# Schema Decryption for Large Extract-Transform-Load Systems

Alexander Albrecht and Felix Naumann

Hasso Plattner Institute for Software Systems Engineering,
Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam, Germany
{alexander.albrecht,felix.naumann}@hpi.uni-potsdam.de

**Abstract.** Extract-Transform-Load (ETL) tools are used for the creation, maintenance, and evolution of data warehouses, data marts, and operational data stores. ETL workflows populate those systems with data from various data sources by specifying and executing a DAG of transformations. Over time, hundreds of individual workflows evolve as new sources and new requirements are integrated into the system. The maintenance and evolution of large-scale ETL systems requires much time and manual effort. A key problem is to understand the meaning of unfamiliar attribute labels in source and target databases and ETL transformations. Hard-to-read attribute labels in schemata lead to frustration and time spent to develop and understand ETL workflows.

We present a schema decryption technique to support ETL developers in understanding cryptic schemata of sources, targets, and ETL transformations. For a given ETL system, our recommender-like approach leverages the large number of mapped attribute labels in existing ETL workflows to produce good and meaningful decryptions. In this way we are able to decrypt attribute labels consisting of a number of unfamiliar few-letter abbreviations, such as `UNP_PEN_INT`, which we decrypt to `UNPAID_PENALTY_INTEREST`. We evaluate our schema decryption approach on three real-world repositories of ETL workflows and show that our approach is able to suggest high-quality decryptions for cryptic attribute labels in a given schema.

**Keywords:** ETL, Data Warehouse and Repository, Data Integration.

## 1 Introduction

ETL systems are visual programming tools that allow the definition of complex workflows to extract, transform, and load heterogeneous data from one or more sources into a target database. Designing and maintaining ETL workflows requires significant manual work, e.g., the effort is up to 70% of the development cost in a typical data warehouse environment [8]. ETL workflows are stored in repositories to be executed periodically, e.g., daily or once a week. In the course of a complex data warehousing project up to several hundred ETL workflows are created by different individuals [1] and stored in such repositories. Moreover, the created ETL workflows get larger and more complex over time. Cryptic

P. Atzeni, D. Cheung, and R. Sudha (Eds.): ER 2012, LNCS 7532, pp. 116–125, 2012.
© Springer-Verlag Berlin Heidelberg 2012

schemata are a well-known problem in the context of data warehousing. The main reason for cryptic schemata is the tendency to assign compact attribute labels consisting of a number of domain-specific abbreviations and acronyms.

*Example 1 (Cryptic Attribute Labels).* Consider a repository of ETL workflows to extract, transform, and load data of an OLTP system with attribute labels from the well-known TPC-E schema. With the to-be-generated decryption pairs ⟨CO ≈ COMPANY⟩ and ⟨SP ≈ STANDARD, POOR⟩, it would be easier for a developer who is unfamiliar with this schema to identify the semantics of attribute labels, such as CO_SP_RATE.

Manually finding decryption pairs is ineffective and time consuming. To illustrate this problem, consider the attribute label CO_SP_RATE from the previous example. As this attribute label is too specific to have a mapped attribute label as decryption in the given ETL repository, the developer has to look up all pairs of mapped attribute labels that give a hint on an appropriate decryption of tokens CO and SP. With over ten thousand of pairs of mapped attribute labels in the evaluated ETL repositories, manual schema decryptions becomes infeasable. Readers are referred to our technical report [2] for a comprehensive overview of schema and ETL workflow characteristics in the given real-world ETL repositories.

In this paper, we regard ETL workflows as transformation graphs of the well-known model introduced by Cui and Widom [7]. This model is generally applicable to ETL workflows from common ETL tools: An ETL workflow is a directed acyclic transformation graph (DAG) and the topologically ordered graph structure determines the execution order of the connected transformations. In ETL, most transformations are a generalization of relational operators supporting multiple inputs and outputs. Two transformations are connected in the graph if one transformation is applied to the output obtained by the other transformation. Accordingly, attributes in the output schema of a transformation are connected to the corresponding attributes in the input schema of the subsequent transformations. We leverage these *connected* attribute labels in the existing ETL workflows as valuable source of information for automated schema decryption. We have observed that *connected* attributes with different labels often contain reasonable decryptions – often not for the entire label but for tokens within the labels. As cryptic attribute labels are often too specific to have a connected attribute label as decryption in the given ETL repository, the problem is to pair portions of the cryptic attribute label with portions of more descriptive attribute labels to produce reasonable decryptions.

