An Automated ETL for Online Datasets

Suzanne McCarthy and Andrew McCarren and Mark Roantree

Abstract

While using online datasets for machine learning is commonplace today, the quality of these datasets impacts on the performance of prediction algorithms. One method for improving the semantics of new data sources is to map these sources to a common data model or ontology. While semantic and structural heterogeneities must still be resolved, this provides a well established approach to providing clean datasets, suitable for machine learning and analysis. However, when there is a requirement for a close to real time usage of online data, a method for dynamic Extract-Transform-Load of new sources data must be developed. In this work, we present a framework for integrating online and enterprise data sources, in close to real time, to provide datasets for machine learning and predictive algorithms. An exhaustive evaluation compares a human built data transformation process with our system's machine generated ETL process, with very favourable results, illustrating the value and impact of an automated approach.

Please cite this version:

Suzanne McCarthy, Andrew McCarren, Mark Roantree. A Method for Automated Transformation and Validation of Online Datasets. 23rd Enterprise Computing Conference (EDOC), October 2019, Paris, France. Volume . ISBN: . IEEE.

An Automated ETL for Online Datasets

Suzanne McCarthy and Andrew McCarren
Insight Centre for Data Analytics
Dublin City University
Dublin 9, Ireland
Suzanne.McCarthy@insight-centre.org,
Andrew.McCarren@dcu.ie

Mark Roantree
School of Computing
Dublin City University
Dublin 9, Ireland
Mark.Roantree@dcu.ie

***Abstract*—While using online datasets for machine learning is commonplace today, the quality of these datasets impacts on the performance of prediction algorithms. One method for improving the semantics of new data sources is to map these sources to a common data model or ontology. While semantic and structural heterogeneities must still be resolved, this provides a well established approach to providing clean datasets, suitable for machine learning and analysis. However, when there is a requirement for a close to real time usage of online data, a method for dynamic Extract-Transform-Load of new sources data must be developed. In this work, we present a framework for integrating online and enterprise data sources, in close to real time, to provide datasets for machine learning and predictive algorithms. An exhaustive evaluation compares a human built data transformation process with our system's machine generated ETL process, with very favourable results, illustrating the value and impact of an automated approach.**

I. Introduction

As the use of data analytics and data warehousing technologies grows in the agricultural (Agri) sector, data integration and heterogeneity is a problem that has received a lot of attention. However, many data sources for the Agri sector are web-based and web data poses challenges for data scientists. While most data is freely available from web-based government sources, they are not easily integrated with their existing systems or datasets. A data warehouse type architecture provides one solution as it manages cleaning, transformation and integration of data into *data marts*. These represent the specific data subset of choice for the analyst. Additionally, a dart mart or *cube* is a powerful structure that facilitates OLAP (OnLine Analytical Processing) to enable scientists and analysts to extract the precise dataset(s) necessary for analysis. An Extract, Transform, and Load (ETL) architecture is used to integrate data acquired from different external sources but this can be an expensive and lengthy process, which must be completed for every new data source.

Early work on the integration of heterogeneous data sources [24] focused on the different types of structural and semantic heterogeneities. These influence how a system can process and manage data and motivated the need for a common (or canonical) data model. By transforming data to this common model, it becomes possible to both manage external data sources in a homogeneous manner and also to integrate these data sources where required. More recent work focused on semantic heterogeneity among data sources [21], [2], which motivated the need for ontologies to overcome complex integrations.

If we consider an ETL system for web data: data acquisition (Extract) is straightforward but the actual processing of data requires some level of knowledge regarding data content and structure. Our approach regards this step as a light-touch process to enable reading and processing of data. The next step (Transform) will require a detailed interpretation of source data in order that querying or integration with other data sources can take place [17]. This process is costly and time-consuming. The final step (Load) constructs the data mart (multi-dimensional dataset) for analysis and data mining. The crucial aspect to this type of architecture is the multi-dimensional data model: it facilitates the extraction of datasets for analysis and mining, in a flexible and powerful manner using an OLAP interface. For this reason, our design decision is to use a star schema common model [17], and map all external sources directly into data marts, which have a star schema representation. However, unlike more traditional ETL approaches, we do not build or populate a data warehouse. Instead, we develop the warehouse model (schema) which assists the creation of user-defined data marts.

Motivation and Contribution. The goal of this research is to develop a methodology that integrates new online source data with existing enterprise or web sources to *automate the creation* of new data marts. In order to deliver this process, it is necessary to build *transformation templates* for each data source so that an accurate integration takes place each time a new source is acquired. However, the creation of each template can be very time-consuming: more than 4 hours for each of the 170 sources used in the development of our common model and vocabulary. This is explained at the begin of section V-A where the creation of mappings for dimensional data is discussed. In addition, it requires a common (domain specific) model and companion vocabulary to interpret the web sources. Our contribution in this paper is the automated construction of the transformation template, creating and populating OLAP-ready data marts, and a robust validation process to ensure accuracy of the overall ETL process. We will show how this greatly reduces the time in which new sources are added to data marts and our experiments demonstrate the accuracy of

This research was funded by Science Foundation Ireland award 12/RC/2289 which is co-funded by the European Regional Development Fund

this approach.

Paper Structure. The remainder of the paper is structured as follows: in §II, we provide a discussion on state of the art; in §III, we provide an outline description of our Extract-Transform-Load architecture which manages new data sources into data marts; in §IV, we present a detailed description of the construction of transformation templates; in §V, we present our evaluation and discussion; and finally in §VI, we conclude the paper.

II. Related Research

The authors of [4] propose a component-based metadata scheme that includes a set of controlled vocabulary, based on the Data Category Registry from International Organization for Standardization [13], with the purpose of solving the problem of semantic interoperability. A metadata component is an atomic field that describes a particular aspect of the data, e.g. title. Alternatively, a component can recursively contain other components [5]. The benefits of this approach are reusability and extensibility as well as providing guidelines for data structuring. However, this work provides neither a mechanism to automatically manage new sources nor a star schema common model to facilitate OLAP.

