SemEQUAL: Multilingual Semantic Matching in Relational Systems

A. Kumaran Jayant R. Haritsa

Database Systems Laboratory, SERC/CSA
Indian Institute of Science, Bangalore 560012, INDIA
{kumaran,haritsa}@dsl.serc.iisc.ernet.in

Abstract. In an increasingly multilingual world, it is critical that information management tools organically support the simultaneous use of multiple *natural languages*. A pre-requisite for efficiently achieving this goal is that the underlying database engines must provide seamless matching of text data across languages. We propose here SemEQUAL, a new SQL functionality for semantic matching of multilingual attribute data. Our current implementation defines matches based on the standard WordNet linguistic ontologies. A performance evaluation of SemEQUAL, implemented using standard SQL:1999 features on a suite of commercial database systems indicates unacceptably slow response times. However, by tuning the schema and index choices to match typical linguistic features, we show that the performance can be improved to a level commensurate with online user interaction.

1 Introduction

Internet demographics are changing dramatically: about two-thirds of current Internet users are non-native English speakers [18] and it is predicted that the majority of webpages will be multilingual by 2010 [22]. In such an increasingly multilingual digital world, it is critical that information management tools, *e-Commerce* portals and *e-Governance* applications, support the simultaneous use of multiple natural languages. A pre-requisite is that the underlying database engines (typically relational), provide similar functionality and efficiency for multi-lingual data as that associated with processing uni-lingual data, for which they are well-known.

From the efficiency perspective, we recently profiled in [14] the performance of standard relational operators on multilingual data and proposed efficient storage formats to make the operators *natural-language-neutral*. Subsequently, from the functionality perspective, we introduced a new SQL operator called LexEQUAL [15], for *phonetic* matching of specific types of attribute data across languages, optimized for supporting e-Commerce environments. In this paper, we take the next logical step, by proposing SemEQUAL, a *semantic* functionality for matching text attribute data across languages based on *meaning*. For example, to automatically and transparently match the English noun *mathematics*, with *mathématiques* in French or கணிதம் (transliterated as *kanitham*, meaning *mathematics*) in Tamil.

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	¥ 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
Franklin	Benjamin	Un Américain Autobiographie	€ 25.00	Autobiographie
Gilderhus	Mark T.	History and Historians	$ 49.95	Historiography
காந்தி	மோகன்தாஸ்	சத்திய சோதனை	INR 250	சுயசரிதம்

Fig. 1. A Multilingual *Books.com*

1.1 The SemEQUAL Operator

The proposed semantic matching functionality is illustrated on a hypothetical *Books.com*, with a sample multilingual product catalog, as shown in Figure 1, where the *Category* attribute stores the classification of the book in the original language of publication. In today's database systems, a query with (Category = 'History') selection condition, 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, however, if all the history books in all the languages (or more likely, in a set of languages specified by her) are returned. A query using the proposed SemEQUAL and a result set, as given in Figure 2, would therefore be desirable.

```
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
Franklin	Un Américain Autobiographie	Autobiographie
Gilderhus	History and Historians	Historiography
காந்தி	சத்திய சோதனை	சுயசரிதம்

Fig. 2. Multilingual Semantic Selection

It should be noted that the SemEQUAL operator shown here is generalized to return not just the tuples that are equivalent in meaning, but also with respect to *semantic generalizations* and *specializations*, as in the last three tuples that are reported in the output[1]. Without the optional ALL directive, only the first three records that are directly equivalent to `History` would be reported.

[1] `Historiography` (*the science of history making*) and `Autobiography` are specialized branches of `History`. The third record in the result has a category value of சரித்திரம் (transliterated as $Charitr\underline{a}m$) in Tamil, meaning `History`, and the last record has a category value of சுயசரிதம் (transliterated as $Suyacharit\underline{a}m$) in Tamil, meaning `Autobiography`.