*Example 2 (Connected Attribute Labels).* Consider the ETL repository from Example 1. Within some ETL workflows, extracted source attributes were renamed in the succeeding transformation to provide a better readability. For example, the attribute label CP_COMP_CO_ID was renamed to COMPETITOR_COMPANY_ID and CO_CEO to COMPANY_CEO. Thus, labels CO_CEO and COMPANY_CEO and labels CP_COMP_CO_ID and COMPETITOR_COMPANY_ID are connected, respectively.

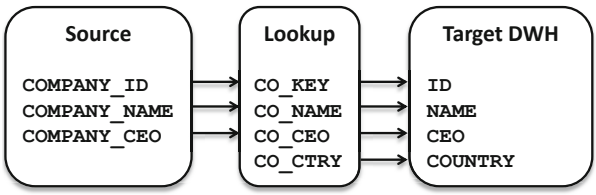


**Fig. 1.** An exemplary ETL workflow

As Etl tools allow the developer to drag-and-drop attribute labels from output to input schemata, there are many connected attributes with equivalent labels. But in large Etl repositories there is also a large number of connected attributes having different labels. There are several reasons for this, such as (1) source-, lookup-, and target-schemata used in an Etl workflow are often created independently and thus contain different attribute labels; (2) a data warehouse schema based on instances of cryptic source schemata uses attributes consisting of more descriptive tokens to provide a better readability; (3) copy-and-paste of entire transformations is a common practice in Etl development, which results in Etl sub-workflows connected to intermediate attributes with different labels.

In this paper, we will focus on attribute pairs between connected transformations for schema decryption. We ignore the connections among attribute labels within a single transformation, because a developer usually avoids using synonyms within a single transformation. Our approach overcomes the weaknesses in existing approaches, such as string and schema matching techniques. These methods lead to poor decryption results due to the use of domain-specific abbreviations, acronyms, and tokens in Etl schemata. Moreover, it is infeasible to exploit data redundancies between different schemata to find pairs of corresponding attribute labels: The data created in the intermediate Etl processing steps is not persisted and we lack this helpful information. Re-executing and storing data from intermediate processing steps is an unrealistic assumption in a typical data warehouse scenario.

## 2 Using Connected Attributes for Decryption

To illustrate our approach upfront we introduce a toy example of an Etl workflow in Fig. 1. The Etl workflow loads company data into a dimension table of a data warehouse. The extracted source data is the input of a lookup transformation. There, a company record is assigned a country from a lookup table using the company identifier as lookup key. Finally, the data is loaded into the data warehouse (DWH).

We observe that (1) attributes can be tokenized based on special characters. We also observe that (2) no two connected attributes have the same label. This is a typical situation if source, lookup, and target schemata were developed independently or for different purposes. Finally, we observe that (3) some attributes

use abbreviations that appear in extended form in connected attribute labels. These observations were made repeatedly in our analysis of three real-world ETL repositories, each with up to several hundred ETL workflows containing thousands of connected attribute pairs with different labels.

Our decryption approach finds reasonable decryptions within the given ETL repository by making use of all three observations: For each ETL workflow in the ETL repository, we first break all labels into tokens, based on case-change or non-alphabetical separators. In the example, we tokenize using the underscore as separator. The second observation allows us to identify attribute labels with same or similar semantics. If data from an attribute in the source or preceding transformation is used as input for some attribute in the target or subsequent transformation, it is reasonable to assume that their two labels are semantically related – in most cases they are semantically equivalent. For instance, `CO_CTRY` and `COUNTRY` in Fig. 1 are such *connected* attribute labels. Finally, using the third observation, we realize that the tokens `CO` and `COMPANY` co-occur in multiple pairs of connected attribute labels, leading us to believe that they are synonymous (and not for instance `CO` and `COUNTRY`).

With the identified decryptions from the ETL repository, we can suggest decryptions for cryptic attribute labels of a given schema. For instance, given a schema with the cryptic attribute label `CO_ID`, it is decrypted to `COMPANY_ID` using the decryption $\langle \texttt{CO} \approx \texttt{COMPANY} \rangle$ derived from ETL workflow in Fig. 1.

