficiently separated. Before evaluating the influence of the primary factor θ, we conducted the experiments multi-times to find an appropriate α for each schema set. The results in Figure 3 prove that schema names are useful in estimating the domain a schema belongs to, but one schema name could not represent a schema.

With appropriate θ and α for each schema set, we run experiments to compare the schema clustering results of hierarchical clustering and our algorithm. We implemented the hierarchical clustering algorithm proposed in (Mahmoud and Aboulnaga, 2010). The comparison results are showed in Table 2.

Table 2: Comparison of clustering results.

		SS1	SS2	ALL
Hierarchical clustering	Precision	0.81	0.80	0.80
	Recall	0.91	0.89	0.89
	NMI	0.77	0.75	0.76
Our algorithm	Precision	0.82	0.78	0.81
	Recall	0.94	0.87	0.89
	NMI	0.79	0.73	0.77

For the three different schema sets, the three metrics of SS_1 are higher than that of SS2, since SS2 contains some ambiguous schemas. It is not surprising that the accuracy of schema set ALL is between that of SS1 and SS2. Through Table 2, we can also see that the three metrics of our algorithm are a bit lower than hierarchical clustering for SS2. It is because that the hierarchical clustering using single linkage is good at clustering ambiguous schemas into isolated clusters. However, the three metrics are higher than hierarchical clustering for SS1 and ALL. To sum up, our algorithm is effective in clustering schemas.

6.3 Query Quality

In this part, we evaluate the quality of queries posed on the mediated schema forest generated by our method so that we can prove the practicability of hierarchical meditated schemas. For each domain in each layer of the schema forest, we created 8 diverse queries. The queries are created by the following principles. Each query contains one to three attributes in the SELECT clause and zero to three predicates in the WHERE clause. The table names in the FROM clause are the mediated schema names. The attributes in the SELECT and WHERE clauses are attributes from the exposed mediated schemas.

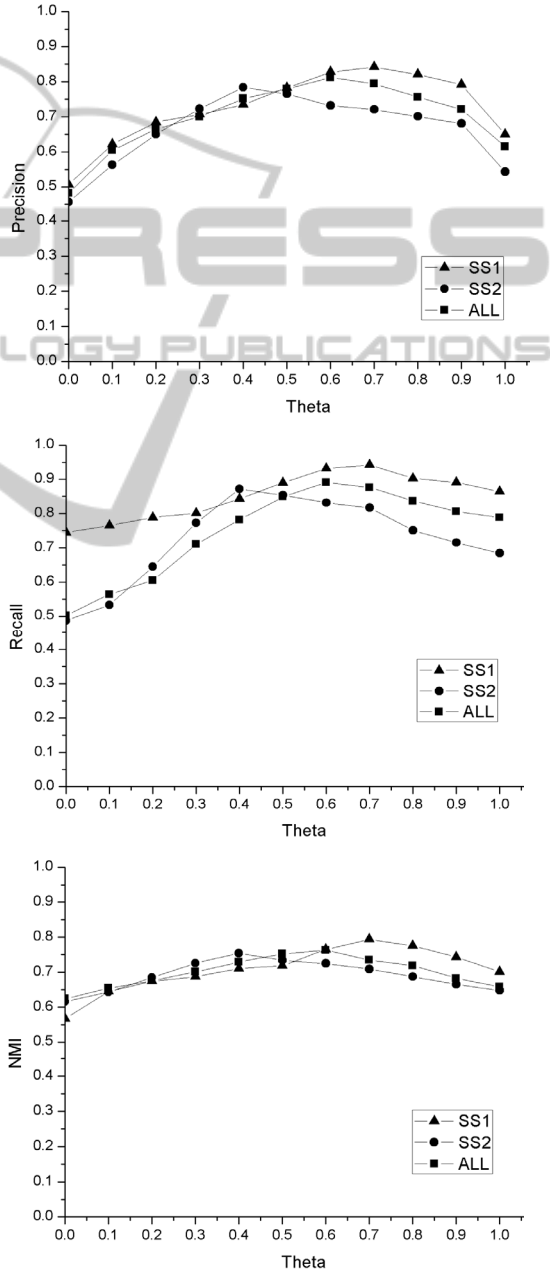

Figure 3: Influence of Schema Name. the Value of Theta (Θ) Indicates the Importance of Schema Name.

