On Semantic Matching of Multilingual Attributes in Relational Systems

A. Kumaran   Jayant R. Haritsa
Database Systems Lab, SERC / CSA
Indian Institute of Science, Bangalore, INDIA
{kumaran,haritsa}@dsl.serc.iisc.ernet.in

Categories and Subject Descriptors: H.2 [Database Management]: Languages-*Query Languages*

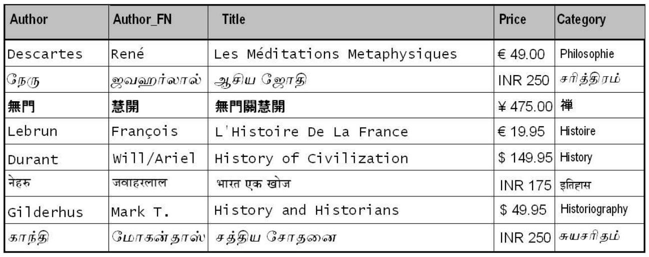
General Terms: Algorithms, Languages

Keywords: Multilingual Databases, Ontology-based Query Processing, Semantic Information Retrieval.

1. INTRODUCTION

In an increasingly multilingual digital world, it is critical that information management tools, such as web search engines, *e-Commerce* portals and *e-Governance* applications, support the simultaneous use of multiple natural languages. An essential pre-requisite is that the underlying database engines (typically relational), provide the functionality for processing multilingual data seamlessly across languages. As a part of our Mira [1] research initiative focusing on functionality and performance aspects of supporting multilingualism in relational database systems, we propose SemEQUAL, a *semantic* operator for matching text attribute data across languages based on *meaning*. For example, to automatically match the English noun *mathematics*, with *mathématiques* in French or கணிதம் (meaning *mathematics*) in Tamil.

1.1 Motivation for the SemEQUAL Operator

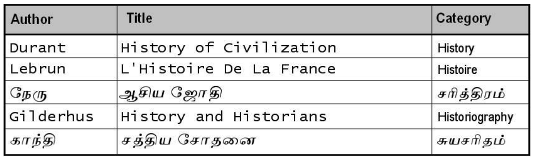


Author	Author_FN	Title	Price	Category
Descartes	René	Les Méditations Metaphysiques	€ 49.00	Philosophie
நேரு	ஜவஹர்லால்	ஆசிய ஜோதி	INR 250	சரித்திரம்
無門	慧開	無門關慧開	¥ 475.00	禅
Lebrun	François	L'Histoire De La France	€ 19.95	Histoire
Durant	Will/Ariel	History of Civilization	$ 149.95	History
नेहरु	जवाहरलाल	भारत एक खोज	INR 175	इतिहास
Gilderhus	Mark T.	History and Historians	$ 49.95	Historiography
காந்தி	மோகன்தாஸ்	சத்திய சோதனை	INR 250	சுயசரிதம்

Figure 1: A Multilingual *Books.com*

The proposed semantic operator is illustrated on a hypothetical *Books.com*, with a sample multilingual product catalog, as shown in Figure 1. Currently, a query with selection condition as (Category = 'History'), would return only those books that have *Category* as `History` in English, although the catalog also contains history books in French, Hindi and Tamil. A multilingual user may be better served, if all the history books in all the languages (or more likely, in a set of languages specified by her) are returned by the SemEQUAL operator, as shown in Figure 2. While only the first three records (matching the English `History`)[1] are reported by basic SemEQUAL, the result set shown contains all records that match with *semantic specializations* (`Historiography` and `Autobiography` are specialized branches of `History`)[2], triggered by the optional ALL directive.

```
SELECT  Author,Title,Category  FROM  Books
WHERE  Category  SemEQUAL ALL  'History'
InLanguages  {English, French, Tamil}
```



Author	Title	Category
Durant	History of Civilization	History
Lebrun	L'Histoire De La France	Histoire
நேரு	ஆசிய ஜோதி	சரித்திரம்
Gilderhus	History and Historians	Historiography
காந்தி	சத்திய சோதனை	சுயசரிதம்