The authors in [1] use procedural metadata for transformations, where templates are composed of skeleton scripts to which parameters are provided. We also use skeleton scripts in the transformation process to which parameters are provided by a set of templates. Although they are intended for reuse, the templates in [1] are developed for specific sources and thus, their approach does not have the ability to automatically process new data sources.

Procedural metadata refers to the storage of the metadata of event, condition, and action-based procedures of smart systems. The authors in [16] provide a proof of concept involving smart-home case studies. While we also use procedural metadata to store transformation functions in the form of a ruleset, we also provide an approach to processing new online sources using a common model to ensure high levels of accuracy.

In another approach to the utilisation of procedural metadata, the authors of [9] discuss the explicit representation of computational procedures performed on their data to facilitate data provenance. To do this, they developed the Chimera virtual data catalog, one of whose functions is to compactly represent data derivation procedures. While our approach uses similar steps to ensure data provenance as a form of validation, we also adopt a multi-dimensional common model to facilitate OLAP for the generation of customised datasets.

In a project with some overlap to the work presented here, the authors propose a configurable approach to creating a customised data warehouse [15]. This work is on a much larger scale as its focus is on the design of the overall warehouse and ETL functionality. However, there are comparisons in that it does not see the warehouse system as static, but as evolving. Indeed, their interpretation of configurability sees warehouse design and construction as building from a common architecture through selection, parameterisation and a synthesising or integration process. While this work provides a solid motivator and platform for some aspects to our design, it does not focus on the specific heterogeneities that make ETL difficult, nor on solutions to resolves these issues using an automated ETL approach.

In the work closest to our approach [25], the authors also automate the process of assigning mappings and transformations required for the ETL process. In this work, the authors use an ontology to provide the domain-specific information to drive the transformation stage of the ETL process. Both this and our approaches propose an automatic transformation method and both use a star schema as the target data model. However, we do not use the more traditional ontology approach as these are very costly to build and require extensive support by domain experts. Instead, we construct a more lightweight domain model and use an extensive analysis of many agri sources to develop our common vocabulary, which delivers significant savings.

III. ETL Architecture

The goal of our system is to have an ETL system robust enough to integrate data from previously unseen sources. Our metamodel driven approach comprises three main constructs: the system data model, which is domain specific; import templates which provide an outline description of source data; and transformation templates which manage the complex process of converting external sources to conform to the system data model. Fig 1 illustrates the workflow in our ETL system and, in the rest of this section, we will provide an outline of each process.

A. Importing a Domain Data Model (Import)

The domain data model provides the system with a description of the system data model: schema facts and dimensions (a star schema). A domain model must be specified in advance and sufficiently exhaustive to represent new data sources. For this research, the common agri trade model [20] was developed after a lengthy analysis of almost 170 agri-trade datasets. Details of the common agri model documented in [18] contains 25 dimensions and 27 facts or measures. In addition, there is a *vocabulary* of 9000 terms in the current version, which are used to assist the schema matching process.

New data sources must conform to the domain data model: all dimensions found in the source data must be mapped to an existing dimension in the model, and all measures found in the source data must be converted to the system format. As a result, all data marts have the same (sub)set of dimensions and facts to facilitate the integration process.

B. Data & Metadata Acquisition (Extract & Analyse)

To provide a dynamic ETL, the system must be able to interpret new data sources. For new sources, a metadata capture component is necessary while existing sources use a previously constructed transformation template. Thus, the

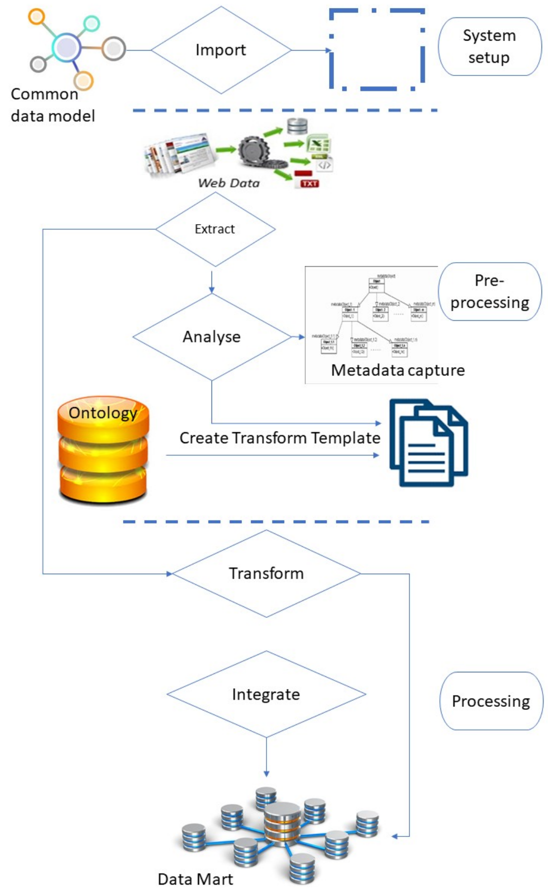

Fig. 1: ETL workflow

Extract component in our ETL system contains both data *and* metadata acquisition functions.