To determine semantic equivalence of word-forms across languages and to characterize the SemEQUAL functionality, we take recourse to WordNet [23], a standard linguistic resource that is available in multiple languages and, very importantly from our perspective, features *inter-lingual* semantic linkages. After integrating WordNet with the database platform, two alternatives arise with regard to the SemEQUAL implementation: a *derived-operator* approach using the standard SQL features, or a *core-operator* implementation that is internally visible to the database engine. While the latter approach may prove more efficient in the long-term, we investigate the derived-operator approach here since it can be implemented immediately on existing commercial database systems using their current SQL capabilities. Specifically, we first analyse the performance of SemEQUAL, expressed using recursive SQL features of the SQL:1999 standard, in relational database systems. A direct implementation on three commercial database systems indicates that supporting multilingual semantic processing is unacceptably slow. However, by applying a few simple optimizations that tune the schema and access structures to match WordNet characteristics, the response times are brought down to *a few milliseconds*, which we expect to be sufficient for current practical deployments. Further, though this paper focuses only on multilingual domain, a functionality defined along the same lines may be generalized for matching in any domain with a well-specified taxonomic hierarchy.

1.2 Our Contributions

To summarize, our main contributions in this paper are:

- Motivating the need for, and formulating the notion of, multilingual semantic equality at the granularity of database attributes.
- Integration of WordNet linguistic resources with relational database systems and a *derived-operator* implementation of SemEQUAL, using standard SQL features.
- Optimizing the performance of SemEQUAL, based on WordNet linguistic features, to a level that appears sufficient for current e-Commerce deployments.

2 Multilingual Semantic Matching

In this section, we provide a brief background on the WordNet linguistic resources, on which the semantics of our current implementation of the SemEQUAL operator is based. Subsequently, we describe our strategy for implementing SemEQUAL as a *derived-operator*, using standard SQL:1999 features that are available in all commercial database systems.

2.1 Overview of WordNet

A word may be thought of as a lexicalized concept; simply, it is the written form of a mental concept that may be an object, action, description, relationship, etc. Formally, it is referred to as a *Word-form*. The concept that it stands for is referred to as *Word-sense*, or in WordNet parlance, *Synset*. The defining philosophy in the design of WordNet is that a synset is sufficient to identify a concept for the user. A short description, similar to the dictionary meaning, called the *Gloss* is provided with synsets, for human understanding. Two words are said to be synonymous, *or semantically the same*, if they

have the same synset and hence map to the same mental concept. WordNet organizes all relationships between the concepts of a language as a semantic network between synsets. A lexical matrix that maps word forms to word senses constitutes the basis for mapping a word-form to synsets. For example, the word-form `bird` corresponds to several different synsets, two of which are {*a vertebrate animal that can typically fly*} and {*an aircraft*}; each of these two synsets is denoted differently with subscripts, in Figure 3. The synsets are divided into five distinct categories and we explore below only the *Nouns* category, as *about a fifth of normal text corpora and majority of query strings are noun-form words* [17].

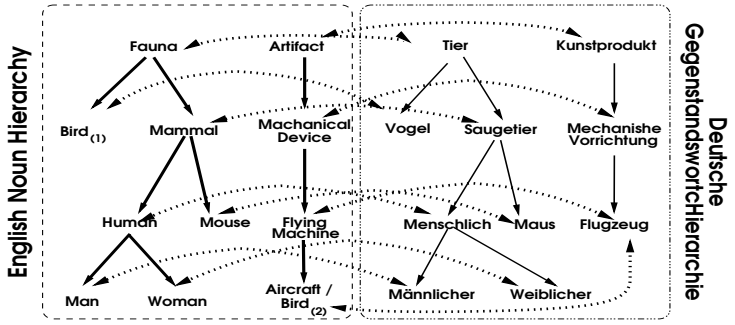

Fig. 3. Sample Inter-linked *WordNet* Noun Hierarchy

Noun Taxonomical Hierarchy The nouns in English WordNet are grouped under approximately twenty-five distinct *Semantic Primes* [7], covering distinct conceptual domains, such as *Animal*, *Artifact*, etc. Under each of the semantic primes, the nouns are organized in a taxonomic hierarchy, as shown in Figure 3, with *Hyponyms* links signifying the `is-a` relationships (shown in solid arrows).

