Comparable datasets in performance benchmarking

David Steier
Price Waterhouse Technology Centre
68 Willow Road
Menlo Park, CA 94025
steier@tc.pw.com

Even if you can't compare apples and oranges, maybe you can still make a nice fruit salad

A number of tasks require gathering information about a collection of similar objects to perform a comparison. When the information needed to perform these tasks comes from a single database, the amount and the type of data retrieved about each object in the collection is likely to be very similar, and the task of comparison relatively straightforward. But when information comes from many sources, information gatherers face a problem of producing a common comparable dataset for each object being compared. This problem is difficult because what should be in a comparable dataset (as we show in this paper) depends on the task for which the information is being gathered, the target collection of objects to report on, and the data available about each object. The purpose of this workshop paper is to highlight the importance of this problem in gathering information from heterogeneous sources and to present some detail about a case study encountered in practice while doing a performance benchmarking study. Aspects of producing compsets have been studied in the database literature within the area of schema integration for heterogeneous databases [Batini et al., 1986], because of the shared concern for semantic comparability at the schematic level. For example, the theory of semantic values developed by Sciore et al. [1994] seems like a promising approach to computing comparable datasets because of the explicit representation of contextual information for each value. We discuss a number of issues involved in using contextual information in this way.

Using SEC filings in benchmarking

The general problem we are trying to solve is how to extract and classify data from financial statements in Securities and Exchange Commission’s (SEC) filings to facilitate valid comparisons for benchmarking. The information source is the SEC’s EDGAR system, which grows by the equivalent of over 10 million pages a year (the EDGAR filings are accessible over the Internet through World-Wide Web servers at http://town.hall.org/edgar and http://edgar.stern.nyu.edu). There is heterogeneity in the data because of the substantial variation among the 15,000 public companies that will file on EDGAR in how they report their financial results. Benchmarking attempts to give a picture of how one company is performing in relation to its competitors on some set of uniformly applied measures. For example, Novell, Inc. might ask a question such as “How are we doing in relation to competitors who produce networking software?”

Benchmarking often centers on financial measures such as profitability, solvency, efficiency, etc., so that a common first step in benchmarking is to extract information from company financial statements. There is usually a large manual component to this step because of the need to review the financial statements by hand. A number of commercial CD-ROM or online databases containing this information provide indexing that is supposed to make the search and comparison easier. Unfortunately, the indexing provided is often very generic, with keys that could apply to all industries represented in the database, so usually industry-specific measures are not indexed. There are also a surprising number of misclassifications and data entry errors (since numbers are often rekeyed manually without automatic integrity checks). Indeed, the searches based on such indexing are so unreliable that most experienced users resort to examining all possibly relevant individual records manually when accuracy is needed. Therefore we set ourselves a goal of building a benchmarker's assistant that could reduce significantly the effort involved in creating industry-specific comparisons.

A sample of the type of question we would like to answer, one which arose in the context of a study we performed for a client, illustrates some of the information gathering challenges in benchmarking. We were asked to benchmark the productivity of a company’s software operations against independent software vendors who sold competing products. In itself, this is an ill-structured problem, as productivity can be mea-
Scaling factors and units: The numbers from specific to benchmarking:

- **思量因素和单位**：涉及的数字是从具体的到用于基准测试的。

We devised a measure that was equivalent to the concept of "operating margin" for a manufacturing company: revenues after costs of software creation as a percentage of total revenues. We intended the costs of software creation to include not just items like packaging, and assembly which would be included in traditional "cost of goods sold" measures, but also the cost of developing and maintaining the software. The higher the percentage operating margin, the more profitable the company would be on an operating basis, and the more productive the software development activities of the company.