When new data sources are acquired, the first step is to construct an *Import Template* to enable the importation of data. As it is not possible to process data at this point, new source data is stored in a data lake. The data first goes through a brief (<5 mins) manual analysis by the developer and an *Import Template* is written for each source. Our approach to metadata acquisition involves a component-based approach similar to [4] and draws from the Dublin Core [6] approach for the specific components used, wherein a subset of the DCMES [6] fields will be split into individual components to add structure. From the subset (Format, Coverage, Subject), the format of the Import Templates is sufficiently generic to manage any data source upon initial acquisition. The user generates the Import Template for each data source. A number of examples are presented in Table I, where the Import Template contains the following three key pieces of information:

- **Version Metadata.** This comprises the data source (`source`); the date of most recent version (`scrape_date`); the granularity (`frequency`) and type of industry (`industry`).
- **Data Mart Management.** This is represented as a set of flags: `transform`, `query`, `load` to indicate the status of each source file in terms of transformation, retrieval of data, and whether or not is is ready for user queries (load). These are 'N' by default and switch to 'Y' when these actions have been completed. The values in Table I show that while four of the data sources have been successfully transformed, only *comtrade* has been loaded for user query or analysis.
- **Source Description.** This comprises the number of rows in the file (`num_rows`); those columns containing dimensional data (`dim_col_list`); those columns containing measures (`fact_col_list`); and fields informing the system how to process the file: `header` and `skip_rows`.

At this point, the values in Table I would report a status N for the columns `transform`, `query` and `load` as data has been imported only. The next step requires a *Transformation Template* to be created for new sources. This is a far more lengthy process which is discussed in detail in §IV. Without this template, the process cannot continue to the next stage, data transformation, as the system is unable to correctly interpret source data.

C. Data Transformation (Transform)

This is the process by which source data is mapped to a target schema and heterogeneities are resolved by providing a mapping plan for the terms in each source file. It is driven by a transformation template, the structure of which is illustrated as follows.

A Transformation Template is a 5-tuple TT $=\langle S_s, CDM, S_t, M, Met\rangle$ where:

- S_s is the set of source schemas, to be imported.
- CDM is the canonical data model with a set of elements $\langle D, DA, DV, F\rangle$ where:
  - D: A set of dimensions $d \in D$.
  - DA: A set of attributes $a \in A$ for each d.
  - DV: A set of values $v \in V$ for each a.
  - F: A set of facts where a fact will have a set of values $v \in V$.
- S_t is the set of target schemas where $S_t \subseteq CDM$
- M is the suite of functions to map S_s to S_t: $\langle AT, SM, DM, RA\rangle$
  - AT: AttributeTyping, an attribute to identify CDM element as D or F (dimension or fact)
  - SM: SchemaMatch, a function to identify S_t
  - DM: DataMap, a function to create the mappings $SO \rightarrow ST$ where
    - $*$ SO is the source original term.
    - $*$ ST is the standard term from S_t.
  - RA: RuleAssign, a function to assign a measure conversion rule $\langle ID, SO, ST, CONV\rangle$ where ID is a unique identifier, SO is the unit to be converted, ST is the unit to be mapped to, and $CONV$ is the formula to convert.

TABLE I: Example Import Templates

scrape_date	source	frequency	industry	transform	query	load	num_rows	header	skip_rows	dim_col_list	fact_col_list
10-01-2019	comtrade	monthly	meat	Y	Y	Y	7655	0	0	2,5,6,8,9,7,12,13	10,11
26-02-2018	kepak	daily	meat	Y	Y	N	5000	0	0	0,1,2,3,4,5	6,7,8,9
11-12-2018	usda	monthly	meat	Y	Y	N	59028	0	0	1,2,3,4,5,9,8,12	11,10
11-12-2018	statcan	monthly	meat	Y	N	N	2599	0	0	0,1,2,3,4,5,6	7,8
11-12-2018	eurostat	monthly	meat	N	N	N	4800	0	0	1,2,3,4,5,6,9,7	8,10

- Met represents the way in which M is created: manual or automated.

D. Data Loading and Integration (Integrate)

After the data is transformed, it is first loaded to a single-source data cube, whose schema is a subset of the (star schema) domain data model imported during the first step. This step is outside the scope of the work presented in the paper.

IV. Constructing Transformation Templates

One of the primary goals of our research is to automate the process of integrating new data sources into the data marts used for queries and analyses. This can only be delivered if a mapping plan exists to transform source data into a format compliant with the system model. As this requires a separate process for each new data source, we present an automated approach, comprising four main parts: *SchemaMatch*, *DataMap*, *RuleAssign*, and *Data Preparation*.

A. SchemaMatch

The process takes each source element and attempts to match with an element in the system model. One of 6 results is possible, where the source element is matched to:

1) A single domain attribute
2) A single measure
3) Multiple domain attributes
4) Multiple measures
5) Both an attribute and measure
6) NULL (unmatched)

For cases 1 and 2, there is no action required and the process continues. For cases 3 and 4 (which are quite rare), the system will use the *best* match, by matching a *set* of source attributes against the selected domains. Any matching errors are detected during validation for each of the 4 cases. For cases 5 and 6, a human decision is required to proceed or the result is *No Match* and the data associated with those attributes will not form part of the final data mart.

Given a blank Transformation Template (TT) with fields `attr_type`, `rule`, `source_original`, `standard_term`, `dimension` and `dim_attr`, a ruleset R, an agri vocabulary O, a common data model CDM and a dataset with a set of attributes A= $\{a_1, ..., a_n\}$ and values V= $\{v_1, ..., v_n\}$ for each attribute, Algorithm 1 shows how TT will be populated. Attribute types and brief descriptions are listed at the start of the algorithm.

In lines 1-21, **SchemaMatch** sees a dimension and its attributes obtained from the common model and added to the Template. In lines 5-7, if an attribute raises one of the types of ambiguity mentioned above (cases 5 or 6), the user can intervene (or not).

In lines 9-18, the source attribute is verified using the vocabulary to determine if it is a dimension or a measure (**AttributeTyping**), and the canonical term is subsequently retreived from the domain model. If the attribute is a measure, the `dimension` and `dim_attr` fields are left blank (lines 11 and 12); if it is a dimension, these fields are filled with canonical terms from the common model (lines 15-18). The same dimension and attribute are added for each value in lines 32 and 33 during the Data Mapping phase.

