Data Cleaning in Microsoft SQL Server 2005

Surajit Chaudhuri, Kris Ganjam, Venky Ganti, Rahul Kapoor, Vivek Narasayya, Theo Vassilakis[1]
{surajitc, krisgan, vganti, rkapoor, viveknar}@microsoft.com
Microsoft Research
One Microsoft Way
Redmond WA 98052, USA

ABSTRACT

When collecting and combining data from various sources into a data warehouse, ensuring high data quality and consistency becomes a significant, often expensive, challenge. Common data quality problems include inconsistent data conventions amongst sources such as different abbreviations or synonyms; data entry errors such as spelling mistakes; missing, incomplete, outdated or otherwise incorrect attribute values. These data defects generally manifest themselves as foreign-key mismatches and approximately duplicate records, both of which make further data mining and decision support analyses either impossible or suspect. We demonstrate two new data cleansing operators, *Fuzzy Lookup* and *Fuzzy Grouping*, which address these problems in a scalable and *domain-independent* manner. These operators are implemented within Microsoft SQL Server 2005 Integration Services. Our demo will explain their functionality and highlight multiple real-world scenarios in which they can be used to achieve high data quality.

1. INTRODUCTION

Data warehousing efforts often aim to consolidate data from heterogeneous sources in hopes of providing a unified view of the data that can be used for business decision support, customer relationship management and a large number of other data analysis tasks. Accuracy of such analyses is crucial and relies upon the accuracy of the data loaded into the warehouse. However, data received at the data warehouse from external sources usually contains errors, e.g., spelling mistakes, inconsistent conventions across data sources, and/or missing fields. Significant amounts of time and money are consequently spent on *data cleaning*, the task of detecting and correcting errors in data.

An effective approach to building a clean data warehouse is to prevent the introduction of errors as early as possible in the process of loading new data into the warehouse. This requires input tuples to be validated and corrected before they are inserted. To perform such cleaning, some information or model of the data is necessary. Traditionally, this has been in the form of hand-crafted rules and logic that apply exclusively to the given domain of the data. For example, many data cleaning products, such as Trillium [TR05], focus on specific domains such as postal addresses. While such solutions can be useful, the wide spectrum of data domains found on the web and in relational databases make construction of manual cleaning routines infeasible for each and every data domain and distribution. Fortunately, data warehouses often contain large amounts of existing data that can be leveraged to clean incoming dirty data. Even when this data has quality issues itself, or when the warehouse has no existing data relevant to the domain of the incoming data, there is information even in the dirty data that can be used to improve the overall data quality. We introduce the *Fuzzy Lookup* and *Fuzzy Grouping* operators and explain how each can be used to perform *domain-independent* data cleaning with the data at hand.

Fuzzy Lookup Scenario: One common cleaning technique validates incoming tuples by looking them up against *reference relations* consisting of known-to-be-clean tuples. These reference relations may be internal to the data warehouse (e.g., customer or product relations) or obtained from external sources (e.g., valid address relations from a postal service). An enterprise maintaining a relation consisting of all its products, for example, may ascertain whether or not a sales record from a distributor describes a valid product by matching the product attributes (e.g., Part Number and Description) of the sales record with the Product reference relation. If the product attributes in the sales record match exactly with a tuple in the Product relation, then the described product is likely to be *valid*. However, due to errors in sales records, the input product tuple often does not match exactly with any record in the Product relation. In such cases, a decision must be made as to whether the input product tuple represents a new product (that should probably be added to the reference relation), or whether it is merely a non-standard variation of one already present.

To address this question, we introduce the *Fuzzy Lookup* operation. Given an input tuple, this operator returns the top-k fuzzy matching tuples in the reference relation and provides numeric measures of both textual similarity and confidence that the top match is indeed the desired result. As illustrated in Figure 1, if an Exact Lookup against the reference table fails, the input tuple is then passed to a Fuzzy Lookup against the reference table. If the resulting match has a textual similarity less than some user-specified threshold, then the input tuple is routed for further cleaning before considering it as referring to an existing entity. A fuzzy match operation that is resilient to input errors can effectively prevent the proliferation of *fuzzy duplicates* [HS95] in a relation, i.e., multiple tuples describing the same real world entity.