Several efforts are underway – such as the *European WordNet* (EWN) [6] that includes all major European languages and the *Indo-WordNet* (IWN) [13, 2] that include all 15 of the official Indian languages – to link up WordNet taxonomic hierarchies of different languages. A Chinese WordNet (CWN) initiative, along the lines of English WordNet, is outlined in [3]. A common feature among such initiatives is that they keep the basic taxonomic hierarchies nearly the same as that of English and provide mapping from their synsets to that of English. Further, inter-linking of semantically equivalent synsets between WordNets of different languages (shown as dotted arrows) is available for some languages currently [6], and is planned for others [13]. Figure 3 shows a simplified interlinked hierarchy in English and German. Such interlinked hierarchy is used for defining semantic matching in the following section.

2.2 Semantic Matching functionality using Interlinked WordNet

Using the lexical matrix function that is a part of the WordNet linguistic resources, the operands (i.e., the the multilingual word-forms), may be mapped on to distinct set of

synsets associated with the languages of the respective operands. Further, the set of synsets corresponding to the RHS operand is augmented with synsets that are reachable using Inter-Lingual-Index (ILI) links, to the target languages. Once augmented, the semantic equality may be defined as follows: A `equivalent` match is `true`, if there is a non-empty intersection between the LHS and RHS sets of synsets. A `generalized` match is `true`, if there is a non-empty intersection between LHS set of synsets and the transitive closure of the RHS set of synsets in the above taxonomic hierarchy. Such a definition ensures that in *at least one word-sense*, the operands may be matched. For example, only in a `generalized` match, the query (*Is* `bird` SemEQUAL `Artifact`?) and (*Is* `bird` SemEQUAL `Fauna`?), both would be `true`.

2.3 Implementing Multilingual SemEQUAL

The summary function implementing SemEQUAL is shown in Figure 4 (details are available in [16]). The SemEQUAL functionality needs two significant steps (both in line 3): computation of the closure of the synsets corresponding to the RHS operand and testing non-empty intersection of the set of synsets corresponding to the LHS operand and the computed closure of the RHS operand.

SemEQUAL ($String_{Data}$, $String_{Query}$, L_D, L_Q, *match*, $\mathcal{T}_\mathcal{L}$)
Input: $String_{Data}$ and $String_{Query}$ in languages L_D and L_Q, *match* flag, Target Languages $\mathcal{T}_\mathcal{L}$
Output: `TRUE` or `FALSE`, [*Optional*] Gloss of Matched Synset

1. ($\mathcal{W}_\mathcal{D}$,$\mathcal{W}_\mathcal{Q}$) ← WordNet Of (L_D,L_Q);
2. ($\mathcal{S}_\mathcal{D}$,$\mathcal{S}_\mathcal{Q}$) ← Synsets of ($String_{Data}$ in $\mathcal{W}_\mathcal{D}$, $String_{Query}$ in $\mathcal{W}_\mathcal{Q}$);
3. **if** $Match$ **is** `EQUIVALENT` **then if** $\mathcal{S}_\mathcal{D} \cap \mathcal{S}_\mathcal{Q} \neq \phi$ **return** true **else return** false;
   else if $Match$ **is** `GENERALIZED` **then**
   $\mathcal{TC}_\mathcal{Q}$ ← TransitiveClosure($\mathcal{S}_\mathcal{Q}, \mathcal{W}_\mathcal{Q}, \mathcal{T}_\mathcal{L}$);
   if $\mathcal{S}_\mathcal{D} \cap \mathcal{TC}_\mathcal{Q} \neq \phi$ **return** true **else ret urn** false;
4. [Optional.] **return** Gloss of the Matched Synset;

Fig. 4. Semantic Matching Algorithm

In the following discussions, we focus on the `generalized` matching that requires a closure computation, which is inefficient in relational systems. The transitive closure is computed using the (intra-language) Is-A relationships and the (inter-language) ILI relationships stored in the database. In our derived operator approach, the transitive closure of the $String_{Query}$ on WordNet taxonomic hierarchy is computed using the standard SQL:1999 recursive SQL constructs. After computing the transitive closure of the RHS operand, each record is checked for intersection of the synsets corresponding to the LHS operand and the computed closure, returning all records for which the intersection is non-empty. While the closure computation may be optimized by generating the closure only up to the point to determine set membership in the second step, such optimizations are not possible in the derived operator approach. Also, we restricted the closure computation only to the target languages, thus keeping the complexity linear

in the number of target languages. Testing the set membership in the second step may be implemented efficiently using well-known hash-table techniques.