In lines 19 to 21, a new row for each data attribute is added to the template.

B. DataMap

The goal of *DataMap* is to complete the creation of the template with precise data values from the source data and generate a mapping and target location for these values. This is an important process as star schemas contain dimensional attribute values (metadata) which may update the existing dimensions. For example, if no data exists for April in 2019, then the value *April* for the month attribute may not exist in the dimension. When a new source contains a sales value of 10,000 for the month of April, the sales value is written as fact data, and the value of *April* is written as dimensional data. This second step is unnecessary if *April* already exists in the dimension. The mapping process is driven by the domain model and assisted by a vocabulary of approximately 9000 terms. The **DataMap** process is captured in lines 22-41.

Between lines 24-30, the system uses the vocabulary to find a canonical term in the data model to which the value v can be mapped. This step sets the `standard_term` attribute to the appropriate canonical term. If that attribute is not found, the NULL value assigned at line 28 is detected during a later step, indicating an (user) action is required. In lines 38 to 41, a new row is added to the template for each dimensional value.

C. RuleAssign

In lines 34-36, if the value is a unit of measure, the system retrieves a conversion ruleset (see Table II) to enable a conversion for the measure data. This returns an ID of the rule so the formula can be used during the transformation process. An example of a conversion rule is shown in Example 4.1 where the source data records the measure as *Ton* but as the canonical measure is *KG*, each value is multiplied by 1,000.

Example 4.1: Ton_KG Sample Rule

Sample rule to convert Ton to KG (using Table II)

Algorithm 1 Construct Transformation Template

TT will have the following attributes:
attr_type: D(dimension) or F(fact);
fact_rule_id: Reference to measure conversion rule;
source_original: Term in data source file;
standard_term: Term converted to;
dimension: Domain model's dimension name;
dim_attr: Domain model's dimension attribute.

```
function Construct(A, R, TT, O, CDM)
    for $a \in A$ do
        source_original $\leftarrow a$
        get $a'$ from O
        if $|a'| > 1$ then
            Request user input
        end if
        standard_term $\leftarrow a'$
        if AttributeTyping($a$.measure) = True then
            attr_type $\leftarrow F$
            dimension $\leftarrow$ null
            dim_attr $\leftarrow$ null
        else
            attr_type $\leftarrow D$
            get $dimension\_cdm$ from CDM
            get $dim\_attr\_cdm$ from CDM
            dimension $\leftarrow dimension\_cdm$
            dim_attr $\leftarrow dim\_attr\_cdm$
            row = [dim_attr, source_original,
            standard_term, dimension, dim_attr]
            TT = TT+row
            get $a.V$
            for v $\in V$ do
                source_original $\leftarrow v$
                try
                    get $v'$ from O
                catch $v'$ = null
                    standard_term $\leftarrow$ null
                finally
                    standard_term $\leftarrow v'$
                end try
                dimension $\leftarrow dimension\_cdm$
                dim_attr $\leftarrow dim\_attr\_cdm$
                if $dimension\_cdm$=dim_unit then
                    get rule_id from R
                    fact_rule_id $\leftarrow$ rule_id
                end if
                row = [attr_type, fact_rule_id,
                source_original, standard_term,
                dimension, dim_attr]
                TT = TT+row
            end for
        end if
    end for
    return TT
end function
```

TABLE II: Sample Rule Type

Attribute	Type	Notes
rule_id	int	I.D. number
source_original	String	Unit in data source file
standard_term	String	Unit from common model
conversion	$x' = f(x)$	Function to convert measure

`rule_id`:1
`source_original`: Ton
`standard_term`: KG
`conversion`: $x' = x * 1000$

The output of the Data Mapping processes is a fully constructed Transformation Template, the first three rows of which are shown in Example 4.2.

Example 4.2: Transformation Template

```
<template>
    <row>
        <attr_type>D</attr_type>
        <fact_rule_id></fact_rule_id>
        <source_original>Ireland,Rep of. </source_original>
        <standard_term>IRELAND</standard_term>
        <dimension>dim_geo</dimension>
        <dim_attr>geo</dim_attr>
    </row>
    <row>
        <attr_type>D</attr_type>
        <fact_rule_id>1</fact_rule_id>
        <source_original>ton</source_original>
        <standard_term>KG</standard_term>
        <dimension>dim_unit</dimension>
        <dim_attr>unit</dim_attr>
    </row>
    <row>
        <attr_type>F</attr_type>
        <fact_rule_id></fact_rule_id>
        <source_original>weight</source_original>
        <standard_term>trade_weight</standard_term>
        <dimension></dimension>
        <dim_attr></dim_attr>
    </row>
</template>
```

D. Data Preparation

The final step in Transformation is the creation of a ready-to-load file (a set of values and their locations), which is used in the Load step to populate the data mart.

V. Evaluation

As the goal of our work is to integrate unseen data sources into our data marts, the main contribution of the work presented here is in the automated creation of the transformation templates for each data source. Without these constructs, automated processing is not possible. However, there is significant overhead in their creation. Thus, the goal of this evaluation is to compare both the speed and accuracy of our automated approach with a manual process. We will begin be describing the experimental setup, then present the results across a number of parameters, before concluding with our interpretation on what the results mean.

TABLE III: Source files

source	num_rows	num_cols
USDA	4999	10
Statcan	2559	9
Bord Bia	24990	5

A. Experiment Setup

To setup a manual validation, we engaged three participants of varying levels of expertise in the areas of databases, data engineering and ETL, using tools built using Microsoft Excel and MySQL. The data sources described below were used for the evaluation. As the users were not familiar with the domain (agriculture), this would add to time taken to create templates. Thus, for these experiments, sources were chosen that had low dimensionality and were not overly large in size.