Figure 2: Basic Semantic Selection

1.2 Defining the SemEQUAL Operator

In semantic matching of multilingual text attributes, we rely on common linguistic resources, specifically *WordNet* [2] that provide rich semantic relationships between the word forms of a language. In addition, the development of WordNets in multiple languages [3, 4] with semantic links between them, provides a way to match them based on meanings. We define the semantic match using the WordNet taxonomic hierarchies in multiple languages (denoted as $\mathcal{H}$), as follows: Given two nodes $\mathcal{A}$ and $\mathcal{B}$ in $\mathcal{H}$, we say $\mathcal{A}$ *is-semantically-equivalent-to* B, iff $\{\mathcal{A}\} \cap \mathcal{T}_{\mathcal{H}}(\mathcal{B}) \neq \phi$, where $\mathcal{T}_{\mathcal{H}}(\mathcal{X})$ is the transitive closure of $\mathcal{X}$ in $\mathcal{H}$.

Consider a canonical SemEQUAL predicate as follows:

$$\{Attr\ a\}\ \texttt{SemEQUAL}\ \{\texttt{Const c}\}\ \texttt{InLang}\ L_1, \ldots, L_N$$

Let S^c_{in} denote the set of synsets of Constant `c` in the input language L_{in}, $S^c_{out_i}$ denote the set of matching synsets of S^c_{in} in target language L_{out_i}. Then, $T_{\mathcal{H}}(S^c_{out_i})$ denotes the transitive closure of $S^c_{out_i}$ in one of the output languages L_{out_i} and $\bigcup_i T_{\mathcal{H}}(S^c_{out_i})$ denotes the union of all such closures. Further, let the value of the Attribute `a`, in the database tuple

[1] The category of the third record of the result is சரித்திரம் (transliterated as, $Charitr\underline{a}m$) in Tamil, meaning `History`.

[2] The category of the last record is சுயசரிதம் (transliterated as, $Suyacharit\underline{a}m$) in Tamil, meaning `Autobiography`.

CIKM'04, November 8–13, 2004, Washington, DC, USA.
ACM 1-58113-874-1-1/04/0011.

currently under consideration, be in the language L_{data}, and the set of synsets of a with respect to L_{data} be $S^a_{L_{data}}$. With this notation, the SemEQUAL operator returns true only if $S^a_{L_{data}} \bigcap (\bigcup_i T(S^c_{L_{out_i}})) \neq \phi$.

2. IMPLEMENTATION

The skeleton of the SemEQUAL function is given in Figure 3. The Str_{Data} is the string from the tuple under consideration (LHS operand) and the Str_{Query} is the query string (RHS operand). We implemented the TransitiveClosure function (line 3) as a *derived operator*, using the standard SQL:1999 features, in line with our objectives.

```
SemEQUAL ($Str_{Data}$, $L_{Data}$, $Str_{Query}$, $L_{Query}$,$\mathcal{T_L}$)

Input:  Strings $Str_{Data}$ in $L_{Data}$, $Str_{Query}$ in $L_{Query}$
        Set of Target Languages $\mathcal{T_L}$
Output: TRUE or FALSE
        [Optionally] Gloss of the matched Synset

1. ($\mathcal{W_D}$,$\mathcal{W_Q}$) $\leftarrow$ WordNetOf ($L_{Data}$,$L_{Query}$);
2. ($\mathcal{S_D}$,$\mathcal{S_Q}$) $\leftarrow$ Synsets ($Str_{Data}$ in $\mathcal{W_D}$, $Str_{Query}$ in $\mathcal{W_Q}$);
3. $\mathcal{T_Q}$ $\leftarrow$ TransitiveClosure ($\mathcal{S_Q}$, $\mathcal{T_L}$);
4. if $\{\mathcal{T_Q} \cap \mathcal{S_D}\} \neq \phi$ then TRUE else FALSE;
5. [Optional] return Gloss of a synset in $\{\mathcal{T_Q} \cap \mathcal{S_D}\}$
```

Figure 3: Semantic Matching Algorithm

The SemEQUAL operator needs two significant steps: computation of the transitive closure of query string Str_{Query} (line 3), and testing if any of the values of $S^a_{L_{data}}$ is a member of the set $\bigcup_i T(S^c_{L_{out_i}})$ (line 4). Implementation-wise, the computation of closure in relational systems is, in general, recognized to be slow [5]. After computing the transitive closure, however, the operator would cycle through the inner table (the LHS operand), outputting all records for which SemEQUAL returns a value `true`. This second step may be implemented efficiently using well-known hash-table techniques. We observed in our experiments that the processing of the IN predicate contributes very little to the overall processing time of the SemEQUAL query ($< 1\%$). Thus, the overall performance of the query primarily depends on the speed of computing the closure, in the current systems.