As reasonable as this measure seemed to us, we found a number of complications in deriving the measure from the source data we had in the 10-K filings. Here are the relevant line items from the income statements of three of the companies in the networking software business we studied:

**Novell:**
- Net sales: 1,122,896
- Cost of sales: 224,531
- Product development: 164,860

**FTP Software:**
- Product revenue: 55629
- Service revenue: 3097
- Total revenue: 58726
- Product cost: 3949
- Service cost: 3979
- Total cost of revenues: 7928
- Gross margin: 50798
- Product development: 9389

**Banyan Systems:**
- Software: 94231
- Hardware: 14980
- Support and training: 18559
- Total revenues: 127770
- Software: 14116
- Support and training: 11002
- Total cost of revenues: 30729
- Product development: 15455

These were extracted by hand, for the purposes of this benchmarking study to understand what data we would have to represent for the benchmarker's assistant. The following items were among those identified as relevant for benchmarking purposes (the first two have been identified previously in the database literature (i.e. [Kim and Seo, 1991], while others are more specific to benchmarking):

- **Cost components:** Just as for revenues, costs should be compared across similar products and types. We had to look at the footnotes to determine what the costs represented: these included documentation, media, distribution, packaging as one would expect, but also royalties and software amortization costs. In some cases, the cost of software support was included as a cost of sales, while in others, it was included as a cost of product development, and in others it was split depending on the type of support (technical support hotlines versus ongoing maintenance and bug fixes). It is in fact so difficult to get a consistent definition of what is considered cost of sales and what is considered product development that our measure of "software operating margin" considered revenues net of the sum of both types of costs.

- **Revenue components:** To ensure valid comparisons for benchmarking, revenue numbers must be compared across similar product areas and income types. Our comparison focused on revenues from networking software, as opposed to hardware or other types of software, such as operating systems, programming tools, or applications. It also focused on revenues from licensing software, as opposed to services such as consulting or training.

- **Capitalized and amortized amounts:** Some software development companies record all development costs as expenses to be charged against revenues on the income statements. But accounting standards (FAS 86) also allow a company the option of recording development costs after technological feasibility is demonstrated as assets, capitalized on the balance sheet. Capitalized amounts are then gradually amortized (charged as expenses) over some number of years. Novell and FTP software record all development costs as expenses, while Banyan Systems capitalizes a small percentage of its costs.
more accurate accounting would compare revenues and costs for each product over its entire life cycle, which could span several years, but companies generally do not break down this information in their public filings for competitive reasons. Instead, we decided to compare costs on a "current year expenditures" basis, adding the capitalized amounts ($703K in Banyan's case) to the reported costs, and subtracting the amortized amounts ($723K for Banyan) from the reported costs.

Taking into account these factors, the comparable dataset we devised computes revenues from licensing networking software minus costs of those revenues on a current fiscal year expenditure basis as a percentage of those revenues. For those who are curious, or wish to replicate this exercise, we found Novell's margin to be 65%, Banyan Systems, 64%, and FTP Software, 71% for fiscal year 1993.

**Issues and Directions**

As we've been constructing the benchmarker's assistant based on our experience, we've thought about possible representations for the contextual information we've identified. A number of database researchers have proposed schemes for representing such information (e.g. [Bhargava et al., 1991]) to detect conflicts in database integration. Recent work by Sciore et. al. [1994] has focused on the use of contextual information to aid in data conversion so that heterogeneous systems can interoperate. Sciore et al. defined a semantic values data value associated with a set of meta-attributes considered to be its context. Comparisons of semantic values are specified with respect to a target context, which is a set of meta-attributes to be considered in the comparison. Conversion functions transform semantic values from one context to another to achieve comparability. For example, if the fiscal year endings differed between two companies, a conversion function could choose a target fiscal year end, and compute "virtual fiscal year results" based on quarterly data. This would assume that data for the four quarters prior to the chosen fiscal year end were available, and each company being studied has a quarter that ends on the chosen target fiscal year end, or else the differences of one or two months are not deemed significant.