- The United States Department of Agriculture (**USDA**) [27] publishes Agri trade data figures which can be downloaded in bulk. We extracted international trade of meat products, capturing weight and currency values, with 10 columns of data and 4,999 rows.
- **StatCan** [26] is the Canadian National Statistics agency and publishes economic, social and census data. From this source, we extracted a dataset on the international trade of meat products, with 9 columns of data and 2,559 rows.
- **Bord Bia** [3] is the Irish Food Board, promoting Irish food sales and horticulture, publishing prices of dairy, pigs etc. We extracted a historical pig price dataset with 5 columns and 24,999 rows of data.

The process for constructing mappings for multidimensional data can be complex and time consuming as a mapping must be created for every *unique* dimensional value. In other words, if (in an unlikely case), the USDA source contained a single data item (fact) for each dimensional value, the transformation template would have 4,999 instances. Thus, it is not simply a case of 10 instances to correspond to the 10 columns of data. This example highlights why it is not scalable to hand craft these templates for every new source. Indeed, a template *cannot* be reused except where dimensional data is identical in a previous template.

The participants were presented with a data file from each of the three sources, and three blank transformation templates. The participants were also provided with:

- The Rules for measure data conversion, such as the example seen in Example 4.1.
- The Canonical Agri Data Model as seen in [18].
- An incognito browsing window to look up help pages for Excel or MySQL, as needed.

Participants were not permitted to write a program in any scripting language, nor to use any MySQL commands other than `SELECT`.

B. Multi-test Results

The key metrics for this evaluation are speed, measured in seconds and minutes, and accuracy, determined by a domain expert who examined both the manual and automated templates after the experiment and calculated an accuracy percentage for each of the following attributes of the templates: `attr_type`, `fact_rule_id`, `standard_term`, `dimension`, `dim_attr`. The domain expert also used the templates to transform the datasets before loading them to data cubes. In the following results tables, user M1 is a relative beginner in the tools needed for this experiment, M2 is an intermediate level and M3 is an expert.

TABLE IV: Time taken to build templates

tester	source	build time	row count
M1	USDA	105 mins	304
M1	Statcan	31 mins	228
M1	Bord Bia	55 mins	1825
M1	Total	191 mins	2357
M2	USDA	215 mins	304
M2	Statcan	112 mins	230
M2	Bord Bia	25 mins	1218
M2	Total	352 mins	1752
M3	USDA	36 mins	307
M3	Statcan	35 mins	230
M3	Bord Bia	37 mins	1218
M3	Total	108 mins	1755
system	USDA	3 secs	304
system	Statcan	5 secs	231
system	Bord Bia	3 secs	1218
system	Total	11 secs	1753

Template Creation Times. Table IV presents the time required for each participant to construct each transformation template (build time) and the number of mapping instances (row count) created for each template. Accuracy is not considered at this point as we are concerned with highlight the times savings by the system-based approach, and to see that templates were of roughly the same size. As expected, the system performs quickest and the expert user was quickest among the 3 testers. Templates were almost equal in size.

Template Accuracy. For this part of the evaluation, an Agri data domain expert created a set of transformation templates for the data sources, using the same manual process and the canonical model as the testers, but in this case, also had the additional use of the system vocabulary. This set of templates is regarded as 100% accurate for the purpose of this evaluation. These templates, as well as the templates created both by the participants and our automated process, were used to transform the source datasets and load the data to data cubes. The data cubes created using the automated templates and each of the participants' templates were then compared against the *gold standard data cubes*, i.e. the data cubes created using the domain expert's templates.

In Table V, the accuracy of each template for each participant when compared to the system generated templates, is shown. For each source data file, testers were required to extract the full set of the attribute names and the unique list of all dimensional values. Each of these terms will be called a source original term. For each source original term, the participants populated the attributes shown in Table V, where `Attr. Type` denotes whether the term is a Dimension (D) or Fact (F); `Fact Rule ID` is the unique identifier of a

TABLE V: Template Accuracy

Tester	Source	Attr. Type	Rule ID	Standard Term	Dimension	Dim. Attr.	Overall
M1	USDA	3.29%	0%	85.85%	100%	100%	
M1	Statcan	3.95%	0%	92.98%	98.68%	98.68%	
M1	Bord Bia	0.41%	0%	88.51%	100%	100%	
AVG		2.55%	0 %	89.11%	99.56%	99.56%	58.16%
M2	USDA	100%	50%	96.05%	99.67%	99.34%	
M2	Statcan	99.57%	50 %	98.7%	98.27%	98.27%	
M2	Bord Bia	100%	100%	99.92%	99.92%	99.92%	
AVG		100%	66.67%	98.22%	99.28%	99.18%	92.64%
M3	USDA	100%	0%	100%	45.4%	45.4%	
M3	Statcan	99.57%	0%	97.84%	99.13%	99.57%	
M3	Bord Bia	100%	0%	100%	100%	99.92%	
AVG		99.85%	0%	99.28%	81.51%	81.63%	72.45%
system	USDA	100%	50%	99.01%	80.59%	80.59%	
system	Statcan	99.57%	100%	99.13%	74.89%	99.57%	
system	Bord Bia	100%	0%	99.92%	100%	100%	
AVG		99.86%	50%	99.35%	85.16%	93.39%	85.55%

rule (they had access to the ruleset) for converting measure data based on the unit of measure; `Standard Term` is the specific dimensional value to which the source original term is mapped; `Dimension` is the dimension from the canonical data model where the `Standard Term` is found; and `Dim. Attr.` is the specific `dimensional` attribute. The accuracy (i.e. percentage of correct values) for each of these attributes is presented in Table V, along with an overall average accuracy, *weighted* by the number of rows in the template.

