Abstract

Efficient storage and query processing of data spanning multiple natural languages are of crucial importance in today's globalized world. A primary prerequisite to achieve this goal is that the principal data repositories, relational database systems, should efficiently and seamlessly support multilingual data. Our survey of current relational systems indicates that while they do support storage and management of multilingual data, querying is restricted to be within a given language, with no crosslingual query support. Further, quantitative performance study of the systems working on different character sets has not been published so far and therefore is an open issue.

In this thesis, we first profile the multilingual performance of a set of current relational database systems, using an environment based on the TPC benchmark suites. The results indicate a significant performance degradation while handling multilingual data. While the differential performance is huge when disk traffic is a factor, it is substantial even when only in-memory processing is considered. To address this inequity, we propose a split representation format that reduces the multilingual storage space and largely eliminates the differential performance for most languages except those with unusually large repertoires.

Next, we propose functionality enhancements that complement the standard lexicographic matching, specifically in the multilingual text space. Two new multilingual join operators – one for joining names across languages and the second for joining multilingual categories based on their meanings – are proposed and formally defined. These operators are implemented in an outside-the-server approach using existing SQL features of relational systems, and using standard linguistic resources. While the performance...
of these basic implementations is too slow for real-world deployments, a host of optimization techniques that tune the schema and index choices to match typical linguistic features are employed and shown to improve the performance to a level sufficient for practical use.

Finally, for a full integration of multilingual functionality with the database engine, we specify a query algebra with a new multilingual storage datatype and the above join operators. The operators are implemented \textit{natively} as first-class features in an open-source database system, along with all components that are required to leverage the relational query optimizer, specifically, the operator cost models and their selectivities. The performance experiments indicate that this native implementation of the multilingual operators improves the performance significantly over the outside-the-server implementation. Further, the power of the algebra is demonstrated through selection of better execution plans for queries using the multilingual operators.

In summary, this thesis presents a multilingual query processing architecture, with a set of functionalities, algorithms, implementation and optimization techniques, all geared towards the goal of developing \textit{natural-language-neutral} database engines.