SIGMOD 2005, June 14–16, 2005, Baltimore, Maryland, USA.

[1] Work done while author was at Microsoft.

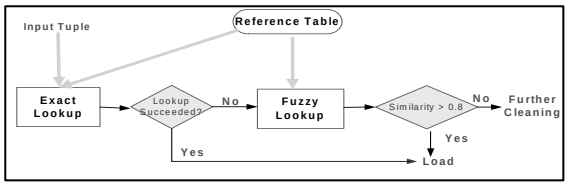

Figure 1: A Template for using Fuzzy Lookup

Fuzzy Grouping Scenario: Consider the scenario where an enterprise has collected or purchased large amounts of customer data, perhaps from several sources, for a targeted mailing campaign. In order to avoid duplicated mails to the same customer and to reduce mailing costs, the enterprise has to detect and eliminate duplicated records from the union of these customer records. For example, the records [Lisa Simpson, Seattle, WA, USA, 98025] and [Simson Lisa, Seattle, WA, United States, 98025] may represent the same individual. In several such scenarios, the same logical real world entity has multiple representations in a relation due to data entry errors, varying conventions, and a variety of other reasons. Such duplicates could also cause incorrect results in analytic queries (say, the number of SuperMart customers in Seattle), and lead to erroneous data mining models. The *Fuzzy Grouping* operation helps solve this problem by taking a relation as input and outputting groups of mutually-similar records along with a suggested canonical record that best represents each group. Like Fuzzy Lookup, a textual similarity score is reported for each record.

1.1 Technical Overview

This section gives a brief overview of some technical details behind the Fuzzy Lookup and Fuzzy Grouping operators. A detailed description of the theoretical basis and performance characteristics of these operations can be found in [CGGM03, CGGNV04]. Both operations are implemented as part of Microsoft SQL Server 2005 Integration Services, a newly rearchitected platform designed to make the development of scalable Extract, Transform and Load (ETL) workflows easier. A typical *data flow pipeline* will involve data sources (such as a flat file or a database query) connected to a number of data *transformations* connected ultimately to data destinations (such as the final warehouse database). Data flows from the sources, is transformed by the transforms and is then output at the destinations. *Fuzzy Lookup* and *Fuzzy Grouping* are data transformations in this framework and can be seamlessly combined with numerous other data transformations to create very powerful ETL solutions.

Fuzzy Lookup: There are two major interacting components which allow Fuzzy Lookup to robustly and scalably retrieve fuzzy matches from large reference relations. The first is a similarity function which returns a numerical measure of how similar one tuple is to another and the second is an indexing structure that allows tuples deemed similar, according to the similarity function, to be found efficiently. The similarity function we chose is based upon notions of string edit-distance combined with information-theoretic weighing of individual tokens and is generally applicable across a wide variety of domains. Essentially, by analyzing the data distribution of the reference relation, we derive a similarity measure that might, for instance, learn that "Microsft Corp" and "Microsoft Corporation" are quite similar, while "Microsoft Corporation" and "Microstar Corporation" are very dissimilar, even though, by naïve edit-distance, the latter two would be more similar than the first two. To efficiently perform error-tolerant retrieval of matches, the operator uses an inverted index structure that is augmented with sampled q-grams of the reference relation tuples. This allows us to quickly locate reference tuples that match on high information-content tokens and still be robust to character-level spelling mistakes.

Fuzzy Grouping: This operator works by first efficiently finding neighbors of each input tuple and then performs a grouping operation that finds clusters of tuples, giving preference to situations where there is a strong central tuple that is similar to all tuples in the group. Internally, the Fuzzy Lookup operation is used to find all relevant records that have textual similarity above a user-specified threshold. This is done by using the entire input to Fuzzy Grouping as the reference relation. This approach allows consistency between both operators in terms of the similarity measure and performs quite well in practice.