The transformers were used to generate data cubes, which were subsequently compared to the golden data cubes. In Table VI, the process of mapping the source attribute names to terms in the domain model (e.g. $area \longrightarrow geo$), is investigated. Each column from every data source is represented in the *Attribute* column. An X in the table shows where either the user or the automated system provided an incorrect attribute mapping. The purpose of this part of the evaluation is to highlight the quality of the data from the point of view of queries. An inaccurate attribute mapping will have a different impact on the quality of the data to an inaccurate mapping of the values. Here, we were provided with clear evidence that term ambiguity resulted in the creation of incorrect mappings by all testers and the system itself. It also illustrated that from this viewpoint, accuracy was very similar across testers and the system.

Table VII illustrates the accuracy of the transformed dimensional values, as a percentage. A dimensional value in the dataset is considered correct if it matches that of the golden data cube so Table VII contains the percentage of dimensional value matches between each data cube and the appropriate golden cube. For this test, with the exception of small number of attributes for one test participant, the results were very high.

C. Analysis and Insights

It is clear that there were some errors in the accuracy of both the participants and the automated system. Categorizing the accuracy errors, one of the following cases may emerge, each with its own fix.

TABLE VI: Data Cube Accuracy (attributes)

Source	Attribute	M1	M2	M3	System
USDA	yearmonth	✓	✓	✓	✓
USDA	reporter	✓	✓	✓	✓
USDA	partner	✓	✓	✓	✓
USDA	product_code	✓	✓	✓	✓
USDA	product_desc	X	X	X	X
USDA	flow	✓	✓	✓	✓
USDA	unit_weight	✓	✓	✓	✓
USDA	unit_value	✓	✓	✓	✓
USDA	trade_weight	X	X	X	X
USDA	trade_value	X	X	X	X
Statcan	yearmonth	✓	✓	✓	✓
Statcan	partner	✓	✓	✓	✓
Statcan	product_code	✓	✓	✓	✓
Statcan	product_desc	X	X	X	X
Statcan	flow	✓	✓	✓	✓
Statcan	unit_weight	✓	✓	X	✓
Statcan	unit_value	✓	✓	✓	✓
Statcan	trade_weight	X	X	X	X
Statcan	trade_value	X	X	X	X
Bord Bia	date	✓	✓	✓	✓
Bord Bia	geo	✓	✓	✓	✓
Bord Bia	product	✓	✓	✓	✓
Bord Bia	unit	✓	✓	✓	✓
Bord Bia	price	✓	✓	✓	✓

- Case 1: A one-off mistake where a single participant assigned an incorrect model term to the source term. The fix is to edit the template before using it in the transformation process, adjusting the standard terms to correct canonical model terms. An example of this is seen in line Statcan,`product_desc` of Table VII, where M2 and M3 assigned correct terms for all values of the `product_desc` attribute for Statcan, but M1 made at least one error.
- Case 2: All test participants assigned a wrong standard term for the source term. This should be examined to see if this is in fact multiple Case 1's, in which case the fix is the same. This is seen in line 5 of Table VII where each of the human participants made errors for the attribute

TABLE VII: Data Cube Accuracy (dimensional data)

Source	Attribute	M1	M2	M3	System
USDA	yearmonth	100%	100%	100%	100%
USDA	reporter	100%	100%	100%	100%
USDA	partner	100%	100%	100%	100%
USDA	product_code	100%	99.5%	100%	100%
USDA	product_desc	40.5%	72.14%	91.82%	100%
USDA	flow	100%	100%	0%	100%
USDA	unit_weight	0%	0%	0%	0%
USDA	unit_value	0%	0%	0%	100%
Statcan	yearmonth	100%	100%	100%	100%
Statcan	partner	92.26%	99.6%	100%	100%
Statcan	product_code	98.28%	100%	100%	100%
Statcan	product_desc	92.7%	100%	100%	100%
Statcan	flow	0%	100%	100%	100%
Statcan	unit_weight	0%	0%	0%	100%
Statcan	unit_value	0 %	100%	100%	100%
Bord Bia	date	56.6%	100%	100%	100%
Bord Bia	geo	90.5%	100%	100%	100%
Bord Bia	product	100%	100%	100%	100%
Bord Bia	unit	0%	0%	0%	0%

`product_desc` for the USDA dataset.

- Case 2a: All participant assigned a wrong standard term to the source original term and it was not multiple Case 1's. This would require a fix in the vocabulary.
- Case 3: Participants could not find a mapping for a term because it did not exist. This requires an extension to the vocabulary. This can be seen in Table VI in USDA,`trade_weight` and `trade_value`, and the same Attributes for Statcan, where there were no terms available to map the names of these measures. This required a post-validation update to the vocabulary.
- Case 3a: Participants could not find a mapping for a source *unit of measure* because it did not exist. This requires an extension to the *ruleset*. This is seen in the last line of Table VII where none of the participants provided a standard term for `unit` for Bord Bia, nor could the system.

The errors are not of equal severity. For example, a transformation template may be less than 100% accurate but only the attribute *names* were incorrectly mapped, while the *values* of the attribute were correctly mapped. If the attribute name is not what is expected, it will have consequences for how the final data cube is queried but will not affect the reliability of the query result. However, an incorrect value will generate incorrect results. Additionally, fixing an incorrect attribute name is a far easier fix than correcting an incorrect value, due to the different numbers of options available in the vocabulary, i.e. relatively few attribute names and many more values.

Although the accuracy of the automated system-built templates did not provide 100% accuracy, it performed better than expected, giving an overall accuracy rating of 85.55% as shown in Table V. This compares favourably with the accuracy ratings of the two more experienced participants. When one also considers the time savings shown in Table IV, the fact that the system achieves these levels of accuracy and takes between 3-5 seconds to produce each template, compared to human efforts of 25-215 minutes, offers solid validation when considering the high volumes of streams that are available for agri data.