## 3 Schema Decryption

Our goal is to suggest "decryption pairs" to provide developers with a better understanding of cryptic schemata and ETL workflows.

**Definition 1 (ETL Workflow).** *An* ETL workflow *comprises a set of transformations $T$ with input and output schemata, interconnected with each other forming a DAG. Let $W = (V, E)$ be a DAG representing an* ETL *workflow consisting of a set of vertices $V$ representing the involved transformations. The edges $e \in E \subseteq V \times V$ connect the output schema of one transformation with the input schema of another transformation, i.e., $e$ represents an ordered pair of transformations.*

In this section we explain how to find decryptions for cryptic schemata leveraging the large number of connections among attribute labels in the given ETL repository.

**Definition 2 (Connected Attribute Labels).** *Two attribute labels are* connected *if there exists at least one* ETL *workflow in which a direct link is established between the corresponding attributes in the output and input schemata of two connected transformations.*

**Table 1.** Sample results for a Spanish ETL repository from the finance industry domain

| **Input Schema** | DEBT_RT, RESID_VAL_AT_RISK, UNP_PEN_INT |
|---|---|
| **Top-5 Decryption Pairs** | ⟨UNP ≈ UNPAID⟩, ⟨INT ≈ INTEREST⟩, ⟨PEN ≈ PENALTY⟩, ⟨RT ≈ RATE⟩, ⟨RESID, VAL ≈ RESIDUAL, VALUE⟩ |

### 3.1 Our Schema Decryption Approach

For a given schema consisting of a set of cryptic attribute labels, our algorithm returns a ranked list of decryptions in descending order of their frequency of occurrence in the given ETL repository. We regard an attribute label as a set of tokens and represent a decryption as a pair of corresponding token sets that appear to be used synonymously within the ETL repository. The algorithm iterates over all attribute labels $l$ from the given schema and returns the set of all applicable decryptions to decrypt $l$. Thus, for each attribute label $l$, we create all possible decryptions leveraging the large number of connected attribute labels in the given ETL repository (see Sec. 3.2). Each decryption is then added to the result. Finally, the algorithm returns a compact list of decryptions ranked in descending order of their frequency of occurrence in the ETL repository.

Let $T_i$ and $T_j$ be disjoint sets of tokens, i.e., $T_i \cap T_j = \emptyset$. We define a decryption pair $\langle T_i \approx T_j \rangle$, where $T_i$ and $T_j$ are synonyms. We regard token sets and not single tokens, because a decryption often applies to multiple tokens or even contains multiple tokens. Table 1 shows a sample schema decryption in which individual tokens but also token sets are decrypted. A decryption pair is applicable to an attribute label only if either all tokens from $T_i$, or all tokens from $T_j$ occur in the (tokenized) attribute label. Tokens from $T_i$ or $T_j$ may occur in any order in the attribute label.

**Definition 3 (Decryption Pair).** *Let $T_i$ and $T_j$ be disjoint sets of tokens. We call $\langle T_i \approx T_j \rangle$ a* decryption pair *if the token set denoted by $T_i$ is synonymous to the token set denoted by $T_j$.*

Finally, to suggest a compact list of decryption pairs, we remove all subsumed decryption pairs from the result list, retaining only maximal decryption pairs.

*Example 3 (Maximal Decryption Pair).* Consider the three created decryption pairs ⟨SP ≈ STANDARD⟩, ⟨SP ≈ POOR⟩ ⟨SP ≈ STANDARD, POOR⟩ derived from the same pairs of connected attribute labels. We only suggest ⟨SP ≈ STANDARD, POOR⟩ and remove the other two subsumed decryption pairs from the result list.

**Definition 4 (Maximal Decryption Pair).** *Let $L = \{(l_m, l_n)\}$ be the set of pairs of connected attribute labels containing decryption pair $\langle T_i \approx T_j \rangle$. We call $\langle T_i \approx T_j \rangle$ a* maximal decryption pair *if there is no decryption pair $\langle T_i' \approx T_j' \rangle$ for every $\{(l_m, l_n)\} \in L$ with $T_i \subseteq T_i'$ and $T_j \subseteq T_j'$.*

### 3.2 Finding Decryption Pairs

We now describe how we identify decryption pairs $\langle T_i \approx T_j \rangle$: Given an attribute label and some contained tokens $T_i$, we want to find all applicable decryption pairs for $T_i$. To this end, we search among all connected attribute labels in the given ETL repository for those that contain $T_i$. More formally, we consider all attribute labels $l$ that contain $T_i$ and are connected to some attribute label containing no subset of $T_i$. Using these pairs of connected attribute labels, we choose candidate decryption pairs $\langle T_i \approx T_j \rangle$, where $T_j$ is some subset of tokens from the other attribute label. A candidate decryption is added to the result if all three of the following observations hold.