2.4 Semantic Matching Example

We present an example to illustrate the *derived-operator* implementation of the SemEQUAL function. The WordNet resource is stored in the $\mathcal{W}_\mathcal{L}$ table. The user query,

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

is mapped to the following query, where the transitive closure on $\mathcal{W}_\mathcal{L}$ is computed using the recursive SQL constructs and the set membership is tested by the SQL `IN` predicate:

```
WITH Descendants (child, lang)
  (SELECT W_L.sub, W_L.lang FROM WordNet W_L WHERE
   W_L.super = 'History' AND W_L.lang IN ('ENGLISH','FRENCH','TAMIL')
  UNION ALL
   SELECT W_L.sub, W_L.lang FROM WordNet W_L, Descendants Dec WHERE
   W_L.parent = Dec.child AND W_L.lang = Dec.lang)
SELECT Author, Title FROM Books
WHERE Category IN (SELECT child FROM Descendants)
```

Thus, the user query effectively translates to the following SQL query:

```
SELECT Author,Title from Books
WHERE Category IN {'History','Memoir','Autobiography', ...
'Histoire','Mémoire', ... 'சரித்திரம்','சுயசரிதம்'...}
```

Here, the values in the IN clause are a few of the subclasses of `History`, in English WordNet, and their equivalents in French and Tamil WordNets. Note that any conjunction (disjunction, respectively) of SemEQUAL predicates can be handled by computing the intersection (union, respectively) of closures for the IN predicate.

3 Experimental Study

In this section, we describe our experimental setup to measure the performance of the SemEQUAL derived operator, on a suite of commercial database systems.

3.1 System Setup

A standard Pentium IV workstation with 512 MB memory running Windows NT operating system, was used as the experimental platform. Three database systems, IBM DB2 Universal Server (Ver. 7.1.0), Microsoft SQL Server (Ver. 8.00.194), and Oracle 9i (Ver. 9.0.1), were installed with default configurations. Of these three, DB2 and Oracle support recursive SQL natively, while the functionality is simulated through scripts in SQL Server. In subsequent sections, the systems are identified randomly as *A*, *B* and *C*, to conceal their identities.

3.2 WordNet Storage

The entire set of noun taxonomic hierarchies of WordNet (Version 1.5), totaling about 110,000 *word forms*, 80,000 *synsets* and about 140,000 relationships between them, was loaded into each of the database systems, in a simple hierarchy table (as `Parent-Child` relationships). We calculate the storage space requirements of each WordNet to be about 4 MB (including index storage), based on the profile of English Wordnet (shown in Table 1). Assuming that the WordNet of each language will be similar to that of English when fully developed, the storage needed to store WordNet in non-Latin script, is about 8 MB, due to the need for Unicode format.

3.3 Query Workload

For profiling the performance of the SemEQUAL operator, we used queries that compute closures of varying sizes, from a few hundreds to a few thousands, on the above taxonomic hierarchy. Queries based on SQL:1999 recursive SQL constructs (as shown in Section 2.4) were used, with appropriate query terms to compute closures of the necessary sizes.

To establish the *likely* closure size (*i.e.*, the average closure size for likely query strings), we selected the top-hundred most used nouns in English [1] and the top-fifty nouns that are used in popular web-search engines [24] and computed the average of *their* closure-sizes in English WordNet, which turned out to be around 625 [16]. Hence, it is realistic to use a figure of around $2,000$ for a representative closure size, assuming that a multilingual user would typically want answers in at most three languages.