It is interesting to note in Tables IV and V the differences in the speed and accuracy between the human participants. For example, M3 (the expert participant) had a lower accuracy than the intermediate tester, M2 - 72.45% as opposed to 92.64% - but was considerably faster. The intermediate user, M2, while slower, was more diligent with the highest accuracy. The beginner, M1, was faster than M2 but these templates had most errors. Meanwhile, the system was obviously the fastest approach, with significant higher accuracy than 2 of the 3 testers.

When creating the template for USDA, our system ran into an issue of term ambiguity that it solved incorrectly. It was presented with a term that may have been one of two domain attributes (Case 3 as mentioned in §IV-A) and incorrectly mapped this term to the wrong dimension - `dim_product` as opposed to `dim_trade_product`. This lowered the overall accuracy for this source and was the reason why the system did not beat the human users, as both the `Dimension` and `Dim. Attr.` fields in the template were wrong. For Bord Bia, the system did not run into this ambiguity. For Statcan, it did but in this case made the correct mapping for Dim. Attr. Improving the system's ability to cope with ambiguity will form part of future work.

For Tables VI and VII, source data has been transformed using the templates and can be queried as a data cube. Separately, a domain expert created templates for the same sources which created *gold standard* data cubes. These are the yardstick measure of accuracy with results for attributes presented in Table VI and dimensional values in Table VII.

It can be seen from Table VI that in terms of mapping the attribute names, the system made the same mistakes as the users. Both were unable to map the names of the measures (`trade_weight` and `trade_value`) as well as the `product_desc` attribute (lines 5 and 14). Regarding the latter case, this was due to ambiguity in the terms and outside of an experimental environment, would result in the program flagging the ambiguity to the developer to provide a correct term. The names of the measures required a fix to the vocabulary.

This term ambiguity was the main cause of any errors in the manually built templates and the sole cause of any errors in the automated templates. Some additional causes of errors in the manual versions included:

- Term duplication: The transformation templates should be a unique list of terms and their matches; two out of three human participants produced templates containing duplicates.
- On the contrary, users may also accidentally omit terms that should be included in the template. This, as well as term duplication, can be seen in Table IV where the `row count` differs between users.
- Due to possible reasons such as fatigue, the participants occasionally missed a step in the constructing of the

templates, e.g. forgot to assign an `Attr Type` to every term in the template.

- Making an incorrect guess. When the automated system was presented with this ambiguity, it would instead map the term to null, allowing a user to easily see where intervention is needed.

Table VII shows the accuracy of both the participants and the system when mapping the dimensional data. This is arguably the most important measure as an incorrect mapping of this data will have the most severe consequences from the point of view of data quality. It can be seen that the results from the system are now very different from that of the human users. Essentially, the system's results are 'all or nothing'. If the system has correctly identified the dimension, then it will either identify the correct values of the attribute or will map any source terms to NULL where a match cannot be found. Users, on the other hand, will always guess and often incorrectly. For example, it can be seen in this table that there were large differences in the mapping of `product_desc` for USDA - the most difficult attribute and the one that would require the most guessing on the part of the participants, M1 guessed correctly 40.5% of the time, M2 guessed correctly 72.14% of the time while M3 identified the 1:1 relationship between `product_desc` and `product_code`, and had the highest accuracy level of 91.82%. However, the system used the vocabulary to achieve 100%. This figure would be lower if the data contained any products not seen during the extensive analysis that resulted in the Agri vocabulary.

In cases where either a participant or the system got 0% of the attribute value mappings correct, the reason is that it is a single-value attribute and the system/user did not correctly map this single value. For example, Table VII shows this happens frequently with attributes `unit_weight` and `unit_value`. These attributes each have a single value for the entire dataset, e.g. 'Tonnes', 'Dollars', etc. Thus, if the user or system does not find a match for these terms, the accuracy for that attribute will be 0. However, fixing this in the transformation template would be straightforward as it is a Case 1 error from the error list at the beginning of this section.

Overall, the accuracy of the automated approach to building templates compared favourably with those manually built by people with high levels of skill in the areas of data warehousing and ETL. They also performed well when compared with a gold standard, while reducing the time spent on build time from hours to seconds.

VI. Conclusions

Often, data mining algorithms rely on data which is both current and accurate, acquired from sources outside their organisation. This presents time challenges to analysts in terms of the processing and preparation of data. Domain areas such as agriculture are heavily dependent on the most current data as decisions are often made based upon events in the last 24 hours. Often, new data sources are identified which will provide greater accuracy to decision making but the addition of new sources requires time and the appropriate expertise for proper integration. In this paper, we presented an automated approach to the acquisition of new data for analysts. Using a domain model with an extended vocabulary, it is possible to almost fully automate the acquisition of new sources to enable the construction of new or extended data marts for use in data mining algorithms. Our experiments show a high degree of accuracy for construction of ETL supporting templates within 3-5 seconds for each template. Our current extensions to this work focuses on the validation of multi-source cubes where challenges with missing data require a probabilistic approach to query answering.

Acknowledgements. The authors would like to acknowledge the support provided by the testing team - Dongyun Nie, Michael Scriney and Congcong Xing - for their manual construction and testing of the templates.

References

[1] P. Aubrecht and Z. Kouba, Metadata driven data transformation in Proceedings of World Multiconference on Systemics, Cybernetics and Informatics, ISAS-SCI (1) pp 332–336, 2001.

[2] M. Bergman, Sources and classification of semantic heterogeneities. AI3:Adaptive Information, Available at http://www.mkbergman.com/232/sources-and-classification-of-semantic-heterogeneities/, 2006.