2. DEMO OUTLINE

In the sections that follow, we proceed to outline the format of the demo of *Fuzzy Lookup* and *Fuzzy Grouping* and detail the specific data cleaning scenarios that will be presented. Both operators will be demonstrated from within the Integration Services Designer of Microsoft SQL Server 2005 Beta 3. This is a graphical user-interface that allows the data flow pipeline to be visualized, easily configured, operated and debugged. See Figure 2 for a screenshot of the Integration Services Designer, where a sample pipeline is using Fuzzy Lookup to match and route dirty input tuples.

We will give a brief overview of data cleaning and the basic ideas behind *Fuzzy Lookup* and *Fuzzy Grouping.* This will be followed by interactive demos of the transformations running live in the designer for each of the following real-world data cleaning scenarios.

2.1 Fuzzy Lookup Product Descriptions

Like most enterprises, Microsoft maintains repositories of sales data for analysis. It accumulates product sales data from thousands of Microsoft software resellers across the country. Each sales entry contains product descriptions along with quantities of how many units of that item were sold. This data is often hand-entered and consequently requires some amount of data cleaning before accurate sales totals for particular software titles can be reliably estimated. Some examples of dirty inputs which all refer to the same underlying software title are given in Table 1.

Table 1: Real-World Product Sales Data

ID	Product Name	Manufacturer
1	Microsoft Windows XP Professional	Microsoft Corp.
2	MS Win XP Prof.	NULL
3	WinXP	MS
4	Win XP #2231143	Micosoft
5	XP Pro OEM	MSFT

We will demonstrate how the Fuzzy Lookup operation can be effectively used to match incoming dirty data against a large reference table containing millions of valid product names, automating what used to be a strictly manual task.

