Simple-ML: Towards a Framework for Semantic Data Analytics Workflows

Simon Gottschalk[1(✉)], Nicolas Tempelmeier[1], Günter Kniesel[2], Vasileios Iosifidis[1], Besnik Fetahu[1], and Elena Demidova[1]

[1] L3S Research Center, Leibniz Universität Hannover, Hanover, Germany
{gottschalk,tempelmeier,iosifidis,fetahu,demidova}@L3S.de
[2] Smart Data Analytics Group (SDA), Universität Bonn, Bonn, Germany
guenter.kniesel@uni-bonn.de

Abstract. In this paper we present the Simple-ML framework that we develop to support efficient configuration, robustness and reusability of data analytics workflows through the adoption of semantic technologies. We present semantic data models that lay the foundation for the framework development and discuss the data analytics workflows based on these models. Furthermore, we present an example instantiation of the Simple-ML data models for a real-world use case in the mobility domain.

1 Introduction

The creation of a *Data Analytics Workflow* (DAW) demands significant data science expertise. This expertise is required to integrate data from heterogeneous sources, to extract features for *machine learning* (ML) tasks, to configure the DAW and to optimize its parameters. The Simple-ML framework, which we currently develop to address these challenges, aims to enable a robust, efficient and reusable DAW configuration through seamless integration of semantic information in all typical DAW components, making it a *Semantic Data Analytics Workflow* (SDAW). The adoption of semantic information, such as a domain model and semantic dataset profiles, substantially differentiates Simple-ML from existing data science frameworks such as RapidMiner or Microsoft Azure.

In this paper we present Simple-ML and illustrate its adoption to data analytics for urban mobility. Popular problems in this domain include short-term road traffic forecasting [5], the prediction of congestion patterns [7] and impact prediction of planned special events [8]. The corresponding SDAWs require a variety of heterogeneous data sources, including but not limited to traffic and mobility data streams, map data (e.g. OpenStreetMap), knowledge graphs containing events and spatial entities (e.g. EventKG [3] and Wikidata), as well as traffic warnings, accidents, weather conditions and event calendars [5,8].

Our contributions are as follows: (i) We propose the Simple-ML framework for SDAWs: a semantic-driven approach that aims at increasing the efficiency of the workflow configuration, as well as robustness and reusability of DAWs using semantic technologies. (ii) We introduce a domain-specific semantic data model

M. Acosta et al. (Eds.): SEMANTiCS 2019, LNCS 11702, pp. 359–366, 2019.
https://doi.org/10.1007/978-3-030-33220-4_26

that provides semantic descriptions of the application domain and domain-specific relevant datasets (i.e. dataset profiles). (iii) We illustrate an application of the Simple-ML framework to a real-world use case in the mobility domain.

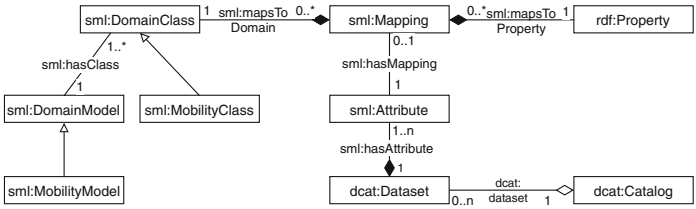

Fig. 1. An UML class diagram illustrating the Simple-ML domain model, the data catalog and a partial instantiation of the domain model in the mobility domain.

2 Semantic Models for SDAWs

The goals of Simple-ML are realized through a domain model (Fig. 1), semantic dataset profiles and the SDAW. We conduct the modeling in RDF[1] reusing existing vocabularies (e.g. `dcat`[2]), where possible. The terms specific to Simple-ML are defined in the Simple-ML vocabulary, denoted using the `sml` prefix[3].

Domain Model: In Simple-ML, the *domain model* describes relevant concepts, their properties and relations in the specific application domain. The class `sml:DomainModel` represents the model of an application domain. The domain-specific concepts are modeled as instances of the class `sml:DomainClass`.

Dataset Profiles: *A dataset profile* is a formal representation of dataset characteristics (features). *A dataset profile feature* is a dataset characteristic. Such features can belong to general, qualitative, provenance, statistical, licensing and dynamics categories [1]. In Simple-ML, the goal of the dataset profiles is to define dataset characteristics required to facilitate SDAWs, including information required for data materialization.

Dataset profile: A dataset profile is modeled as an instance of `dcat:Dataset`. General dataset profile features as well as provenance and licensing features are described using the DCMI Vocabulary (`dcterms`) Statistical dataset profile features (e.g. the number of instances) can be provided at the dataset and the attribute levels.

