ASIA: a Tool for Assisted Semantic Interpretation and Annotation of Tabular Data

Vincenzo Cutrona, Michele Ciavotta, Flavio De Paoli, and Matteo Palmonari

University of Milano - Bicocca, Milan, Italy
{name.surname}@unimib.it

Abstract. Enriching datasets with additional information to build robust models is an essential task in many data science applications. Also, the huge availability of Linked Data encourages to reuse and integrate such high-quality information. The ASIA tool assists users in annotating tabular data both at schema- and instance-level, in such a way to enable data extension. This demo paper presents its core capabilities.

Keywords: Semantic annotation · Data enrichment · Linked Data

1 Introduction

Table interpretation and annotation is the process where a table, e.g., a CSV file or an HTML `<table>`, is annotated with semantic pieces of information such as *types* (ontology classes or data types), *properties* and resource *identifiers*. Consider for instance the case where the columns of the table are annotated with types specifying the class of entities or literal values contained in the column. Columns can also be associated with ontology properties, which specify a relation that is implicitly represented in the column; in this case, the column can be interpreted as a source of RDF triples `<subject, predicate, object>`, one per row, such that the values of the annotated column, i.e., the *target column*, are interpreted as objects of the triple, values contained in a different column, specified as the *source column* (of the relation), are interpreted as subjects, and the property specified in the annotation defines the predicate of the triples. In addition to these *schema-level annotations*, *instance-level annotations* match values in the columns (interpreted as mentions to entities) to identifiers in a Knowledge Base (KB), e.g., identifiers of DBpedia resources. Several approaches have been proposed to automate this interpretation and annotation process; we suggest two recent papers for a review of techniques proposed in these approaches [1, 4]. Among these approaches, we also mention semantic labeling approaches, where the distinction as mentioned above between class-based and property-based annotations of columns is less strict than in our definition [3]. The automatic table interpretation and annotation approaches discussed above target two main kinds of applications: mapping tables to known vocabularies and instances so as to generate RDF data from the table; execute structured queries on a large amount of data available in web tables.

In this paper, we showcase ASIA (Assisted Semantic Interpretation and Annotation Tool)[1], a tool designed to support users in annotating data by providing assistance with three main tasks: i) schema-level annotation, to map tabular data to existing vocabularies and generate RDF data; ii) instance-level annotation, to perform data linking while generating the new data; iii) data extension, to use the links established with instance-level annotations to fetch additional data from third-party sources (e.g., after linking a column to DBpedia cities, additional data about these cities can be fetched from DBpedia). Thanks to the combination of instance-level annotations and data extension features, both implemented to work with third-party reconciliation and extension services[2], ASIA targets a new type of application that is crucial to support analytics workflows at scale: **semantic enrichment of tabular data** to help users analyzing their proprietary data once they are enriched with third-party data sources. Applications of this semantic enrichment task can be found in real-world data analytic projects in domains such as Digital Marketing[3], and eCommerce.

ASIA is built on top of the D-a-a-S application DataGraft and its data manipulation tool Grafterizer [6]. From the latter ASIA borrows the capability to transform the annotations into full-fledged data transformation scripts, which can be applied in batch mode to transform data into RDF or enrich large volumes of data. Moreover, ASIA provides features to streamline the annotation task by supplying cross-lingual vocabulary suggestions services based on data profiling systems, which provide information about the usage of vocabularies in existing data (currently, ABSTAT [5] and Linked Open Vocabularies (LOV) are supported). As a result, ASIA table interpretation and annotation is offered as part of an end-to-end solution for semantic data preparation.

We can summarize the novelties of ASIA by comparing it with other table annotation tools (the comparison with table interpretation tools or techniques that do not offer a UI is out of the scope of this paper).[4] Compared to Karma, ASIA provides also reconciliation of column values as well as data extension as features; in addition, (*cross-lingual*) schema-level annotation is implemented as a service and is currently performed using vocabulary usage statistics, rather than one full-fledged ontology (otherwise, Karma uses more sophisticated schema-level annotation techniques). Compared to OpenRefine, ASIA supports more sophisticated schema-level annotations and RDF data generation; it also supports, natively, batch execution of data transformations. Odalic and MantisTable support schema-level annotation, but - to the best of our understanding - does not support data enrichment. RMLEditor supports the editing of rules to generate RDF data, but does not perform table annotation and data enrichment.

[1] `http://inside.disco.unimib.it/index.php/asia/`

[2] The latest release of ASIA includes several reconciliation services: GeoNames, Google GeoTargets, Wikifier, and Google ProductsServices Categories.

[3] Examples of ASIA-supported enrichment pipelines in this domain can be found in [2].

[4] A more complete comparison can be found as a resource at `https://ew-shopp.github.io/eswc2019-tutorial/`, the tutorial's page where ASIA has been presented and compared with several other tools. This is the first work that illustrates the tool.