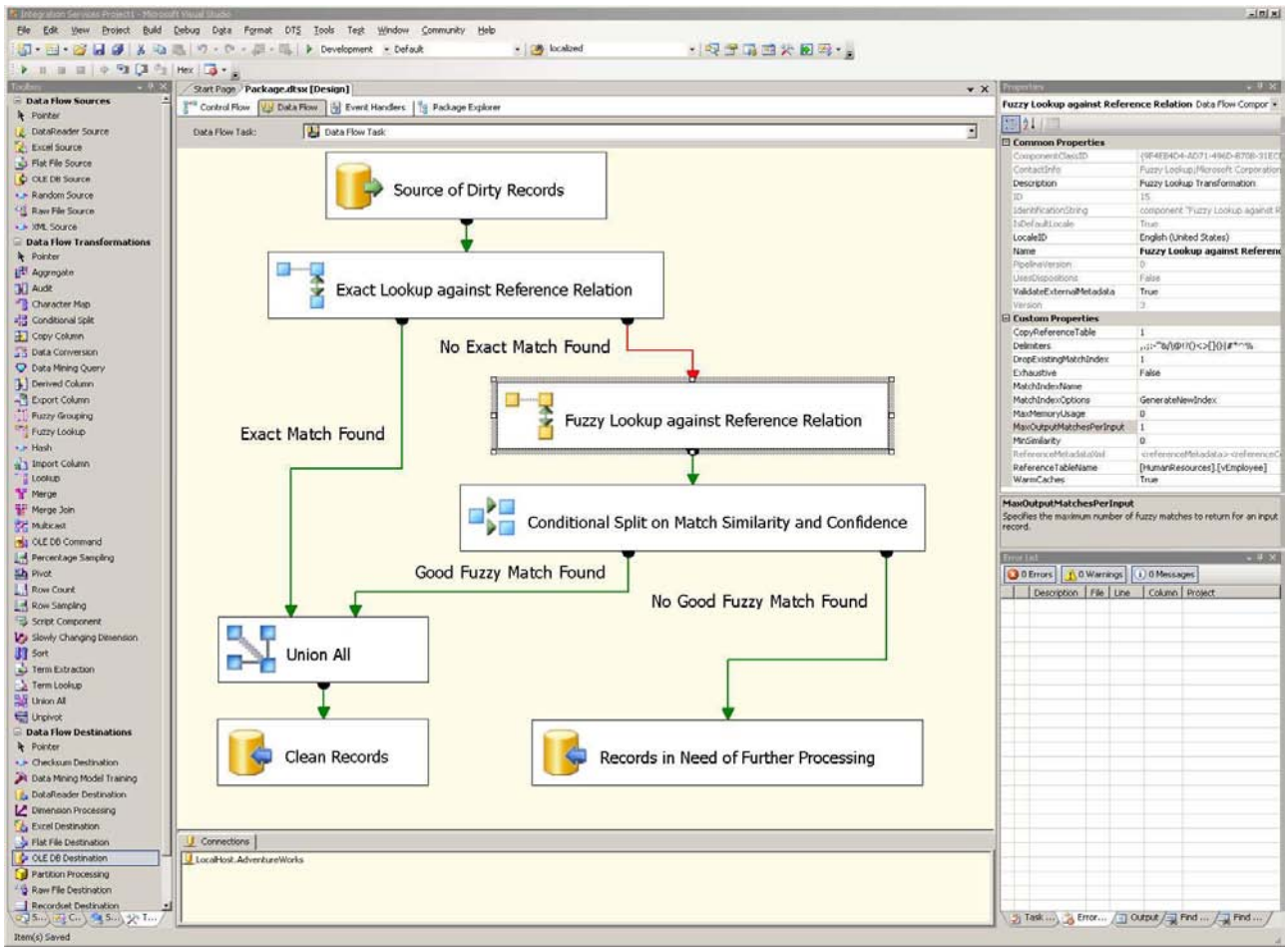

Figure 2: Screenshot of Microsoft SQL Server 2005 Integration Services Designer illustrating a data flow pipeline involving Fuzzy Lookup. Input tuples flow from a source and are first exact matched against a reference relation. If no match was found, the input tuple is fuzzy matched against a reference relation and, depending upon the match results, subsequently routed to either a destination for clean records or a destination for records requiring further processing.

We will highlight several important aspects of the *Fuzzy Lookup*. First, for each fuzzy match result returned, both a *similarity* and a *confidence* score are returned. The first is a measure of how similar the input tuple is to the match result according to our information-theoretic similarity measure. The latter is a measure of how confident we are that the returned result is the *best* match for the input from amongst all the tuples in the reference table. These scores can be used later in the data flow pipeline to decide whether or not to automatically load a particular tuple into the warehouse or whether it should be set aside for further processing or manual review. More details on these scores can be found in [CGGNV04].

2.2 Fuzzy Grouping Windows Media Catalog

Media databases containing song listings and other metadata for hundreds of thousands of compact discs can be purchased from media companies or downloaded from web sources. Users on the internet enter such data on music albums by hand. As one would expect, there are often hundreds of variations entered for a given album, each with a different set of spelling mistakes and data entry conventions.

We will demonstrate how *Fuzzy Grouping* is used to successfully locate and consolidate fuzzy duplicates in this dataset. We will explain various parameters that control how strict or loose the groupings are and how we have overcome issues that are peculiar to datasets where small differences in attribute values can mean that two tuples are in fact distinct entities and should not be considered duplicates.

3. REFERENCES

[CGGM03] S. Chaudhuri, K. Ganjam, V. Ganti, and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the ACM SIGMOD*, June 2003.

[HS95] M. Hernandez and S. Stolfo. The merge/purge problem for large databases. In *Proceedings of the ACM SIGMOD*, San Jose, CA, May 1995.

[CGGNV04] Whitepaper on Fuzzy Lookup and Fuzzy Grouping. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsql90/html/FzDTSSQL05.asp

[TR05] http://www.trilliumsoft.com