3. PERFORMANCE STUDIES

For the performance experiments, – details in [6], we used a standard workstation running a popular commercial database system. The entire set of noun taxonomic hierarchies of WordNet (Version 1.5), totaling about 110,000 *word forms*, 80,000 *word senses* and about 140,000 relationships between them, was loaded on the database systems, occupying about 4 MB of storage. Since different WordNets are in different stages of development, we simulated linked WordNets, by replicating English WordNet in Unicode, and creating an equivalence link between corresponding synsets. The queries to compute a closure size of approximately $2,000$[3] on the linked WordNet database were run, in an SQL environment. Our objective is to devise methods to speed up the performance of the SemEQUAL query, to a level acceptable for online interactions.

Closure Size	Basic No Index	Basic Index	PreComp Index	ReOrg Index
2,000	40.0	3.4	0.87	0.03

Table 1: Transitive Closure Performance

Table 1 provides the runtime of the queries, measured as the wall-clock time of a given query on the given data set. The closure computation takes a few seconds (with a B+ index) to tens of seconds (without an index), making the performance unsuitable for *e-Commerce* deployments.

We implemented two different optimizations that improve the closure computation performance, as follows: First, in the **Pre-computed Closure** approach, the transitive closures and computed and stored, along with each node in the hierarchy $\mathcal{H}$, so that the closures could be found with a simple linear scan of the enhanced table. The same baseline query takes about 0.87 seconds (column 4, in Table 1), but such an improvement comes at an enormous storage cost: the storage of precomputed closures and the indexes on them take about 200 MB, nearly 50 times the normal storage of $\mathcal{H}$. In the **Re-organized Schema** approach, we analysed the structural characteristics of WordNet hierarchies and since *less than* 10% have *more than* 16 children, we inlined up to 16 children for each node, leaving the other records in the original $\mathcal{H}$ table. All the queries were re-written to match the new organization. With the reorganized schema, the runtime is improved to 0.03 seconds (column 5, in Table 1), nearly 3 orders of magnitude better than the baseline performance. More significantly, the storage requirement did not go up in this approach, indicating that closures can be computed efficiently, without excessive space tradeoffs.

4. CONCLUSIONS

We proposed a new SQL operator – SemEQUAL, to support seamless multilingual semantic matching of text attribute data, by integrating the WordNet linguistic resource with the database query processing engine. We defined the operator semantics and outlined a *derived-operator* implementation for SemEQUAL. Our experiments on a commercial database systems underscored the inefficiency of the SemEQUAL operator, in computing transitive closure, an essential component for semantic matching. While the runtime was unsuitable for practical deployments, we proposed optimization techniques that speeded up the closure computation by 2 to 3 orders of magnitude – to *a few milliseconds* – without excessive space tradeoff, making the operator efficient and viable for supporting online query processing.

Acknowledgements We thank Prof. P. Bhattacharyya of IIT-Bombay, for his help on Hindi WordNet. This work was supported in part by a Swarnajayanthi Fellowship from Dept. of Science and Technology of Govt. of India.

5. REFERENCES

[1] MIRA Project. *dsl.serc.iisc.ernet.in/projects/mira.html*

[2] The WordNet. *www.cogsci.princeton.edu/w̃n.*

[3] Euro-WordNet. *www.illc.uva.nl/EuroWordNet/*

[4] Indo-WordNet. *www.cfilt.iitb.ac.in/*

[5] R. Agrawal *et. al.* Direct Transitive Closure Algorithms: Design and Performance Evaluation. *ACM TODS*, 1990.

[6] A. Kumaran and J. R. Haritsa. Multilingual Semantic Matching Operator in SQL. *Technical Report TR-2004-3, DSL/SERC, Indian Institute of Science*, Aug 2004.

[3] The average closure size corresponding to the most frequently used query terms is 625. We assumed an average number of output languages to be 3, for a multilingual user.