Dataset attributes: The attributes of the `dcat:Dataset` are modeled as instances of `sml:Attribute`. An attribute is described through its statistical characteristics at the instance level (e.g. the mean value `sml:meanValue`), along with the access

[1] Resource Description Framework (RDF): https://www.w3.org/RDF/.
[2] Data Catalog Vocabulary (DCAT): https://www.w3.org/TR/vocab-dcat/.
[3] The list of the adopted namespaces and the data catalog are available online: https://simple-ml.de/index.php/data-catalog/.

information to the underlying data source (e.g. the column name in a relational database) to facilitate data access and materialization.

Dataset access: Simple-ML supports access to datasets through dedicated attributes that represent physical storage location and data format (e.g. `sml:fileLocation` and `csvw:separator`). Currently, relational databases (`sml:Database`) and text files (`sml:TextFile`) are supported.

Mapping between the Dataset Profile and the Domain Model: Dataset attributes are mapped to the concepts in the domain model (`sml:DomainClass`) through the `sml:Mapping` class, as illustrated in Fig. 1. This mapping adds domain-specific semantic description to the dataset attributes and facilitates their use in the SDAWs. The class `sml:Mapping` provides two properties: `sml:mapsToProperty` to map a dataset attribute to a property in the domain model, and `sml:mapsToDomain` to specify the `rdfs:domain` of this property, which is an instance of `sml:DomainClass`.

Data Catalog: Dataset profiles are organized in a domain-specific data catalog. The extensible Simple-ML data catalog is modeled as an instance of `dcat:Catalog`. The data catalog schema including representations of dataset profiles and the mapping to the domain model is illustrated in Fig. 2.

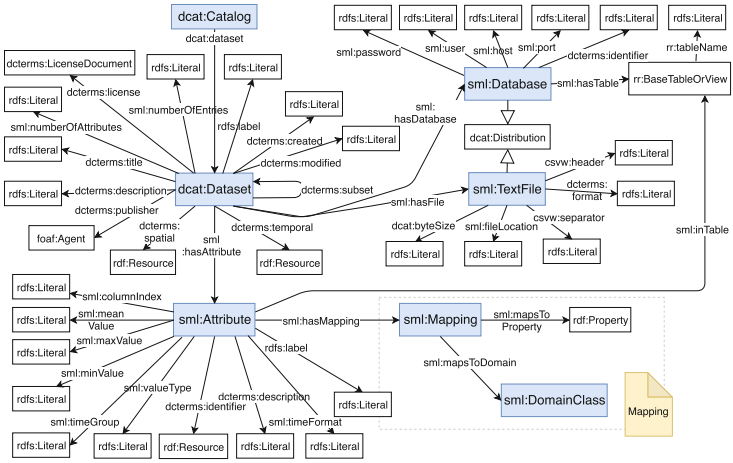

Fig. 2. The data catalog schema based on the `dcat` vocabulary. Arrows with an open head denote the `rdfs:subClassOf` properties. Regular arrows denote the `rdfs:domain` and `rdfs:range` restrictions. Blue boxes denote the key `dcat` and `sml` classes. (Color figure online)

3 Semantic Data Analytics Workflow (SDAW)

Figure 3 depicts an overview of a *Semantic Data Analytics Workflow* (SDAW). A SDAW consists of several steps discussed in the following.

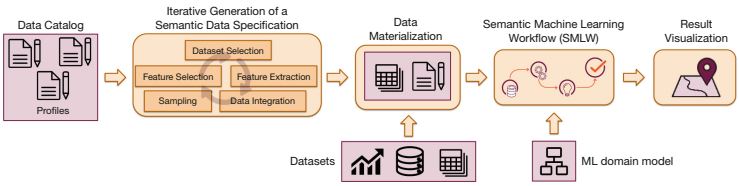

Fig. 3. An overview of the Simple-ML Semantic Data Analytics Workflow (SDAW).

Iterative Generation of a Semantic Data Specification: In this first step, the user defines the semantic specification of the data to be used in the workflow. The input in this step is the data catalog. The specification is defined through the selection of the operations to be applied to the dataset(s) in the data catalog and their attributes. Possible operations include dataset selection, sampling, feature selection, feature extraction and data integration. These operations can be applied iteratively in a user-defined order. The Semantic data specification is defined at the metadata level using dataset profiles and does not require any physical data access. The specification can be stored to facilitate reusability.