Semantic values (and perhaps also other meta-data schemes we haven't examined as closely) seem to have the expressive power we need for representing our contextual information. Semantic values are hierarchical, so values for meta-attributes can have their own contexts, and so on. Those trained in AI will find some of this familiar, as semantic values can be viewed as a kind of frame representation. Expressiveness aside, however, we found serious challenges with contextual information that we are still struggling with:

- **Deciding what contextual information to represent:** There is no shortage of possible meta-attributes for financial data, given the complexity of the domain. It is an open question whether there is a systematic basis for generating possible meta-attributes, and deciding which to include could be crucial in a database with hundreds of thousands of data items. We found design of the scheme to be a difficult process, consisting of reviewing the source filings in collaboration with domain experts, designing and testing a proposed scheme, and revising it, limited primarily by the time available.

- **Acquisition of contextual information:** Little is known about how the values for complex meta-attributes can be computed efficiently, especially when there are already gigabytes of pre-existing data. We are developing a system, called RELATE (for REAL-time Access To Edgar) that extracts the values of such meta-attributes from the SEC filings automatically, which is possible because EDGAR filings are instances of structured documents. In such documents, the expectations about what information will be in the document are strong enough to make information extraction far more tractable than would be the case for unrestricted natural language (although not as straightforward as querying a relational database). We have built parsers that create annotated internal representations of the content of financial tables very quickly, much as a natural language parser would produce a parse tree for a sentence. The requirement for efficiency is crucial, since at peak filing periods the SEC has received as many as 300 10-K filings an hour, and 10-K filings range from several dozen to several hundred pages in length. RELATE processes a 10-K on a Sparc 2 in an average of eight seconds, which is fast enough to handle the peak data rates. We are currently generalizing our system to apply to other types of structured documents.

- **Task influences on use of contextual information:** The literature generally posits fixed schemes for using contextual information. For example, the convention of Sciore et al., is to compute the target context as the intersection of the sets of meta-attributes available for the items being compared. An alternative suggested by Sciore, et al. is to use an explicit generalization hierarchy instead of intersection to find the generalization of the items being compared. This approach towards generalization and specialization is used to reformulate requests for information source selection in the SIMS project [Arens et al., 1993]. Unfortunately, we found that whether a generalization is valid varies from task to task, as does the strategy for converting values. One possible generalization strategy we found was to produce a common profile of each data object taking into account the maximal amount of comparable detail available, the maximal comparable dataset. Thus, if one had to, one could compare "net sales" and "income from operations" for Novell to "interest income" and "net interest income" for the Bank of
San Francisco, respectively. Dividing the latter by the former in each case yields a comparable "operating margin." But if certain types of income (e.g. "software licensing") and expenses (e.g. "product development") were common within an industry and all companies compared were within that industry, then a more detailed breakdown and comparison would be possible. An alternative to a maximal comparable dataset is a directed comparable dataset, in which one company is designated as the primary object of the study and data from other companies is converted to that the context of that company. This is especially true if detailed segment data is available for the primary target company, enabling one to carve out "virtual companies" out of larger ones. This approach enables comparisons of diverse companies to companies with more focused product lines by considering results for only directly competing product lines, i.e. what would Novell's results look like if it only produced network software products that directly compete with Banyan's. Given the experience of AI with representations such as frames and MOPS, there may be some leverage in exploiting AI techniques such as unification or case-based reasoning to compute the necessary generalizations. Again, the primary issue is how to make the processing in such techniques sensitive to constraints from the task environment, such as requirements on accuracy, resource usage, etc.

We believe that these issues would be found in some form whether we used semantic values or other schemes to use contextual information to assess comparability. Despite all the progress that has been made in knowledge representation and processing, it seems that the bulk of the effort in creating a benchmark's assistant will be in instantiating the representation scheme and creating the conversion processes for the complexities of a particular domain. We welcome assistance in finding appropriate technology that would make acquiring the domain-specific knowledge easier for the next domain.

Acknowledgments

Others at the Price Waterhouse Technology Centre have provided a great deal of the contextual information used to write this paper. Susan Darwin and Mike Katz performed much of the benchmarking study upon which the observations are based, and Scott Huffman assisted with both the content and form of this draft. Bill Kornfeld was the primary implementor of the RELATE system as a consultant to Price Waterhouse.

References