3.4 Metrics Measured

In all the experiments, we measured the wall-clock runtime of a given query on the given data set. The queries were run in an SQL or a programming language environment, as appropriate. The test machine was quiesced except for the database system under study and the queries were run cold. The average runtime from several identical runs was taken as the runtime of a specific query (the graphs show mean values with relative half-widths about the mean of less than 5% at the 90% confidence interval).

It should be noted here that the *quality* of the retrieval is determined solely by the coverage (for *recall*) and the resolution power (for *precision*) of the WordNet taxonomic hierarchy. Measurement of such quality is in the domain of behavioral and linguistic experts, and beyond the scope of our research, which focuses solely on optimizing the database performance, given the linguistic hierarchies.

4 Results and Analysis

In this section, we report on the performance of a suite of commercial database systems in computing the SemEQUAL operator, as per the SQL queries described in Section 2. To profile the performance of SemEQUAL working with *fully developed* linguistic resources, we used the following strategy: We first profiled the structural characteristics of WordNets, as they exist now, and the results are given in Table 1.

Characteristic	English	French	German	Spanish	Hindi
Word Forms (Words)	114,648	32,809	20,453	50,526	22,522
Word Sense (Synsets)	80,000	22,745	15,132	23,378	7,868
Average Synsets per Word Form	2.236	2.176	2.301	2.360	3.889
Average Word Forms per Synset	1.985	1.442	1.352	2.162	2.286
*E*quivalence Relations per Synset(to English)	1.000	0.999	1.080	0.908	Not Available

Table 1. Statistical Profile of WordNets [2, 6]

The statistics of the individual taxonomic hierarchies indicate a very close match between the WordNets. In addition, since both Euro and Indo WordNets have conformance to English WordNet as their stated design goal, it is reasonable to expect their structures to be similar to that of English WordNet, when fully developed. Since the English WordNet is the most developed at this point of time, we replicated English WordNet in Unicode format and created ILI links between every English synset and its corresponding synset in Unicode. The resulting taxonomic hierarchy is used in the performance experiments.

4.1 Closure Computation – Baseline

For the baseline performance experiments, the interlinked WordNet taxonomic hierarchy (in Unicode format to simulate multilingual environments, as discussed earlier) was stored and queried, as specified in Section 3. The query strings for the experiment were chosen so as to result in the computation of closures of varying sizes. The *SQL-Baseline* performance (in seconds) for the basic closure computation in the three database systems (with out and with B+ tree index) is given in Figure 5 (shown in *log-log* scale). As can be observed here, the closure computations for all the systems take up to hundreds of seconds without index support and up to a few seconds even with an index. Though the variations in performance may be attributed to the respective algorithms and optimization techniques – details in [16], the net result is that the performance is unsuitable for *e-Commerce* deployments, if the size of the closure exceeds a few hundred items.

In the following sections, we outline two different (and mutually exclusive) optimization techniques that improve the performance in *System B*, which exhibits the worst indexed performance.

4.2 Optimization #1: Precomputed Closure

First, we used a standard optimization technique – *pre-computing* the closures of every element in WordNet and *storing* them explicitly as the immediate children of the corresponding element; thus, the closures could be found with a simple scan of the enhanced table. We also explored the possibility of further reducing the cost of computation by building an index on the parent attribute of the pre-computed table.

We ran the transitive closure query on the resulting data set, and the performance, with and without the index, is presented in Figure 6 (the graph is shown in *log-log* scale).

We observe here an improvement in performance, to about 7 seconds (without index) for the Unicode WordNet. Understandably, the closure computation takes approximately the same time for all sizes of the closure, since only a table scan is needed. With the index, as expected, the runtime is reduced by an order of magnitude from the baseline index performance, to just under one second. However, this gain comes with the penalty of enormous storage costs: the space requirements of the taxonomic tables are increased by about 20 times, to roughly 120 MB (and an additional 45 MB for index).