Our first observation is that connected attribute labels often share tokens, i.e., such tokens appear in both connected attribute labels. In Example 2 in Sec. 1, connected attribute labels `CP_COMP_CO_ID` and `COMPETITOR_COMPANY_ID` share token `ID` and connected attribute labels `CO_CEO` and `COMPANY_CEO` share token `CEO`. Considering shared tokens for decryption makes no sense, since their counterpart is the same token in the other label. Thus, we do not create decryption pairs containing a shared token. In the example we would not create a decryption pair such as $\langle$`CO` $\approx$ `CEO`$\rangle$; the token `CEO` is already 'taken' by its counterpart `CEO` in the other attribute label.

Our second observation (and assumption) is that synonymous token sets are never used together in a single attribute label, as it would be useless to label a single attribute with synonyms. That is, if tokens $x$ and $y$ appear together in one attribute label, there is no decryption pair $\langle T_i \approx T_j \rangle$ with $x \in T_i$ and $y \in T_j$ or vice versa. Considering the attribute labels in Example 2 in Sec. 1, we do not create decryption pair $\langle$`CO` $\approx$ `COMP`$\rangle$ from a corresponding pair of connected attribute labels, because both tokens appear together in the attribute label `CP_COMP_CO_ID` and thus are very unlikely to represent synonyms.

Our last observation is that a decryption is consistently used between two connected transformations. To determine the consistency of a decryption pair derived from a pair of connected attribute labels, we determine its correctness (*confidence*) and frequency of occurrence (*hit-ratio*) throughout the corresponding schemata of the two connected transformations: Let $L_{T_i} = \{(l_m, l_n)\}$ be the set of pairs of connected attribute labels in which all tokens of $T_i$ appear either in $l_m$ or $l_n$ (but not both, as these are the trivial cases). These pairs represent the *positive class* for the decryption of $T_i$. Further, let $L_{T_i,T_j}$ be the set of pairs of connected attribute labels in which $T_i$ appears in one label and $T_j$ in the other label. These pairs represent the *true positive class* for the decryption. Note that $L_{T_i,T_j} \subseteq L_{T_i}$. Then we define confidence as

$$\mathit{confidence}_{T_i,T_j} = \frac{|L_{T_i,T_j}|}{|L_{T_i}|}$$

and we define the hit-ratio for decryption pair $\langle T_i \approx T_j \rangle$ as

$$\mathit{hit\text{-}ratio}_{T_i,T_j} = \frac{|L_{T_i,T_j}|}{|L_{T_j}|}.$$

Note that both confidence and hit-ratio have to be considered. A high hit-ratio may result in a poor confidence, i.e., the decryption from $T_i$ to $T_j$ may occur frequently, but $T_i$ also occurs frequently with other tokens in connected attribute labels. Similarly, a high confidence, e.g., achieved by returning only correct decryptions producing no false positives, may result in a poor hit-ratio.

*Example 4 (Quality of Decryption Pairs).* Consider the connected attribute labels from Example 2 in Sec. 1. Decrypting `CO` to `COMPETITOR` might have a high hit-ratio in the corresponding schemata of the two connected transformations, if `COMPETITOR` often co-occurs with `CO`. As `CO` also occurs frequently with tokens different from `COMPETITOR`, such as `COMPANY`, decrypting `CO` to `COMPETITOR` results in a poor confidence. On the other hand, decrypting `COMP` to `COMPANY` might have a high confidence: labels with the token `COMP` are almost always connected to labels containing `COMPANY`, but labels containing `COMPANY` might also often be connected with labels containing `CO` (but not `COMP`). Thus the decryption of `COMP` to `COMPANY` has a low hit-ratio.