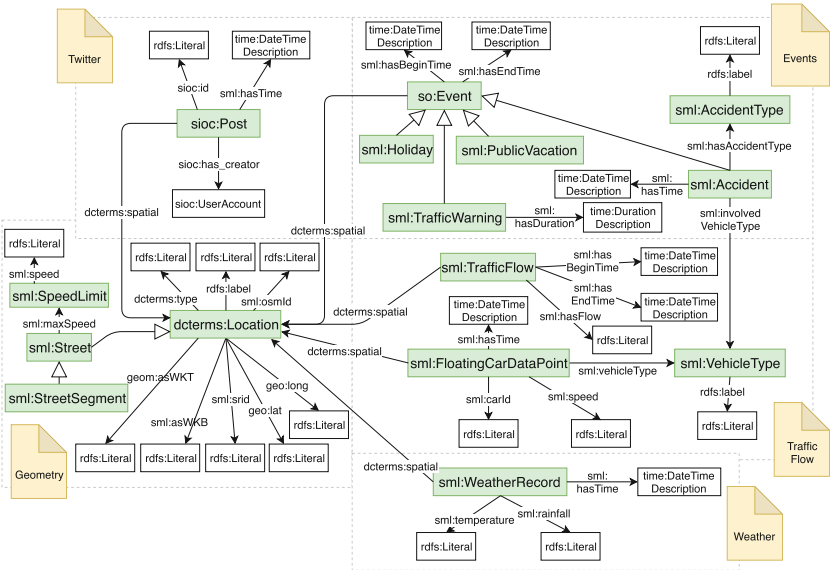

Fig. 4. An example domain model for the mobility domain. The arrows with an open head denote the `rdfs:subClassOf` properties. Regular arrows denote the `rdfs:domain` and `rdfs:range` restrictions. Classes in green boxes are sub classes of `sml:MobilityClass`.

Data Materialization: The data specification configured during the previous steps is applied to the physical datasets to materialize the integrated data.

Semantic Machine Learning Workflow (SMLW): The domain model is complemented with a ML domain model that captures the essential properties of ML concepts and their implementation in specific frameworks. A domain specific language (DSL) for SDAWs and SMLWs will include an advanced type system that will use metadata from the application domain to describe datasets and the intermediate results of data processing on one hand, and the metadata of the ML domain to describe the ML processing steps. This will enable statically checking the correctness of applying particular ML methods to particular data. To this extent, we will build upon previous approaches aiming to integrate ontologies into existing type systems (see e.g. [4]). We will go one step further, by designing a language dedicated to the data analytics and ML domain and including data models both for the data and also for the ML processes.

Result Visualization: The domain model can be used to automatically suggest suitable visualizations for specific data types.

4 Domain Model for Mobility

Figure 4 exemplifies an instantiation of the domain model in the mobility domain. This model includes the following classes:

- `sml:FloatingCarDataPoint`: A vehicle's type, position, time and speed.
- `sml:TrafficFlow`: Vehicle count statistics (e.g. from road sensors [7]).

```
sml:SimpleMLCatalog a dcat:Catalog ;
    dcat:dataset sml:FCDDataset .
sml:FCDDataset a dcat:Dataset ;
    dcterms:title "Floating Car Data" ; sml:hasFile sml:FCDDatasetFile ;
    dcterms:temporal [  so:startDate "2017-08-01"^^xsd:date ;
                        so:endDate "2017-12-31"^^xsd:date ] ;
    sml:hasAttribute sml:FCDDatasetAttribute1 .
sml:FCDDatasetFile a sml:TextFile ;
    dcterms:format "text/comma-separated-values" ; csvw:separator ";" .
sml:FCDDatasetAttribute1 a sml:Attribute ;
    rdfs:label "vehicle id"@en ; sml:columnNumber "0"^^xsd:integer ;
    sml:hasMapping [    sml:mapsToProperty sml:carId ;
                        sml:mapsToDomain sml:FloatingCarDataPoint ] .
```

Fig. 5. An excerpt of an example data catalog in the mobility domain.

```
SELECT ?columnNumber  ?attrName ?mapProperty ?mapDomain WHERE {
    sml:FCDDataset sml:hasAttribute ?attribute .
    ?attribute dcterms:identifier ?attrName .
    ?attribute sml:columnNumber ?columnNumber .
    OPTIONAL { ?attribute sml:hasMapping [
        sml:mapsToProperty ?mapProperty ; sml:mapsToDomain ?mapDomain ; ] . } }
```

Fig. 6. SPARQL query to select attributes of a given dataset (here: `sml:FCDDataset`).