4.3 Optimization #2: Reorganizing Schema

We now move on to an alternative performance optimization strategy with much smaller space overheads. This strategy is based on leveraging the *distribution* of synsets in the WordNet hierarchy to reduce the calls to the expensive recursive SQL statements. We first computed and plotted the fan-out of subclasses for every parent node in English WordNet, as shown in Figure 7. The plot of the fan-out exhibits a characteristic *power-law* distribution with an exponent of -2.75. Further analysis indicated that only a small number of synsets (*less than* 10%) have a large number of children (*more than* 16), with the large majority having only a few children[2]. This distribution suggests a new, more efficient organization of WordNet hierarchy, where a certain number of sub-classes may be *inlined*. We chose to inline those synsets with upto 16 subclasses in a new taxonomy table, reducing the number of records in the new taxonomy table to about a tenth of that of the original table. All synsets with greater than 16 subclasses remained in the original table. The closure computation algorithm is modified to access the inlined table for all synsets with less than 16 children, or the original table, otherwise. The overall size of the table (in terms of number of tuples) reduces by about 90%, though the storage size remains about the same as the Baseline (about 8 MB for Unicode WordNets).

For the above schema, the performance of the closure queries – with and without indexes – are shown in Figure 8 (the graph is drawn to a *log-log* scale). As can be observed from the figure, the performance with reorganized schema is speeded up by 2 orders of magnitude on the plain table, and by 3 orders of magnitude on the indexed table, with *no perceptible increase in storage requirements from the baseline*.

4.4 Scaling of Performance with Languages

Finally, we explore how the performance behaves as function of the number of languages being considered for query processing. The runtimes for the typical query, computing a transitive closure of approximately ≈ 600 is shown in Figure 9. We observe a near-linear increase in both *pre-computed closure* and *re-organized tables* methodologies, with the number of languages. Further, even with about 8 languages, the index-based runtimes for the typical query remained within a few tens of milliseconds, which appears sufficiently small to support online interaction for a multilingual user.

Thus, we show that a new semantic multilingual matching functionality may be added to current relational database systems by integrating standard linguistic resources, and leveraging only on existing SQL features. Further, we show the performance of this

[2] The fan-outs in Hindi and English WordNets (in Figure 7) exhibit a very similar profile differing only in scale, suggesting the applicability of power-laws in linguistic domains as well.

matching may be sufficiently optimized to support online-user interactions for multilingual e-commerce applications.

5 Related Research

To the best of our knowledge, multilingual semantic matching of attribute data – by integrating standard linguistic resources with the database engine, has not been discussed, previously in the literature. With respect to **Semantic Query Processing**, no standards have been specified in SQL and hence there is no uniformity among systems in such support. All systems support some level of semantic querying, based on NLP techniques, but are un-suitable for for attribute level matching. The WordNet based approach was used for semantic information retrieval in [19], where the emphasis was on *quality* of the results and not performance; our work on performance of such retrievals is complementary to this research. There are vast amounts of literature in the **Information Retrieval** Research community in the areas of Knowledge-based and Natural-language based retrieval. The techniques employed are diverse, ranging from syntactic and morphological analysis [8] to Machine Translation [5], statistical techniques [9], and Latent Semantic Indexing [4] for semantic querying in a single language, and to paired dictionaries [20] techniques for handling cross-language querying. We refer to the Multilingual Information Retrieval Track of the ACM SIGIR conference for a survey of current techniques. Such techniques do not perform well on attribute level data in OLTP type environments. Initiatives, such as the **Semantic Web** [21] are appropriate for meta-data management in the web domain, but not for database query processing. Finally, the existence of several **International WordNet** initiatives [3, 6, 13], with a stated objective of following similar taxonomical structures, is an enabling resource, for realizing our proposal.

6 Conclusions

In this paper, we proposed a new SQL functionality – SemEQUAL – to support seamless multilingual text data matching, based on semantics, to cater to increasingly multilingual user requirements in e-commerce deployments. Our proposal outlines a light-weight approach for implementing this feature by adopting and integrating the WordNet linguistic resource in the database system. Multilingual text attribute data are matched after transforming them to a canonical semantic form, leveraging on the rich cross-linked taxonomic hierarchies in WordNets. As a side effect, such a methodology provides a repeatable and consistent result set for a given data set across different database systems.