As the quality of a decryption pair depends on both measures, we choose the harmonic mean of confidence and hit-ratio to determine the quality of a decryption pair. The harmonic mean is a typical way to aggregate measures:

$$harmonicMean = \frac{2 \cdot confidence \cdot hit\text{-}ratio}{confidence + hit\text{-}ratio}$$

As the reverse decryption of $T_j$ to $T_i$ results in the same harmonic mean value, we can ignore order. In our experiments, we choose a threshold value of 80% for the harmonic mean to suggest consistent decryptions from pairs of connected attribute labels.

## 4 Experimental Study

We evaluated our schema decryption approach on three real-world ETL repositories. These repositories were created separately by different departments of a banking organization in Switzerland (CH), Germany (DE), and Spain (ES) using Informatica PowerCenter[1]. Informatica provides ETL workflow specifications in a proprietary XML format, which our schema decryption algorithm takes as input. Schemata and connections between attribute labels are pre-indexed offline and are used to compute schema decryptions in an online fashion. Our algorithm operates efficiently and typically returns a ranked list of decryption pairs for a given schema in under a second.

### 4.1 Evaluation Technique

We have successfully tested our schema decryption approach on all ETL workflows from the three given ETL repositories. To evaluate the accuracy of our

[1] `www.informatica.com`

**Table 2.** Calculating average precision for top-3 decryption pairs

| Rank | Decryption Pair | *rel(i)* | Precision |
|---|---|---|---|
| 1 | ⟨UNP ≈ UNPAID⟩ | 1 | 1 |
| 2 | ⟨INT ≈ OUTPUT⟩ | 0 | 1/2 |
| 3 | ⟨PEN ≈ PENALTY⟩ | 1 | 2/3 |

schema decryption, we randomly selected three schemata consisting of at least 20 attribute labels from each repository and use schema decryption to generate ranked lists of decryption pairs for the selected nine schemata.

In our evaluation we consider the top-$k$ decryption pairs $p_i$ in the ranked list. Let $i$ be the position of $p_i$ in the ranked list, i.e., $i \leq k$. Then, we manually determine whether $p_i$ is relevant/correct or not for the given schema, i.e., we set $rel(i)$ to 0 or 1, respectively. We consider a decryption pair to be accurate if it helps to understand the underlying semantic domain of the original attribute label. Then, we calculate the average precision measure for the top-$k$ decryption pairs. The average precision is the average of the precision values for the seen accurate decryption pairs. Average precision is a widely-used evaluation measure in information retrieval to indicate ranking accuracy [4]:

**Definition 5 (Average Precision).** *Let $P(i)$ be the precision of the first $i$ suggested decryption pairs. Then, the average precision at position $k$ is*

$$AvP_k = \frac{\sum_{i=1}^{k} P(i) \cdot rel(i)}{\sum_{i=1}^{k} rel(i)}$$

*where precision is defined as* $P(i) = \frac{\sum_{j=1}^{i} rel(j)}{i}$

*Example 5 (Average Precision).* Table 2 shows an illustrative top-3 example of ranked decryption pairs. The examples are from the ETL repository from Spain (ES). The precision values after each new accurate decryption is observed are 1 and $\frac{2}{3}$. Thus, the *average precision* of the top 3 results (with two seen accurate decryptions) is given by $(1 + \frac{2}{3}) \ / \ 2 = 83\%$.

### 4.2 Results

Fig. 2 shows the accuracy of our schema decryption approach. We measure the mean average precision for each of the experiments and show the top-5, top-10, top-15, and top-20 results. For all three repositories the algorithm achieves an accuracy of above 90%. For the CH repository the algorithm provides the best accuracy. This is expected, because if there is a pair of connected attributes with different labels in the CH repository, it often contains an accurate decryption. The experiments demonstrate the advantages of identifying decryption pairs based on tokens and based on their characteristics. Additional experiments confirmed that our approach results in a significantly lower number of incorrect,