In our experiments we used two metrics *precision* and *recall* to measure the query accuracy. The two metrics are different with that used in clustering evaluation. For all queries, let A be the set of answers generated by our approach and B be the set of true answers. We estimate query precision as $\frac{|A \cap B|}{|A|}$ and recall as $\frac{|A \cap B|}{|B|}$. We compare the average query accuracy of our algorithms with that of one mediated schema method and key word query method. The results are showed in Table 3.

Table 3: Comparison of query quality.

		SS1	SS2	ALL
One mediated schema	Precision	0.87	0.81	0.83
	Recall	0.78	0.72	0.76
keyword	Precision	0.85	0.75	0.82
	Recall	0.74	0.68	0.72
Our algorithm	Precision	0.97	0.88	0.86
	Recall	0.88	0.84	0.84

Table 3 states obvious that the query precision and recall of our algorithm gain a notable increase than that of another two methods. Not surprisingly, keyword query method has the worst results since keyword search engines of database offer a simple and general solution for searching any kind of information. One mediated schema method treats all schemas as schemas in one domain. That is the reason why its results are worse than our algorithms. We have observations that the precision of our method is high. It is because the schema mappings we designed are correct mappings between mediated schemas and source schemas due to the schema ID. Generally speaking, the results demonstrate that queries on the mediated schemas generated by our algorithm could achieve answers with good accuracy.

Finally, we compare the average query accuracy of one level mediated schemas and hierarchical mediated schemas. For the source schemas of *Sports* in the bottom of Table 1, we need to write 6 queries in the domain of the source schema clusters, but only one query clause in the highest level domain. These two situations obtain exactly the same answers. The similar behaviours of less query clauses with the same answer are presented for the other source schemas. This is due to the query forward mechanism which transforms the query from the highest level domain to the lowest level domain transparently for users. In one word, the cost of writing query clauses for users is reduced without losing query accuracy.

7 CONCLUSIONS

We believe that users need hierarchical mediated schemas with names when facing a large number of data sources from different domains. Therefore, in this paper, we presented an approach of generating hierarchical mediated schema forest with schema name assigned to each mediated schema for data sources of multiple domains. We also explained the compatible query execution process against the hierarchical mediated schemas. The experimental results on data sources of different domains validate the feasibility and effectiveness of our approach. Users can easily find the target mediated schema and obtain the answers with good accuracy and less query effort.

In future, we will investigate more on when we should stop the process of producing hierarchical mediated schema to reduce the time cost and applied the approach to very large scale source schemas.

REFERENCES

Sarma, D.A., Dong, X. and Halevy, A., 2008. Bootstrapping pay-as-you-go data integration systems. In *SIGMOD*, ACM.

Dong, X., Halevy, A. and Yu, C., 2007. Data integration with uncertainty. In *VLDB*, ACM.

Aboulnaga A. and Gebaly, K.E., 2007. µbe: User guided source selection and schema mediation for internet scale data integration. In *ICDE*, IEEE.

Mahmoud, H.A. and Aboulnaga, A., 2010. Schema clustering and retrieval for multi-domain pay-as-you-go data integration systems, In *SIGMOD*, ACM.

He, B., Tao, T. and Chang, K. C.-C., 2004. Organizing structured web sources by query schemas: a clustering approach. In *CIKM*, ACM.

Madhavan, J., Cohen, S., Dong, X., Halevy, A., Jeffery, S., Ko, D. and Yu, C., 2007. Web-scale data integration: you can afford to pay as you go. In *CIDR*.

Murphy, K., 2012. *Machine learning: a probabilistic perspective*. MIT Press, London.

Wand, Y. and Weber, R., 2002. Research Commentary: Information Systems and Conceptual Modeling--A Research Agenda. *Information Systems Research*, vol.13(4), pp.363-376.

Carmel, D., Roitman, H. and Zwerdling, N., 2009. Enhancing Cluster Labeling Using Wikipedia. In *SIGIR*, ACM.

Strehl, A. and Ghosh, J, 2003. Cluster ensembles: a knowledge reuse framework for combining multiple partitions, *The Journal of Machine Learning Research*, vol.3, pp.583-617.