We outlined a *derived-operator* approach for implementing the SemEQUAL operator, using standard SQL:1999 constructs. Our performance experiments with real WordNet data on three popular commercial database systems, underscored the inefficiencies in computing transitive closure, an essential component for semantic matching. The runtimes are in the order of a few seconds, unsuitable for practical deployments. We proposed optimization techniques, by tuning the storage and access structures to match the characteristics of linguistic resources, and demonstrated that the closure computation may be speeded up by nearly 3 orders of magnitude – to *a few milliseconds*

– to make the operator efficient enough for supporting online user query processing. These results underscore the viability of the SemEQUAL functionality for immediate practical use. Finally, we expect that for specific applications, semantic matching using domain-specific ontological hierarchies, may also benefit from a similar approach to those outlined in this paper.

Acknowledgements We thank Dr. P. Bhattacharyya, Coordinator of Center for Indian Language Technology at IIT-Bombay, for providing us with details on Hindi WordNet.

References

1. The British National Corpus, Oxford University Press. *http://www.comp.lancs.ac.uk*.
2. Centre for Indian Language Technology, IIT-Bombay. *http://www.cfilt.iitb.ac.in*.
3. H. Chen, C. Lin and W. Lin. Building a Chinese-English WordNet for Translingual Applications. *ACM Transactions on Asian Languages Information Processing*, 2002.
4. S. Deerwester, S. T. Dumais and W. C. Ogden. Indexing by Latent Semantic Analysis. *Jour. of American Soc. of Information Sciences*, September 1990.
5. The EuroSpider. *http://www.eurospider.ch*.
6. The Euro-WordNet. *http://www.illc.uva.nl/EuroWordNet*.
7. C. Fellbaum and G. A. Miller. WordNet: An electronic lexical database (language, speech and communication). *MIT Press*, 1998.
8. C. Fluhr *et al.* Multilingual Database and Crosslingual Interrogation in a Real Internet Application. *AAAI Sym. on Crosslanguage Text and Speech Retrieval*, 1997.
9. F. Gey, A. Chen, M. Buckland and R. Larson. Translingual Vocabulary Mapping for Multilingual Information Access. *Proc. of 25th ACM SIGIR Conf.*, 2002.
10. The Global WordNet Association. *http://www.globalwordnet.org*.
11. J. Han et al. Some Performance Results on Recursive Query Processing in Relational Database Systems. *Proc. of 2nd ICDE Conf.*, 1986.
12. Y. Ioannidis. On the Computation of TC of Relational Operators. *Proc. of 12th VLDB Conf.*, 1986.
13. B. D. Jayaram and P. Bhattacharyya. Report on Indo-WordNet Workshop. *Central Institute of Indian Languages*, January 1999.
14. A. Kumaran and J. R. Haritsa. On Multilingual Performance of Database Systems. *Proc. of 29th VLDB Conf.*, 2003.
15. A. Kumaran and J. R. Haritsa. Supporting Multiscript Matching in Database Systems. *Prof. of 9th EDBT Conf.*, 2004.
16. A. Kumaran and J. R. Haritsa. Multilingual Semantic Operator in SQL. *Technical Report TR-2004-03, DSL/SERC, Indian Institute of Science*, 2004.
17. M. Liberman and K. Church. Text Analysis and Word Pronunciation in TTS Synthesis. *Advances in Speech Processing*, 1992.
18. The Computer Scope Ltd. *http://www.NUA.ie/Surveys*.
19. R. Richardson and A. F. Smeaton. Using WordNet in a Knowledge-based Approach to Information Retrieval. *Working Paper CA-0395, Dublin City University*, 1999.
20. D. Soergel. Multilingual thesauri in cross-language text and speech retrieval. *AAAI Sym. on Cross-Language Text and Speech Retrieval*, March 1997.