[3] Bord Bia Irish Food Board. https://www.bordbia.ie/Pages/Default.aspx, 2019.

[4] D. Broeder, M. Kemps-Snijders, D. Van Uytvanck, M. Windhouwer, P. Withers, P. Wittenburg and C. Zinn, A data category registry- and component-based metadata framework, in Proceedings of the 7th conference on International Language Resources and Evaluation (LREC 2010), pp 43–47.

[5] D. Broeder, O. Schonefeld, T. Trippel, D. Van Uytvanck and A. Witt, A pragmatic approach to XML interoperability - the Component Metadata Infrastructure (CMDI), in Proceedings of Balisage Series on Markup Technologies In Balisage: The Markup Conference (7), 2011. Available at https://doi.org/10.4242/BalisageVol7.Broeder01.

[6] Dublin Core. http://dublincore.org/documents/dces/, 2019.

[7] E. Duval, W. Hodgins, S. Sutton and S. Weibel, Metadata Principles and Practicalities, D-Lib Magazine, 2002, vol. 8, no. 4, [online] http://www.dlib.org/dlib/april02/weibel/04weibel.html

[8] J. Farinha, M.J. Trigueiros and O. Belo, Using inheritance in a metadata based approach to data quality assessment, in Proceedings of the first international workshop on Model driven service engineering and data quality and security (MoSE+DQS '09) ACM 2009, pp 1–8.

[9] I. Foster, J. Vckler J, M. Wilde and Y. Zhao, Chimera: A virtual data system for representing, querying, and automating data derivation, in Proceedings of the 14th International Conference on Scientific and Statistical Database Management, IEEE 2002, pp 37–46.

[10] M Goodchild, Beyond metadata: Towards user-centric description of data quality, International Symposium on Spatial Data Quality, 2007, pp 1–11.

[11] O. Hartig and J. Zhao, Publishing and consuming provenance metadata on the web of linked data, In: McGuinness D.L., Michaelis J.R., Moreau L. (eds) Provenance and Annotation of Data and Processes, IPAW 2010, LNCS, vol 6378. Springer, Berlin, Heidelberg. pp 78–90, 2010.

[12] UN Stats. Harmonized Commodity Description and Coding Systems (HS). Available at https://unstats.un.org/unsd/tradekb/Knowledgebase/50018/Harmonized-Commodity-Description-and-Coding-Systems-HS, 2019.

[13] ISO. Computer Applications in Terminology - Data Categories - Specification of Data Categories and Management of a Data Category Registry for Language Resources. ISO, Geneva, Switzerland. Available at https://www.iso.org/standard/37243.html

[14] Y. Kargin, H. Pirk, M. Ivanova, S. Manegold and M. Kersten, Instant-on scientific data warehouses - lazy ETL for data-intensive research, In Enabling Real-Time Business Intelligence - 6th International Workshop, BIRTE 2012, Held at the 38th International Conference on Very Large Databases. Revised Selected Papers. pp 60–75, 2012.

[15] S Khouri and L Bellatreche, Design Life-Cycle-Driven Approach for Data Warehouse Systems Configurability, Journal on Data Semantics, Springer-Verlag Berlin Heidelberg, vol. 6, pp 83–111, 2017.

[16] N Kim, Procedural metadata: Structured guide for data interoperability in support of web of things, ITU Journal: ICT Discoveries (Special Issue 2), 2018.

[17] R. Kimball and M. Ross, The Data Warehouse Toolkit (2nd ed), Wiley, 2002.

[18] S. McCarthy, A. McCarren. and M Roantree, Predicting prices, volume and trade for meat and dairy products: an agri food data model, Technical Report 22386, pp 1–24, Available at http://doras.dcu.ie/22386/ 2017.

[19] S. McCarthy, A. McCarren and M. Roantree, Combining web and enterprise data for lightweight data mart construction, in Proceedings of the 29th DEXA (2), LNCS 2018, pp 138–146.

[20] A. McCarren, S. McCarthy, C. O'Sullivan, and M. Roantree, Anomaly detection in agri warehouse construction, in Proceedings of the ACSW, ACM press 2017, pp 1–17.

[21] C. Pluempitiwiriyawej and J. Hammer, A classification scheme for semantic and schematic heterogeneities in XML data sources, Technical Report TR00-004. 2000. Available at https://www.cise.ufl.edu/j̃hammer/publications/TechReports/tr00-004.pdf

[22] G. Rogova and L. Snidaro, Considerations of context and quality in information fusion, in 2018 Proceedings of the 21st International Conference on Information Fusion (FUSION). IEEE 2018, pp 1925–1932.

[23] M. Scriney, S. McCarthy, A. McCarren, P. Cappellari and M. Roantree, Automating data mart construction from semi-structured data sources, The Computer Journal, vol. 62, no. 3, pp 394-413, 2019.

[24] A. Sheth and J. Larson, Federated database systems for managing distributed, heterogeneous, and autonomous databases, ACM Computing Surveys, vol. 22, no. 3, pp 183–236, 1990.

[25] D. Skoutas, A. Simitsis and T.K. Sellis, Ontology-driven conceptual design of ETL processes using graph transformations, Springer Journal on Data Semantics (JoDS), Special issue on Semantic Data Warehouses (JoDS XIII), LNCS 5530, pp 119–145, 2009.

[26] Statistics Canada. https://www.statcan.gc.ca/eng/start, 2019.

[27] United States Department of Agriculture. http://www.ers.usda.gov/data-products/chart-gallery/detail.aspx?chartId=40037, 2019.

[28] S. Zhang, J. Xu, S. Choi, J. Tang, P. Varshney and Z. Chen, A parallel platform for fusion of heterogeneous stream data, in 2018 Proceedings of the 21st International Conference on Information Fusion (FUSION). IEEE 2018, pp 588–594.