- `so:Event`: Mobility-relevant events, their time and geographical location.
- `sioc:Post`: Social media posts modeled using the SIOC ontology[4].
- `sml:WeatherRecord`: Temperature and rainfall at location and time.
- `dcterms:Location`: Spatial information with geographical coordinates.
- `sml:SpeedLimit`, `sml:AccidentType`, `sml:VehicleType`: Classes that represent categorical values for speed limits, accident types and vehicle types.

These classes are sub classes of `sml:MobilityClass`, which is a sub class of `sml:DomainClass` and thus allows the use of `sml:Mapping` as shown in Fig. 2.

Figure 5 provides an excerpt of an example Simple-ML mobility data catalog.

5 Simple-ML Application to Traffic Speed Prediction

We illustrate the iterative generation of a semantic data specification for the problem of traffic speed prediction for a specific road segment at a given time.

Dataset Selection: The user selects a Floating Car Data (F) and OpenStreetMap (O) datasets. Figure 6 shows the SPARQL query to retrieve F's profile.

Data Specification: (i) *Feature Selection*: The user selects four features based on the domain model: `sml:maxSpeed`, `sml:hasTime` from (F) (class `sml:FloatingCarDataPoint`), and `rdf:type` and `sml:maxSpeed` from (O) (class `sml:StreetSegment`). (ii) *Feature Extraction*: The user selects the following temporal features that are suggested by the system: week day, hour of day from (F). (iii) *Data Integration*: A mapping between the vehicle positions in (F) and the street segment coordinates in (O) is suggested by the system and chosen by the user.

Data Materialization: Using the data specification, relevant features are materialized, with example instances shown in Table 1. The resulting data can then be used in the SMLW to train a supervised traffic speed prediction model.

Table 1. Example instances generated using the semantic data specification

FloatingCarDataPoint (F)				StreetSegment (O)	
Type	Speed	Time (day)	Time (hour)	Type	maxSpeed
1	74	Sunday	23	motorway_link	80
0	84	Sunday	16	motorway	*none*
1	17	February	8	secondary	70

[4] https://www.w3.org/Submission/sioc-spec/.

6 Related Work

Recent works [2,4,6] aim to combine semantics and ML to address a variety of real-world problems. Simple-ML goes one step further and makes use of semantics in the entire DAW. Simple-ML employs dataset profiles and domain-specific data models. The survey [1] provides a comprehensive overview of RDF dataset profiling methods, tools, vocabularies and features partially utilized by Simple-ML. We illustrate the use of Simple-ML in the mobility domain. Mobility has seen many challenges and use cases for data analytics [5,7,8]. In Simple-ML, the mobility domain is modeled in a light-weight, data-driven manner that facilitates compatibility and reusability of the SDAWs across use cases and datasets.

7 Conclusion

In this paper we presented our current development towards the Simple-ML framework. Simple-ML adopts semantic technologies to support the efficient creation, configuration and reusability of robust data analytics workflows. We illustrated an application of the framework to a real-world use case in the mobility domain.

Acknowledgements. This work was partially funded by the Federal Ministry of Education and Research (BMBF), Germany under Simple-ML (01IS18054) and Data4UrbanMobility (02K15A040).

References

1. Ellefi, M.B., et al.: RDF dataset profiling - a survey of features, methods, vocabularies and applications. Semantic Web **9**(5), 677–705 (2018)
2. Esteves, D., et al.: MEX vocabulary: a lightweight interchange format for machine learning experiments. In: Proceedings of the SEMANTiCS (2015)
3. Gottschalk, S., Demidova, E.: EventKG: a multilingual event-centric temporal knowledge graph. In: Proceedings of the ESWC (2018)
4. Hartenfels, C., Leinberger, M., Lämmel, R., Staab, S.: Type-safe programming with OWL in Semantics4J. In: Proceedings of the ISWC (2017)
5. Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., Zhou, X.: LC-RNN: a deep learning model for traffic speed prediction. In: Proceedings of the IJCAI 2018 (2018)
6. Merkle, N., Zander, S.: Using a semantic simulation framework for teaching machine learning agents. In: SEMANTiCS, pp. 78–89 (2018)
7. Nguyen, H., Liu, W., Chen, F.: Discovering congestion propagation patterns in spatio-temporal traffic data. IEEE Trans. Big Data **3**(2), 169–180 (2017)
8. Tempelmeier, N., Dietze, S., Demidova, E.: Crosstown traffic – supervised prediction of event impact on urban traffic. GeoInformatica (2019)