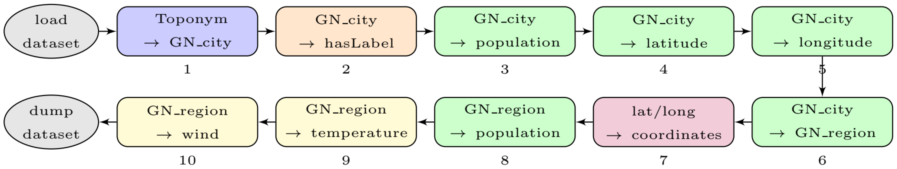

Fig. 1. The enrichment pipeline

2 Demonstration

ASIA's prime objective is to support users in annotating semantically and extending datasets in a tabular format. In the following, we consider a scenario where a user is interested in running analyses requiring information about cities and their regions (such as population and coordinates), and weather forecasts about those regions. The dataset used for this demo has been provided by the JOT Internet Media company[5] and contains data about digital marketing campaigns performance. Particularly, it comes with a column "CityStr", featuring city *toponyms*. We demonstrate how ASIA can help the user in extending the working dataset. First, the user relies on ASIA's matching functionalities to disambiguate the toponyms with non-ambiguous identifiers (URIs) from a reference KB, e.g., GeoNames in the example. These identifiers are then used to query the reference KB to retrieve additional information.

Figure 1 depicts the whole enrichment pipeline. The blocks refer to: a reconciliation step (blue), an annotation step (orange), a transformation step (purple), and extensions steps, namely, KB-based extensions (green), and weather extensions (yellow). The first reconciliation step includes an important user validation step mediated by an interface (wrong reconciliations lead to wrong extensions). Statistics help the user understand the quality of the results returned by automatic reconciliation; the user can modify the results by i) choosing an alternative URI, or ii) manually inserting the URI himself. Consequently, a new reconciled column (named "GN_city") is appended in the working dataset and is automatically annotated with the type of the entities listed therein.[6] In step 2, with the support of the schema-level annotation form, the user specifies that toponyms are associated as labels with the GeoNames entities. Subsequently, the user exploits some KB-based extensions to extend the working dataset with information from GeoNames: the extension form allows to select as many properties as the user needs, and then retrieves all the properties' objects from the KB. In the pipeline depicted in Figure 1, the user applies four consecutive KB-based extension steps starting from the "GN_city" column (steps 3 to 6): all these steps can be accomplished at once by selecting four properties in the extension form. The sixth step, "GN_city → GN_region", adds a new reconciled column to the dataset, which contains the region entity wherein the city entity is located. At this point, the user may want to slightly modify the extension results, for ex-

[5] `https://www.jot-im.com`

[6] Since GeoNames uses the type gn:Feature for all its instances, we adopted the gn:featureCode property as type, which is more significant.

ample by merging the latitude and longitude columns into a new "coordinates" column (step 7). Starting from the "GN_region" column, the user applies new KB-based extensions and appends the population of each region to the dataset (step 8). Lastly, the user retrieves information about weather (temperature and wind) at region level.

Weather extensions become available in ASIA when i) the dataset contains one column annotated as `xsd:date`, or ii) the dataset contains a column reconciled to GeoNames. Thus, the user obtains weather data by extending the "GN_region" column. In the Weather extension form, the user selects the observation dates (that can be kept from another column - ASIA can recognize the most common date formats) and the day offset, i.e., the weather forecast for the next x days using the observation date as base. The user has also to select which aggregation function to apply to the daily weather observations (avg, min, max, cumulative). In the example pipeline, the user chooses to add information about temperature and wind (steps 9 and 10); as a result, the Weather extension appends $n \times m \times p$ new columns, where n is the number of selected parameters, m is the number of selected offsets, and p the number of selected aggregation functions. Finally, the user downloads the enriched dataset in CSV format. Alternatively, she can generate a KB in RDF, or download the whole pipeline as an executable JAR to perform the same manipulations locally on larger volumes of data compared to those that can be managed from the UI.

A video demonstration of ASIA for building an enrichment pipeline that extends the one described above can be found at `https://youtu.be/Z7M2_SjN2xo`[7]. The demonstration can be replicated using the online version of Datagraft at `https://datagraft.io/`.

Acknowledgment. This research has been partly supported by EU H2020 projects EW-Shopp - Grant n. 732590, and EuBusinessGraph - Grant n. 732003.

References

1. Chen, J., Jimenez-Ruiz, E., Horrocks, I., Sutton, C.: Colnet: Embedding the semantics of web tables for column type prediction. In: AAAI (2019)
2. Cutrona, V., De Paoli, F., Košmerlj, A., Nikolov, N., Palmonari, M., Perales, F., Roman, D.: Semantically-enabled optimization of digital marketing campaigns (2019), Accepted for ISWC2019 In-Use track.
3. Pham, M., Alse, S., Knoblock, C.A., Szekely, P.A.: Semantic labeling: A domain-independent approach. In: ISWC. pp. 446–462 (2016)
4. Ritze, D., Bizer, C.: Matching web tables to dbpedia - A feature utility study. In: EDBT. pp. 210–221 (2017)
5. Spahiu, B., Porrini, R., Palmonari, M., Rula, A., Maurino, A.: Abstat: Ontology-driven linked data summaries with pattern minimalization. In: The Semantic Web. pp. 381–395 (2016)
6. Sukhobok, D., Nikolov, N., Pultier, A., Ye, X., Berre, A., Moynihan, R., Roberts, B., Elvesæter, B., Mahasivam, N., Roman, D.: Tabular data cleaning and linked data generation with grafterizer. In: ESWC (Posters & Demos). pp. 134–139 (2016)

[7] Other videos are available at `https://www.youtube.com/playlist?list=PLy7SznldqqmezwdL4QcxQYy2Fz1HVOwMS`.