WInte.r - A Web Data Integration Framework

Oliver Lehmberg, Alexander Brinkmann, Christian Bizer

Data and Web Science Group, Universität Mannheim
B6 26, 68159 Mannheim, Germany
{oli,chris}@informatik.uni-mannheim.de
albrinkm@mail.uni-mannheim.de

Abstract. The Web provides a plethora of structured data, such as semantic annotations in web pages, data from HTML tables, datasets from open data portals, or linked data from the Linked Open Data Cloud. For many use cases, it is necessary to integrate such web data with existing local datasets. This integration entails schema matching, identity resolution, as well as data fusion. As an alternative to using a combination of partial or ad hoc solutions, this poster presents the Web Data Integration Framework (*WInte.r*), which supports end-to-end data integration by providing algorithms and building blocks for data pre-processing, schema matching, and identity resolution, as well as data fusion. While being fully usable out-of-the box, the framework is highly customisable and allows for the composition of sophisticated integration architectures such as T2K Match, which is used to match millions of web tables against DBpedia. A second use case for which WInte.r was employed is the task of stitching (combining) web tables from the same web site into larger tables as a preprocessing step before matching. The WInte.r framework is written in Java and is available as open source under the Apache 2.0 license.

1 Introduction

Many web-based systems need to combine data from various sources, such as semantic annotations in web pages, data from HTML tables, datasets from open data portals, or linked data from the Linked Open Data Cloud. These data often do not use the same schema and miss explicit links between their entities. Such heterogeneous data sources must hence be integrated before they can be used for any further use case.

While there are tools available to solve specific data integration problems, such as schema matching, identity resolution or data fusion, using these tools in order to realise a complete data integration workflow can be cumbersome and requires researchers to spend quite some time on writing glue code. Also, scalability is often a problem as many tools are designed for use cases with only a few datasets and not for web-scale scenarios where thousands of data sources need to be integrated. As an alternative, this poster presents the **W**eb Data **Integ**r**a**tion (*WInte.r*) Framework. The WInte.r framework supports end-to-end data integration processes by providing algorithms and building blocks for web data pre-processing, schema matching, identity resolution, as well as data fusion. As an example, the integration of millions of tables collected from web pages with the DBpedia knowledge base has been implemented using the WInte.r framework in the T2K Match [2] project.[1]

The WInte.r framework is fully usable out-of-the box with implementations for all mentioned data integration tasks and provides highly customisable functionality that allows for the composition of sophisticated integration architectures. This allows researchers to focus on the actual task instead of starting from scratch and repeatedly implementing the same functionality in different projects. The WInte.r framework is written in Java and is available as open source under the Apache 2.0 license.[2]

2 Functionality

The WInte.r framework covers all steps of the data integration process, including data loading, pre-processing, schema matching, identity resolution, and data fusion. This section gives an overview of the methods that are provided for each of these steps. Figure 1 shows the components that are involved in the integration process.

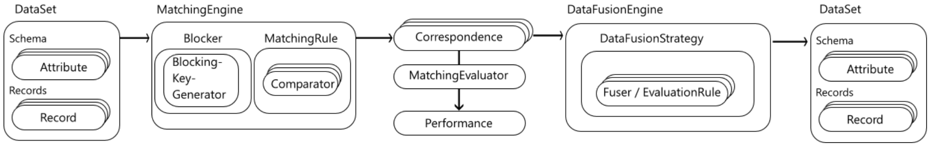

Fig. 1: The components of the data integration process

2.1 Data Management

The WInte.r data sets act as an interchangeable data management and processing component for the rest of the framework. They implement essential processing operations such as iteration, filtering, transformations, aggregations and joins. By exchanging the data set implementation, a user can, for example, switch between single-threaded or parallel execution.

Data Loading. WInte.r provides readers for standard data formats such as CSV, XML and RDF. In addition, WInte.r implements a specialized JSON format for representing tabular data from the Web together with meta-information about the origin and context of the data, as used by the Web Data Commons (WDC) web tables corpora.[3] Specifically, the context represents data such as the original URL, page and section headings and the surrounding text from the web page from which the tables were extracted.