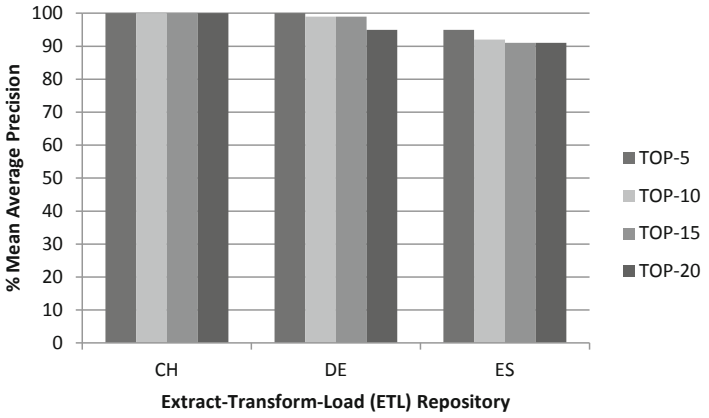


**Fig. 2.** Schema Decryption Accuracy for ETL Repositories (CH), (DE), (ES)

conflicting and redundant decrpytion pairs compared to other approaches. We compared our approach to a straightforward alternative of choosing entire labels of two connected attributes as decryption pair. In addition, we compared our approach against different string-similarity measures, such as Levenshtein distance [9].

## 5 Related Work

Our work is related to research on schema normalization in the field of data integration [10], attribute-synonym finding for relational tables and spreadsheet data in web pages [6] and string and schema matching techniques [9,3,5].

Sorrentino et al. present a semi-automatic technique for schema normalization and motivate the importance of incorporating individual examples in the process of schema normalization [10]. In contrast, ours is the first work that incorporates connected attribute labels from *complementing schemata* as source of information for fully-automated schema decryption.

The authors of [6] point out that distance metrics and global dictionaries, as often used for string and schema matching, are not appropriate to automatically find synonyms for arbitrary attribute labels. This observation is supported by our experimental results: Common distance metrics result in poor decryptions and global dictionaries lead to a relatively poor coverage of domain-specific abbreviations, acronyms, and tokens. The authors of [6] propose a large-scale discovery method on 125 million relational tables extracted from 14.1 billion HTML tables. Their approach is based on pairs of attribute labels co-occurring in tables with same context attributes. As already pointed out in the introduction, those data-driven approaches are infeasible for ETL systems. Furthermore, with our approach we can identify accurate decryptions from a substantially smaller corpus of examples compared to data-driven approaches relying on a large set of web-scale example data [3,6].

## 6 Conclusion

With this paper we presented a fully-automated schema decryption method leveraging the large number of mapped attribute labels in a given ETL repository. Our work is motivated by observing the need of easy-to-understand schemata during ETL development and maintenance. Cryptic schemata significantly increase the amount of time to understand unfamiliar data, as many readers might have experienced themselves. We demonstrated that our schema decryption approach provides helpful suggestions for three different real world ETL repositories. An ETL developer is now able to quickly grasp the underlying semantics of data records in cryptic schemata.

**Acknowledgment.** This research was funded by InfoDyn AG; we want to thank Benjamin Böhm and Dieter Radler of InfoDyn AG for supplying real-world data.

## References

1. Agrawal, H., Chafle, G., Goyal, S., Mittal, S., Mukherjea, S.: An Enhanced Extract-Transform-Load System for Migrating Data in Telecom Billing. In: Proceedings of the International Conference on Data Engineering (ICDE), Cancún, México (2008)
2. Albrecht, A., Naumann, F.: Understanding Cryptic Schemata in Large Extract-Transform-Load Systems. Tech. rep., Hasso Plattner Institute for Software Systems Engineering (October 2012)
3. Arasu, A., Chaudhuri, S., Kaushik, R.: Learning String Transformations from Examples. In: Proceedings of the International Conference on Very Large Databases (VLDB), Lyon, France (2009)
4. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Boston (1999)
5. Bernstein, P.A., Madhavan, J., Rahm, E.: Generic Schema Matching, Ten Years Later. VLDB Journal 4 (2011)
6. Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E., Zhang, Y.: WebTables: Exploring the Power of Tables on the Web. In: Proceedings of the International Conference on Very Large Databases (VLDB), Auckland, New Zealand (2008)
7. Cui, Y., Widom, J.: Lineage Tracing for General Data Warehouse Transformations. VLDB Journal 12(1) (2003)
8. Dayal, U., Castellanos, M., Simitsis, A., Wilkinson, K.: Data Integration Flows for Business Intelligence. In: Proceedings of the International Conference on Extending Database Technology (EDBT). Saint Petersburg, Russia (2009)
9. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10(8) (1966)
10. Sorrentino, S., Bergamaschi, S., Gawinecki, M., Po, L.: Schema Normalization for Improving Schema Matching. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 280–293. Springer, Heidelberg (2009)