21. The Semantic Web. *http://www.w3.org/2001/sw*.
22. The WebFountain. *http://www.almaden.ibm.com/WebFountain*.
23. The WordNet. *http://www.cogsci.princeton.edu/w̃n*.
24. Word Discover. *http://www.worddiscover.com*.

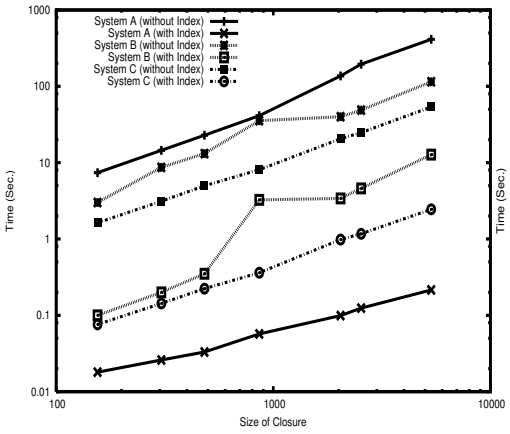

Fig. 5. Baseline Performance

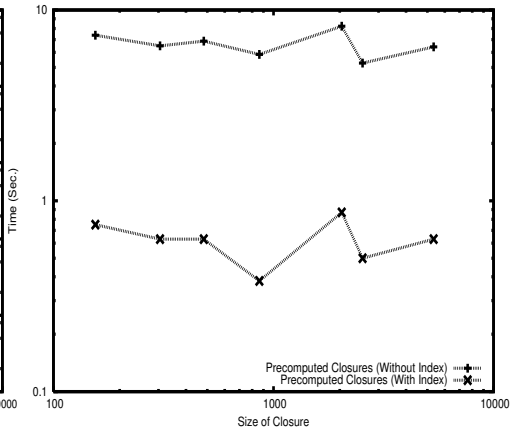

Fig. 6. Precomputed Closures

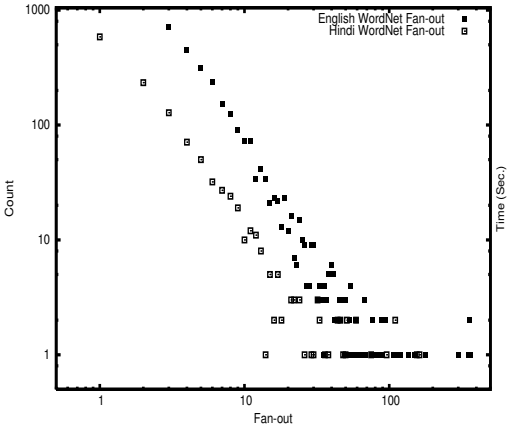

Fig. 7. WordNet Fan-out Plot

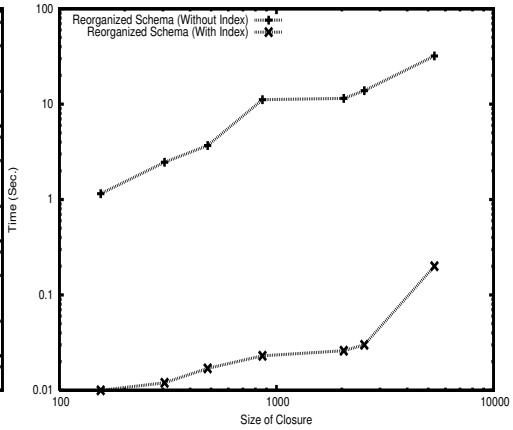

Fig. 8. Re-Organized Schema

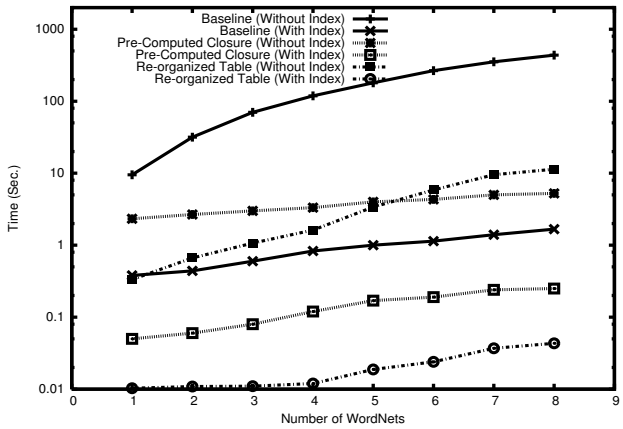

Fig. 9. Scaling of Computing Closure