[1] https://github.com/olehmberg/T2KMatch

[2] https://github.com/olehmberg/winter

[3] http://webdatacommons.org/webtables/

Pre-processing. During pre-processing, data is prepared for the methods that are applied later on in the integration process. WInte.r provides specialized pre-processing methods for data with missing schema information, such as: data type detection, unit of measurement normalization, header detection, and entity name detection (In cases where no explicit “rdfs:label” property is available, the entity name detection finds the property that most likely contains the entity names).

2.2 Matching

The matching components are used for schema matching and identity resolution. First, an optional blocking step generates candidate pairs of records to reduce the total number of comparisons. Then, a matching rule evaluates each candidate pair and decides whether or not to create a correspondence. Matching rules are specified by the user and can, for example, calculate the weighted sum of different similarity values and apply a threshold. Alternatively, matching rules can be learned using supervised machine learning. Given a labelled training set, the framework will apply a user-specified machine learning algorithm from the WEKA machine learning library to learn the matching rule.

Schema Matching. Schema matching methods find attributes in two schemata that have the same meaning. WInte.r provides three pre-implemented schema matching algorithms which either rely on attribute labels (label-based schema matching) or data values (instance-based schema matching), or exploit an existing mapping of records (duplicate-based schema matching) in order to find attribute correspondences.

Identity Resolution. Identity resolution methods (also known as data matching or record linkage methods) identify records that describe the same real-world entity. The pre-implemented identity resolution methods can be applied to a single dataset for duplicate detection or to multiple datasets in order to find record-level correspondences. WInte.r provides the following pre-implemented methods: standard blocking by single/multiple blocking key(s), Sorted-Neighbourhood Method, token-based identity resolution and rule-based identity resolution.

2.3 Data Fusion

Data fusion methods combine the data from multiple sources into a single, consolidated dataset. For this task, they rely on the schema- and record-level correspondences that were discovered in the previous steps of the integration process. However, different sources may provide conflicting data values. WInte.r provides a mechanism to resolve such data conflicts (i.e., deciding which value to include in the final dataset) by applying different conflict resolution functions for strings, numbers, lists of values, data type independent functions, and functions that consider metadata such as creation time or dataset specific trust scores. A user can specify a fusion strategy that specifies how data conflicts should be handled for each property.

3 Use Cases

This section gives an overview of existing work based on the WInte.r framework.

Integration of Large Numbers of Data Sources: Augmenting the DBpedia Knowledge base with Web Table Data. Many web sites provide data in the form of HTML tables. Data from these tables can be used to fill missing values in large cross-domain knowledge bases such as DBpedia. An example of how pre-defined building blocks from the WInte.r framework are combined into an advanced, use-case specific integration method is the T2K Match algorithm [2].[4] The algorithm matches millions of web tables against a central knowledge base describing millions of instances belonging to hundreds of different classes [3].

Preprocessing for Large-Scale Matching: Stitching Web Tables for Improving Matching Quality. Tables on web pages ("web tables") cover a diversity of topics and can be a source of information for different tasks such as knowledge base augmentation or the ad-hoc extension of datasets. The challenges that matching methods for this purpose have to overcome are the high heterogeneity and the small size of the tables. To counter these problems, web tables from the same web site can be stitched (combined) before running any of the existing matching systems, which improves the matching quality especially for small tables [1].[5]

Data Search 4 Data Mining (DS4DM). Analysts increasingly have the problem that they know that some data which they need for a project is available somewhere on the Web or in the corporate intranet but they are unable to find the data. The goal of the Data Search 4 Data Mining (DS4DM) project is to extend the data mining platform Rapidminer with data search and data integration functionalities which enable analysts to find relevant data in potentially very large data corpora and to semi-automatically integrate the discovered data with existing local data.[6]

References

1. O. Lehmberg and C. Bizer. Stitching Web Tables for Improving Matching Quality. *PVLDB*, 10(11):1502–1513, 2017.
2. D. Ritze, O. Lehmberg, and C. Bizer. Matching HTML Tables to DBpedia. In *Proc. of the 5th Int. Conference on Web Intelligence, Mining and Semantics*, page 10, 2015.
3. D. Ritze, O. Lehmberg, Y. Oulabi, and C. Bizer. Profiling the potential of web tables for augmenting cross-domain knowledge bases. In *Proceedings of the 25th WWW*, pages 251–261, 2016.

[4] https://github.com/olehmberg/T2KMatch

[5] https://github.com/olehmberg/WebTableStitching

[6] http://ds4dm.de/en/